

基于评分矩阵与评论文本的深度推荐模型

冯兴杰 曾云泽

(中国民航大学计算机科学与技术学院 天津 300300)

摘 要 基于评分矩阵的矩阵分解模型被广泛研究与应用,但是数据稀疏性问题严重制约了该模型的推荐效果. 基于评论文本的推荐模型能够从文本信息中刻画用户偏好和商品特征,有效缓解了评分数据的稀疏性,但忽略了评分矩阵中用户和商品的潜在因子. 为了进一步提高推荐质量,融合评分矩阵和评论文本的推荐模型被相继提出,但其仅仅局限在浅层线性特征层面,而且用户特征与商品的高级抽象特征未被充分挖掘,因此本文提出深度学习模型 DeepCLFM(Deep Collaborative Latent Factor Model). 该模型基于预训练的 BERT 模型,结合双向 GRU 和注意力机制从用户评论和商品评论中提取用户和商品的深层非线性特征向量,并根据用户和商品的编号映射出用户和商品的潜在隐向量. 为了充分融合深层非线性特征和隐特征,DeepCLFM 将用户和商品的深层特征向量与潜在隐向量以一、二阶特征项的方式产生深度特征项来预测出用户对商品的评分. 在 5 组公开数据集上,以推荐结果的均方误差 MSE 作为评估指标进行对比实验,结果表明 DeepCLFM 的预测误差比多个优秀的基准算法更低,且平均预测误差最大降低了 6.402%.

关键词 推荐系统;评论文本;评分矩阵;神经网络;冷启动

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2020.00884

Joint Deep Modeling of Rating Matrix and Reviews for Recommendation

FENG Xing-Jie ZENG Yun-Ze

(School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300)

Abstract With the growing popularity of the Internet and smart mobile devices, people's online time is rising. In order to improve office efficiency and consumption experience, the company provides a variety of products and services to meet the different needs of users, but it is also more difficult for users to quickly make satisfactory decisions from a large amount of information. Due to it can help different users to find out the items they are interested in through their historical behavior, the recommender system has become an extremely important part of online activities, such as online shopping, reading articles, and watching movies. To provide a personalized recommendation service, how to accurately predict the user's rating of the item is a key issue that the recommender system needs to solve. Based on rating matrix, one of the most outstanding methods is matrix factorization, which has been widely studied and applied to model user preferences and item characteristics through rating data. However, the performance of these methods is severely restricted by the data sparsity problem, which can be seen as a phenomenon of the shortage of trainable data. To overcome this limitation, the recommendation models based on the review text can capture the user preferences and item features from the text data, effectively alleviating the sparsity of the rating data, but they ignore the latent factors of users and items in the rating matrix. With the comprehensive consideration of the above models and to further improve the

recommendation quality, the model of combining rating matrix and review text has been proposed one after another. However, they are only limited to the linear latent feature level, in which the high-level abstract features of users and items fail to be fully explored. Therefore, this paper proposes deep learning model DeepCLFM (Deep Collaborative Latent Factor Model). First, the pre-trained BERT is used as the encoder of review text, which is a general-purpose “language understanding” model trained on a large text corpus like Wikipedia. Second, with the purpose of considering the latent relationship between different reviews in a review set, DeepCLFM extracts deep nonlinear feature vectors of users and items from review embeddings through a bidirectional GRU. Additionally, DeepCLFM introduces attention mechanism to measure the contribution of each review, and adopts matrix factorization module to learn latent factors according to the IDs of users and items. Finally, to fully integrate deep nonlinear features and latent factors, DeepCLFM generates deep interaction of them in the first and second order fashion to predict the user’s rating of the item. Experiments are conducted on five public datasets called Amazon Product Review, in which each sample contains user ID, item ID, user’s rating on the item (1~5 points), and user’s review text on the item. The mean square error (*MSE*) of the recommendation results is used as the evaluation metric. The results show that the prediction error of DeepCLFM is lower than that of many excellent benchmark algorithms, and the prediction error is reduced by a maximum of 6.402%. Moreover, DeepCLFM achieves a better performance than traditional matrix factorization in the “cold start” scenario.

Keywords recommender systems; review text; rating matrix; neural network; cold start

1 引言

随着互联网和智能移动设备的普及,信息爆炸的大数据时代已经来临.作为缓解信息过载的重要技术,推荐系统目前已经在健康^[1]、智能交通^[2]、教育^[3]、电子商务^[4]等领域有相当成功的应用.特别在电子商务领域,推荐系统能够根据用户的历史行为数据帮助不同用户快速找出其感兴趣的物品,已经成为人们网上购物、阅读文章、观看电影等线上活动极其重要的组成部分,也是企业核心竞争力的重要组成部分.为了提供更好的个性化推荐服务,如何准确地预测用户对物品的评分是推荐系统需要解决的关键问题.

当前杰出的推荐系统大多都是基于协同过滤技术,其主要思想是以往有相同偏好的人群会在未来做出相似的选择.其中大放异彩的协同过滤算法是基于矩阵分解的 LFM (Latent Factor Model)^[5],该算法将每个用户和物品分别映射成两个向量进行表达(向量的每一维就是用户或物品的隐因子,分别表示用户不同方面的偏好或物品不同方面的特征),最后将这两个向量的点积结果作为该用户对物品的预

测评分. LFM 不仅能找出这些隐因子,并且能学习到不同隐因子间的相互影响.

尽管基于矩阵分解的协同过滤算法在很多情况下有优越的性能,但是评分数据的稀疏性一直是其性能进一步提升的瓶颈.例如用户只对千万分之一的物品进行过打分,而我们期望模型能够对另外用户没接触过的大量物品进行评分预测,由于没有足够多的信息指导模型进行合理地推算,模型的评分预测远远没达到令人满意的程度,因此评分数据稀疏性问题实际上是一种数据信息贫乏的表现.

一种简单有效缓解评分数据稀疏的方法是利用评论文本数据,以增加更多信息的方式来指导模型学习出更为合理的参数,进而提高模型预测精度^[6-7].相比于信息单一的评分数据,评论文本天然具有丰富的信息,能够从以下两方面来提高模型性能:(1)评论文本中直观地叙述了物品获得该评分的原因,这能够帮助模型理解该物品的特征;(2)不同用户对同一物品的评价侧重点及叙述方式都不同,因此评论文本也能让模型更有效地刻画用户偏好.

近年来深度学习技术得到长足发展,目前最新的推荐算法都采用了深度学习技术.例如 DeepCoNN^[8]、

D-Attn^[9]、WCN^[10] 和 NARRE^[11], 它们都是使用 CNN 将评论编码至相应的隐空间, 并取得了不错的效果. 但是, 它们也面临以下 4 点局限性: (1) 它们将多个单词拼接成一个长序列文本, 采用基于局部视野(卷积核大小固定)的 CNN 进行编码, 其所丢失的长距离特征较为严重; (2) 它们都使用 word2vec^[12] 或 glove^[13] 预训练词向量, 然而该词向量属于静态编码的一种, 同一单词在不同的上下文环境中是相同的参数表达, 导致模型对语义的理解有偏差. 例如“该手机具有‘很长’的使用寿命”、“该手机需要‘很长’的开机时间”. 这两种情况中的‘很长’, 前者表示积极的评价, 后者表示负面的评价. 针对该问题, 目前较好的解决方法是采用在大语料下预训练的 BERT 模型^[14], 每个单词的表示会根据不同的上下文信息而动态变化; (3) 在某用户或商品的评论集中, 它们(DeepCoNN、D-Attn、WCN)将每条评论对用户建模的贡献视为同等重要; 实际上如果能区分每条评论的贡献度则能进一步提升推荐性能^[11]; (4) 它们(DeepCoNN、D-Attn、WCN)仅使用评论数据作为模型输入, 未能充分挖掘评论数据与评分数据的内在联系. 为了解决上述局限性, 本文提出深度推荐模型 DeepCLFM (Deep Collaborative Latent Factor Model).

本文的贡献可以总结为以下 3 点:

(1) 本文结合评论文本及评分矩阵提出了深度学习模型 DeepCLFM, 该模型能以一、二阶特征项的方式, 联合评论文本及评分矩阵共同学习出用户偏好和商品属性的深度特征, 进一步提升模型推荐效果;

(2) 以预训练的 BERT 模型取代以往模型的词嵌入层, 直接将每条评论送入 BERT 模型得到各评论的隐表达. 基于评论级别的隐表达, 采用双向 GRU 并结合注意力机制来度量各评论对用户(或商品)特征的贡献程度, 使得模型能够较准确理解评论中的语义及情感, 有效构建评分、商品、用户这三者间的潜在关系, 进一步提升评分预测精度;

(3) 在五组公共数据集上进行对比实验, 结果表明 DeepCLFM 不仅能利用评论文本来降低评分矩阵稀疏性的影响, 而且预测结果的均方误差 MSE 比目前多个优秀的模型^[5,7-8,11,15-17]更低.

本文第 2 节将介绍相关工作的研究进展, 包括基于评分数据的推荐方法、融入评论文本的推荐方法; 第 3 节介绍本文所提出的模型 DeepCLFM; 第 4 节主要进行模型性能、结构的实验与分析; 第 5 节对

本文模型进行相关讨论; 最后在第 6 节给出总结与展望.

2 相关工作

2.1 基于评分矩阵的推荐方法

在基于评分数据的推荐方法中, 最常见的方法是基于邻域的模型. 其中基于用户的协同过滤算法是最早期的基于邻域的推荐方法, 该方法通过相似度计算找出一群需求类似的用户的评分记录来估计未知评分. 随后基于商品相似度的方法流行起来, 并为推荐结果提供了直观的解释.

目前基于评分矩阵的协同过滤方法中, 基于矩阵分解的隐变量模型是最为成功的一类模型. Koren 等人^[5,16,18]提出的矩阵分解模型 LFM 在 Netflix 比赛中表现突出, 为日后的许多工作提供了值得借鉴的思路: 用户偏好和商品属性可以分别用两个隐向量表达, 最终用户对商品的打分等于这两个向量的点积. 随后, Salakhutdinov 等人^[19]提出了基于概率的矩阵分解方法 PMF, 通过添加特征的高斯概率分布来进一步改进算法. 多项研究的实验结果表明基于矩阵分解模型的推荐效果明显优于基于邻域的推荐方法.

随着深度学习的兴起, 模型构造的特征不仅仅局限于以往模型所构造的浅层特征, 越来越多基于深度学习的方法被运用在推荐任务上. 文献^[20-22]使用去噪自动编码器或者 RBM (Restricted Boltzmann Machines) 从评分矩阵对用户和商品进行建模, 但是它们都仅仅使用了评分数据而忽略了评论文本.

尽管矩阵分解推荐算法的推荐效果比较好, 但是评分数据的稀疏性问题一直是制约该类模型进一步发展的主要因素. 为了缓解评分数据稀疏性, 近年来的研究热点是通过挖掘评论文本来对用户和商品建模, 从评论文本语义信息及内在情感来提高模型预测精确度.

2.2 融入评论文本的推荐方法

2.2.1 基于主题模型的方法

最早使用评论文本来降低评分预测误差的工作是文献^[23], 其发现用户对电影的评论是宝贵的资源, 认为评论文本中关于价格、服务、正负情感的叙述能够提高模型的评分预测性能. 但是该工作停留在这种能够一一对应的显性特征, 未能对潜在的隐因子特征进一步挖掘.

随后文献[24]使用 LDA 模型从 Yelp 评论数据集中挖掘子话题来构建隐因子特征并进行评分预测, 开启了运用概率主题模型从评论文本来构建用户和商品隐因子的研究方向. 文献[25]提出 CTR 模型, 这是一种结合概率主题模型的协同过滤算法, 其利用文本信息在推荐文章的任务上表现出优越的性能. 为了进一步与早期基于用户的协同过滤思想相结合, 文献[26]先利用评论文本构建用户偏好特征, 然后将相似的用户进行聚类, 最后使用 LFM 进行评分预测. 文献[7]通过将评分数据的潜在维度(与 LFM 相似的用户和商品隐因子)与评论文本的主题分布进行结合, 提出表现优秀的 HFT 模型. 随后相似的研究陆续出现, 文献[27]提出基于协同过滤和主题模型的概率模型来进行评分预测, 但该方法对评论文本建模时并没有使用评分数据. 为了利用评分数据, 文献[6]提出的 RMR 模型将主题模型与基于评分数值的混合高斯模型相结合, 进一步提高推荐性能. 但是文献[28]认为 LDA 只能挖掘词语级别的主题分布, 不能准确表达复合主题的分布, 因此提出 TopicMF 模型. TopicMF 通过非负矩阵分解得到每条评论的潜在主题, 并与用户和商品的隐因子建立映射关系, 最后主题分布反映了用户偏好和商品特性. 文献[29]指出结合评论文本使用主题模型对评分矩阵进行挖掘隐因子的模型都归类为主题矩阵分解模型. 文献[30]通过手动建立部分主题词将评论文本主题与评分矩阵分解模型融合. 文献[31]首先利用主题模型挖掘主题分布来刻画用户偏好和商品画像, 通过逻辑回归构建主题与评分之间的关系. 文献[32]提出基于隐概率和随机漫步的混合模型, 概率模型负责挖掘用户潜在偏好及商品的隐特征, 随机漫步能够构建全局的潜在关联来预测用户对未评分商品的偏好.

上述的相关工作, 大都是通过概率主题模型来对评论文本挖掘出用户或商品潜在的特征分布, 然而基于词袋模型的主题模型不能保留词序信息, 忽略了在情感分析中极其重要的局部上下文信息^[33]. 另外, 从深度学习的角度来看, 这些方法学习到的都是浅层线性特征, 未能充分挖掘出深度的非线性隐因子特征.

2.2.2 基于深度模型的方法

目前, 深度学习模型 CNN、RNN 能有效保留词序信息, 并能结合不同的注意力机制来提高文本信息的提取质量, 主题模型已逐渐被 CNN 和 RNN 所

取代.

2016 年 Kim 等人^[34]提出 ConvMF(Convolution Matrix Factorization), 利用卷积神经网络从商品描述文本中产生更深层的隐表达, 这种方法考虑了文本的局部词序信息, 因此产生出更准确的商品隐空间向量. 然而, ConvMF 仅仅考虑使用了商品的文本信息, 而忽略了用户的文本信息. 针对该问题, 2017 年 Zheng 等人^[8]提出 DeepCoNN(Deep Cooperative Neural Network), 其使用两个并行的 CNN 网络分别处理用户评论集和商品评论集, 这启发了后来很多相关的研究. 2017 年 8 月, Catherine 等人^[35]基于 DeepCoNN 的双 CNN 结构提出 TransNet(Transformational Neural Networks), 其能先重构出待预测评分对应的评论来预测评分. 同样采用 DeepCoNN 的双 CNN 结构, 但是评分预测性能的提升并不明显的研究共有三个: (1) Seo 等人^[9]提出的 D-Attn(CNNs with Dual Attention Model), 其通过同时结合局部注意力机制和全局注意力机制来对评论文本中的各个单词进行打分, 找出与用户对商品的评分最相关的单词, 来提高推荐结果的可解释性; (2) 2018 年 Wang 等人^[10]提出的 WCN(Word-driven and Context-aware Networks), 其先使用主题模型对评论文本进行预处理, 使其能弥补 CNN 可能丢失的词频信息; (3) 2018 年 Wu 等人^[36]提出的 DRMF(Dual-Regularized Matrix Factorization with Deep Neural Networks), 其分别在两个 CNN 网络之后添加一层双向 GRU 网络结构来提高评论特征的提取质量.

但是上述这些方法都仅使用评论文本作为模型输入来学习用户和商品的隐表达, 未能充分利用评分矩阵中用户及商品的隐因子. 针对上述问题, 与本文最相近的工作是 Chen 等人^[11]提出的 NARRE(Neural Attentional Regression model with Review level Explanations), 其同时将评论文本及评分矩阵作为输入, 评分预测性能得到较大的提升. 但是, NARRE 在提取评论特征时, 仍采取静态词向量(word2vec 或 glove)结合 CNN 的结构, 而本文则采取 BERT 模型结合双向 GRU 的结构. 另外, NARRE 在学习商品的评论特征过程中, 已经和评分数据的用户隐因子产生了交互, 本文认为这样会影响评论特征的表达质量. 相反, 本文所提模型 DeepCLFM 使评论数据特征与评分数据特征的学习相对独立, 二者的交互仅发生在融合层模块.

3 模 型

3.1 问题与符号定义

给定一个包含 N 个样本的数据集 D , 其中每个样本 (u, i, r_{ui}, w_{ui}) 表示用户 u 对商品 i 写了一条评论 w_{ui} 及相应的评分 r_{ui} . 现在的任务是训练出一个模型能够根据用户 u 、商品 i 、用户 u 的评论集(用户 u 除了 w_{ui} 以外的任何历史评论)、商品 i 的评论集(商品 i 除了 w_{ui} 以外的任何历史评论)来预测出评分 \hat{r}_{ui} , 使得 \hat{r}_{ui} 与 r_{ui} 之间的误差最小. 其中本文使用的数学运算符号的定义如表 1 所示.

表 1 符号定义	
符号	定义
\odot	给定两个同等维度的向量 a, b , 则 $a \odot b$ 表示 a 和 b 之间对应元素相加
sum	给定一个 k 维向量 a , 则 $\text{sum}(a)$ 表示将 a 中 k 个元素进行相加, 最终得到一个实数
multiply	给定两个同等维度的向量 a, b , 则 $\text{multiply}(a, b)$ 表示 a 和 b 之间对应元素相乘
\oplus	给定两个 k 维的向量 a, b , 则 $a \oplus b$ 的结果是 a 和 b 进行拼接的 $2k$ 维向量
*	卷积运算
\times	矩阵乘法

3.2 模型结构

在 3.2 节中将会详细介绍本文提出的模型 DeepCLFM. 如图 1 所示, DeepCLFM 共包含 3 个模块, 模块一是文本处理模块, 其中包含的两个并行网络 Net_u 和 Net_i , 它们分别负责从用户评论文本及商品评论文本对用户和商品进行建模. 最后两个网络 Net_u 和 Net_i 的输出分别为用户和商品的深层特征向量 $TEXT-U$ 和 $TEXT-I$; 模块二是矩阵分解模块, 该模块的输出是评分矩阵中分解出得到的用户及商品的隐向量 $LFM-U$ 和 $LFM-I$, 其属于浅层特征向量; 模块三是融合层, 其主要负责将上游模块(文本特征提取模块、矩阵分解模块)得到的深层特征与浅层特征进行不同的组合, 得到不同的高级复合特征(一阶特征、二阶特征和深度特征), 然后通过全连接层将高级复合特征以不同权重的方式进行线性组合, 最终回归出用户对商品的预测评分 $Y_{predict}$.

3.2.1 文本处理模块

文本处理模块结构如图 2 所示. 由于文本处理模块是由两个结构相似的平行网络组成, 因此本小节只介绍负责对用户评论建模的网络 Net_u 进行说明(对商品评论进行建模的 Net_i 同理).

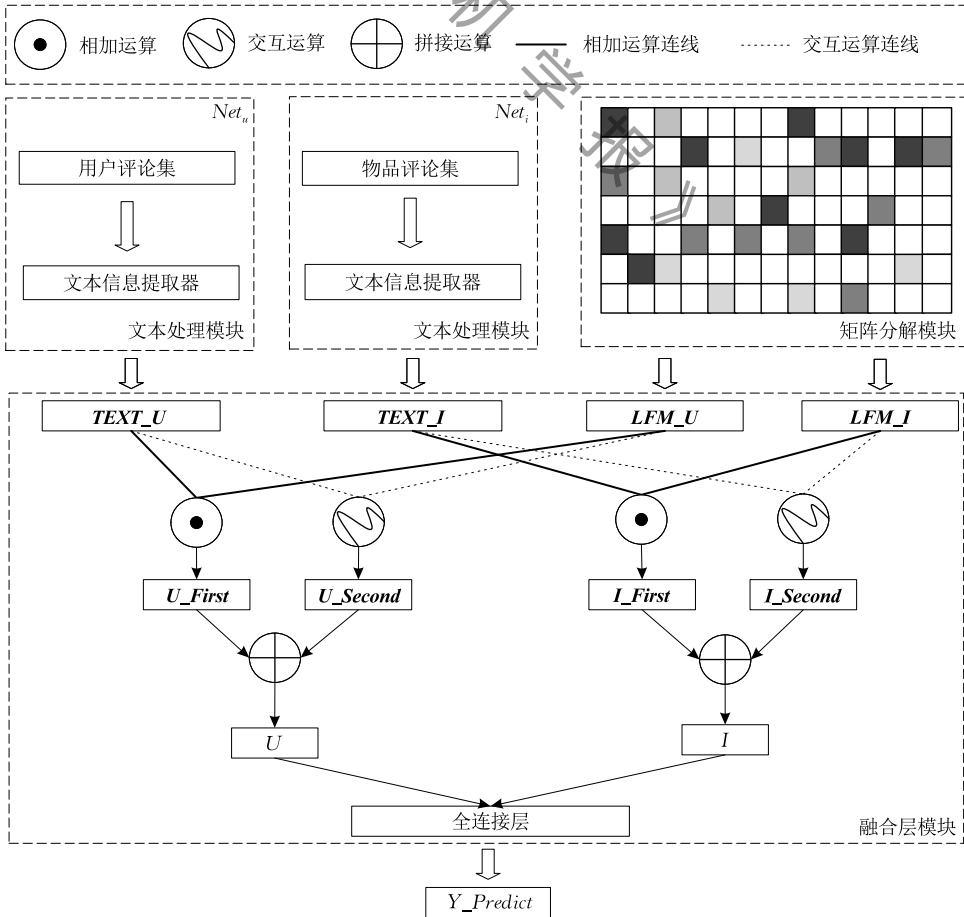


图 1 模型结构

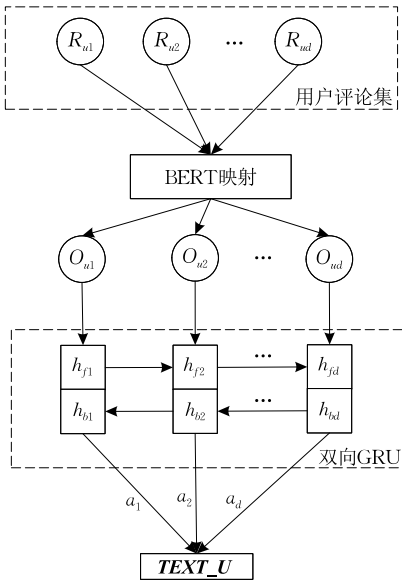


图2 文本处理模块结构

给定某用户 u 的评论集, 即该用户各条评论组成的列表 $\{R_{u1}, R_{u2}, \dots, R_{ud}\}$, 其中 d 表示模型输入最大的评论数. 经过 BERT^[14] 映射后, 每条评论映射为 c 维的向量表达, 得到 BERT 输出列表 $\{O_{u1}, O_{u2}, \dots, O_{ud}\}$. 如果用户评论集的历史评论数少于 d , 则先将该用户的全部历史评论逐条经过 BERT 映射, 得到长度小于 d 的 BERT 输出列表. 然后通过填充若干条 c 维的零向量, 使得 BERT 输出列表的长度为 d ; 如果用户评论集的历史评论数大于 d , 则只将该用户的前 d 条评论逐条经过 BERT 映射, 得到长度为 d 的 BERT 输出列表.

得到每条评论的隐表示后, NARRE 直接采用注意力机制将它们汇聚成一个评论集的隐表示. 但其忽略了评论之间的内在联系, 因此本文再使用一层双向 GRU 来对 $\{O_{u1}, O_{u2}, \dots, O_{ud}\}$ 进行编码, 其能从前向和后向来对 BERT 输出列表提取更高级的信息. 前向 GRU_f (forward) 从 O_{u1} 到 O_{ud} 的方向进行处理, 后向 GRU_b (backward) 从 O_{ud} 到 O_{u1} 的方向进行处理:

$$h_{fi} = GRU_f(O_{ui}), i \in [1, d] \quad (1)$$

$$h_{bi} = GRU_b(O_{ui}), i \in [d, 1] \quad (2)$$

为了得到 O_{ui} 对应的隐藏状态 $h_i \in \mathbb{R}^{2l}$, 这里采用拼接策略:

$$h_i = h_{fi} \oplus h_{bi} \quad (3)$$

其中每个 GRU 的隐藏单元数为 l , $\mathbf{H}_o \in \mathbb{R}^{d \times 2l}$ 表示整个双向 GRU 组件的隐藏状态:

$$\mathbf{H}_o = (h_1, h_2, \dots, h_d) \quad (4)$$

直观来看, 用户评论集中每条评论能体现用户

偏好的贡献是不同的, 这里采用自注意力机制^[37]对 \mathbf{H}_o 进行信息汇总, 得到注意力向量 $\mathbf{a} \in \mathbb{R}^{1 \times d}$:

$$\mathbf{a} = \text{softmax}(\mathbf{w}_1 \times \tanh(\mathbf{w}_2 \times \mathbf{H}_o^T)) \quad (5)$$

其中 $\mathbf{w}_1 \in \mathbb{R}^{1 \times l}$, $\mathbf{w}_2 \in \mathbb{R}^{l \times 2l}$, l 是可设置为任意维度的超参数. $\text{softmax}()$ 用于将注意力权重进行归一化处理.

注意力向量各个维度度量了每条评论对用户偏好的不同贡献, 将每条评论的隐藏状态 h_i 进行加权求和, 得到用户评论集的整体表达 $\text{doc}_u \in \mathbb{R}^{1 \times 2l}$:

$$\text{doc}_u = \mathbf{a} \mathbf{H}_o \quad (6)$$

随后将 doc_u 送入全连接层, 得到用户 u 在评论数据中的最终偏好表达:

$$\text{TEXT_U} = \mathbf{W}' \times \text{doc}_u + b' \quad (7)$$

其中矩阵 \mathbf{W}' 是全连接层的权重参数, b' 是全连接层的偏倚项. 最后, 我们可以分别得到用户网络 Net_u 和商品网络 Net_i 的输出 TEXT_U , TEXT_I .

3.2.2 矩阵分解模块

由于 DeepCLFM 中的矩阵分解模块沿用了 LFM 的思路, 因此在介绍矩阵分解模块前, 首先简单介绍 LFM. LFM 是一种基于矩阵分解的算法, 其预测评分的方法如式(8):

$$\hat{r}_{ui} = \mathbf{q}_u \mathbf{p}_i^T + b_u + b_i + \mu \quad (8)$$

式(8)主要包含四部分: 全局偏倚项 μ 、用户偏倚项 b_u 、商品偏倚项 b_i 和用户-商品交互项 $\mathbf{q}_u \mathbf{p}_i^T$. 其中 \mathbf{q}_u 和 \mathbf{p}_i 是根据评分矩阵将用户和商品分别分解成两个 k 维隐向量得到, 它们分别表示用户偏好及商品特征. 如果用户 u 未知, 则 \mathbf{q}_u 和 b_u 为 0 (对于未知商品 i 同理).

从式(8)可以看出, LFM 实际上是根据分解得到的 \mathbf{q}_u 和 \mathbf{p}_i 通过内积运算进行组合, 然后回归出预测的评分. 但是本文 DeepCLFM 的回归部分主要由融合层模块 (详见 3.3 节) 负责, 因此在 DeepCLFM 的矩阵分解模块中, 只需要将评分矩阵中各个用户和各个商品分解成两个 k 维向量表达即可, 也就是将评分矩阵分解成两个隐向量矩阵来表示, 其具体如式(9):

$$\mathbf{R}_{m \times n} \approx \mathbf{U} \mathbf{I}^T \quad (9)$$

其中式(9)的评分矩阵 \mathbf{R} 表示 m 个用户对 n 个商品的评分, 它约等于用户隐向量矩阵 \mathbf{U} 和商品隐向量矩阵 \mathbf{I} 的乘积. 其中 \mathbf{U} 的每个用户都可以用一个 k 维向量 LFM_U_i 进行表示, \mathbf{I} 中的每个商品都可以用一个 k 维向量 LFM_I_j 表示, 具体如式(10)和(11)所示:

$$U_{m \times k} = \begin{Bmatrix} \mathbf{LFM-U}_1 \\ \vdots \\ \mathbf{LFM-U}_m \end{Bmatrix} \quad (10)$$

$$I_{n \times k} = \begin{Bmatrix} \mathbf{LFM-I}_1 \\ \vdots \\ \mathbf{LFM-I}_n \end{Bmatrix} \quad (11)$$

其中 $\mathbf{LFM-U}$ 和 $\mathbf{LFM-I}$ 可以通过融合层结合从文本处理模块得到的深度特征 $\mathbf{TEXT-U}$, $\mathbf{TEXT-I}$ 共同训练得到. 接下来将详细介绍融合层模块.

3.2.3 融合层模块

如图 1 所示, 融合层是 DeepCLFM 的第 3 个模块, 同时也是最重要的模块. 其主要负责将文本处理模块的 $\mathbf{TEXT-U}$, $\mathbf{TEXT-I}$ 与矩阵分解模块的 $\mathbf{LFM-U}$, $\mathbf{LFM-I}$ 进行不同层次、不同粒度的融合, 得到 3 个不同的组合特征向量: 一阶特征 ($\mathbf{U-First}$, $\mathbf{I-First}$)、二阶特征 ($\mathbf{U-Second}$, $\mathbf{I-Second}$) 和深度特征 (\mathbf{U} , \mathbf{I}). 最后通过线性回归出预测评分.

在 DeepCLFM 的计算过程中, $\mathbf{TEXT-U}$, $\mathbf{TEXT-I}$ 是使用预训练 BERT、双向 GRU 和注意力机制等深度学习方法从评论文本中得到, 它们是属于用户及商品在评论文本中体现的深度非线性特征向量. $\mathbf{LFM-U}$, $\mathbf{LFM-I}$ 是从评分矩阵中得到的, 它们是属于用户及商品在评分数据中体现的浅层线性特征向量. 我们受到因子分解机的启发, 类似地先将特征进行一阶组合和二阶组合, 再将一阶、二阶特征和深度特征进行线性组合回归出预测评分. 而且为了深层特征与浅层特征更好地融合并产生交互, $\mathbf{TEXT-U}$, $\mathbf{TEXT-I}$ 与 $\mathbf{LFM-U}$, $\mathbf{LFM-I}$ 都设置为 k 维向量. 由于用户和商品的特征计算过程相同, 因此下文仅详细介绍用户的一阶特征、二阶特征和深度特征的计算过程.

(1) 一阶特征

文本隐因子与评分隐因子最直接的融合策略为两向量对应维度相加, 这在相关工作中^[11,38]已表明能获得更好的推荐性能. 这里, 用户的一阶特征是将两种特征向量中对应维度的隐因子进行叠加, 具体的计算过程如式(12)所示:

$$\mathbf{U-First} = \mathbf{TEXT-U} \odot \mathbf{LFM-U} \quad (12)$$

其中 $\mathbf{U-First}$ 是一个 k 维向量, 由 $\mathbf{TEXT-U}$ 和 $\mathbf{LFM-U}$ 两个 k 维向量对应元素相加得到. 同理可得商品的一阶特征 $\mathbf{I-First}$.

(2) 二阶特征

在因子分解机 FM (Factorization Machines)^[39]

中, 二次项 $\sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j$ 可以转化为式(13a)来降低运算复杂度, 而且为了方便实现, 本文继续进行变换得到式(13c), 如式(13)所示:

$$\begin{aligned} & \sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j \\ &= \frac{1}{2} \times \sum_{j=1}^k \left(\left(\sum_{i=1}^n v_{ij} x_i \right)^2 - \sum_{i=1}^n (v_{ij} x_i)^2 \right) \quad (13a) \end{aligned}$$

$$\begin{aligned} &= \frac{1}{2} \times \text{sum} \left(\left[\text{multiply} \left(\sum_{i=1}^n v_i x_i, \sum_{i=1}^n v_i x_i \right) - \sum_{i=1}^n \text{multiply}(v_i x_i, v_i x_i) \right] \right) \quad (13b) \end{aligned}$$

$$\begin{aligned} &= \text{sum} \left(\frac{1}{2} \times \left[\text{multiply} \left(\sum_{i=1}^n v_i x_i, \sum_{i=1}^n v_i x_i \right) - \sum_{i=1}^n \text{multiply}(v_i x_i, v_i x_i) \right] \right) \quad (13c) \end{aligned}$$

其中 $\text{multiply}()$ 和 $\text{sum}()$ 运算参见 3.1 节的表 1.

从式(13a)的可知, 因子分解机是将特征向量的每个维度特征 x_i 转化为一个 k' 维向量 $v_i x_i$ 去表达 (其中 $v_i \in \mathbb{R}^{k'}$), 然后计算原特征向量中任意两维特征之间的交互. 沿用该思想, DeepCLFM 首先将用户的两种特征 $\mathbf{TEXT-U}$ 和 $\mathbf{LFM-U}$ 拼接成一个向量 $\mathbf{z} = \mathbf{TEXT-U} \oplus \mathbf{LFM-U}$, 然后借鉴因子分解机来计算用户的二次项特征 $\mathbf{U-Second}$:

$$\begin{aligned} \mathbf{U-Second} &= \frac{1}{2} \times \left[\text{multiply} \left(\sum_{i=1}^n v_i z_i, \sum_{i=1}^n v_i z_i \right) - \sum_{i=1}^n \text{multiply}(v_i z_i, v_i z_i) \right] \quad (14) \end{aligned}$$

由于 $\mathbf{U-Second}$ 进行 sum 运算后就是标准的二次项结果 (见式(13c)), 因此 $\mathbf{U-Second}$ 已经包含了二次项结果的全部信息, 在这种向量形式二次项 $\mathbf{U-Second}$ 的基础上, 能更好地通过全连接层将 $\mathbf{U-First}$, $\mathbf{U-Second}$ 中各隐因子进行线性组合, 充分发挥深度学习的性能来挖掘一次项特征、二次项特征和深度特征之间的内在关联. 注意, $\mathbf{U-Second} \in \mathbb{R}^{k'}$, k' 是可以设置为任意值的参数, 同理可以得到 $\mathbf{I-Second} \in \mathbb{R}^{k'}$.

值得一提的是, DeepCLFM 和 DeepCoNN 在计算二次项时都借助了因子分解机技术, 但是 DeepCoNN 是将用户 \mathbf{U} 和商品 \mathbf{I} 这两种特征向量拼接成一个向量送入因子分解机, 来产生用户与商品之间的二阶交互来预测评分. 由于 DeepCoNN 仅仅使用了评论数据, 用户隐向量 \mathbf{U} 和商品隐向量 \mathbf{I} 是同源的 (即 \mathbf{U} 仅来自于 $\mathbf{TEXT-U}$, \mathbf{I} 仅来自于 $\mathbf{TEXT-I}$),

因此该做法能很好地使得用户和商品产生交互。但是 DeepCLFM 处理的是同时含有同源和非同源数据的情景, 即 \mathbf{U} 来自于 $\mathbf{TEXT_U}$ 和 $\mathbf{LFM_U}$ 、 \mathbf{I} 来自于 $\mathbf{TEXT_I}$ 和 $\mathbf{LFM_I}$ 。在面面对该情景下, 本文认为该做法会使得含有非同源特征的用户隐向量 \mathbf{U} 和商品隐向量 \mathbf{I} 的学习互相干扰, 限制二者表达质量的提高。相反, DeepCLFM 仅计算用户(商品)自身特征向量中的任意二维特征的二阶交互, 保证了用户(商品)的表达质量, 比 DeepCoNN 这种方式取得了更好的结果(具体分析见 4.7 节)。

(3) 深度特征

用户的深度特征 \mathbf{U} 同时包含了评论文本和评分数据中用户的偏好信息, 它可以通过拼接一、二阶特征向量得到:

$$\mathbf{U} = \mathbf{U_First} \oplus \mathbf{U_Second} \quad (15)$$

其中 \mathbf{U} 是一个 $k + k'$ 维向量, 同理得到商品的深度特征 \mathbf{I} 。

(4) 评分预测

假设现在要预测用户 u 对商品 i 的评分, 根据上文的计算过程可以得到用户深度特征 \mathbf{U} 和商品深度特征 \mathbf{I} , 在融合层将会通过全连接层将它们进行线性组合, 最终预测的评分 $Y_predict$ 为

$$Y_Predict = \mathbf{W} \times \text{multiply}(\mathbf{U}, \mathbf{I}) + b_u + b_i + \mu \quad (16)$$

其中矩阵 \mathbf{W} 是全连接层的权重参数, μ 是全局偏倚项, b_u 是用户偏倚项, b_i 是商品偏倚项。显然, 将 \mathbf{W} 固定为 1, 式(16)可以还原为标准的 LFM(见式(8))。

如果用户 u 未知(冷启动情况), 则 b_u 、 \mathbf{U} 都为 0, 这时 $Y_predict$ 仅依靠全局偏倚 μ 、商品偏倚 b_i 得到(商品 i 同理)。实验证明(见 4.4 节), 通过全连接层得到的 $Y_predict$ 能够充分利用评论和评分数据的浅层隐因子、深度隐因子, 并且能使这些隐因子充分产生交互, 更好地降低预测评分的误差。这种通过评论文本来指导偏移项的学习方式, 能有效应对用户(商品)冷启动的情况, 具体实验分析详见 5.3 节。

3.3 模型学习

实际上通过对用户和商品建模来进行评分预测是回归问题, 常用的目标函数为平方损失函数^[40-42]:

$$L = \sum_{u, i \in \Omega} (\hat{r}_{ui} - r_{ui})^2 \quad (17)$$

其中 Ω 表示训练集中的样本, \hat{r}_{ui} 是用户 u 对商品 i 的预测评分, r_{ui} 是训练集中的真实评分。

为了最小化目标函数, 我们采用 Adam(Adaptive Moment Estimation)^[43] 作为优化器进行优化。其主

要优势是能够在训练过程中自己调整恰当的学习率, 可以省去人工选择学习率的步骤, 使得模型能够更快的收敛。

3.4 模型复杂度

由于 DeepCLFM 主要由文本处理模块、矩阵分解模块、融合层模块组成, 因此本小节将从该三部分进行叙述。

假设数据集共含有 n_u 个用户, n_i 个商品, 模型的隐因子数量为 k 。对于每个样本 (u, i, r_{ui}, w_{ui}) , 用户 u 和商品 i 对应的 one-hot 编码分别为 $oh_u \in \mathbb{R}^{n_u}$ 和 $oh_i \in \mathbb{R}^{n_i}$ 。用户映射成隐向量的过程实际为 oh_u 与映射矩阵 $\mathbf{W}_{LFM_U} \in \mathbb{R}^{n_u \times k}$ 相乘, 共需要进行 $n_u k$ 次运算。因此, 矩阵分解模块的空间复杂度为 $\mathcal{O}(k(n_u + n_i))$, 时间复杂度为 $\mathcal{O}(k(n_u + n_i))$ 。

在文本处理模块, 用户 u 评论集(评论最多保留 d_u 条)、商品 i 评论集(评论最多保留 d_i 条)经过 BERT 映射后, 每条评论映射为 c 维的向量表达。Net_u 的参数主要由双向 GRU 单元、注意力层和全连接层构成, 其中一个 GRU 单元的参数为 $\mathcal{O}(3(c+l+1)l)$, 注意力层的参数为 $\mathcal{O}(2lt+t)$, 全连接层的参数为 $\mathcal{O}(2l+1)$, 因此 Net_u 的空间复杂度为 $\mathcal{O}(6(c+l+1)l+2lt+t+2l+1)$ (与 Net_i 参数相同), Net_u 的时间复杂度为 $\mathcal{O}(d_u(6(c+l)l+2ltd+2lk))$, 同理将 d_u 替换成 d_i 可以得到 Net_i 的时间复杂度。

对于融合层模块, 其参数主要由两个因子分解机和评分预测的全连接层参数所构成, 因子分解机的空间复杂度为 $\mathcal{O}(2(2k \times k'))$, 全连接层的空间复杂度为 $\mathcal{O}(k+k'+n_u+n_i+1)$ 。融合层的时间复杂度主要由二阶项的交互运算和评分预测全连接层构成 $\mathcal{O}(2kk'+k+k')$ 。

4 实验

4.1 数据集与评价指标

为了验证 DeepCLFM 的有效性, 我们使用亚马逊评论公共数据集 Amazon 5-core 中的 5 个子类别数据集 Toys-and-Games、Movies-and-TV、Kindle-Store、Instant-Video、Digital-Music 进行实验, 下文简称为 TG、MV、KS、IV、DM。这些数据集具有不同的主题和大小, 其中 Movies-and-TV 是最大型的数据集(共包含大约 160 万条评论), 而 Instant-Video 是最小的数据集(共包含 37 000 条评论)。

在这些数据集中, 对于每个样本, 本文使用到的特征共有 4 个: 用户编号、商品编号、用户对商品的

打分(1~5 分)、用户对商品的评论文本. 其中基本的数据统计信息如表 2 所示.

表 2 数据集基本信息				
	用户数	商品数	样本数	稀疏度/%
TG	19412	11924	167597	92.75
MV	123960	50052	1697533	97.26
KS	68223	61934	982619	97.67
IV	5130	1685	37126	99.57
DM	5541	3568	64706	99.67
平均	111133	25832	589916	97.38

如表 2 所示,上述 5 个数据集虽然都是亚马逊评论数据集的子数据集,但是平均的样本数为 589916,平均的用户数和商品数分别为 111133 和 25832,因此它们都属于大型数据集,拥有足够多的样本来验证模型性能.

尽管用户和商品的总数量巨大,但是实际上每个用户购买过的商品相对于商品总数只是很小的一部分,因此我们对商品的被评分数量进行了统计,如表 3 所示.

表 3 商品评分统计信息			
	≤20	≤15	≤10
TG	0.817	0.730	0.554
MV	0.644	0.549	0.385
KS	0.798	0.702	0.506
IV	0.734	0.645	0.456
DM	0.738	0.645	0.472
平均	0.746	0.6542	0.474

总体来看,平均 74.6%的商品评分数量少于 20 个,平均 65.4%的商品评分数量少于 15 个,平均 47.4%的商品评分少于 10 个,表明这 5 个数据集的评分数据是极其稀疏的,这将会影响到矩阵分解模型对预测评分的精度. 在以往的研究中,通过使用数据初始化填充、基于内容或商品的推荐等方式来缓解该问题对推荐性能的影响. 但是本文主要是通过结合评论文本的数据来缓解数据稀疏,因此对商品评论中的单词数进行了统计,如表 4 所示.

表 4 评论数据统计信息						
	≥70	≥90	R-p-U	W-p-U	R-p-I	W-p-I
TG	0.81	0.67	8.63	875.99	14.05	1426.09
MV	0.82	0.69	13.69	2259.92	33.91	5596.96
KS	0.84	0.72	14.40	1616.22	15.86	1780.33
IV	0.85	0.74	7.23	673.38	22.03	2050.12
DM	0.85	0.74	11.67	2374.29	18.14	3687.21
平均	0.83	0.71	11.12	1559.96	20.79	2908.14

其中,R-p-U(Reviews per User)表示平均每用户发表的评论数,W-p-U(Word per User)表示每用户多

条评论所包含的单词的平均数. 同理,R-p-I(Reviews per Item)表示平均每商品的评论数,W-p-I(Word per Item)表示每商品多条评论所包含的单词的平均数. 从表 4 看出,平均 83%的商品评论文本的单词数大于 70 个,平均 71%的商品评论文本的单词数大于 90 个. 同时 R-p-U、W-p-U、R-p-I 和 W-p-I 等指标表明,大部分用户和商品具有一定长度的评论文本,即评论文本中存在足够多的信息来指导建模. 下文的实验结果表明,通过融入评论文本信息,能够有效缓解评分数据的稀疏性且提高评分预测精度.

在下文的实验中,我们采用均方误差(MSE)作为模型性能的评价指标,这是一个在相关文献[6-8]中被广泛使用的指标. MSE 的值越低表示模型的性能越好. 给定一个用户 u 对商品 i 的预测评分 \hat{r}_{ui} 和真实评分 r_{ui} ,则 MSE 的定义如下:

$$MSE = \frac{1}{N} \sum_{u,i} (\hat{r}_{u,i} - r_{u,i})^2$$

(18)

其中 N 表示测试集中样本的个数.

4.2 对比模型

本文开展对比实验的目的有两个:(1)验证结合评论文本和评分矩阵是否能够缓解数据稀疏给推荐性能带来的影响;(2)相比同样结合评论文本的算法,验证本文提出的 DeepCLFM 是否能够通过融合不同粒度的深层特征进一步降低预测误差. 为了达到上述两个实验目的,我们将 DeepCLFM 与 7 个目前较为先进的模型进行对比,也就是 NMF、LFM、SVD++、HFT、DTMF、DeepCoNN 和 NARRE.

如表 5 所示,前 3 个模型 NMF、LFM、SVD++ 在参数学习阶段仅仅使用了评分矩阵,而后面的 HFT、DTMF 是同时使用了评分矩阵和评论文本的杰出模型. 接下来一个模型是仅仅使用了评论文本的深度学习模型 DeepCoNN. 最后的一类模型是本文提出的 DeepCLFM 和 NARRE,它们是同时使用评分矩阵和评论文本作为输入的深度学习模型.

表 5 模型的比较			
	Rating Matrix	Textual Reviews	Deep Learning
NMF	✓	\	\
LFM	✓	\	\
SVD++	✓	\	\
HFT	✓	✓	\
DTMF	✓	✓	\
DeepCoNN	\	✓	✓
NARRE	✓	✓	✓
DeepCLFM	✓	✓	✓

(1) NMF^[15]. 非负矩阵分解模型, 其输入仅仅用到了评分矩阵.

(2) LFM^[5]. 隐语义模型 (Latent Factor Model), 通过矩阵分解 (SVD) 对未知商品预测评分, 是最经典的矩阵分解算法.

(3) SVD++^[16]. 基于 LFM 的改进算法, 在 SVD 的基础上引入隐式反馈, 使用用户的历史数据来进一步提高模型预测精度.

(4) HFT^[7]. 结合评分数据与评论文本最经典的模型. 其将用户评论集或商品评论集作为其输入, 将评论中的主题与矩阵分解中的隐因子进行融合. 根据文献中的报告, 使用商品评论集作为输入会比用户评论集效果要好, 因此在实验中选择使用商品评论集作为输入, 也就是使用模型 HFT(item).

(5) DTMF^[17]. 基于 HFT 的改进模型, 在将评论主题与矩阵分解隐因子相结合的基础上, 能够同时将用户评论集和商品评论集作为模型的输入.

(6) DeepCoNN^[8]. 深度协同神经网络模型 (Deep Cooperative Neural Networks) 是首个同时结合用户评论集和商品评论集的深度学习方法, 其性能优越.

(7) NARRE^[11]. 该模型在 DeepCoNN 的基础上引入注意力机制来识别无用的评论. 由于注意力机制能将评论数据和评分矩阵数据有效结合, 其评分预测性能得到一定提高.

(8) DeepCLFM. 本文提出的模型 DeepCLFM, 能够同时利用评分矩阵和评论文本, 而且通过融合一、二阶特征的方式使得用户和商品的隐因子产生交互, 使得性能进一步提升.

4.3 实验细节

我们随机将实验数据集划分为训练集 (80%)、验证集 (10%)、测试集 (10%), 在验证集上进行超参数选取, 最终使用测试集进行性能评估. 另外, 由于

Movies-and-TV 过大, 我们随机选取一部分进行多次实验取平均值.

其他对比算法根据对应的论文进行参数初始化, 并微调使其达到最佳性能. 对于矩阵分解算法 NMF、LFM、SVD++, 我们使用格子搜索法从 {8, 16, 32, 64} 寻找最佳的隐因子个数, 从 {0.001, 0.01, 0.1, 1.0} 寻找最佳的正则化系数. 根据文献 [17] 的报告, 基于主题模型的 HFT、DTMF 随着主题数 K 的增大, 算法运算速度持续下降, 但是其效果相差不大, 因此我们参照该文献将主题个数 K 都设置为 5.

对于深度学习模型 DeepCoNN、NARRE、DeepCLFM, 其超参数众多, 针对不同的数据集其最佳表现的对应参数都不同, 因此我们进行了参数灵敏度分析实验 (详见 4.5 节). 注意, 本文实验的隐因子数指的是 $TEXT-U$ 、 $TEXT-I$ 、 $LFM-U$ 和 $LFM-I$ 的维度, 它们的维度都为 k . 在实验过程中, DeepCoNN 和 NARRE 使用 glove 预训练的 300 维词向量; DeepCLFM 使用 BERT 的预训练模型版本为 uncased_L-12_H-768_A-12. 如果没有特别说明, CNN 组件的卷积核大小为 3、个数为 50. 其中 DeepCLFM 的 GRU 隐藏单元数 l 为 200, 注意力权重向量 α 的维度 t 为 400, 因子分解机参数 v_i 的维度 k' 为 6. 同时为了实验的公平性, 本文参照 NARRE 的实验设置, 使得每个用户评论集的文本截取长度覆盖 90% 的用户 (即将每个用户对应的文本长度从小到大排序, 对应 90% 分位点的用户评论文本长度作为当前数据集的用户评论集文本截取长度). 商品评论集的文本截取长度同理. 每个用户评论集最多包含 10 条评论, 每个商品评论集最多包含 20 条评论.

4.4 性能评估

本文提出的 DeepCLFM 和其他对比模型的评分预测结果如表 6 所示, 分析实验结果我们有如下结论.

表 6 结果对比

	a: NMF	b: LFM	c: SVD++	d: HFT	e: DTMF	f: DeepCoNN	g: NARRE	h: DeepCLFM	提升/% h vs. e	提升/% h vs. f	提升/% h vs. g
Toys-and-Games	1.429	0.807	0.802	0.801	0.813	0.796	0.769	0.751	7.626	5.653	2.341
Movies-and-TV	1.415	1.281	1.273	1.211	1.103	1.019	0.998	0.981	11.06	3.729	1.703
Kindle-Store	0.995	0.623	0.616	0.622	0.619	0.614	0.611	0.602	2.746	1.954	1.472
Instant-Video	1.205	0.949	0.943	0.951	0.934	0.930	0.916	0.901	3.533	3.118	1.637
Digital-Music	1.228	0.851	0.837	1.049	0.826	0.806	0.796	0.787	3.631	2.357	1.130
平均	1.254	0.902	0.894	0.926	0.859	0.833	0.818	0.804	6.402	3.362	1.656

(1) 考虑评论文本的模型 (HFT, DTMF, DeepCoNN, NARRE 和 DeepCLFM) 一般会比仅仅考虑评分数据的传统协同过滤模型 (NMF, LFM, SVD++) 表现更佳. 这与我们的直观理解相符, 评论信息可以

看作是评分数据的补充, 它能够提高隐因子的表达质量.

(2) 在同样考虑评论文本的模型中, 深度学习模型 (DeepCoNN、NARRE、DeepCLFM) 的性能会

比传统模型(HFT、DTMF)更优. 我们认为当中的原因有 3 点: ①传统模型处理文本数据的方式大都是基于主题模型 LDA, 而之前的工作^[8,34]已经证明深度学习技术在分析文本方面比主题模型 LDA 更优; ②传统模型的局限性在于其仅仅学习线性特征, 而深度学习模型能以非线性的方式来对用户和商品建模; ③一些深度学习如 dropout, batch normalization 等技巧能有效抑制过拟合, 进一步挖掘模型的潜能.

(3) 如表 6 所示, 本文提出的方法 DeepCLFM 是同时结合评分矩阵与评论文本的深度学习模型, 其平均预测误差相比 DTMF 和 DeepCoNN 分别提升了 6.402% 和 3.362%. 而且最重要的是, 相比仅仅使用评论文本的深度学习模型 DeepCoNN, DeepCLFM 和 NARRE 通过添加评分数据的用户和商品隐因子, 能更深入地挖掘数据来得到更好的表现. 这表明基于深度学习, 同时考虑评论文本和评分矩阵进行建模是一个值得研究的方向. 从结果可以看出, 尽管 NARRE 取得不错的效果, 但是其仍使用 word2vec 这种静态词向量结合 CNN 结构. 静态词向量不能准确表达同一单词在不同上下文环境的含义, CNN 结构不能有效捕获评论集中的长距离特征. 相反, DeepCLFM 采用预训练的 BERT 模型来应对静态词向量的问题, 并结合双向 GRU 组件来对评论集中各条评论进行信息概括. 实验表明, DeepCLFM 取得了比 NARRE 更优的性能.

4.5 参数灵敏度分析

在本小节, 我们展示不同参数不同模型在验证集上的表现. 由于篇幅有限, 我们仅仅展示模型在数据集 Toys_and_Games 和 Instant_Video 上的表现. 下文首先探索隐因子数量对模型的影响. 对于主题模型(HFT, DTMF), 其主题数量等价于隐因子数量. 对于矩阵分解模型和深度模型, 隐因子数量就是用户和商品的隐向量长度. 根据文献^[17]报告, HFT 和 DTMF 对于不同主题数的性能变化不大, 且 NMF 表现糟糕, 因此下文仅对 LFM, SVD++, DeepCoNN, NARRE, DeepCLFM 进行实验. 如图 3 所示, 本文提出的 DeepCLFM 能同时结合评分矩阵和评论文本中的用户(商品)隐因子进行建模, 在不同的隐因子数下都拥有优秀的表现. 另外, 随着隐因子数的上升, 矩阵分解模型 LFM 和 SVD++ 的误差逐步上升, 我们认为这是由于参数过多而导致的过拟合现象. 这里我们可以得出结论, 随着参数

的增多, 基于矩阵分解的传统模型比基于深度学习模型更容易出现过拟合现象.

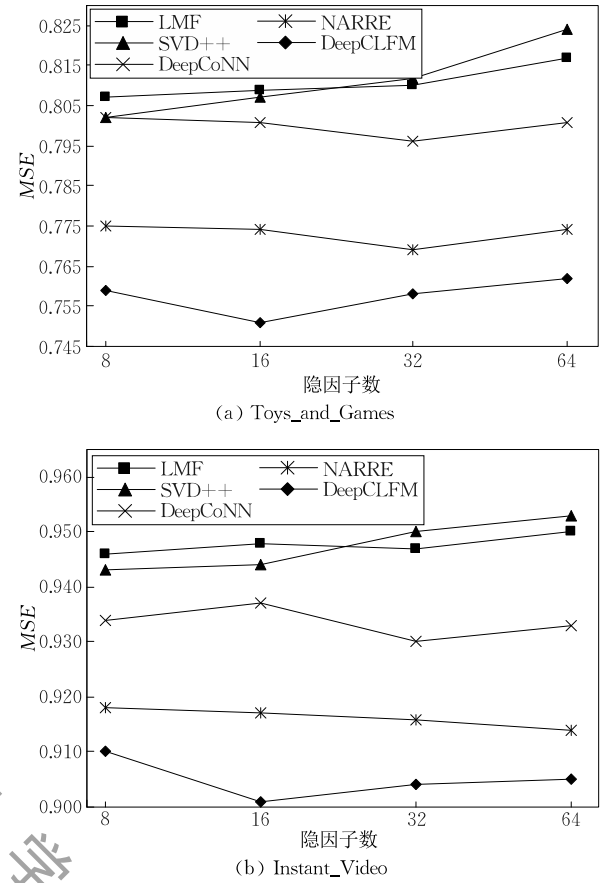


图 3 隐因子数量对模型性能的影响

为了防止过拟合, 在深度学习中可以通过 dropout 技术来解决. 我们在图 4 中展示了不同 dropout 比率对模型性能的影响. 从结果可知, dropout 比率设置为恰当的值能使模型的性能得到提升, 这是因为数据过拟合得到缓解, 模型得到更良好的泛化性能. 经过对比 DeepCoNN、NARRE、DeepCLFM 在两个数据集的表现, 发现 3 个模型在 Toys_and_Games 的结果要比在 Instant_Video 上要稳定, 我们认为这是因为 Toys_and_Games 的语义信息没有那么复杂(从表 4 可知, Toys_and_Games 的评论文本长度整体要比 Instant_Video 更短). 另外, 3 个模型在 Instant_Video 上的表现都不稳定, 这是因为 Instant_Video 数据样本数少(从表 2 可知), 不足以支撑模型学习出性能稳定的参数. 而且随着 dropout 比率的上升, DeepCLFM、NARRE 的性能大幅下降(MSE 持续上升), 这符合我们之前的结论: 由于 DeepCLFM、NARRE 引入矩阵分解模型的做法来处理评分矩阵数据, 具有矩阵分解模型容易产生过拟合的特性.

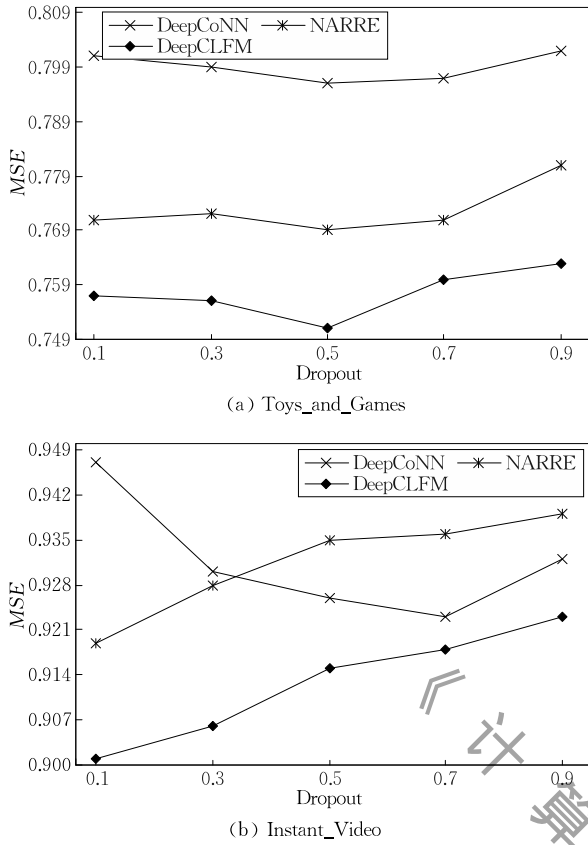


图 4 dropout 比率对模型性能的影响

4.6 文本处理模块结构分析

近年来基于评论文本进行评分预测的研究大多是使用自然语言处理领域相对老旧的技术:静态词向量结合 CNN,特别是以 DeepCoNN 和 NARRE 为代表的模型.而本文则采用目前最先进的预处理技术(BERT),并结合双向 GRU 和自注意力机制来提取评论集特征.在本小节,为了探究 DeepCLFM 文本处理模块对模型性能的影响,分别对文本处理模块进行不同变换,得到以下变体模型并进行对比实验:

- (1) Word-CNN. 使用预训练的词向量并结合 CNN 做评论集的特征抽取.
- (2) Word-GRU. 使用预训练的词向量并结合双向 GRU 做评论集的特征抽取.
- (3) BERT-CNN. 使用 BERT 提取每条评论的特征,然后使用 CNN 做评论集的特征抽取.
- (4) BERT-GRU. 使用 BERT 提取每条评论的特征,然后使用双向 GRU 做评论集的特征抽取.但是相比 DeepCLFM,取消自注意力机制,将每条评论平等对待,即式(6)的 $\alpha=1$.
- (5) BERT-NARRE. 使用 BERT 得到每条评论的特征,并仿照 NARRE 的做法在学习商品评论集

的文档特征时,添加用户隐因子.即将上文的式(5)转化为

$$\mathbf{a} = \text{softmax}(\mathbf{w}_1 \times \tanh(\mathbf{w}_2 \times \mathbf{H}_o^T + \mathbf{w}_3 \mathbf{u}^T)).$$

需要注意的是,式(5)描述的是用户评论集网络,这里描述的是商品评论集网络.因而, $\mathbf{H}_o \in \mathbb{R}^{d \times 2l}$ 表示某一商品的评论集的 d 条评论经过双向 GRU 的输出. $\mathbf{u} \in \mathbb{R}^{d \times k}$ 表示发表该 d 条评论对应用户编号的隐向量矩阵,每个用户的隐向量维度是 k .其余的 $\mathbf{w}_1 \in \mathbb{R}^{1 \times t}$, $\mathbf{w}_2 \in \mathbb{R}^{t \times 2l}$, $\mathbf{w}_3 \in \mathbb{R}^{t \times k}$ 为模型参数.

(6) DeepCLFM. 使用 BERT 提取每条评论的特征,基于双向 GRU 捕获的评论间的联系,结合注意力机制区别对待每条评论.

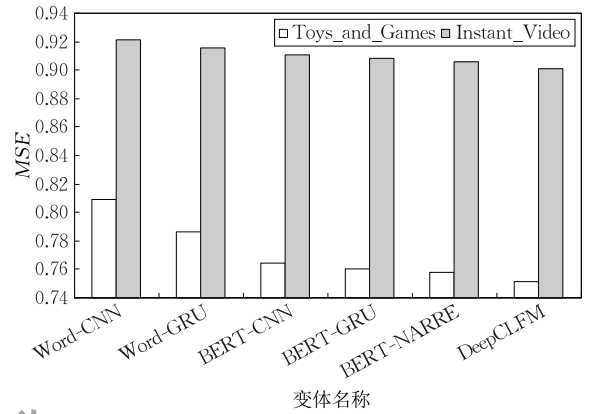


图 5 文本处理模块结构分析

文本处理模块变体性能的对比如图 5 所示.将基于预训练词向量的模型(Word-CNN、Word-GRU)和基于 BERT 的模型(BERT-CNN、BERT-GRU、BERT-NARRE)进行比较分析得出:使用预训练词向量的模型的性能进一步提高,主要受多义词的影响,其不能正确理解同一单词在不同语义环境下的具体含义.而使用 BERT 的模型则解决了该问题,其能基于更准确的语义理解来映射出更优质的评论表达,使性能得到进一步提升.在使用相同的预训练模型下(同样使用预训练词向量或 BERT),采用双向 GRU 的模型总体比采用 CNN 的模型性能更佳,这是因为:处理由多个评论组成的评论集实际上属于处理序列问题,RNN 的递归结构天然比 CNN 更擅长捕获序列信息,且在双向结构的作用下,其能更有效地学习评论文本之间的联系.然而,并不是每条评论都能有效刻画用户(或商品)的特点,因此需要一个权值来分别刻画每条评论的贡献度. BERT-NARRE 和 DeepCLFM 使用注意力机制区别对待每条评论,整体效果比 BERT-GRU 更优.但是 BERT-NARRE 在计算注意力向量时添加用

户隐因子 ($LFM-U$), 使得学习商品评论集特征时 ($TEXT-I$), 过早地与用户隐因子产生了交互, 影响了商品评论文本特征的学习. 相反, DeepCLFM 中学习 $TEXT-I$ 和 $LFM-U$ 的过程相对独立, 它们二者的交互仅发生在融合层模块, 保证了它们的表达质量, 因而取得了最好的效果.

4.7 融合层模块结构分析

在本节中, 我们将会在数据集 Toys_and_Games 和 Instant_Video 上对 DeepCLFM 融合层模块的各种变体进行对比实验, 验证分析 DeepCLFM 融合层结构的合理性. 融合层涉及处理分别来自评论数据和评分数据的特征 (本文称之为非同源特征). 更具体地, 同源特征指的是 $\{TEXT-U$ 和 $TEXT-I\}$ 、 $\{LFM-U$ 和 $LFM-I\}$. 非同源特征指的是 $\{TEXT-U$ 和 $\{LFM-U, LFM-I\}\}$ 、 $\{TEXT-I$ 和 $\{LFM-U, LFM-I\}\}$.

通过修改融合层的结构, 可以得到如下对比模型:

(1) $U \times I$. 式 (16) 是将用户和商品的深度特征 U 和 I 进行对应元素相乘后经过全连接层进行回归. 这里模仿 LFM , 直接将 U 和 I 进行点积运算得出预测评分, 即将式 (16) 的 W 固定为 1.

(2) Fully-Connected. 为了验证融合层以一二阶特征的方式来处理从评论文本和评分矩阵得到的用户 (商品) 特征的合理性, 这里将融合层简单地以全连接层进行替代, 即将各种同源和非同源的 $TEXT-U, TEXT-I, LFM-U, LFM-I$ 拼接成一个向量 e , 最后预测的评分为 $Y_{Predict} = Fully_Connected(e)$.

(3) FM. 该变体采取 DeepCoNN 中因子分解机 FM 的融合方式, 直接将各种同源和非同源的 $TEXT-U, TEXT-I, LFM-U, LFM-I$ 送入 FM, 即 $Y_{Predict} = FM(e)$.

(4) AFM. AFM (Attentional Factorization Machines) 是增加了注意力机制的 FM, 其能对 FM 中不同的交互项 (interaction) 赋予不同的权值^[42]. 该变体将各种同源和非同源的 $TEXT-U, TEXT-I, LFM-U, LFM-I$ 送入 AFM, 即 $Y_{Predict} = AFM(e)$.

(5) DeepCLFM-1. 标准版 DeepCLFM, 详见第 3 节. 先将用户 (商品) 隐因子进行一阶和二阶特征处理, 然后通过全连接层线性变换得到预测评分.

DeepCLFM 和其各种变体在数据集 Toys_and_Games 和 Instant_Video 的性能如图 6 所示.

第一种变体 $U \times I$ 借用 LFM 的思想直接将非同源的隐因子向量进行点积运算, 但实验结果证明点积并不能有效处理非同源隐因子. 相比 $U \times I$,

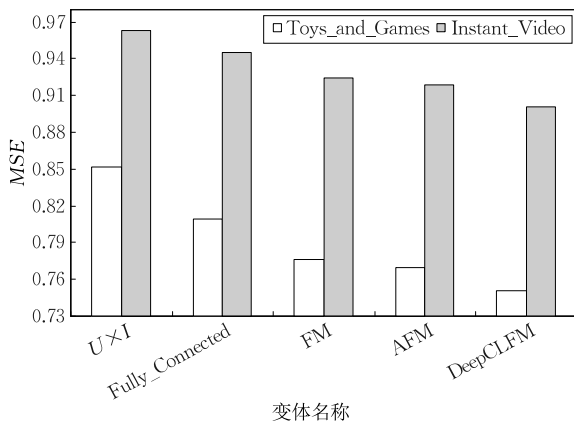


图 6 融合层结构比较

DeepCLFM 简单地将非同源隐因子进行线性变换就能得到较好的结果. 因此对于非同源隐因子, 使用全连接进行线性变换比点积运算更优.

变体 Fully-Connected 相当于省略了 DeepCLFM 融合层中的一二阶特征处理过程, 直接将从未处理过的各种同源和非同源特征 $TEXT-U, TEXT-I, LFM-U, LFM-I$ 进行回归预测评分, 其忽略了隐藏在二阶数据特征中的内在结构, 效果不如 DeepCLFM. 由于 DeepCoNN 在处理同样来自评论的用户和商品隐因子时, 使用因子分解机取得了较好效果. 因此, 变体 FM 借鉴 DeepCoNN 将各种同源和非同源特征拼接成一个向量 e , 利用因子分解机将用户-用户 ($TEXT-U-LFM-U$)、商品-商品 ($TEXT-I-LFM-I$)、用户-商品 ($TEXT-U-TEXT-I, TEXT-U-LFM-I, TEXT-I-LFM-U, LFM-U-LFM-I$) 之间的二阶交互都考虑在内. 实验表明, 变体 FM 比变体 Fully-Connected 取得更好的结果, 表明二阶交互有利于隐因子质量的提高.

为了更直观地探索同源和非同源二阶项的贡献, 下文使用变体 AFM 对 6 种组合的二阶项进行注意力打分. 由于隐因子数量 k 为 16 时模型取得最好效果, 因此向量 $TEXT-U, TEXT-I, LFM-U, LFM-I$ 的维度都设置为 16, 由它们拼接而成的向量 e 的维度为 64, 即 $e = [TEXT-U, TEXT-I, LFM-U, LFM-I] = [x_1 - x_{16}, x_{17} - x_{32}, x_{33} - x_{48}, x_{49} - x_{64}] = [C1, C2, C3, C4]$, 其中 $x_1 - x_{16}$ 被本文定义为类别 $C1$, $x_{17} - x_{32}$ 以此类推. e 中的二阶组合共有 6 种, 具体信息如表 7 所示. 从 $C1$ 和 $C2$ 中各随机选取一个维度就能形成一个二阶项 (例如 $x_3 x_{25}$), 重复选出 10 个不同的二阶项, 则能得到 $C1-C2$ 类型的 10 个二阶项. 从 6 种组合中都分别随机选出 10 个二阶项, 它们的注意力得分如图 7 所示.

表 7 二阶交互类型

组合类型	特征	数据源	交互对象
C1-C2	TEXT-U-TEXT-I	同源	用户-商品
C1-C3	TEXT-U-LFM-U	非同源	用户-用户
C1-C4	TEXT-U-LFM-I	非同源	用户-商品
C2-C3	TEXT-I-LFM-U	非同源	用户-商品
C2-C4	TEXT-I-LFM-I	非同源	商品-商品
C3-C4	LFM-U-LFM-I	同源	用户-商品

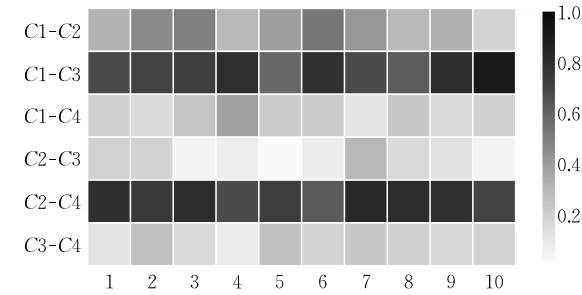


图 7 二阶项的注意力得分

图 7 中颜色越浅,表示注意力得分越低,对模型的贡献越少,反之亦然.从图中可以看出,C1-C4 和 C2-C3 的注意力得分稳定在 0~0.2 之间,表明非同源的用户-商品的二阶交互项对模型产生消极影响.另一类组合是 C1-C2 和 C3-C4,它们分别被 DeepCoNN 和 LFM 所采用并取得不错的效果,但是它们的注意力得分分布在 0.2~0.5 之间,表明在当前同时含有同源和非同源隐因子的情况下,同源的用户-商品的二阶组合对模型未能产生积极的影响.相反,C1-C3 和 C2-C4 的注意力得分最高,分布在 0.7~0.9 之间,表明尽管它们是非同源,但是用户-用户和商品-商品的二阶交互是最为重要的,其能对模型性能产生积极效果.

综上所述,能区分不同二阶组合贡献度的 AFM 性能比 FM 更优.然而,AFM 将 6 种二阶组合都考虑在内,实际上只有用户-用户(C1-C3)和商品-商品(C2-C4)的二阶交互能产生积极效果.因此,仅考虑用户-用户和商品-商品二阶交互项的 DeepCLFM,能排除其余二阶组合的干扰,取得比 AFM 更好的效果.通过比较变体 *Full-Connected*、FM、AFM,我们可以得出结论:在处理同时含有同源和非同源隐因子时,DeepCLFM 基于用户-用户、商品-商品的交互项构建深度特征,并使用特制的全连接层来预测评分的方式是可取的(见式(16)),其取得了最好的结果.

5 相关讨论

5.1 词序信息

在评分预测领域中,使用评论文本对用户或商

品建模的经典推荐算法大都是基于主题模型^[7,17].但是主题模型的局限性在于其基于词袋模型的假设而忽略了词序信息,而且文献[44]证明文本处理中词序信息是非常关键的.相反,DeepCLFM 采用 BERT 模型对评论进行预处理提取特征,BERT 的特征表示由前向和后向的上下文信息得到,充分保留了词序信息.

5.2 在线学习

在线学习中,可扩展性和如何处理商品种类和用户数量的动态变化被认为是许多推荐系统的核心需求.另外,推荐系统的时间敏感性要求对于以在线学习的方式来获得隐因子的效率提出了新的挑战.相比其它基于主题模型的方法,基于神经网络的 DeepCLFM 对训练数据的规模具有可伸缩性,可以很容易地接受新数据,进行快速地训练和更新参数,并能独立于历史数据来更新商品或用户的隐因子.

5.3 冷启动问题

冷启动问题广泛存在于推荐系统中,特别是当新用户或新商品加入到系统中时,其可用的信息极其有限,仅仅基于评分数据的推荐系统难以构建合适的隐因子进行准确的评分预测.特别是对于 LFM,当有新用户(商品)上线时,其用户隐向量 q_u 和用户偏倚项 b_u 都为 0(见式(8)),其评分预测仅依靠历史数据训练得到的全局偏倚项 μ .但是 DeepCLFM 能将评论文本的信息作为全局偏倚项 μ 的补充,能更有效地应对冷启动情景.

为了展示 DeepCLFM 应对用户(商品)冷启动的能力,首先需要构建模拟用户(商品)刚上线时的冷启动数据集.在用户冷启动情景(商品冷启动同理)下,随着用户购买了越来越多的商品,其产生了交互(已评论、评分)的商品数越来越多.为此,本文将每个用户产生过交互的商品数分别限制为 1~5 个,即每个用户可训练的样本数为 1~5 个.本小节在不同规模的训练数据(也就是对于每个用户(商品)可用作训练的样本数, x 轴)下,对 DeepCLFM 和 LFM 进行对比实验,实验结果如图 8 所示.需要注意的是,当每用户的交互数限制为 1 时,输入模型的用户评论集为空,而商品评论集不一定为空.我们在 y 轴展示的是 LFM 与 DeepCLFM 结果 MSE 的差值.因此如果 y 轴的正,则表明 DeepCLFM 性能更佳.特别是,当每个用户(商品)的可用训练样本极其缺乏时,本文提出的 DeepCLFM 相比 LFM 的实验结果有更大的提升.随着可训练样本的增加,DeepCLFM 的优势逐渐减弱.但是,在可训练样本

达到最大值 5 时,曲线的 y 值始终大于 0. 该实验结果表明:融入评论文本的 DeepCLFM 能利用评论信息指导全局偏移项的学习,比仅仅使用评分数据的 LFM 更有效地应对冷启动情况.

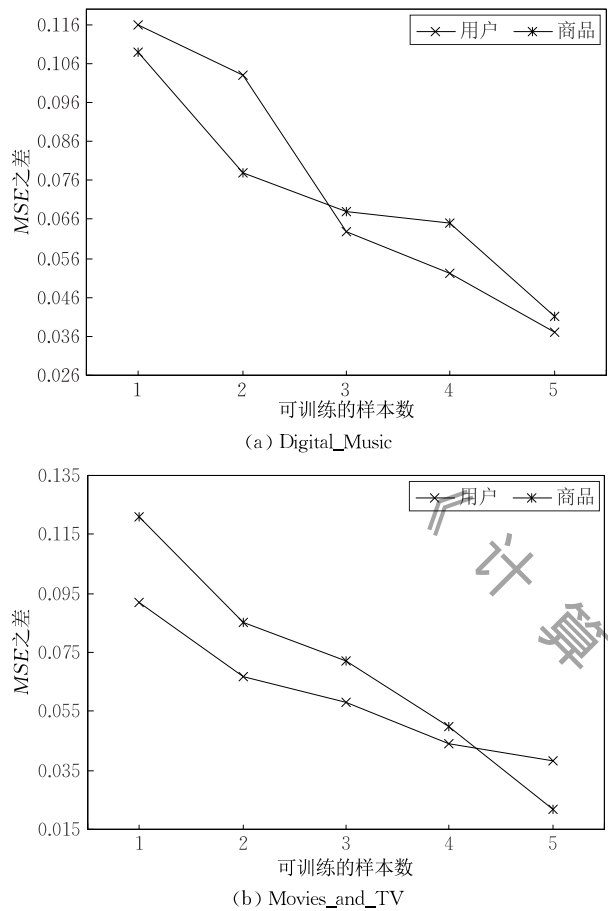


图 8 用户(商品)冷启动情景模拟实验

6 总 结

本文基于评分矩阵和评论文本,提出了能个性化预测评分的深度学习模型 DeepCLFM.

DeepCLFM 能同时从评分矩阵和评论文本中学习到用户和商品的潜在特征,并通过一二阶特征项的方式将它们进行融合,各种对比实验表明本文的方法能进一步降低预测评分的误差.此外,我们针对冷启动问题进行了相关实验,证实 DeepCLFM 以融合评论文本的方式比仅使用评分矩阵的方法能更有效地应对冷启动情况.

对于今后的工作,有几个方向有待探索.(1)将用户和商品交互产生的评论文本按照发表时间的顺序进行排列,使得模型能够考虑用户偏好随时间变化所发生的转变;(2)本文 DeepCLFM 学习到的用

户偏好向量是一种静态向量(同一用户对不同商品的偏好向量是相同的),如果能学习出用户的动态偏好向量,推荐性能或许能进一步提升.

参 考 文 献

[1] Subramaniaswamy V, Manogaran G, Logesh R, et al. An ontology-driven personalized food recommendation in IoT-based healthcare system. The Journal of Supercomputing, 2018, 75(6): 1-33

[2] Huang Z H, Shan G X, Cheng J J, et al. TRec: An efficient recommendation system for hunting passengers with deep neural networks. Neural Computing and Applications, 2018, 31(S-1): 209-222

[3] Chien T C, Chen Z H, Chan T W. Exploring long-term behavior patterns in a book recommendation system for reading. Journal of Educational Technology & Society, 2017, 20(2): 27-36

[4] Grbovic M, Radosavljevic V, Djuric N, et al. E-commerce in your inbox: Product recommendations at scale//Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Sydney, Australia, 2015: 1809-1818

[5] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009, 42(8): 30-37

[6] Ling G, Lyu M R, King I. Ratings meet reviews, a combined approach to recommend//Proceedings of the 8th ACM Conference on Recommender Systems. California, USA, 2014: 105-112

[7] McAuley J, Leskovec J. Hidden factors and hidden topics: Understanding rating dimensions with review text//Proceedings of the 7th ACM Conference on Recommender Systems. Hong Kong, China, 2013: 165-172

[8] Zheng L, Noroozi V, Yu P S. Joint deep modeling of users and items using reviews for recommendation//Proceedings of the 10th ACM International Conference on Web Search and Data Mining. Cambridge, UK, 2017: 425-434

[9] Seo S, Huang J, Yang H, Liu Y. Interpretable convolutional neural networks with dual local and global attention for review rating prediction//Proceedings of the 11th ACM Conference on Recommender Systems. Como, Italy, 2017: 297-305

[10] Wang Q, Li S, Chen G. Word-driven and context-aware review modeling for recommendation//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. Torino, Italy, 2018: 1859-1862

[11] Chen C, Zhang M, Liu Y, Ma S. Neural attentional rating regression with review-level explanations//Proceedings of the 2018 World Wide Web Conference on World Wide Web. Lyon, France, 2018: 1583-1592

- [12] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality//Proceedings of the Advances in Neural Information Processing Systems. California, USA, 2013: 3111-3119
- [13] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar, 2014: 1532-1543
- [14] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
- [15] Lee D D, Seung H S. Algorithms for non-negative matrix factorization//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2001: 556-562
- [16] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model//Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Nevada, USA, 2008: 426-434
- [17] Li Lin, Liu Jin-Xing, Meng Xiang-Fu, et al. Commodity recommendation model based on fusion scoring matrix and comment text. Chinese Journal of Computers, 2018, 41(7): 1559-1573(in Chinese)
(李琳, 刘锦行, 孟祥福等. 融合评分矩阵与评论文本的商品推荐模型. 计算机学报, 2018, 41(7): 1559-1573)
- [18] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets//Proceedings of the IEEE International Conference on Data Mining. Pisa, Italy, 2008: 263-272
- [19] Mnih A, Salakhutdinov R R. Probabilistic matrix factorization//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2008: 1257-1264
- [20] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering//Proceedings of the 24th International Conference on Machine Learning. Corvalis, USA, 2007: 791-798
- [21] Wu Y, DuBois C, Zheng A X, et al. Collaborative denoising auto-encoders for top- n recommender systems//Proceedings of the 9th ACM International Conference on Web Search and Data Mining. California, USA, 2016: 153-162
- [22] Li S, Kawale J, Fu Y. Deep collaborative filtering via marginalized denoising auto-encoder//Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. Melbourne, Australia, 2015: 811-820
- [23] Jakob N, Weber S H, Müller M C, et al. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations//Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion. Hong Kong, China, 2009: 57-64
- [24] Huang J, Rogers S, Joo E. Improving restaurants by extracting subtopics from yelp reviews//Proceedings of the iConference 2014 (Social Media Expo). Berlin, Germany, 2014: 1-5
- [25] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles//Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. California, USA, 2001: 448-456
- [26] Ganu G, Kakodkar Y, Marian A. Improving the quality of predictions using textual information in online user reviews. Information Systems, 2013, 38(1): 1-15
- [27] Diao Q, Qiu M, Wu C Y, et al. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA, 2014: 193-202
- [28] Bao Y, Fang H, Zhang J. TopicMF: Simultaneously exploiting ratings and reviews for recommendation//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Québec, Canada, 2014: 2-8
- [29] Hu G N, Dai X Y, Song Y, et al. A synthetic approach for recommendation: Combining ratings, social relations, and reviews//Proceedings of the 24th International Joint Conference on Artificial Intelligence. Buenos Aires, Argentina, 2015: 1756-1762
- [30] Li X, Xu G, Chen E, et al. Learning user preferences across multiple aspects for merchant recommendation//Proceedings of the 2015 IEEE International Conference on Data Mining. Atlantic City, USA, 2015: 865-870
- [31] Gao Yi-Fan, Yu Wen-Zhe, Chao Ping-Fu, et al. Analyzing reviews for rating prediction and item recommendation. Journal of East China Normal University (Natural Science), 2015, (3): 80-90(in Chinese)
(高伟藩, 余文喆, 晁平复等. 基于评论分析的评分预测与推荐. 华东师范大学学报(自然科学版), 2015, (3): 80-90)
- [32] Feng S, Cao J, Wang J, et al. Recommendations based on comprehensively exploiting the latent factors hidden in items' ratings and content. ACM Transactions on Knowledge Discovery from Data, 2017, 11(3): 35
- [33] Wang S, Manning C D. Baselines and bigrams: Simple, good sentiment and topic classification//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistic. Jeju Island, Korea, 2012: 90-94
- [34] Kim D, Park C, Oh J, et al. Convolutional matrix factorization for document context-aware recommendation//Proceedings of the 10th ACM Conference on Recommender Systems. Boston, USA, 2016: 233-240
- [35] Catherine R, Cohen W. TransNets: Learning to transform for recommendation//Proceedings of the 11th ACM Conference on Recommender Systems. Como, Italy, 2017: 288-296
- [36] Wu H, Zhang Z, Yue K, et al. Dual-regularized matrix factorization with deep neural networks for recommender systems. Knowledge-Based Systems, 2018, 145: 46-58
- [37] Al-Sabahi K, Zuping Z, Nadher M. A hierarchical structured self-attentive model for extractive document summarization (HSSAS). IEEE Access, 2018, 6: 24205-24212

[38]

Cheng Z, Ding Y, He X, et al. A³NCF: An adaptive aspect attention model for rating prediction//Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm, Sweden, 2018; 3748-3754

[39]

Rendle S. Factorization machines//Proceedings of the 2010 IEEE International Conference on Data Mining. Sydney, Australia, 2010; 995-1000

[40]

He X, Chua T S. Neural factorization machines for sparse predictive analytics//Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. Tokyo, Japan, 2017; 355-364

[41]

Li P, Wang Z, Ren Z, et al. Neural rating regression with

abstractive tips generation for recommendation//Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. Tokyo, Japan, 2017; 345-354

[42]

Xiao J, Ye H, He X, et al. Attentional factorization machines: Learning the weight of feature interactions via attention networks. arXiv preprint arXiv:1708.04617

[43]

Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

[44]

Wallach H M. Topic modeling: Beyond bag-of-words//Proceedings of the 23rd International Conference on Machine Learning. Pennsylvania, USA, 2006; 977-984



FENG Xing-Jie, Ph. D. , professor. His research interests include recommender system, database and data warehouse, intelligent information processing theory and technology.

ZENG Yun-Ze, M. S. His research interests include recommender system and machine learning.

Background

Predicting a user’s rating of an item is a key issue for the recommender system to provide a personalized recommendation service. In the past, the matrix factorization algorithm based on rating matrix has superior performance, but it is limited by the sparsity of rating data. Then, the recommendation algorithm based on the review text alleviates the impact of sparsity of rating matrix through the text data, and further improves the accuracy of the personalized recommendation. However, most of the current reviews-based algorithms fail to fully integrate with the rating matrix, and are limited to the latent linear feature level, which can not to further exploit the more abstract nonlinear features.

In this paper, inspired by previous researches, we propose a deep learning model DeepCLFM which is combing with

review text and rating matrix data. This model can capture the nonlinear features of users and items through first and second order features to further reduce the rating prediction error. The experimental results on five public large datasets show that DeepCLFM has lower error prediction than current methods, and is more able to cope with problems such as online learning and cold start.

The work described in this study is supported in part by the National Natural Science Foundation of China and China Civil Aviation Administration Joint Fund Project under Grant Nos.U1233113 and U1633110, and the National Natural Science Youth Foundation of China under Grant Nos.61301245 and 61201414.