

关注长尾物品的推荐方法

秦 婧*, 张青博, 王 斌

(东北大学 计算机科学与工程学院, 沈阳 110169)

(* 通信作者电子邮箱 qinj_job@126.com)

摘要:针对推荐系统算法中覆盖率和多样性偏低所带来的长尾问题,提出了一种长尾物品的推荐框架以及关注长尾物品的推荐算法 FLTI。长尾物品的推荐框架是基于卷积神经网络(CNN)模型构建的,分为数据处理层、推荐算法层和推荐列表生成层。将 FLTI 算法加入到了框架中的推荐算法层,该算法首先计算了频繁推荐项以及非频繁推荐项,然后采用使用长尾物品替换频繁推荐项的方法来满足不同系统中指定的长尾比例。实验结果表明,在 Movielens 1M 和 BookCrossing 数据集上,FLTI 算法比传统的基于用户的协同过滤(UserCF)算法、基于物品的协同过滤(ItemCF)算法、奇异值分解(SVD)推荐算法以及协同去噪自动编码(CDAE)算法在覆盖率指标上最多提高了 51%,多样性指标上最多提高了 59%。

关键词:推荐系统;长尾推荐;长尾物品;频繁推荐项;卷积神经网络

中图分类号:TP183 **文献标志码:**A

Recommendation method with focus on long tail items

QIN Jing*, ZHANG Qingbo, WANG Bin

(School of Computer Science and Engineering, Northeast University, Shenyang Liaoning 110169, China)

Abstract: To solve the long tail problem caused by low coverage and diversity in recommendation system, a recommendation framework for long tail items and a recommendation algorithm named FLTI (Focusing on Long Tail Item) were proposed. The recommendation framework for long tail items was built based on Convolutional Neural Network (CNN) model with three layers, including data processing layer, recommendation algorithm layer and recommendation list generation layer. The FLTI algorithm was added to the recommendation algorithm layer of the framework, and calculated the frequent recommendation items and the infrequent recommendation items at first, then replaced the frequent recommendation items by the long-tail items to meet the specified proportion of long-tail items in the system. Experimental results on Movielens 1M and BookCrossing datasets show that compared to traditional User-Based Collaborative Filtering (UserCF) algorithm, Item-based Collaborative Filtering (ItemCF) algorithm, Singular Value Decomposition (SVD) recommendation algorithm and Collaborative Denosing Auto-Encoder (CDAE) algorithm, the coverage of FLTI algorithm is improved up to 51%, and the diversity of FLTI algorithm is improved up to 59%.

Key words: recommendation system; long tail recommendation; long tail item; frequent recommendation item; Convolutional Neural Network (CNN)

0 引言

从 1992 年的第一个邮件推荐系统^[1]问世至今 20 多年中,推荐系统在电商、医疗、商品、新闻等各领域都得到了广泛应用。推荐系统中物品被用户评分或选择的次数是一种典型的长尾现象,符合帕累托法则(Pareto's principle)。从图 1 中可以看出,MovieLens 1M^[2]数据集中的电影观看数量即为长尾现象,大多数的电影被观看的次数少于 500 次。此外,用户仅对 3 706 部电影进行了评分,而余下的近 200 部电影无人评分。由于用户对电影的评分次数存在长尾现象,因此,推荐系统在向用户推荐物品时也随之出现了长尾现象,即推荐物品的覆盖率偏低。传统的推荐方法包括基于近邻的推荐方法、基于内容的推荐方法、矩阵分解的推荐方法以及混合的推荐方法等,由于数据集中数据稀疏问题严重,导致推荐结果会出现推

荐物品的总量大多不足 30%,甚至低于 10%。此外,推荐物品的总量并没有因为推荐列表的长度变化而使推荐物品数量显著增加,反而呈现出缓慢增长趋势。以基于用户的协同过滤(User-based Collaborative Filtering, UserCF)算法^[1]、基于物品的协同过滤(Item-based Collaborative Filtering, ItemCF)^[3]算法和奇异值分解(Singular Value Decomposition, SVD)推荐算法为例,使用 MovieLens 1M 实现电影推荐,假设目标用户数量为 100,在推荐列表长度分别为 10、15、20 时,推荐结果中不重复电影的数量随推荐列表长度的变化如图 2 所示。

从图 2 中可以看出:ItemCF 算法在推荐列表长度为 10 时推荐了 188 部不重复的电影,在推荐列表长度为 20 时推荐了 286 部不重复电影;但推荐列表长度为 10 时 ItemCF 算法应为 100 个用户推荐的电影总量(包含重复的电影)为 1 000 部,推荐列表长度为 20 时,则 ItemCF 算法应为 100 个用户推荐的电

收稿日期:2019-08-12;修回日期:2019-09-29;录用日期:2019-10-24。 基金项目:国家自然科学基金资助项目(61572122, U1736104)。

作者简介:秦婧(1981—),女,辽宁沈阳人,博士研究生,CCF 会员,主要研究方向:推荐系统、数据挖掘;张青博(1994—),女,河南三门峡人,博士研究生,主要研究方向:推荐系统、人工智能;王斌(1972—),男,辽宁沈阳人,副教授,博士生导师,博士,主要研究方向:数据库、人工智能。

影总量为2000部。显然,推荐列表的长度从10增长到20,ItemCF算法推荐的不重复电影数量增长量(1.5倍)低于向用户推荐的电影总量的增长量(2倍)。在图2中的3个算法里,UserCF算法推荐的不重复电影数量最多,但在推荐列表长度为20时,它向用户推荐的不重复电影数量也仅为494,仅占MovieLens数据集中电影总量的10%左右。因此,推荐系统中物品推荐数量的总体覆盖率是严重偏低的,从而导致了推荐系统中推荐物品缺乏多样性。由此,用户的观影、阅读以及购买物品等行为受到了局限性的引导,影响了用户的知识扩展和对新事物的适应能力等。

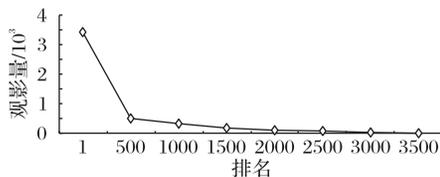


图1 观影量的长尾效应

Fig. 1 Long tail effect of movie viewing

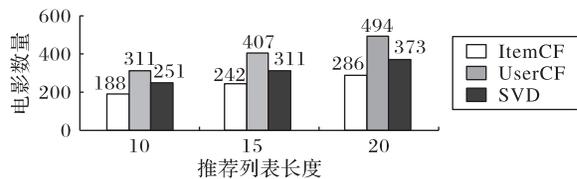


图2 不重复推荐电影数量随推荐列表长度(L)的变化

Fig. 2 Number of non-repeating recommended movies varies with L

为提高推荐结果中物品的总体覆盖率和多样性,特别是增长长尾物品的推荐数量,本文提出了一种基于卷积神经网络(Convolutional Neural Network, CNN)建立的推荐模型以及关注长尾物品的推荐算法FLTI(Focusing on Long Tail Item)。

本文工作如下:1)分析了推荐列表中热门物品与长尾物品的关系,建立了热门物品与长尾物品的对应关系;2)基于CNN建立推荐框架,并基于此框架提出了关注长尾物品的推荐算法FLTI;3)在公开的数据集MovieLens1M和BookCrossing上进行实验,并与传统的UserCF、ItemCF、SVD推荐算法以及协同去噪自动编码(Collaborative Denoising Auto-Encoder, CDAE)^[4]方法在召回率、覆盖率、多样性等方面进行了比较,结果显示本文方法在覆盖率和多样性方面表现更佳。

1 相关工作

1.1 长尾推荐

推荐系统中的长尾问题研究主要集中在流行度低的物品推荐中,流行度低的物品主要指用户对物品的评分数量少的物品。长尾推荐包括在线推荐和离线推荐:在线推荐主要是指查询推荐^[5-6],根据用户的查询信息并结合用户的历史信息为用户推荐符合其偏好的物品;而离线推荐是在未提供查询关键字以及其他实时信息时,将用户的历史信息作为依据为用户推荐符合其偏好的物品。本文主要研究的是离线推荐算法中的长尾问题。从近年来的研究文献中可以看出,离线的长尾推荐方法主要有使用多目标函数的优化算法^[7-9]、二部图方法^[10]、三部图方法^[11-13]、聚类方法^[14]等。多目标函数优化包括评价模型准确率、覆盖率、多样性等目标的函数,如Hamedani等^[7]分别提出了在AICF(Attribute-aware Item-based Collaborative Filtering)方法和AMOSA(Archived Multi-Objective Simulated Annealing)方法基础上优化推荐列表的方法,使得推荐列表的准确度、多样性方面有所提高,并减少了

推荐列表中的流行度;Pang等^[8]提出了基于NSGA-II(Non-dominated Sorting Genetic Algorithm II)算法的权重相似度计算方法,并同时准确率覆盖率作为目标函数优化推荐算法,实验表明该推荐算法在保证准确率的同时提高了覆盖率;Wang等^[9]提出了将准确度和新颖度作为目标函数建立推荐模型,实验表明该推荐算法相比传统的推荐算法在准确率、新颖度、多样性等方面均有所提高。

Yin等^[10]提出了二部图算法解决长尾推荐问题。Johnson等在文献^[11]中提出了使用三部图方法用于长尾推荐,即使用user-genre和genre-item的形式来构建三部图。Luke等^[12]在Johnson提出的三部图方法^[11]的基础上,优化了user-genre的关联关系,在genre-item关系图中建立了更多的路径,将Johnson提出的“full genre”关系转换为“basic genre”关系,即将由多个类别构成的类别信息转换为单一类别分别表示。比如,一部电影的类别是“Action, Adventure, Comedy”,则拆分为“Action”“Adventure”“Comedy”三个基本类别,通过提高图的联通性,从而改善了推荐效果。Johnson等在文献^[13]中基于三部图的推荐算法加入了马尔可夫过程方法用于表示长尾物品的区域,从而提高了推荐物品的多样性。Grozin等^[14]使用基于session距离的商品聚类方法来实现交叉销售的长尾推荐,与直接使用类型、产品相似度聚类相比在归一化折损累积增益(Normalized Discounted Cumulative Gain, NDCG)指标评测上效果更优。

此外,Takama等^[15]还提出了基于用户的个人价值观来建立推荐模型,从而改善长尾物品的推荐;Abdollahpouri等^[16]提出了基于物品流行度权重的方法来解决长尾物品的推荐。

尽管以上方法在覆盖率、多样性、准确性等各方面都有不同程度的改善,但很少考虑长尾物品的推荐率,以及针对推荐列表中物品重复率高且频繁推荐项较多等现象的改善。

1.2 基于深度学习的推荐模型

目前,深度学习在推荐系统中得到了广泛的应用^[17]。深度学习主要包括循环神经网络(Recurrent Neural Network, RNN)^[18]和CNN^[19]。其中,基于RNN技术,研究人员提出了长短时记忆网络(Long Short-Term Memory, LSTM)和门限循环单元(Gated Recurrent Unit, GRU)。RNN适用于序列数据的应用,比如用户的点击流数据、用户的实时位置数据等,它可以根据序列中的上一个状态预测下一个状态;CNN则主要适用于图像、文本等内容的特征提取和分析。由于本文研究的推荐算法是离线的推荐算法,并且在推荐数据源中包含了大量的文本信息,因此,选用CNN处理文本特征。

2 问题描述及相关定义

为了明确本文中长尾物品的推荐方法,本章将具体化问题描述及相关定义。关注长尾物品的推荐,问题描述如下:

给定数据集 D ,为给定的目标用户 $Users=\{U_1, U_2, \dots, U_n\}$ 推荐长度为 L 的物品集合 $Recommended_Items=\{I_1, I_2, \dots, I_n\}$, p 为长尾物品数量占推荐列表中总物品数量的比例,使 $Recommended_Items$ 满足以下条件:1)在 $Recommended_Items$ 中含有指定比例 p 的长尾物品;2)符合用户的历史偏好。

其中,长尾物品是指物品的评分次数低于用户对物品评分次数中位数的物品。

为了明确后续算法中所提及的定义,先介绍本文中用到的频繁推荐项(Frequent Recommended Item, FRItem)、非频繁推荐项(Infrequent Recommended Item, IRItem)、物品长尾系数(Item Long Tail Coefficient, ILTC)等概念的定义。

定义 1 FRItem 和 IRItem。FRItem 是指为目标用户 *Users* 所推荐的物品集 *Recommended_Items* 中出现的次数高于物品出现次数中位数的项目, 低于物品出现次数中位数的项目则为 IRItem。

定义 2 ILTC。由物品的推荐次数和评分次数共同加权得出, 第 *i* 个物品的物品长尾系数计算公式如下:

$$ILTC_i = \alpha(rec_times_i/rec_total_i) + (1 - \alpha)(rat_times_i/rat_total_i) \quad (1)$$

其中: *i* 代表物品集合 *Items* 中一个物品, α 的值代表推荐次数的权重值, *rec_times* 和 *rec_total* 分别代表物品 *i* 的推荐次数和推荐物品的总次数, *rat_times* 和 *rat_total* 分别代表物品 *i* 的评分次数和评分总次数。

3 长尾物品的推荐框架及算法

3.1 长尾物品推荐框架

长尾物品推荐框架主要由数据处理层、推荐算法层、推荐列表生成层构成, 如图 3 所示。从图 3 可以看出, 长尾物品推荐框架的核心是推荐算法层, 输入是目标用户群, 而输出则是向目标用户群推荐的物品列表。

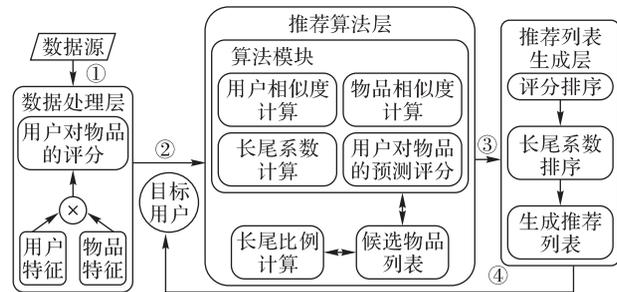


图 3 长尾物品推荐框架

Fig. 3 Recommend framework for long tail items

数据处理层 首先从数据源中分别选取用户和物品的特征, 然后将用户和物品的特征作标准化处理, 并将用户和物品的各特征作全连接。其中, 将含有文本内容多的特征使用 CNN 的方式来处理文本内容以提高相似度计算的准确性, 如在 MovieLens 数据集中, 将电影名称特征使用 CNN 方法得到文本的向量表示。最后, 将用户对物品的评分作为目标, 通过设置损失函数 Huber 来训练得到用户对物品的预测评分值。Huber 损失函数适用于回归问题, 对于离群点不敏感, 降低了离群点对于损失函数的权重值, 能有效避免模型过拟合, 如式 (2) 所示。

$$L_{\delta} = \begin{cases} 0.5(y - f(x))^2, & |y - f(x)| \leq \delta \\ \delta|y - f(x)| - 0.5\delta^2, & \text{其他} \end{cases} \quad (2)$$

其中: δ 是边界值, 用于判断数据是否为异常点; *y* 为预测评分值; *f(x)* 为实际评分值。

数据处理层的评分回归模型的整体结构如图 4 所示。

推荐算法层 在数据处理层得到的回归模型基础上实现推荐算法, 包括使用模型的正向传播可以获得用户对物品的预测评分, 计算物品的相似度和计算用户的相似度。关注长尾物品的推荐算法 FLTI 首先根据目标用户已评分的物品, 找到与其相似的物品产生推荐物品候选集; 然后根据物品候选集物品的推荐次数的中位数拆分为频繁推荐项集和非频繁推荐项集, 并查找非候选集与频繁推荐项集相似的物品; 最后, 将符合用户偏好的非候选集中的物品推荐给选择过频繁推荐项中物品的目标用户, 并满足长尾比例 *p* 的要求。具体的过

程如图 5 所示。

相似度计算方法根据数据处理层模型的表示, 使用矩阵乘法并结合使用向量相似度计算的余弦相似度方法来计算物品的相似度和用户相似度。

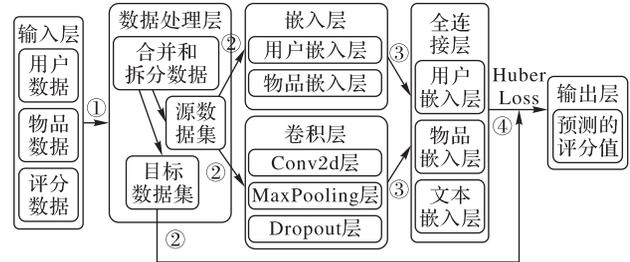


图 4 数据处理层模型

Fig. 4 Data processing layer model

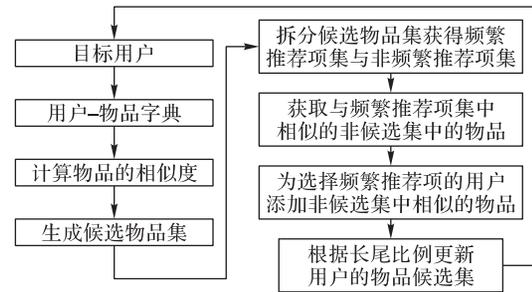


图 5 FLTI 算法的推荐流程

Fig. 5 Recommendation flow of FLTI algorithm

推荐列表生成层 为了让目标用户能更多关注长尾物品, 在推荐列表排序中引入了长尾系数。根据用户对候选列表的预测评分以及长尾系数对由推荐算法模型得到的候选物品列表进行排序, 对于预测评分相同的物品, 按照长尾系数升序排列。

3.2 FLTI 算法描述

为了提高 FLTI 算法的效率, 通过数据处理层得到的模型, 将用户对物品的评分结果保存到文件中, 在推荐算法中直接读取该文件即可获得用户对物品的评分。

FLTI 算法主要是通过替换频繁推荐项的方法实现的, 并在替换时选择在候选项中用户预测评分最高的项目作为替换项, 具体描述如下:

算法 1 关注长尾物品的推荐算法 FLTI。

输入 数据集 *D*, 长尾比例 *P*, 推荐列表长度 *L*, 目标用户 *Users*;

输出 目标用户的候选列表项 *CandidateDict*。

for *u* in *Users*

 获取与用户 *u* 相似的按照评分值降序排列的前 *L* 个物品

 将用户和对应的物品列表添加到推荐列表候选集字典 *temp_items* 中

end for

根据物品推荐次数的中位数, 将 *temp_items* 拆分为频繁推荐项字典 (*frequent_items*) 和非频繁推荐字典 (*infrequent_items*)

for *f_item* in *frequent_items*

 在非候选集中, 获取与 *f_item* 相似的 *L* 个物品存放到

temp_frequent_items 字典中

end for

获得用户推荐列表中不符合长尾比例 *p* 的用户集合

candidate_users_items

for *u, i* in *candidate_users_items*

 if *i* 是 *frequent_items* 中的物品

 将 *i* 替换为 *temp_frequent_items* 中用户评分最高的物品, 且用

户 u 的候选推荐列表中没有该物品

重新计算用户推荐列表中物品的长尾比例,若未满足长尾比例,则继续更新候选推荐列表。

若 $candidate_users_items$ 中仍有不满足长尾比例的用户,则将 $candidate_users_items$ 中推荐次数最多的非长尾项目替换为与其相似度最高的长尾项目,且不在用户的候选推荐列表中。

return CandidateDict

在算法1中,为了减少内存使用,将数据集中的长尾物品以及运算结果的中间步骤保存到文件中。此外,算法1中重点关注了长尾项目的推荐比例,长尾比例的高低将直接影响推荐效果,若将长尾比例设置为1,则表示推荐列表中的所有物品均为长尾物品,此时,长尾项目会成为频繁推荐项,而之前的频繁推荐项则会成为长尾物品,依然会导致推荐结果中覆盖率和多样性下降;若将长尾比例设置为0或者0.1,则所有的推荐列表均能满足要求,不能提升长尾物品在推荐列表中的数量。因此,为了避免由于长尾比例设置过大而导致的新的长尾现象,建立统计频繁推荐项的替换次数以及替换的长尾物品数量的字典,并在加入到候选推荐列表时,选择替换次数少且与要替换的物品相似度高的物品优先替换。

此外,通过算法1完成推荐候选集的筛选后,为了保证推荐列表中最大限度满足用户的偏好并优先选择长尾物品,首先将候选集中用户对物品的预测评分排序,若评分相同则按照物品的长尾系数排序。

4 实验与结果分析

4.1 数据集与评价标准

实验环境:CPU为Inter Core i5-4690K@3.5 GHz,内存为16 GB;操作系统为Windows 10;软件为TensorFlow1.13.1和Python 3.6.4。

实验数据集为公开的数据集MovieLens1M (<https://grouplens.org/datasets>)和BookCrossing。其中,MovieLens 1M数据集中存放了6040位用户对3883部电影的评分信息,包含了用户表(users)、评分表(ratings)、电影表(movies);BookCrossing数据集中存放了278858位用户对271379本书的评分信息,包含了用户表(BX-Users)、评分表(BX-Book-Ratings)以及图书表(BX-Books)。

为了验证FLTI算法的有效性,选择了三类方法进行对比:一类是基于近邻的推荐方法,包括基于用户的协同过滤(UserCF)算法和基于物品的协同过滤(ItemCF)算法;一类是基于矩阵分解的推荐方法,即基本的SVD推荐算法;一类是基于神经网络的方法CDAE。除了CDAE算法外,其余算法都采用基于Python语言的surprise (<https://surprise.readthedocs.io/en/stable/>)库中所提供的协同过滤算法(KNNBasic)和SVD算法。在KNNBasic方法中通过更改“sim_option”中的属性值切换UserCF和ItemCF两种算法:当“sim_option”中“user_based”的值为“True”则为基于用户的协同过滤算法;当“user_based”的值为“False”则为基于物品的协同过滤算法。实验方法说明如表1所示。

评价标准包括召回率($Recall@L$)^[10]、覆盖率($Coverage$)^[5]和多样性($Diversity$)^[10]。

召回率用于验证在推荐列表的准确性,如式(3)所示:

$$Recall@L = \frac{1}{N} \sum_{n=1}^N hit@L \quad (3)$$

其中: $hit@L$ 表示推荐列表 L 中为目标用户推荐的正确项目个数, N 表示目标用户在测试数据集中所有正确推荐项目的数量。

覆盖率指标和多样性指标用于验证推荐结果的长尾情况,分别如式(4)和式(5)所示。

$$Coverage = \frac{1}{|T|} \left| \bigcup_{u \in U} R(u) \right| \quad (4)$$

其中: $R(u)$ 表示为用户推荐的物品数量, T 表示数据集中物品总量。

$$Diversity = \frac{1}{|I|} \left| \bigcup_{u \in U} R(u) \right| \quad (5)$$

其中: $R(u)$ 表示为用户推荐的物品数量, I 表示向用户推荐物品的总量。比如,向500个用户推荐10部电影,则推荐物品的总量为5000。

表1 实验中用到的方法说明

Tab. 1 Explanation of methods used in experiment

方法	说明
KNNBasic (User-Based)	基于用户的协同过滤方法,相似度计算使用余弦相似度
KNNBasic (Item-Based)	基于物品的协同过滤方法,相似度计算使用余弦相似度
SVD	将用户-物品评分表使用SVD方法分解

4.2 实验步骤与分析

为了提高模型预测的准确性,所有方法均使用五折交叉法来训练模型。针对数据集MovieLens1M和BookCrossing中数据特征不同的情况,推荐模型的参数设置除了目标函数(即损失函数)和激活函数都设置为Huber和ReLU外,初始学习率(Learning_rate)分别设置为0.01和0.001,初始批处理参数(batch_size)分别设置为256和100。为了保证验证结果的一致性,长尾比例 p 均设置为0.35。

1)召回率。将推荐列表长度分别设置为10、15、20,各方法在MovieLens 1M数据集和BookCrossing数据集上的召回率测试结果如图6所示。从 $Recall@L$ 的测试结果可以看出,FLTR方法的评测值略高于其他对比方法。由于BookCrossing数据集的稀疏程度高于MovieLens数据集,实验中的方法均用到了用户对物品的评分表,因此, $Recall@L$ 的结果普遍要低于MovieLens上的评测结果。

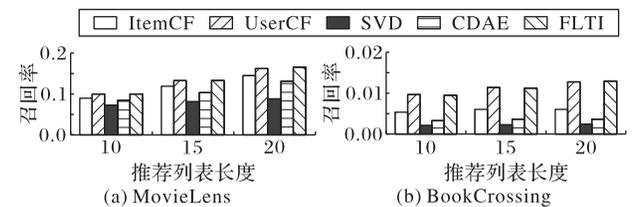


图6 两个数据集上不同方法的召回率对比

Fig. 6 Comparison of recall rate of different methods on two datasets

2)覆盖率和多样性。在验证覆盖率和多样性时,仍设置推荐列表长度为10、15、20。验证覆盖率时,由于对比方法中只考虑了评分数据中的物品,因此,总物品数量的计算是参与推荐的总物品数量。在MovieLens和BookCrossing数据集上的覆盖率结果如图7所示,多样性的实验结果如图8所示。从图7.8可以看出,FLTI相比对比方法有明显的优势,特别是在MovieLens数据集上,当FLTI方法在推荐列表长度为20时,覆盖率到达了40%以上,而ItemCF方法的覆盖率仅为20%。

综上所述,FLTI算法在MovieLens数据集和BookCrossing数据集上的覆盖率和多样性的结果都有所提高,当长尾比例为0.35时,在MovieLens数据集上且推荐列表长度为10时的覆

盖率和多样性提升比例最多,分别为51%和59%;并且在数据稀疏程度高的BookCrossing数据集上效果更优。在FLTI算法中,随着长尾比例的提高,覆盖率和多样性也将不断提高,但准确率和召回率会有所降低,因此,在实际应用中,需要根据不同数据集的稀疏情况以及推荐系统的实际需求来调整长尾比例。

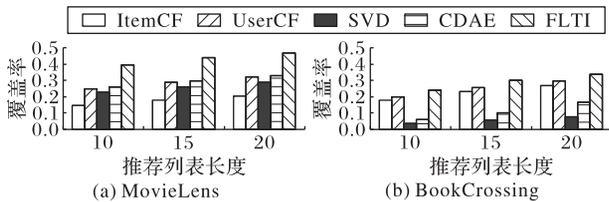


图7 两个数据集上不同方法的覆盖率对比

Fig. 7 Comparison of coverage rate of different methods on two datasets

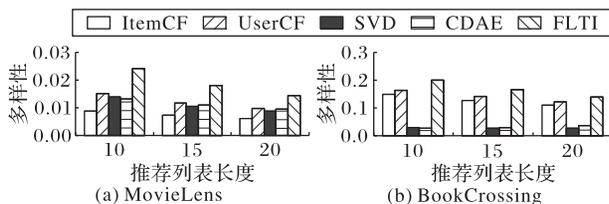


图8 两个数据集上不同方法的多样性对比

Fig. 8 Comparison of diversity of different methods on two datasets

5 结语

本文针对推荐系统中的推荐物品存在的长尾现象,提出了一种关注长尾物品的推荐框架以及推荐算法FLTI。在关注长尾物品的推荐框架中使用三层结构实现了长尾物品推荐:在数据预处理层中使用深度学习的方法全面地将用户特征、物品特征结合在一起,并将数据特征中的文本内容部分通过CNN卷积,以便更好地实现对特征的语义分析;在推荐模型层中,关注了长尾物品推荐的比例,并使长尾物品尽可能符合用户的偏好;在推荐列表生成层中,使用了用户对物品的预测评分以及长尾系数对推荐列表中的物品排序。在公开数据集MovieLens1M和BookCrossing上与推荐系统中的基本算法UserCF、ItemCF、SVD以及CDAE进行了对比,结果显示FLTI算法在覆盖率和多样性方面较对比方法占有较大优势。在后续的研究工作中,将重点研究使用深度学习的方法通过学习长尾物品的特征,使得推荐结果中出现更多的长尾物品,且提高推荐物品的准确率。

参考文献 (References)

- [1] GOLDBERG D, NICHOLS D, OKI B M, et al. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM, 1992, 35(12):61-70.
- [2] HARPER F M, KONSTAN J A. The MovieLens datasets: history and context [J]. ACM Transactions on Interactive Intelligent Systems, 2016, 5(4): No. 19.
- [3] LINDEN G, SMITH B, YORK J. Amazon.com recommendations: item-to-item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1):76-80.
- [4] 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 on Web Search and Data Mining. New York: ACM, 2016: 153-162.
- [5] HUANG Z, CAUTIS B, CHENG R, et al. Entity-based query recommendation for long-tail queries [J]. ACM Transactions on Knowledge Discovery from Data, 2018, 12(6): No. 64.

- [6] HUANG Z, CAUTIS B, CHENG R, et al. KB-enabled query recommendation for long-tail queries [C]// Proceedings of the 25th ACM International Conference on Information and Knowledge Management. New York: ACM, 2016: 2107-2112.
- [7] HAMEDANI E M, KAEDI M. Recommending the long tail items through personalized diversification [J]. Knowledge-Based Systems, 2019, 164:348-357.
- [8] PANG J, GUO J, ZHANG W. Using multi-objective optimization to solve the long tail problem in recommender system [C]// Proceedings of the 23rd Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNCS 11441. Cham: Springer, 2019: 302-313.
- [9] WANG S, GONG M, LI H, et al. Multi-objective optimization for long tail recommendation [J]. Knowledge-Based System, 2016, 104:145-155.
- [10] YIN H, CUI B, LI J, et al. Challenging the long tail recommendation [J]. Proceedings of the VLDB Endowment, 2012, 5(9): 896-907.
- [11] JOHNSON J, NG Y K. Using tripartite graphs to make long tail recommendations [C]// Proceedings of the 8th International Conference on Information, Intelligence, Systems and Applications. Piscataway: IEEE, 2017:1-6.
- [12] LUKE A, JOHNSON J, NG Y K. Recommending long-tail items using extended tripartite graphs [C]// Proceedings of the 2018 IEEE International Conference on Big Knowledge. Piscataway: IEEE, 2018: 123-130.
- [13] JOHNSON J, NG Y K. Enhancing long tail item recommendations using tripartite graphs and Markov process [C]// Proceedings of the 2017 International Conference on Web Intelligence. New York: ACM, 2017: 761-768.
- [14] GROZIN V, LEVINA A. Similar product clustering for long-tail cross-sell recommendation [C]// Proceedings of the 6th International Conference on Analysis of Images, Social Networks, and Texts. Berlin: Springer, 2017: 273-280.
- [15] TAKAMA Y, CHEN Y S, MISAWA R, et al. Analyzing potential of personal values-based user modeling for long tail item recommendation [J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2018, 22(4): 506-513.
- [16] ABDOLLAHPOURI H, BURKE R, MOBASHER B. Value-aware item weighting for long-tail recommendation [EB/OL]. [2019-01-10]. <https://arxiv.org/pdf/1802.05382v1.pdf>.
- [17] 黄立威,江碧涛,吕守业,等. 基于深度学习的推荐系统研究综述 [J]. 计算机学报, 2018, 41(7): 1619-1647. (HUANG L W, JIANG B T, LYU S Y, et al. Survey deep learning based recommender systems [J]. Chinese Journal of Computers, 2018, 41(7): 1619-1647.)
- [18] ZHOU Y, ZHOU J, LIU L, et al. RNN-based sequence-preserved attention for dependency parsing [C]// Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2018:5738-5746.
- [19] LECUN Y, BENGIO Y, HINTON G E. Deep learning [J]. Nature, 2015, 521(7553): 436-444.

The work is partially supported by the National Natural Science Foundation of China (61572122, U1736104).

QIN Jing, born in 1981, Ph. D. candidate. Her research interests include recommendation system, data mining.

ZHANG Qingbo, born in 1994, Ph. D. candidate. Her research interests include recommendation system, artificial intelligence.

WANG Bin, born in 1972, Ph. D., associate professor. His research interests include database, artificial intelligence.