感知用户年龄的 Item-based 协同过滤推荐算法

张彩廷,祝永志
(曲阜师范大学 信息科学与工程学院,山东 日照 276826)

摘 要:随着大数据时代的到来,推荐系统为人们寻找自己感兴趣的物品或事件提供了捷径。协同过滤推荐算法分为 User-based 协同过滤算法和 Item-based 协同过滤推荐算法。传统 Item-based 协同过滤推荐算法只关注 Item 间的相似度,与目标用户特征无关,因此传统算法相似度不能有效反映 Item 间的相似程度,推荐准确率低。并且传统 Item-based 协同过滤算法需要基于所有用户的历史行为数据进行计算,随着数据量的快速增长计算量不断增大,推荐时效性差。针对以上问题,提出了一种感知用户年龄的 Item-based 协同过滤推荐算法,基于用户年龄特征对用户进行分类,在类内采用加权相似度对 Item 间的相似度进行计算,并且在 Spark 分布式计算平台上运行测试。实验结果显示,该算法不仅保证了推荐准确率,而且大幅度提高了推荐效率,提升了推荐系统的实时性。

关键词:用户年龄;实时;Item-based 协同过滤;Spark
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2019)06-0095-05
doi:10.3969/j.issn.1673-629X.2019.06.020

User's Age-aware Item-based Collaborative Filtering Recommendation Algorithm

ZHANG Cai-ting, ZHU Yong-zhi
(School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China)

Abstract: With the advent of the big data era, the recommendation system provides a shortcut for people to find objects or events that they are interested in. Collaborative filtering recommendation algorithms are divided into user-based collaborative filtering algorithm and item-based collaborative filtering recommendation algorithm. The traditional Item-based collaborative filtering recommendation algorithm only pays attention to the similarity between the items and has nothing to do with the characteristics of the target user. And the similarity of traditional algorithms cannot effectively reflect the degree of similarity between items, which leads to inaccurate recommendations. The traditional Item-based collaborative filtering algorithm needs to be calculated based on all users' historical behavior data. With the rapid increase of the amount of data, the amount of calculation continues to increase and the recommendation timeliness is poor. For this, we propose a user's age-aware item-based collaborative filtering recommendation algorithm. Users are classified based on user age characteristics. The similarity between items is calculated by weighted similarity within the class and the running test is implemented on the Spark distributed computing platform. Experiment shows that the proposed algorithm can greatly improve the recommendation efficiency and the real-time performance of the recommendation system while ensuring the accuracy of the recommendation.

Key words: user's age; real-time; Item-based collaborative filtering; Spark

0 引 言

在被称之为"大数据时代"[1]的今天,电子商务琳琅满目的商品、休闲娱乐软件数不尽的电影和音乐、每天大大小小的新闻事件以及社交网站中充满的无限可能,使得人们的生活丰富多彩,但如何从如此大量的信息中快速找到自己需要的或感兴趣的信息成为互联网平台的挑战。当前,推荐系统成为人们互联网生活中的"引路人",其在电子商务中的主要功能有:吸引新用户,即向潜在的新客户推荐物品将访客转换为购买者;激励老用户,即根据他们之前购买的物品向老客户推荐更多他们可能喜欢的物品;改善客户服务,提高系统与用户的交互[2]。目前,在推荐系统中应用最广泛

收稿日期:2018-07-16 **修回日期**:2018-11-20 **网络出版时间**:2019-03-06
基金项目:山东省自然科学基金(ZR2013FL015);山东省研究生教育创新资助计划(SDYY12060)
作者简介:张彩廷(1994-),女,硕士研究生,研究方向为分布式计算、大数据;祝永志,教授,硕导,通讯作者,CCF 高级会员(12490S),研究方向为并行与分布式计算、网络数据库。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20190306.0938.058.html

的是协同过滤推荐算法。

文中首先简要介绍了相关研究,在对传统算法的基本思想和相关技术进行简单描述的基础上提出了一种改进算法,并通过实验进行了验证。

1 相关研究

对于推荐算法,已经有不少学者针对其各方面存在的问题提出了各种各样的改进方法。例如,文献[3]为了缓解数据稀疏性问题,提出了综合用户特征及专家信任的协作过滤推荐算法;文献[4]主要考虑用户特征随时间的动态变化,进行精确相似度的计算,并解决冷启动问题;文献[5]提出对用户评分矩阵进行两个维度联合聚类,然后在类内进行矩阵分解预测评分的两阶段联合聚类协同过滤算法,以提升推荐实时性;文献[6]将基于用户的协同过滤算法运行在 Hadoop 平台,将多任务映射到不同的处理器上,以解决算法扩展性的问题。

文中受上述文献启发,提出一种感知用户年龄的 Item-based 协同过滤推荐算法,该算法在推荐系统的实时性、精确性和可扩展性上均有所改善。

2 传统 Item-based 协同过滤推荐算法

2.1 相关技术

(1)相似度计算。

Item-based 协同过滤算法常用有三种相似度计算方法[7]:

余弦相似度:

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U_{ij}} R_{ui}*R_{uj}}{\sqrt{\sum_{u\in U_{ij}} R_{ui}^2}\sqrt{\sum_{u\in U_{ij}} R_{uj}^2}} \tag{1}$$

修正余弦相似度:

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U_{ij}} (R_{ui}-\overline{R_u})(R_{uj}-\overline{R_u})}{\sqrt{\sum_{u\in U_{ij}} (R_{ui}-\overline{R_u})^2}\sqrt{\sum_{u\in U_{ij}} (R_{uj}-\overline{R_u})^2}} \tag{2}$$

Pearson 相关性相似度:

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U_{ij}} (R_{ui}-\overline{R_i})(R_{uj}-\overline{R_j})}{\sqrt{\sum_{u\in U_{ij}} (R_{ui}-\overline{R_i})^2}\sqrt{\sum_{u\in U_{ij}} (R_{uj}-\overline{R_j})^2}} \tag{3}$$

其中,$\mathrm{sim}(i,j)$ 表示项目 i 和项目 j 的相似度;U_{ij} 是参与评分的所有用户集合;R_{ui} 是用户 u 对项目 i 的评分;R_{uj} 是用户 u 对项目 j 的评分;$\overline{R_u}$ 是用户 u 打分的平均分;$\overline{R_i}$ 和 $\overline{R_j}$ 分别表示 U_{ij} 中全部用户对项目 i 和 j 的评分均值。

(2)项目邻居选择。

将所有 Item 对的相似度组成一个项目相似度矩阵,每行为一个项目 i 的相似度向量,将每行的相似度都从大到小降序排列,选出相似度最高的前 k 个项目,作为该行项目 i 的邻居项目,所组成的集合标记为 $\mathrm{KNNI}(i)$。集合之外的相似度不用于目标用户预测评分的参考[8],根据需求可以调整 k 的大小来控制算法计算量和精度。

(3)评分预测。

$$P_{ui}=\overline{R_i}+\frac{\sum_{j\in \mathrm{KNNI}(i)} \mathrm{sim}(i,j)(R_{uj}-\overline{R_j})}{\sum_{j\in \mathrm{KNNI}(i)} |\mathrm{sim}(i,j)|} \tag{4}$$

其中,P_{ui} 表示用户 u 对项目 i 的预测评分;$\overline{R_i}$ 为所有用户对项目 i 评分的平均值;$\mathrm{KNNI}(i)$ 表示项目 i 的邻居项目集合,与预测评分相关的项目 j 都包含于该集合;$\mathrm{sim}(i,j)$ 为项目 i,j 之间的相似度;R_{uj} 为用户 u 对项目 j 的评分;$\overline{R_j}$ 为所有用户对项目 j 的评分平均值。

该评分预测算法是在该项目评分平均值的基础上,利用相似度和目标用户对其他项目的评分的加权平均值来计算预测评分[6]。

2.2 算法描述

Item-based 协同过滤算法的基本思想为:根据用户的历史行为信息计算项目间的相似度,从而预测用户的其他喜好以及喜好程度,根据预测为其进行推荐,算法主要步骤[9]如下:

(1)利用原始数据生成用户-项目评分矩阵;

(2)选择相似度计算方法,计算所有项目之间的相似度,并筛选出每个项目相似度最高的 k 个邻居项目;

(3)依据目标用户对该项目的邻居项目的历史评分,利用评分预测算法来预测目标用户对该项目的评分;

(4)将对目标用户预测出评分的项目进行排序,从高到低选择一定数量对目标用户进行推荐。

3 感知用户年龄的 Item-based 协同过滤推荐算法

3.1 算法改进

(1)冷启动问题是指新加入的用户和项目因为没有相关历史数据而导致无法进行相似度的相关计算,因此不能对其进行推荐。相对于新项目冷启动问题,新用户冷启动问题在现实的推荐系统中表现得更为突出。为解决用户冷启动问题,文中在传统的用户评分数据集的基础上加入用户特征数据集。用户特征数据

集包含的用户特征有:年龄、性别、职业和邮政编码,相关专家给出四个特征的权重比值为 4 ∶ 3 ∶ 2 ∶ 1[4],可以看出年龄特征所占比重最大。因此,首先根据用户年龄特征对用户评分数据集进行预处理,将不同年龄段的用户分成不同的组。在组内进行项目间的相似度计算,这些局部项目间的相似度的计算大大减小,提高了推荐实时性[10]。

(2)传统 Item-based 协同过滤算法的相似度计算并不能准确反映项目间的相似程度,针对该问题,文中使用加权相似度。共同评分的用户数量越多,则该相似度有效度越高,故权重选取为同一 Item 对打分的用户数量[11],加权相似度计算公式如下:

加权余弦相似度:

$$\mathrm{sim}(i,j)=n\frac{\sum_{u\in U_{ij}}R_{ui}*R_{uj}}{\sqrt{\sum_{u\in U_{ij}}R_{ui}^2}\sqrt{\sum_{u\in U_{ij}}R_{uj}^2}} \tag{5}$$

加权 Pearson 相关性相似度:

$$\mathrm{sim}(i,j)=n\frac{\sum_{u\in U_{ij}}(R_{ui}-\overline{R_i})(R_{uj}-\overline{R_j})}{\sqrt{\sum_{u\in U_{ij}}(R_{ui}-\overline{R_i})^2}\sqrt{\sum_{u\in U_{ij}}(R_{uj}-\overline{R_j})^2}} \tag{6}$$

其中,n 为项目 i 和项目 j 共同打分的用户数量;其余元素与式 1 ~ 3 中含义相同。

(3)随着推荐系统使用时间的增长和用户数量的增多,系统数据规模会快速扩展,传统的单机推荐算法对于海量的用户和项目历史数据是无能为力的,不论是存储还是计算无疑都成为了难题。分布式计算平台的出现解决了推荐系统可扩展性的难题。文中利用 Spark 分布式计算平台和 HDFS 分布式存储系统相结合的方式,Spark 基于内存计算,HDFS 具有高容错性、适合批处理等特点,能够满足大数据计算和存储的需求[12]。

Spark 是基于内存的分布式计算框架,通过对弹性分布式数据集(RDD)的操作来进行计算,这些计算会在集群上自动并行执行[13]。每个应用需要一个驱动器程序来发起,它通过一个 SparkContext 对象来访问 Spark,这个对象代表对计算集群的一个连接[14],驱动器程序一般要管理多个执行器节点,计算会被分配到所有的节点上执行。Spark 在集群上的运行如图 1 所示。

3.2　算法描述

输入:用户评分数据集,用户特征数据集,目标用户年龄标识;

输出:对目标用户的预测评分集。

(1)用户特征矩阵提取年龄特征列,数据集已对年龄进行分段标识;

(2)用户评分矩阵提取 User ID,Movie ID,评分对应的列,不使用时间戳数据;

(3)根据目标用户年龄段,提取该类别用户的评分数据,并格式化为非嵌套元组,并将数据按 4 ∶ 1 分为训练集和测试集;

(4)计算类内训练集项目相似度矩阵,记录各个 Item 的最近邻集;

(5)预测评分算法对训练集的空缺数据进行预测。

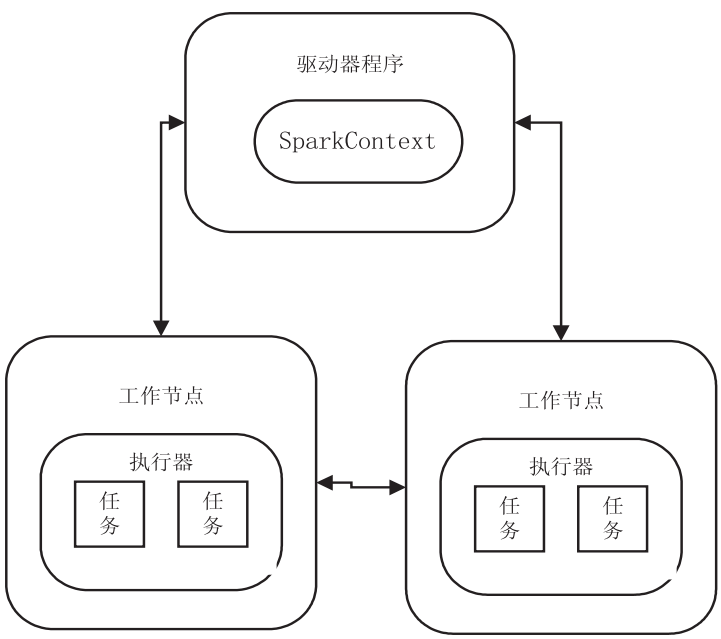

图 1　Spark 运行原理

4 实验及结果分析

4.1 实验数据集与实验环境

实验采用 MovieLens 网站提供的 1 MB 数据集,文中使用了其中的 ratings. dat 和 users. dat 两个数据集,ratings. dat 是一百多万条用户电影评分数据,users. dat 是参与评分的六千多名用户特征信息。

实验环境为虚拟机上架设的三个节点的 Hadoop 集群,系统为 Ubuntu16. 04,Spark 为 2. 1. 0 版本,运行在 Hadoop 集群,其依赖于 Yarn,HDFS 作为存储平台,采用 Python 语言进行实验编程。

4.2 实验设计与结果分析

首先用传统 Item-based 推荐算法,分别用余弦相似度和 Pearson 相关性相似度对数据集进行实验。实验数据共 1 000 209 条评分数据,按 4∶1 分为训练数据和测试数据,由于数据量过大,仅取邻居数为 10,对此数据集进行了三次实验,运行时间都在 3 000 s 以上,每次实验都需要耗费大量时间。MAE(平均绝对误差)如表 1 所示。

表 1 传统算法实验结果

MAE	第一次实验	第二次实验	第三次实验
余弦相似度	1.020 2	1.018	1.019 2
Pearson 相关性相似度	1.020 6	1.022 8	1.022 9

如表 1 所示,两种相似度推荐算法测试出的 MAE 都大于 1,可见传统推荐算法推荐效果并不理想。

感知用户年龄的 Item-based 协同过滤推荐算法中分类标识规则为:18 岁以下、18 岁 ~ 24 岁、25 岁 ~ 34 岁、35 岁 ~ 44 岁、45 岁 ~ 49 岁、50 岁 ~ 55 岁、56 岁以上,分别标识为 1、18、25、35、45、50 和 56。每组用户数量与总数量相比大大减少,比例最大的不到 40%,最小的为 2.7%。实验结果显示不同用户组实验运行时间最长为 2 934 s,最短为 82 s,可见改进算法在很大程度上提高了推荐的实时性。

利用改进算法同样进行了两组实验,实验结果如图 2 ~ 图 5 所示。

图 2 和图 3 是第一组采用传统相似度的各年龄段实验结果,18 岁以下和 18 ~ 24 岁这两组用户的 MAE 在 0. 7 ~ 0. 9 之间,其余用户组实验的 MAE 结果可以控制在 0. 7 ~ 0. 8 之间,比 18 岁以下和 18 至 24 岁两组用户的推荐结果更为精确。这说明,24 岁以下用户爱好并不稳定,该分组数据质量不高,随着年龄增长用户兴趣逐渐趋于稳定,推荐准确度也相应提高[15]。从项目邻居数的选取角度来看,随着邻居数目的增大,MAE 变化率不断减小。与传统推荐算法相比,MAE 都在 1 以下,而且最小达 0. 706 0,精确度有大幅提高。

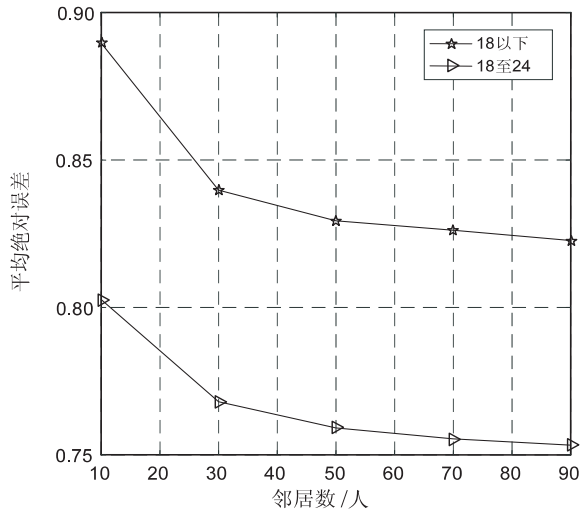

图 2 第一组实验结果(1)

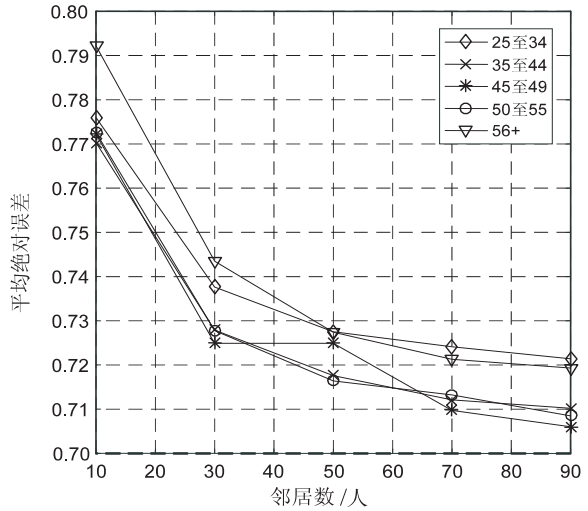

图 3 第一组实验结果(2)

图 4 和图 5 是第二组实验结果,相似度采用 Pearson 相关性相似度,实验结果同样显示 18 岁以下和 18 ~ 24 岁这两组用户 MAE 偏高,其余用户组使用该算法 MAE 控制在 0. 65 ~ 0. 8 之间,推荐精度大大提高。

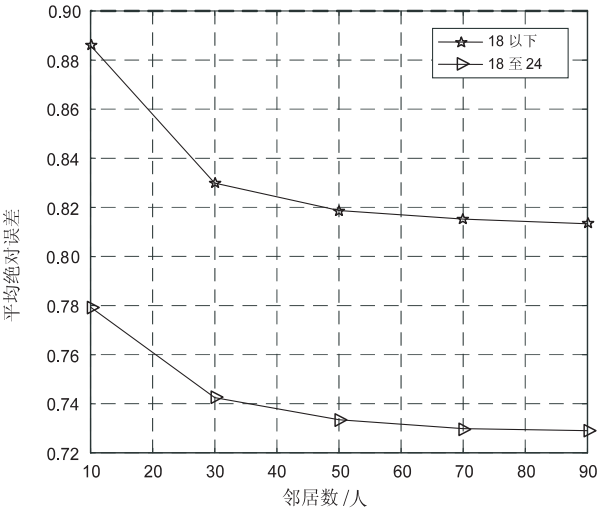

图 4 第二组实验结果(1)

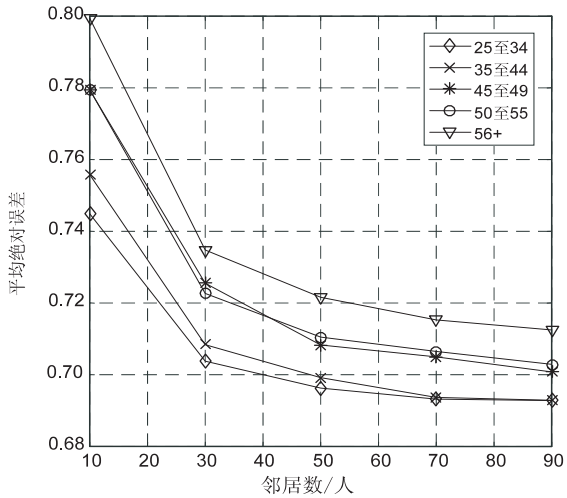

图 5　第二组实验结果(2)

由上述实验结果可以说明该算法能够保证推荐精度,提高推荐实时性,并且运行在分布式集群上具有可扩展性。该算法适用于计算资源有限,需要减少计算量,能够与用户有效交互,为用户提供精准个性化推荐的推荐系统。

5　结束语

针对传统 Item-based 协同过滤算法存在的相关问题,提出一种感知用户年龄的 Item-based 协同过滤推荐算法。新用户进入系统时根据年龄信息,利用相应组内的用户历史信息为其推荐,降低了计算相似度阶段的计算量;采用加权相似度提高了推荐准确度;并且该算法运行在 Spark 分布式集群,具有可扩展性。通过实验表明,该算法在计算效率和推荐精度方面都有一定程度的改善。

参考文献:

[1] NAIMI A I, WESTREICH D J. Big data: a revolution that will transform how we live, work, and think[J]. American Journal of Epidemiology, 2014, 179(9): 1143-1144.

[2] BA Qilong, LI Xiaoyong, BAI Zhongying. Clustering collaborative filtering recommendation system based on SVD algorithm[C]//IEEE international conference on software engineering and service science. Beijing, China: IEEE, 2013: 963-967.

[3] 高发展,黄梦醒,张婷婷. 综合用户特征及专家信任的协作过滤推荐算法[J]. 计算机科学,2017,44(2):103-106.

[4] 谢霖铨,梁博群. 结合用户特征分类和动态时间的协同过滤推荐[J]. 计算机工程与应用,2017,53(6):80-84.

[5] 吴　湖,王永吉,王　哲,等. 两阶段联合聚类协同过滤算法[J]. 软件学报,2010,21(5):1042-1054.

[6] ZHAO Zhidan, SHANG Mingsheng. User-based collaborative-filtering recommendation algorithms on Hadoop[C]//International conference on knowledge discovery and data mining. Phuket, Thailand: IEEE, 2010: 478-481.

[7] 田保军,胡培培,杜晓娟,等. Hadoop 下基于聚类协同过滤推荐算法优化的研究[J]. 计算机工程与科学,2016,38(8):1615-1624.

[8] 罗　辛,欧阳元新,熊　璋,等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报,2010,33(8):1437-1445.

[9] 明小红. 基于用户聚类的协同过滤推荐算法研究[D]. 北京:北京交通大学,2017.

[10] 徐江辉. 基于 Hadoop 的聚类协同过滤推荐算法研究及应用[D]. 长沙:湖南大学,2016.

[11] 唐积益. 推荐系统中相似度计算方法的研究[D]. 镇江:江苏科技大学,2015.

[12] 廖　彬,张　陶,国冰磊,等. 基于 Spark 的 ItemBased 推荐算法性能优化[J]. 计算机应用,2017,37(7):1900-1905.

[13] JIN Chen, LIU Ruoqian, CHEN Zhengzhang, et al. A scalable hierarchical clustering algorithm using spark[C]//IEEE first international conference on big data computing service and applications. Redwood City, CA, USA: IEEE, 2015: 418-426.

[14] KARAU H. Spark 快速大数据分析[M]. 北京:人民邮电出版社,2015.

[15] KURDIJA A S, SILIC M, VLADIMIR K, et al. Efficient global correlation measures for a collaborative filtering dataset[J]. Knowledge-Based Systems, 2018, 147(1): 36-42.