

基于特征和项目近邻的混合推荐算法研究

苏晓云,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:针对传统的协同过滤算法在推荐过程中存在的可扩展性差、推荐准确性低等问题,提出了一种基于动态加权的混合协同过滤算法(ItemBase_ALS collaborative filter, IACF)。该算法将基于项目的协同过滤算法(ItemBase CF)与基于矩阵分解的ALS推荐算法按照一定的权重进行混合,并在分布式平台Spark上得以实现,有效解决了算法扩展性问题。该混合算法首先分别利用ItemBase CF和ALS算法进行初步预测,然后选取能够反映其各自特性的因素,即项目近邻和隐藏特征,按照权重公式进行融合从而得到最终预测结果。通过调整权重比例,可以突出某一算法的特性,满足不同的推荐需求。实验选用MovieLen电影评分数据集,实验结果表明,混合协同过滤算法较之传统单个算法,既能体现其各自特点及变化规律,在可扩展性、准确性上也有所改善。

关键词:协同过滤;扩展性;Spark平台;动态加权

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2019)09-0071-05

doi:10.3969/j.issn.1673-629X.2019.09.014

Research on Hybrid Recommendation Algorithm Based on Feature and Item Nearest Neighbor

SU Xiao-yun, ZHU Yong-zhi

(School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China)

Abstract: A hybrid collaborative filtering algorithm (ItemBase_ALS collaborative filtering, IACF) based on dynamic weighting is proposed to solve the problems of poor scalability and low recommendation accuracy in the traditional collaborative filtering algorithm. The algorithm combines the item-based collaborative filtering algorithm (ItemBase CF) and the matrix factor-based ALS recommendation algorithm according to certain weights and is implemented on the distributed platform Spark which effectively solves the problem of scalability. The hybrid algorithm first uses ItemBase CF and ALS algorithms to make preliminary prediction respectively, and then selects the factors that can reflect their respective characteristics, that is, item nearest neighbor and hidden feature, and fuses them according to the weight formula to get the final prediction results. By adjusting the weight ratio, the characteristics of an algorithm can be highlighted to meet different recommendation requirements. The experiment on MovieLen dataset shows that the hybrid collaborative filtering algorithm can not only reflect their own characteristics and change rules, but also improve the scalability and accuracy.

Key words: collaborative filtering; scalability; Spark platform; dynamic weighting

0 引言

在互联网和信息技术飞速发展的背景下,网络中的数据呈现爆炸性增长的趋势,同时也存在大量冗余,如何从这些过量数据中高效准确地提取出所需信息成为当前亟待解决的难题。推荐系统正是解决这一难题的重要手段。

推荐系统^[1]旨在为用户提供个性化的产品或服务推荐,通过处理日益增多的过载信息来改善当前难

以从大量冗余数据中检索有效信息的状况,从而提供个性化服务。早期研究人员就开始关注明确依赖评级结构的推荐问题^[2],基于项目间或项目与用户间交互的相关信息来对用户兴趣进行预测,进而向特定的用户推荐最适合的项目,以生成个性化推荐。

1 相关研究

推荐算法是推荐系统的核心,协同过滤^[3]是推荐

收稿日期:2018-11-01

修回日期:2019-03-06

网络出版时间:2019-04-24

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:苏晓云(1996-),女,硕士研究生,研究方向为分布式计算、大数据;祝永志,教授,硕士,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.20190424.1051.052.html>

算法中最流行的方法,它从用户行为中收集大量的数据来进行预测,依赖于用户和项目之间的关系;混合推荐算法是指综合不同推荐算法的优势从而提高推荐算法准确性的一种推荐方法。

近年来,国内外研究人员对推荐算法的研究呈现出多样化的趋势。如文献[4]在传统矩阵分解模型 SVD 的基础上提出了最小二乘 ALS 算法;文献[5]将协同过滤应用到 Spark 平台上,实现了算法并行化;文献[6]基于用户相似度定义了一种社交网络中的属性相似度,改进了协同过滤算法;文献[7]提出了一种基于用户行为特征的动态加权混合推荐算法,有效降低了推荐的误差,提高了推荐精度。

文中将基于项目的协同过滤推荐算法(ItemBase CF)与基于矩阵分解的最小二乘法(ALS)进行动态加权混合,并将该混合算法在 Spark 平台上进行实现,以提高算法的并行性和扩展性,以及推荐准确性。

2 协同过滤推荐算法

2.1 ItemBase 协同过滤推荐算法

2.1.1 相似度计算

ItemBase CF 根据项目间的相似性,对于给定的某个用户根据其历史行为信息,将相似度最高的物品推荐给用户^[8]。计算物品之间相似度的方法^[9]主要有:

(1) 余弦相似度。

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

(2) 修正的余弦相似度。

$$\text{sim}(i, j) = \frac{\sum_{u \in U_{ij}} (R_{ui} - \bar{R}_u)(R_{uj} - \bar{R}_u)}{\sqrt{\sum_{u \in U_{ij}} (R_{ui} - \bar{R}_u)^2} \sqrt{\sum_{u \in U_{ij}} (R_{uj} - \bar{R}_u)^2}} \quad (2)$$

(3) Pearson 相关性相似度。

$$\text{sim}(i, j) = \frac{\sum_{u \in U_{ij}} (R_{ui} - \bar{R}_i)(R_{uj} - \bar{R}_j)}{\sqrt{\sum_{u \in U_{ij}} (R_{ui} - \bar{R}_i)^2} \sqrt{\sum_{u \in U_{ij}} (R_{uj} - \bar{R}_j)^2}} \quad (3)$$

其中, $\text{sim}(i, j)$ 为项目 i 和项目 j 之间的相似度; U_{ij} 为参与评分的所有用户的集合; R_{ui} 、 R_{uj} 分别为用户 u 对项目 i 和项目 j 的评分; \bar{R}_i 和 \bar{R}_j 分别为项目 i 和项目 j 的平均评分; \bar{R}_u 表示用户 u 对所有项目打分的平均值。

2.1.2 评分预测

在 ItemBase CF 中采用基于项目均值的加权平均值来计算预测评分^[10],公式如下:

$$P_{ui} = \bar{R}_i + \frac{\sum_{j \in \text{KNN}_i} \text{sim}(i, j)(R_{uj} - \bar{R}_j)}{\sum_{j \in \text{KNN}_i} |\text{sim}(i, j)|} \quad (4)$$

其中, P_{ui} 为用户 u 对项目 i 的预测评分; \bar{R}_i 为项目 i 的平均分; KNN_i 为项目 i 的近邻项目集合; $\text{sim}(i, j)$ 为项目 i, j 之间的相似度; R_{uj} 为用户 u 对项目 j 的评分; \bar{R}_j 为所有用户对项目 j 的评分平均值。

ItemBase CF 通过计算项目间的相似度,将相似度较高的前 k 个项目作为该项目的近邻集合^[11],通过调整 k 值的大小可以改善推荐结果的精确度。

2.2 基于 ALS 的协同过滤推荐算法

基于矩阵分解的 ALS 协同过滤推荐算法^[12]的原理是通过最小化 Frobenius 损失函数找到两个低秩矩阵 U 、 V 来最大程度逼近原矩阵 R ,即:

$$R_{m \times n} \approx U_{m \times d} V_{d \times n}^T \quad (5)$$

其中, $R_{m \times n}$ 为原始评分矩阵; $U_{m \times d}$ 为用户特征矩阵; $V_{d \times n}$ 为项目特征矩阵, $d \ll \min(m, n)$ 。

Frobenius 损失函数如下:

$$L(U, V) = \sum_{u, v} (r_{ij} - u_i v_j^T)^2 \quad (6)$$

为防止过拟合问题,加入正则化参数,如下:

$$L(U, V) = \sum_{u, v} (r_{ij} - u_i v_j^T)^2 + \lambda (\|u_i\|_F^2 + \|v_j\|_F^2) \quad (7)$$

在求解过程中,通过交替迭代的方法,先固定 V ,将 $L(U, V)$ 对 u_i 求偏导,得到:

$$u_i = (V^T V + \lambda I)^{-1} V^T r_i \quad (8)$$

同理,对 v_j 求偏导得到:

$$v_j = (U^T U + \lambda I)^{-1} U^T r_j \quad (9)$$

式 8、式 9 中, r_i 表示由用户 i 的评分组成的向量; r_j 表示由项目 j 组成的评分向量; I 表示 $d \times d$ 的单位矩阵。

在上述过程中不断迭代,直到 RMSE 值最小或达到最大迭代次数^[13]为止,将迭代产生的特征向量相乘得到 $R' = UV^T$ 。

3 混合协同过滤推荐算法 IACF

3.1 由单机转向 Spark 分布式平台

传统的单机环境目前已不能满足大数据日益增长的需求,在运行具有较高复杂性的推荐算法时,实际效率是差强人意的。利用分布式平台将规模较大较复杂的作业分配给不同的节点,能够充分利用其存储和计算性能。

Spark 分布式平台是基于内存的通用并行大数据计算引擎,在迭代速度上优于 MapReduce^[14],通过弹性分布式数据集(RDD)进行数据操作,将算法流程中

的任何一个点映射到集群节点的内存中,后续过程不必反复从磁盘中读取数据,适用于多次迭代算法^[15]。

Spark 的工作原理如图 1 所示。

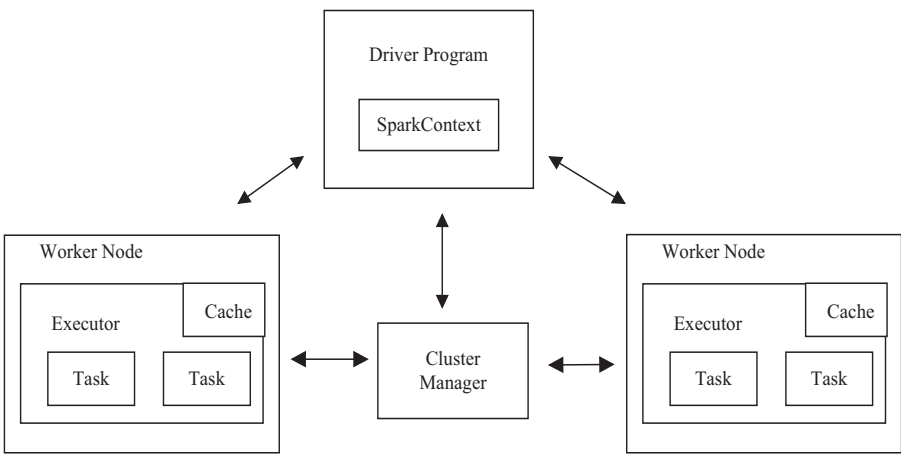


图 1 Spark 工作原理

3.2 混合推荐原理

基于项目的协同过滤算法与基于矩阵分解的 ALS 算法各有其优势,文中采用动态加权的方式将两种算法的预测结果进行混合,得到混合协同过滤推荐算法 IACF(ItemBase_ALS collaborative filter)。该算法的主要原理是根据加权公式将 ItemBase CF 的评分 I_{ui} 和 ALS 算法的评分 A_{ui} 进行混合,得到最终预测评分 R_{ui} ,加权公式如下:

$$R_{ui} = \lambda I_{ui} + (1 - \lambda) A_{ui}$$
 (10)

其中, λ 是加权因子,其取值会影响最终混合结果的精确性。

基于项目的协同过滤算法与项目近邻的选取有关;ALS 算法与迭代次数、隐含特征因子的个数及正则项系数 lambda 有关。通过多次实验验证得知,隐含特征因子对算法准确性影响较大,因此文中选择隐含特征因子作为关键影响因素,固定迭代次数和 lambda。在混合算法中,通过调整项目近邻 n 和隐藏特征因子 r 的取值来确定 λ ,如下:

$$\lambda = \frac{n}{r + n}$$
 (11)

$$1 - \lambda = \frac{r}{r + n}$$
 (12)

3.3 算法描述

ItemBase CF 和 ALS 混合协同过滤推荐算法 (IACF)描述如下:

输入:用户-项目评分矩阵;

输出:预测结果评分。

(1)利用修正的余弦相似度公式(式 2)计算项目间相似度;

(2)根据步骤 1 中的结果选择相似度最高的 k 个项目作为项目近邻,通过评分预测公式(式 4)进行预测,得出预测结果 I_{ui} ;

(3)调整隐藏特征 r 的个数,选择 ALS 算法进行预测,得出预测结果 A_{ui} ;

(4)根据式 11 确定 λ ;

(5)根据加权公式(式 10)将步骤 2、3 的结果进行混合,得出最终预测结果。

4 实验及结果分析

4.1 实验数据与环境

采用 Grouplens 提供的 MovieLens 数据集中约 700 位用户对 9 000 部电影的最新评分数据集,约 10 万条,评分范围为 1 ~ 5 分,分值越高表示用户对该电影的喜好程度越高。评分数据集格式如表 1 所示。

表 1 评分数据表格式

UserID	MovieID	Rating	Timestamp
用户编号	电影编号	评分	时间戳

实验环境是在 VirtureBox 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 集群上,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 评价指标

预测评分的精准度主要通过均方根误差 (RMSE) 和平均绝对误差 (MAE) 两种指标体现,两者的值越小,表示预测准确度越高。若 p_{ui} 表示预测评分, r_{ui} 表示实际评分, N 表示测试集中评分个数,有如下定义:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (p_{ui} - r_{ui})^2}{N}}$$
 (13)

$$MAE = \frac{\sum_{i=1}^n |p_{ui} - r_{ui}|}{N}$$
 (14)

4.3 实验设计与结果分析

实验首先将评分数据集按照 8 : 2 的比例划分为

训练集和测试集,将训练集用于预测评分,测试集中的数据作为参考来评估算法预测的准确性。

设置两组实验,涉及到 ALS 算法时按经验选取迭代次数为 20 进行迭代直到收敛,lambda=0.01。

第一组实验比较传统的 ItemBase CF 与混合协同过滤算法 IACF,固定 ALS 算法的隐藏特征数 r ,改变项目近邻数 n 分别进行实验,结果如图 2 和图 3 所示。

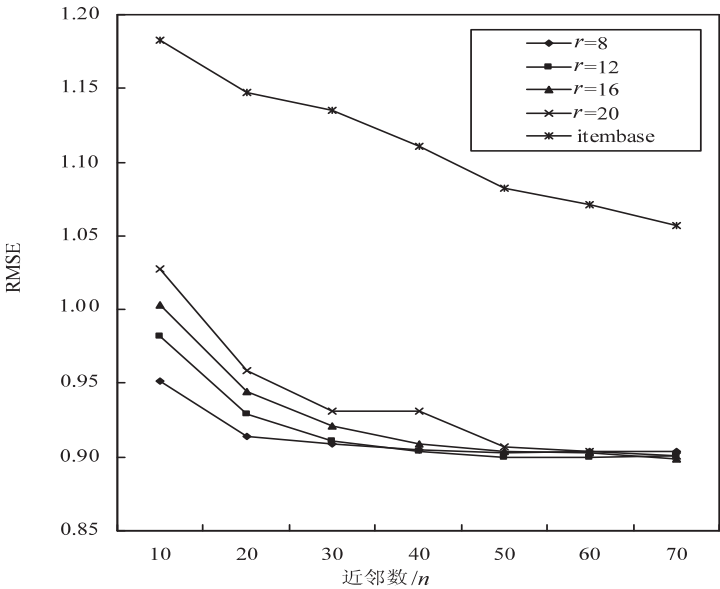


图 2 Itembase CF 与 IACF 算法的 RMSE 变化趋势

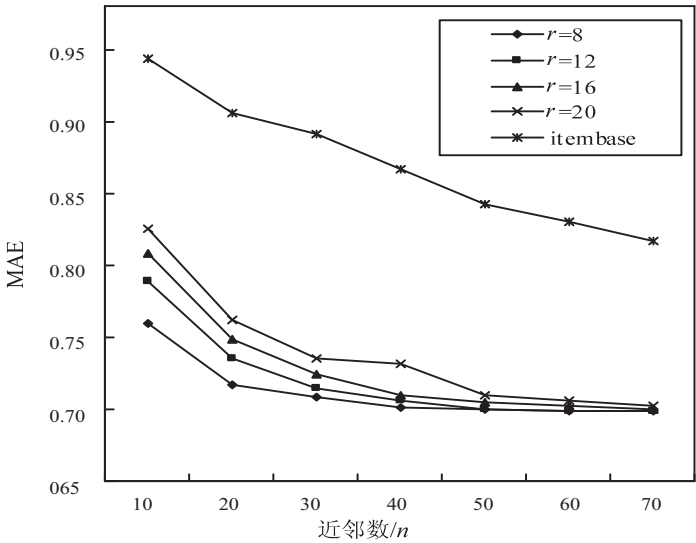


图 3 Itembase CF 与 IACF 算法的 MAE 变化趋势

从图中可以看出,传统 ItemBase 的 RMSE 最小值为 1.053 7,MAE 最小值为 0.816 6,其变化趋势大致相同,都随近邻数 n 的增大而减小;而混合算法 IACF 的 RMSE 最小值为 0.899 0,MAE 最小值为 0.698 2,二者均大幅度下降,推荐准确度明显提高,且与传统 ItemBase 算法的变化规律基本相同,随近邻数 n 的增加而减小,即 λ 的值越大,其 RMSE 和 MAE 越小,达到一定程度时趋于平稳。

第二组实验比较 ALS 协同过滤算法与混合协同过滤算法 IACF,通过确定的项目近邻数 n ,调整隐藏特征因子 r 来分别进行实验,实验结果如图 4 和图 5 所示。

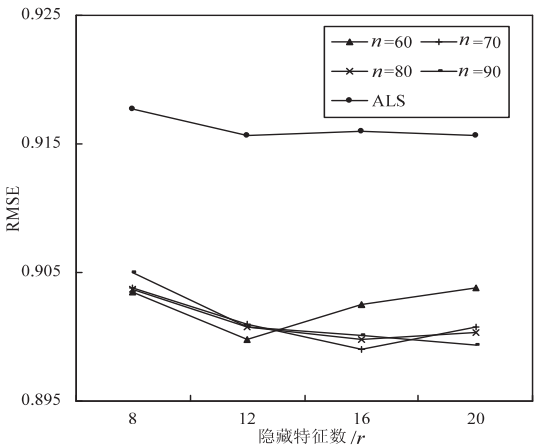


图 4 ALS 与 IACF 算法的 RMSE 变化趋势

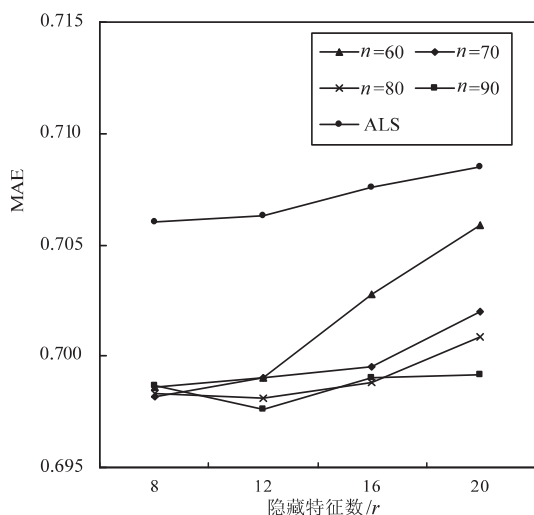


图5 ALS与IACF算法的MAE变化趋势

从图中可以看出, $r=12$ 时,传统 ALS 算法 RMSE 最小值为 0.915 6, $r=8$ 时, MAE 最小值为 0.706;融合了 ItemBase 的 IACF 算法,其 RMSE 和 MAE 值均有所减小, RMSE 范围在 0.899 0 ~ 0.905 0 之间, MAE 值保持在 0.697 6 ~ 0.705 9 间,推荐准确度明显提高。

综合两组实验结果,混合协同过滤算法 IACF 在 Spark 上的实现保证了算法的可扩展性,动态加权混合提高了算法推荐的准确性。

5 结束语

文中实现了由单机环境到分布式平台的转变,并提出了一种混合协同过滤推荐算法 IACF。该算法将 ItemBase CF 和 ALS 算法根据各自特点选取不同特征组合构造加权公式进行融合,通过调节不同参数来动态改变不同维度的权重,从而突出某一算法的特性,使得改进后的 IACF 能够满足不同需求。实验结果表明, IACF 算法在分布式平台 Spark 上的实现提高了算法的并行性与可扩展性,同时在评分预测及推荐精度上也有所改善。

参考文献:

[1] LU Jie, WU Dianshuang, MAO Mingsong, et al. Recommender system application developments [J]. Decision Support Systems, 2015, 74 (C): 12-32.

[2] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17 (6): 734-749.

[3] 陈孝通. 基于 Spark 的混合协同过滤推荐系统的研究与实现 [D]. 秦皇岛: 燕山大学, 2017.

[4] 李改, 李磊. 基于矩阵分解的协同过滤算法 [J]. 计算机工程与应用, 2011, 47 (30): 4-7.

[5] 徐新瑞, 孟彩霞, 周雯, 等. 一种基于 Spark 时效化协同过滤推荐算法 [J]. 计算机技术与发展, 2015, 25 (6): 48-55.

[6] 荣辉桂, 火生旭, 胡春华, 等. 基于用户相似度的协同过滤推荐算法 [J]. 通信学报, 2014, 35 (2): 16-24.

[7] 刘沛文, 陈华锋. 基于用户行为特征的动态权重混合推荐算法 [J]. 计算机应用与软件, 2017, 34 (4): 316-321.

[8] 李现伟. 基于 Spark 的推荐系统的研究 [D]. 杭州: 浙江理工大学, 2017.

[9] AHN H J. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem [J]. Information Sciences, 2008, 178 (1): 37-51.

[10] 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.

[11] 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.

[12] ZHOU Yunhong, WILKINSON D, SCHREIBER R, et al. Large-scale parallel collaborative filtering for the netflix prize [C]// International conference on algorithmic applications in management. Berlin: Springer, 2008: 337-348.

[13] 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.

[14] BHANDARKAR M. MapReduce programming with apache Hadoop [C]// IEEE international symposium on parallel & distributed processing. Atlanta, GA, USA: IEEE, 2010: 1.

[15] 杨志伟. 基于 Spark 平台推荐系统研究 [D]. 合肥: 中国科学技术大学, 2015.