融合用户对项目和属性偏好的协同过滤算法

王云超 刘 臻
(北京师范大学教育学部 北京 100875)

摘 要 协同过滤推荐算法是目前推荐系统领域中十分常用的方法。余弦相似度和 Pearson 相关系数是目前协同过滤推荐算法中计算相似度的两种常用算法。为提高协同过滤推荐算法的准确性,对相似度计算问题进行了研究,针对目前常用的余弦相似度和 Pearson 相关系数这两种相似度计算方法的不足,通过设计和引入调节因子,分别考虑用户在评分习惯和项目选择上的差异性,以对这两种传统的相似度算法进行优化和改进。另外,考虑到用户的偏好往往与项目所具有的属性有关,设计了衡量用户对属性偏好的参数,通过加权的方式将其与改进后的相似度算法进行融合,提出了一种融合用户评分习惯、项目选择差异及属性偏好的协同过滤推荐算法。在 MovieLens 数据集上进行的实验表明,相比于传统算法,提出的改进算法更为精确,平均绝对误差和均方根误差得到了明显的降低。

关键词 推荐系统,协同过滤,用户相似度,属性偏好,调节因子
中图法分类号 TP391 **文献标识码** A

Collaborative Filtering Algorithm Based on User's Preference for Items and Attributes

WANG Yun-chao LIU Zhen
(Faculty of Education, Beijing Normal University, Beijing 100875, China)

Abstract Collaborative filtering algorithm is one of the most successful and useful technologies in recommendation systems. Cosine similarity and Pearson correlation coefficient are two of the most widely used traditional algorithms to calculate the similarity in collaborative filtering algorithm. In order to reduce the error, an improved collaborative filtering recommendation algorithm was proposed in view of the disadvantages of the two traditional similarity algorithms. The two traditional algorithms were improved by importing two parameters, one of them was proposed for considering the rating habits of users, and the other was imported to measure the difference of items chosen by users. User's preference is related to project attributes, therefore, a parameter was designed to measure it. The new algorithm was constructed by the improved traditional algorithm and user's preference for attributes. The results of experiment on MovieLens dataset show that the proposed algorithm has lower mean absolute error (MAE) and root mean square error (RMSE), and has better performance by using the two parameters compared with traditional algorithms.

Keywords Recommendation system, Collaborative filtering, User similarity, Preference for attributes, Parameter for adjustment

伴随着互联网技术的快速发展,信息量的爆炸性增长使人们带来了大量可获取的资源,但同时也使得用户越来越难以从纷繁复杂的信息海洋中获取自己感兴趣的信息。当前人们主动获取需求信息最常用的方法是通过搜索引擎检索信息,但在用户潜在的兴趣目标以及满足用户个性化需求等方面,推荐系统的优势大大超越了搜索引擎。现今的推荐系统已经被广泛应用到了如电子商务、社交、影视和音乐、新闻等诸多领域。

高精确度的推荐算法是推荐系统的核心,协同过滤推荐算法是当前应用十分广泛的推荐算法,其原理是根据与目标用户相似的其他用户的历史行为记录来预测目标用户对目标项目的态度。在实际环境中,协同过滤推荐算法有了许多成功的应用,如进行音乐 CD 推荐的 Ringo 系统[1],对网络新闻进行推荐的 GoodNews 系统[2]和 GroupLens 系统[3]等。

由于协同过滤推荐算法在实际应用中发挥出了显著的作用,近年来越来越多的研究人员开展了对协同过滤推荐算法的研究。Bobadilla 等[4]提出了一种考虑用户或项目重要性的改进方法;Ortega 等[5]提出了使用 Pareto 优先的方式改善了用户最近邻居的选择问题;Wang 等[6]提出了一种利用汉明距离进行聚类的方法;邢春晓等[7]提出了一种基于时间和项目相似度的适应用户兴趣变化的改进方法;王吉源等[8]将活跃近邻作为考虑对象,提出了用户属性加权活跃近邻的改进方法;赵文涛等[9]对多种用户属性分配权重,提出了基于用户多属性与兴趣的改进方法。这些方法都从不同的角度进行了改进,提高了推荐的准确性。

1 基于用户的协同过滤推荐算法

目前,协同过滤推荐算法主要包括两种,即基于用户的协

本文受国家 863 科技支撑计划项目(2012AA12AA407),赛尔网络下一代互联网技术创新项目(NGII20170628)资助。
王云超(1991-),男,硕士生,主要研究方向为机器学习、推荐算法,E-mail:chester91@foxmail.com;**刘 臻**(1972-),男,博士,教授级高工,博士生导师,主要研究方向为数字化学习环境与工程、高分辨率图像变化检测方法与应用,E-mail:liuzhen@bnu.edu.cn(通信作者)。

同过滤推荐算法和基于项目的协同过滤推荐算法。基于用户的协同过滤推荐算法的思想是：相似的用户对于同一个项目具有相似的态度。利用已有用户的历史评价数据，可以通过相似度算法计算出各个用户之间的相似度，之后找出与目标用户相似度最高的 k 个邻居（k 个最近邻居），通过这些最近邻居来预测目标用户对目标项目的态度，最后根据预测得出的目标用户对各个目标项目的态度好恶情况的排序，向目标用户进行项目推荐。

1.1　算法流程

（1）构建用户-项目评分矩阵

各用户对各项目的评分数据集可以看作是一个矩阵 $R_{m\times n}$（见表 1），用集合 $U=\{u_1,u_2,u_3,\cdots,u_m\}$ 表示数据集中的 m 个用户，用集合 $V=\{v_1,v_2,v_3,\cdots,v_n\}$ 表示数据集中的 n 个项目，用 $r_{i,j}$ 表示 U 中第 i 个用户 u_i 对 V 中第 j 个项目 v_j 的评分。

表 1　用户-项目评分矩阵

	v_1	v_2	…	v_j	…	v_n
u_1	$r_{1,1}$	$r_{1,2}$	…	$r_{1,j}$	…	$r_{1,n}$
u_2	$r_{2,1}$	$r_{2,2}$	…	$r_{2,j}$	…	$r_{2,n}$
…	…	…	…	…	…	…
u_i	$r_{i,1}$	$r_{i,2}$	…	$r_{i,j}$	…	$r_{i,n}$
…	…	…	…	…	…	…
u_m	$r_{m,1}$	$r_{m,2}$	…	$r_{m,j}$	…	$r_{m,n}$

评分通常有 3 种模式（用户对项目没有评分的 $r_{i,j}$ 置空）：用 0 或 1 的二值法来标注用户对项目喜欢或不喜欢；用 -1，0，1 三值法来标注用户对项目不喜欢、中性、喜欢；用多值法来标注用户对项目的喜欢程度（数值越高表示越喜欢）。

（2）计算用户相似度，确定最近邻居

用户 u_a 和用户 u_b 的相似度用 $sim(u_a,u_b)$ 来表示。假设目标用户为 u，首先根据（1）中的用户-项目评分矩阵来计算所有用户与 u 的相似度（具体计算方法会在后面进行详述），相似度的取值范围为 $[-1,1]$，相似度越趋近于 1，表明两者的相似度越高，相似度越趋近于 -1，表明两者的喜好情况越相反。之后选取与目标用户 u 相似度最高的 k 个邻居组成近邻集 $N=\{u_1,u_2,u_3,\cdots,u_k\}(u\notin N)$，并将其作为对用户 u 产生推荐的依据。选择邻居的方法通常有两种：一种是将相似度大于某一阈值的用户的集合作为 u 的最近邻居，生成近邻集 N；另一种是将相似度最高的 k 个用户的集合作为 u 的最近邻居，生成近邻集 N。

（3）产生推荐

根据近邻集 N 以及 N 中的各个最近邻居，可以对目标用户 u 对其他未评分项目 v 的评分进行预测，预测分值用 $p_{u,v}$ 表示，可以生成预测分值集合 $P=\{p_{u,v_1},p_{u,v_2},\cdots,p_{u,v_n}\}$，根据 P 中各分值的排序情况，向用户推荐其可能感兴趣的项目。预测计算[10]：

$$P(u,v)=\overline{r_u}+\frac{\sum\limits_{u'\in N} sim(u,u')\cdot(r_{u',v}-\overline{r_{u'}})}{\sum\limits_{u'\in N} sim(u,u')}$$

其中，$\overline{r_u}$ 表示用户 u 对所有曾经评分过的项目的平均评分值，$\overline{r_{u'}}$ 表示用户 u' 对所有曾经评分过的项目的平均评分值。

1.2　传统的相似度计算方法

1.2.1　余弦相似度

在基于用户的协同过滤推荐算法中，使用余弦相似度来计算用户间的相似度：

$$sim(a,b)_{\cos}=\frac{\vec{a}\times\vec{b}}{\|\vec{a}\|\times\|\vec{b}\|}=\frac{\sum\limits_{v\in S} r_{a,v}\cdot r_{b,v}}{\sqrt{\sum\limits_{v\in S} r_{a,v}^2}\sqrt{\sum\limits_{v\in S} r_{b,v}^2}}$$

其中，a 和 b 分别表示两个不同的用户，$sim(a,b)_{\cos}$ 表示用户 a 和用户 b 的相似度，S 为 a 和 b 共同评分的项目的集合，$r_{a,v}$ 表示用户 a 对项目 v 的评分，$r_{b,v}$ 表示用户 b 对项目 v 的评分。

1.2.2　Pearson 相关系数

Pearson 相关系数是一种常用的计算相似性的方法，在基于用户的协同过滤推荐算法中，计算用户间相似度的公式为：

$$sim(a,b)_{\text{Pearson}}=\frac{\sum\limits_{v\in S}(r_{a,v}-\overline{r_a})(r_{b,v}-\overline{r_b})}{\sqrt{\sum\limits_{v\in S}(r_{a,v}-\overline{r_a})^2}\sqrt{\sum\limits_{v\in S}(r_{b,v}-\overline{r_b})^2}}$$

其中，a 和 b 分别表示两个不同的用户，$sim(a,b)_{\text{Pearson}}$ 表示用户 a 和用户 b 的相似度，S 为用户 a 和用户 b 共同评分的项目的集合，$r_{a,v}$ 表示用户 a 对项目 v 的评分，$r_{b,v}$ 表示用户 b 对项目 v 的评分，$\overline{r_a}$ 表示用户 a 对所有曾经评分过的项目的平均评分值，$\overline{r_b}$ 表示用户 b 对所有曾经评分过的项目的平均评分值。

2　改进的协同过滤推荐算法

2.1　传统相似度算法的改进

传统的相似度算法在实际使用过程中暴露出一些弊端。如果两个用户 u_a 和 u_b 对项目集合 $V=\{v_1,v_2,v_3,v_4,v_5\}$ 的评分集合分别为 $R_a=\{1,1,1,1,1\}$ 和 $R_b=\{5,5,5,5,5\}$，则使用余弦相似度算法计算得到的用户相似度为 1，这显然不符合真实情况；如果用户 u_c 对项目集合的评分集合为 $R_c=\{1,1,2,1,2\}$，显然 u_a 和 u_c 之间的相似度要高于 u_a 和 u_b，但使用余弦相似度计算的结果却相反。如果两个用户 u_a 和 u_b 对项目集合 $V=\{v_1,v_2,v_3,v_4,v_5\}$ 的评分集合分别为 $R_a=\{1,1,1,2,2\}$ 和 $R_b=\{4,4,4,5,5\}$，则使用 Pearson 计算得到的用户相似度为 1，但真实情况显然也并非如此。

在传统的余弦相似度和 Pearson 相关系数的相似度计算方法中，都采用的是两个用户共同评分的项目集合作为计算依据，并没有考虑到两个用户项目选择上的差异。此外，这两种传统的相似度计算方法也没有考虑到由于两个用户评分习惯不同而可能造成的评分情况不同，导致了上述相似度计算结果发生偏差。

为解决上述传统相似度计算方法的弊端，本研究提出和引入了两个参数（参数 p 和 Jaccard 系数）来对余弦相似度和 Pearson 相关系数的计算方法进行修正。

上述两种传统的相似度计算方法都没有考虑两个用户之间的评分差异，由此造成了误差。针对两个用户之间存在评分标准、评分离散度等评分习惯的差异，设计了参数 p：

$$p=2\times\left(1-\frac{1}{1+\mathrm{e}^{-\left|\frac{\sum\limits_{v\in S}|r_{a,v}-r_{b,v}|}{|S|}\right|}}\right)$$

其中，a 和 b 分别表示两个不同的用户，S 为 a 和 b 共同评分的项目集合，$|S|$ 表示 a 和 b 共同评分的项目个数，$r_{a,v}$ 表示用户 a 对项目 v 的评分，$r_{b,v}$ 表示用户 b 对项目 v 的评分。参数 $p(p\in(0,1])$ 与两个用户的评分差异呈负相关关系，当用户 a 和用户 b 对相同的项目普遍评分差异较大时，参数 p 的值较小；当用户 a 和用户 b 对相同的项目普遍评分差异较小时，参数 p 的值较大。

由于上述两种传统相似度计算方法中都是将两个用户共同的评分项目作为计算集合，没有考虑到在实际情况中两个

用户各自进行过评分的项目集合往往并不完全相同。由于用户对项目进行过评分，本身就说明用户对该项目可能有兴趣，因此对项目的选择的异同，也是判断用户相似度的重要依据。据此引入 Jaccard 系数：

$$J(A,B)=\frac{|I_A\cap I_B|}{|I_A\cup I_B|}$$

其中，I_A 表示用户 a 进行过评分的项目集合，I_B 表示用户 b 进行过评分的项目集合。当用户 a 和用户 b 共同评分过的项目占比较大时，参数 $J(A,B)$ 也较大；当用户 a 和用户 b 共同评分过的项目占比较小时，参数 $J(A,B)$ 也较小。$J(A,B)\in[0,1]$。

最后，计算出改进后的余弦相似度和 Pearson 相关系数。

$$sim(a,b)_{\cos'}=p\times J(A,B)\times\frac{\sum_{v\in S}r_{a,v}\cdot r_{b,v}}{\sqrt{\sum_{v\in S}r_{a,v}^2}\sqrt{\sum_{v\in S}r_{b,v}^2}}$$

$$=2\times\left(1-\frac{1}{1+e^{-\left|\frac{\sum_{v\in S}|r_{a,v}-r_{b,v}|}{|S|}\right|}}\right)\times\frac{|I_A\cap I_B|}{|I_A\cup I_B|}\times\frac{\sum_{v\in S}r_{a,v}\cdot r_{b,v}}{\sqrt{\sum_{v\in S}r_{a,v}^2}\sqrt{\sum_{v\in S}r_{b,v}^2}}$$

$$sim(a,b)_{\text{Pearson}'}=p\times J(A,B)\times\frac{\sum_{v\in S}(r_{a,v}-\overline{r_a})(r_{b,v}-\overline{r_b})}{\sqrt{\sum_{v\in S}(r_{a,v}-\overline{r_a})^2}\sqrt{\sum_{v\in S}(r_{b,v}-\overline{r_b})^2}}$$

$$=2\times\left(1-\frac{1}{1+e^{-\left|\frac{\sum_{v\in S}|r_{a,v}-r_{b,v}|}{|S|}\right|}}\right)\times\frac{|I_A\cap I_B|}{|I_A\cup I_B|}\times\frac{\sum_{v\in S}(r_{a,v}-\overline{r_a})(r_{b,v}-\overline{r_b})}{\sqrt{\sum_{v\in S}(r_{a,v}-\overline{r_a})^2}\sqrt{\sum_{v\in S}(r_{b,v}-\overline{r_b})^2}}$$

2.2 用户对属性的偏好

用户对项目的偏好往往与项目自身所具有的某些属性有关，这一特点表现在两个方面：1)用户对具有某种属性的项目的选择数量在该用户所有选择数量中的占比，即用户对具有某种属性的项目的打分次数在该用户所有打分次数中的占比；2)用户对具有某种属性的项目的偏好程度。据此，可以将用户 u 对于属性 c 的偏好定义为：

$$h_{u,c}=\frac{t_{u,c}}{t_u}\times\overline{r_{u,c}}$$

其中，$t_{u,c}$ 表示用户 u 对具有属性 c 的项目进行打分的次数，即用户 u 所有打分项目中具有属性 c 的项目的个数；t_u 表示用户 u 打分过的项目数；$\overline{r_{u,c}}$ 表示在用户 u 对其打分项目中具有属性 c 的项目的评分的平均值。

用户 u 对于各属性的好恶程度可以用向量 $\overrightarrow{h_u}=(h_{u,1},h_{u,2},\cdots,h_{u,w})$ 表示，其中 w 是属性的总个数。由此可以构建出用户-属性评分矩阵，如表 2 所列。

表 2 用户-属性评分矩阵

	c_1	c_2	…	c_j	…	c_w
u_1	$h_{1,1}$	$h_{1,2}$	…	$h_{1,j}$	…	$h_{1,w}$
u_2	$h_{2,1}$	$h_{2,2}$	…	$h_{2,j}$	…	$h_{2,w}$
…	…	…	…	…	…	…
u_i	$h_{i,1}$	$h_{i,2}$	…	$h_{i,j}$	…	$h_{i,w}$
…	…	…	…	…	…	…
u_m	$h_{m,1}$	$h_{m,2}$	…	$h_{m,j}$	…	$h_{m,w}$

两位用户 a 和 b 之间对于属性偏好的相似度可以定义为：

$$sim_2(a,b)=\frac{\sum_{i=1}^{w}\overrightarrow{h_a}\times\overrightarrow{h_b}}{\sqrt{\sum_{i=1}^{w}\overrightarrow{h_a}^2}\sqrt{\sum_{i=1}^{w}\overrightarrow{h_b}^2}}$$

2.3 融合相似度算法

对于上述提出的传统相似度算法的改进和针对用户对属性偏好的考虑，可以通过加权的方式进行融合，计算出最终的用户相似度。融合后的用户相似度计算公式为：

$$sim(a,b)=\beta\cdot sim_1(a,b)+(1-\beta)\cdot sim_2(a,b)$$

其中，$sim_1(a,b)$ 为上述改进后的余弦相似度算法或改进后的 Pearson 相关系数算法。β 的取值范围为 $[0,1]$，该参数通过人工调控的方式进行设置，选取适当的值将两者进行更加完美的融合，使得计算结果更加准确。

3 实验结果及分析

3.1 数据集

本实验采用目前在推荐算法研究领域常用的 MovieLens 数据集[11]中 100k 的版本。其中包含 1997 年 9 月 19 日至 1998 年 4 月 22 日期间，943 名用户对 1682 部电影的 10 万条评分记录，此数据集中，每位用户至少对其中的 20 部电影进行过打分，分值为 1 分、2 分、3 分、4 分、5 分中的一项，用来表示用户对电影的喜爱程度，评分越高，表示用户越喜欢该电影。

本实验使用 Python 语言，在 Pycharm 开发环境下进行算法代码的编写和测试。将 Movielens 数据集中 100K 的版本中的数据按照 8∶2 的比例进行随机划分，其中 80%的数据作为训练集数据，另外 20%的数据作为测试集数据。

3.2 评价指标

本实验主要采用平均绝对误差(Mean Absolute Error，MAE)作为评价指标，另外还将采用均方根误差(Root Mean Square Error，RMSE)进行辅助检验。MAE 和 RMSE 是推荐算法研究领域被广泛使用的评价指标，具体计算公式如下：

$$MAE=\frac{1}{N}\sum_{i=1}^{N}|p_i-q_i|$$

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}|p_i-q_i|^2}$$

其中，N 表示预测评分集合中的评分个数，p_i 表示通过推荐算法计算得到的用户对项目的预测评分，q_i 表示数据集中用户对该项目的真实评分。MAE 和 RMSE 的大小均与推荐的准确性呈负相关，MAE 和 RMSE 越小，表明推荐的准确性越高。

3.3 实验结果及分析

3.3.1 实验 1

基于上述对 MovieLens 数据集划分后的实验集和测试集，对比余弦相似度进行改进前后的 MAE 和 RMSE 结果，如图 1 和图 2 所示；同时，对比 Pearson 相关系数进行改进前后的 MAE 和 RMSE 结果，如图 3 和图 4 所示。

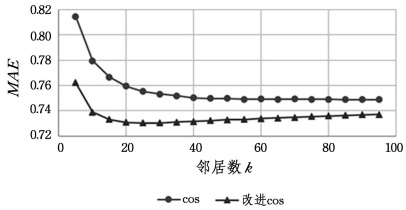

图 1 余弦相似度改进前后 MAE 的对比

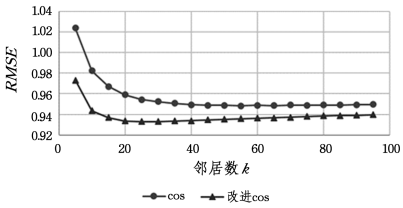

图 2　余弦相似度改进前后 RMSE 的对比

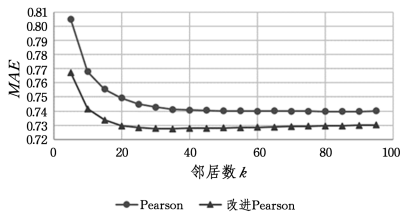

图 3　Pearson 相关系数改进前后 MAE 的对比

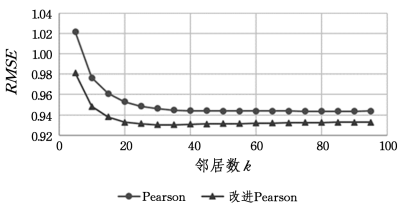

图 4　Pearson 相关系数改进前后 RMSE 的对比

通过以上余弦相似度和 Pearson 相关系数改进前后的 MAE 和 RMSE 对比图可以看出，使用本研究设计的参数对这两种传统的相似度计算方法进行改进后，MAE 和 RMSE 这两个误差指标均有所降低。

在本实验结果中，当选取的最近邻居数 k 超过一定数量时，误差反而有所增加，这是由于选用的数据集中总用户数为 943，与目标用户真正相似的用户数可能有限，低于实验中设定的 k 值，因此将一些与目标用户并不相似的用户也计算在内，从而对预测结果产生了负面影响。在本实验的数据集中，选取邻居数 k 在区间(20，40)时，MAE 和 RMSE 的综合表现最佳。

3.3.2　实验 2

将上述 MovieLens 数据集按照 8∶2 的比例划分成实验集和测试集，选取最近邻居 k 为 50、$sim_1(a,b)$ 为 $sim_{\text{Pearson}'}(a,b)$ 进行实验，对取值进行调节(分别取值 0，0.1，0.2，0.3，0.4，0.5，0.6，0.7，0.8，0.9，1.0)，生成预测评分。通过与真实评分的对比计算，获得令 MAE 最小的 β 值，实验结果如图 5 所示，由此可以确定相似度改进算法的参数为 0.7。

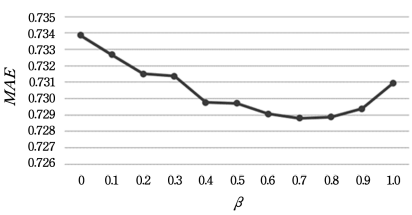

图 5　不同 β 值对应的 MAE

将本文提出的融合相似度算法(MUPA)与传统的余弦相似度和 Pearson 相关系数算法进行对比实验，实验结果如图 6 和图 7 所示。结果表明，本研究提出的融合后的相似度算法的误差低于传统相似度算法，说明相似度计算结果在准确性上有所提高。

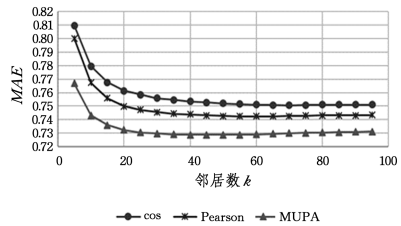

图 6　不同邻居数下相似度算法的 MAE 对比

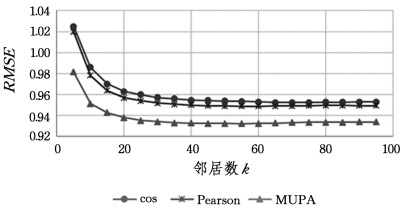

图 7　不同邻居数下相似度算法的 RMSE 对比

结束语　本研究通过设计和引入参数，对传统的协同过滤推荐算法中的相似度计算方法进行改进，降低了传统相似度算法由于没有考虑用户评分习惯的差异和用户对项目选择的差异带来的误差。利用用户对相同项目评分的分差，设计参数来对用户之间评分习惯的差异进行量化；利用 Jaccard 系数来对用户之间项目选择的差异性进行量化。将这两个参数与传统的相似度算法进行融合，对传统的相似度计算方法进行修正。结合用户对项目的偏好程度受项目所具有属性的影响这一因素，提出了衡量用户对属性偏好的相似度计算方法，并通过加权的方式将二者进行融合。实验结果表明，本研究提出的改进算法能够降低误差，能有效地提高相似度计算的准确度。

参 考 文 献

[1] SHARDANAND U，MAES P. Social information filtering：algorithms for automating "word of mouth"[C]// Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co.，1995：210-217.

[2] MALTZ D A. Distributing information for collaborative filtering on Usenet net news[OL]. http://dspace. mit. edu/bitstream/handle/1721. 1/36454/31314097-MIT. pdf.

[3] RESNICK P，IACOVOU N，SUCHAK M，et al. GroupLens：an open architecture for collaborative filtering of netnews[C]// Proceedings of the 1994 ACM Conference on Computer Supported Cooperative work. ACM，1994：175-186.

[4] BOBADILLA J，HERNANDO A，ORTEGA F，et al. Collaborative filtering based on significances[J]. Information Sciences，2012，185(1)：1-17.

[5] ORTEGA F，SÁNCHEZ J L，BOBADILLA J，et al. Improving collaborative filtering-based recommender systems results using Pareto dominance[J]. Information Sciences，2013，239(4)：50-61.

[6] WANG S，ZHAO Z，HONG X. The Research on Collaborative Filtering Recommendation Algorithm Based on Improved Clustering Processing[C]// IEEE International Conference on Computer and Information Technology；Ubiquitous Computing and Communications；Dependable，Autonomic and Secure Computing. Pervasive Intelligence and Computing. IEEE，2015：1012-1015.

[7] 邢春晓，高凤荣，战思南，等. 适应用户兴趣变化的协同过滤推荐

算法[J]. 计算机研究与发展,2007,44(2):296-301.
[8] 王吉源,黎晨,王婵娟. 用户属性加权活跃近邻的协同过滤算法[J]. 计算机应用研究,2016,33(12):3625-3629.
[9] 赵文涛,王春春,成亚飞,等. 基于用户多属性与兴趣的协同过滤算法[J]. 计算机应用研究,2016,33(12):3630-3633.
[10] 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 & Data Engineering,2005,17(6):734-749.
[11] HARPER F M,KONSTAN J A. The MovieLens Datasets:History and Context[J]. Acm Transactions on Interactive Intelligent Systems,2015,5(4):19.

(上接第 411 页)

定的支持度阈值,基于不同数量的计算节点,得到如图 6 所示的运行结果。

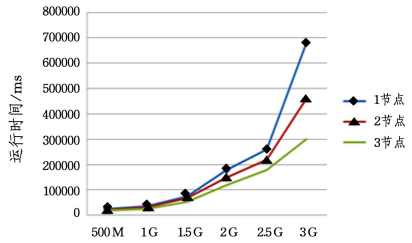

图 6 数据片与运行时间

从图 6 中看出,随着数据集大小的增加,运行时间开销逐渐增加。对于小数据集,MapReduce 计算模型并未充分发挥优势,图中 500 M 与 1 G 数据集运行 PST-Apriori 算法的时间差异性不大,这主要是因为在 Hadoop 平台下,数据在存入 HDFS 之前仅被分割为几个数据分片,且单个节点 DataNode 已经满足数据存储,mapper 操作大部分在内存中完成,不会频繁写入 HDFS,因此节点数量并不占太大优势。但是,随着数据集的增大和计算节点的增加,数据集实现了分布式存储与计算,算法的运行效率明显提升。

5.2.2 真实数据

采用 R 语言 arules 软件包中的 Groceries 数据集进行测试,该数据集是某一食品杂货店一个月的真实交易数据,共包含 169 种数据项。通过改变支持度阈值(min_sup=0.005~0.5),比较传统 Apriori 算法与 PST-Apriori 算法的执行效率,得到如图 7 所示的运行结果。

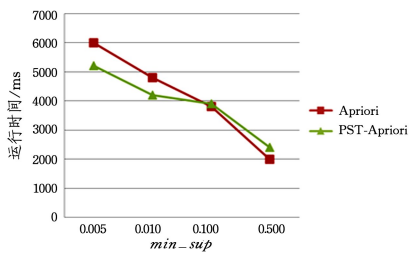

图 7 Apriori 与 PST-Apriori

从图 7 可以分析出,随着支持度阈值的增加,两个算法的时间开销均减少。当支持度阈值较大时,传统的 Apriori 算法比 PST-Apriori 算法的执行效率高,这是因为 Apriori 算法生成的候选项集少,迭代的次数少,而 PST-Apriori 算法需要逐层生成候选项集,并压缩到 PST 中,且需要分布式计数,增加了时间开销。当支持度阈值较小,PST 树中压缩的节点数量较多时,因为 PST-Apriori 算法采取广度优先搜索的策略,且按照 key 值排序,加快了 Mapper 和 Reducer 端的 shuffle 操作,充分发挥了 MapReduce 分布式计算的优势,因此提高了算法的执行效率。相比之下,对于较小的支持度阈值,PST-Apriori 算法的执行效率有了明显的提升。

结束语 本文提出的 PST-Apriori 算法主要用于解决传统并行关联规则算法多次启动 MapReduce 任务带来较大性能开销的问题,算法只需要两个 MapReduce 任务,第一个 MapReduce 任务用于分布式计数,统计频繁 1 项集;第二个 MapReduce 任务将每个事务 T 生成的候选项集逐层压缩到 PST 中,PST 中的每个节点对应 map 函数的键值对,由 MapReduce 框架自动实现聚集。最后调用 reduce 函数进行分布式汇总,得到满足支持度阈值的频繁项集。模拟和真实数据的实验证明,PST-Apriori 面对较大数据集时能有效地挖掘关联规则。

参 考 文 献

[1] AGRAWAL R,SRIKANT R. Fast algorithms for mining association rules(3rd ed)[M]// Readings in Database Systems. Morgan Kaufmann Publishers Inc.,1998:2299-2308.
[2] HAN J,PEI J,YIN Y. Mining frequent patterns without candidate generation[C]// ACM SIGMOD International Conference on Management of Data. ACM,2000:1-12.
[3] LI L,ZHANG M. The Strategy of Mining Association Rule Based on Cloud Computing[C]// International Conference on Business Computing and Global Informatization. IEEE,2011:475-478.
[4] LI N,ZENG L,HE Q,et al. Parallel Implementation of Apriori Algorithm Based on MapReduce[C]// Acis International Conference on Software Engineering,Artificial Intelligence,Networking and Parallel & Distributed Computing. IEEE,2012:236-241.
[5] ZHOU X,HUANG Y. An Improved Parallel Association Rules Algorithm Based on MapReduce Framework for Big Data[C]// International Conference on Fuzzy Systems and Knowledge Discovery. IEEE,2014:284-288.
[6] 郝天曙. 基于 Hadoop 的并行数据挖掘的研究[D]. 南京:南京邮电大学,2017.
[7] 张玲. 基于 Hadoop 平台并行关联规则挖掘算法研究[D]. 西安:西安科技大学,2017.
[8] 荀亚玲. 集群环境下的关联规则挖掘及应用[D]. 太原:太原科技大学,2017.
[9] 于跃. 基于 Hadoop 平台的并行化分布式关联规则挖掘算法研究[D]. 吉林:吉林大学,2017.
[10] 李若晨. 基于并行的 Apriori 数据挖掘算法的研究[D]. 吉林:吉林大学,2017.
[11] 叶璐. 基于 Spark 的改进关联规则算法研究[D]. 太原:太原科技大学,2017.
[12] 马连灯. 基于 HADOOP 平台的并行关联规则算法研究[D]. 天津:天津工业大学,2017.