

融合巴氏系数的加权 Slope One 算法

刘金梅,舒远仲,张尚田,唐小敏,刘文祥

(南昌航空大学 信息工程学院,江西 南昌 330063)

摘要:针对用户共同评分很少甚至没有时,传统相似度计算性能较差的问题,以及传统加权 slope one 算法在进行预测评分时未考虑项目相似度对预测结果的影响,提出一种融合巴氏系数的加权 slope one 算法(BCWSOA)。该算法主要针对上述两个问题进行改进,一是利用巴氏系数对用户相似度进行改进。首先用皮尔逊相关系数计算用户局部相似度,Jaccard 相似性计算用户全局相似度,巴氏系数分析用户相关性,然后将巴氏系数作为权重因子优化用户局部相似度,最后使用参数 α 组合优化用户局部相似度和用户全局相似度,从而获得融合相似度。参数 α 用来凸显不同相似度在融合相似度中的权重;二是利用巴氏系数改进预测评分,考虑项目相似度对预测结果的影响,计算项目相似度并将其作为权重改进加权 slope one 算法预测评分公式。通过在 MovieLens100k 数据集上的实验表明,相比于其他算法,提出的 BCWSOA 算法准确度有所提高。

关键词:巴氏系数;加权 slope one 算法;相似度;BCWSOA;评分预测

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2020)11-0074-06

doi:10.3969/j.issn.1673-629X.2020.11.014

A Weighted Slope One Algorithm Based on Bhattacharyya Coefficient

LIU Jin-mei, SHU Yuan-zhong, ZHANG Shang-tian, TANG Xiao-min, LIU Wen-xiang

(School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China)

Abstract: For the problem of poor performance of traditional similarity calculation when the user common rating is few or not, and the traditional weighted slope one algorithm does not take into account the effect of item similarity on the prediction results when making the prediction rating, a weighted slope one algorithm (BCWSOA) based on Bhattacharyya coefficient is proposed. The algorithm is mainly aimed at the above two problems. One is to use Bhattacharyya coefficient to improve the user similarity. Firstly, the local similarity is calculated by the Pearson correlation coefficient, the global similarity is calculated by the Jaccard similarity and the user correlation is analyzed by the Bhattacharyya coefficient which is used as a weighting factor to optimize the user's local similarity. Finally, the parameter α is used to optimize the user's local similarity and the user's global similarity, thus the fusion similarity is obtained. The parameter α is used to highlight the different similarities in the fusion similarity weights. The other one is to improve the prediction score by using the Bhattacharyya coefficient. Based on the impact of project similarity on prediction results, the similarity of items is calculated by using the Bhattacharyya coefficient, which is used as a weight to improve the weighted slope one algorithm prediction score formula. Experiments on MovieLens100k dataset show that compared with other algorithms, the accuracy of proposed BCWSOA is improved.

Key words: Bhattacharyya coefficient; weighted slope one algorithm; similarity; BCWSOA; rating prediction

0 引言

传统的推荐算法主要有基于内容的算法、基于知识的算法、协同过滤算法和混合算法^[1]。协同过滤算法应用最广泛,主要包括基于用户和基于物品的协同过滤^[2]。slope one 算法^[3-4]是一种基于物品协同过滤算法,它简洁、容易实现、执行效率高,主要依据项目评分差值预测,当面对庞大的数据集时,无法较好地处理稀疏数据。因此,许多学者对其进行了改进,如:刘毓等人^[5]考虑到评分用户数量对预测结果存在影响,提

出一种加权 slope one 算法,该算法提高了预测精度;向小东等人^[6]利用项目偏差对未评分的数据进行填充,再使用协同过滤算法;由于加权 slope one 算法未考虑用户与用户或项目与项目自身关系,王潘潘等人^[7]将协同过滤算法与加权 slope one 算法相结合,通过用户相似度寻找与目标用户最相近的 K 个邻居,根据邻居对项目的评分进行预测;李桃迎等人^[8]不仅考虑了用户相似度还考虑到用户兴趣变化,用遗忘函数与邻居相似度改进加权 Slope one 算法,进一步提高了

收稿日期:2019-12-24

修回日期:2020-04-24

基金项目:国家自然科学基金(71761028)

作者简介:刘金梅(1995-),女,硕士生,研究方向为信息挖掘、推荐算法;舒远仲,教授,通讯作者,研究方向为物联网、软件工程。

精确度;文献[9-10]考虑项目相似性对结果的影响,根据不确定 K 近邻矩阵中的项目相似性,动态选择每个项目的邻域,然后计算邻域中的项目评分偏差,最后再用线性模型进行预测;陶志勇等人^[11]通过用户与项目属性改进相似度,引入用户属性-项目类型偏好权重因子,利用天牛须搜索算法调整兴趣度预测;Zhao 等人对上述三人的算法进行了融合,考虑用户和项目相似度并将其都作为权重因子融合到加权 slope one 算法^[12]。

上述改进算法中,相似度的度量都依赖共同评分。当数据稀疏时,皮尔逊相关系数计算性能较差;Jaccard 系数仅考虑两者是否有共同特性,不能衡量具体评分数值差异。Bobadilla 等人^[12]通过获取奇异性信息,提出奇异性相似度测量,相比传统的相似度测量,提高了预测质量,但在数据稀疏时,计算的相似度较差。

因此,该文对相似度计算方法进行改进,提出一种融合巴氏系数的相似度计算方法。首先用 Jaccard 相似度计算全局相似度, Pearson 相关系数计算用户局部相似度,其次用巴氏系数优化用户局部相似度,然后再引入参数 α 将两者融合获取最终的用户相似度,参数 α 的引入可以更好地衡量各个相似度在最终相似度中所占的比重。由于要对未知项目进行预测,所以再根据相似度排序选择前 K 个用户作为目标用户的近邻,再计算近邻用户对项目的评分偏差,最后结合目标用户的评分用加权 slope one 算法进行评分预测,由于项目相似度对预测也存在影响,因此用巴氏系数计算项目相似度并将其作为权重对预测评分进行改进。

1 几种 slope one 算法

1.1 slope one 算法

slope one 算法是一种线性模型,即 $f(x) = ax + b$, 可由项目 x 的值预测出 $f(x)$, b 为项目间评分差异值。如表 1 所示,预测 U_3 对 A 的评分。

表 1 用户对项目的评分

项目	A	B
U_1	2	4
U_2	5	1
U_3	?	3

首先,根据 U_1 、 U_2 对 A、B 的评分,计算出 A、B 之间的评分偏差 b , $b = \frac{(4-2) + (1-5)}{2} = -1$, 然后预测评分, $f(x) = 3 + (-1) = 2$, 即 U_3 对 A 的评分为 2 分。

slope one 算法的工作流程如下:

(1) 计算偏差。

项目之间的偏差计算如下:

$$\text{Dev}_{ji} = \sum_{u \in S_j(X)} \frac{u_j - u_i}{\text{Num}(S_j(X))} \quad (1)$$

其中, u_i 、 u_j 表示用户 u 对项目 i 、 j 的评分; $S_j(X)$ 表示评价项目 i 、 j 的用户集合; $\text{Num}(S_j(X))$ 表示用户数量。

(2) 预测。

根据步骤 1 计算出的评分偏差,再加上用户 u 对项目的评分预测出目标项目 j 的评分,公式如下:

$$P(u)_j = \frac{1}{\text{Num}(R_j)} \sum_{i \in R_j} \text{Dev}_{ji} + u_i \quad (2)$$

其中, $\text{Num}(R_j)$ 表示所有用户评过的项目集合。

1.2 Weighted slope one 算法

slope one 算法仅考虑评分,未考虑评分用户数量对偏差计算的影响。对同一个项目评分用户数量越多,计算的偏差越精确,因此为平衡每个用户对评分的影响,进行加权处理,改进后的公式如下:

$$P(u)_j = \frac{\sum_{i \in \text{Num}(R_j)} (\text{Dev}_{ji} + u_i) \text{Num}(S_{ji}(X))}{\sum_{i \in \text{Num}(R_j)} \text{Num}(S_{ji}(X))} \quad (3)$$

2 改进的加权 slope one 算法

2.1 相似度计算方法分析

传统相似度计算方法^[13-14]有余弦相似度、修正余弦相似度、皮尔逊相关系数、Jaccard 相似度。

设 m 个用户 $U = \{U_1, U_2, \dots, U_m\}$ 和 n 个项目 $I = \{I_1, I_2, \dots, I_n\}$ 组成的用户-项目评分矩阵,用户 U_i 对项目 I_j 的评分用 R_{ij} 表示,具体如表 2 所示。

表 2 用户-项目评分

	I_1	...	I_j	...	I_n
U_1	R_{11}	...	R_{1j}	...	R_{1n}
...
U_i	R_{i1}	...	R_{ij}	...	R_{in}
...
U_m	R_{m1}	...	R_{mj}	...	R_{mn}

(1) 余弦相似度。

余弦相似度(cosine similarity)用向量夹角的余弦值表示个体间差异。

设两个用户 u 、 v , I_w 为用户所有评分的项目集, R_{ui} 、 R_{vi} 为用户 u 、 v 对项目 i 的评分,则两个用户间的余弦相似度的计算公式如下:

$$\text{Sim}(u, v) = \frac{\sum_{i \in I_w} R_{ui} \times R_{vi}}{\sqrt{\sum_{i \in I_w} R_{ui}^2} \times \sqrt{\sum_{i \in I_w} R_{vi}^2}} \quad (4)$$

(2) 修正余弦相似度。

由于余弦相似度未考虑评分数值对结果的影响,所以产生了修正余弦相似度(adjusted cosine

similarity), 本质是在计算时减去用户评分的平均值。计算公式如下所示:

$$\text{Sim}(u, v) = \frac{\sum_{i \in I_{uv}} (R_{ui} - \bar{R}_u)(R_{vi} - \bar{R}_v)}{\sqrt{\sum_{i \in I_u} (R_{ui} - \bar{R}_u)^2} \times \sqrt{\sum_{i \in I_v} (R_{vi} - \bar{R}_v)^2}} \quad (5)$$

(3) 皮尔逊相关系数。

皮尔逊相关系数 (Pearson correlation coefficient, PC) 用来衡量两个变量间的相关程度, 取值在 $[-1, +1]$ 之间。计算公式如下:

$$\text{Sim}_{\text{pcc}}(u, v) = \frac{\sum_{i \in I_{uv}} (R_{ui} - \bar{R}_u)(R_{vi} - \bar{R}_v)}{\sqrt{\sum_{i \in I_u} (R_{ui} - \bar{R}_u)^2} \times \sqrt{\sum_{i \in I_v} (R_{vi} - \bar{R}_v)^2}} \quad (6)$$

其中, I_{uv} 为用户 u 、 v 的共同评分集合; R_{ui} 、 R_{vi} 为用户 u 、 v 对项目 i 的评分; \bar{R}_u 、 \bar{R}_v 分别为用户 u 、 v 对所有项目评分的均值。

(4) Jaccard 相似度。

Jaccard 相似度仅考虑个体间是否具有共同特征, 没有考虑评分数值大小, 具体公式如下:

$$\text{Sim}_{\text{jac}}(u, v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (7)$$

其中, I_u 、 I_v 分别表示用户 u 、 v 对项目评分的集合。

(5) 巴氏系数。

巴氏系数 (Bhattacharyya coefficient, BC)^[15-17] 在信号、图像以及模式识别等领域使用较多, 可以用来测量样本间的相关性。设 $p_1(x)$ 和 $p_2(x)$ 为同一离散区间 X 上的两个概率, 则计算公式如下:

$$\text{BC}(P_1, P_2) = \sum_{x \in X} \sqrt{p_1(x)p_2(x)} \quad (8)$$

其中, $p_1(x)$ 、 $p_2(x)$ 是评分矩阵中的已知评分数据, P_u 、 P_v 分别表示用户 u 和用户 v 的评分, 则用户 u 和用户 v 的巴氏系数相似度计算公式如下:

$$\text{BC}(u, v) = \sum_{s=1}^n \sqrt{(P_{us})(P_{vs})} \quad (9)$$

其中, n 为用户对项目评分种类数量; P_{us} 、 P_{vs} 分别表示用户 u 和用户 v 在评分类别为 s 情况下的评分概率, 其计算公式为: $p_{us} = \frac{n(s)}{n(u)}$, 其中 $n(s)$ 为用户 u 评分为 s 情况下的项目数量, $n(u)$ 为用户 u 评分的项目总数量。

设用户 u 对项目的评分为 $u = (2, 3, 0, 1, 1, 4)^T$, 用户 v 对项目的评分为 $v = (1, 0, 3, 2, 1, 5)^T$, 评分范围为 0~5 分, 则用户 u 和用户 v 使用巴氏系数计算的相似度为:

$$\begin{aligned} \text{BC}(u, v) &= \sum_{s=1}^5 \sqrt{(P_{us})(P_{vs})} = \\ &= \sqrt{\frac{2}{5} \times \frac{2}{5}} + \sqrt{\frac{1}{5} \times \frac{1}{5}} + \sqrt{\frac{1}{5} \times \frac{1}{5}} + \\ &= 0 + 0 = 0.8 \end{aligned} \quad (10)$$

通过对上述相似度计算方法的分析, 发现皮尔逊相关系数在数据稀疏时计算性能较差; Jaccard 相似性虽然计算了全局相似度, 但是仅考虑项目间的共同特征, 没有考虑评分数值大小; 而余弦和修正余弦相似度都依赖共同评分; 巴氏系数不仅可以在数据稀疏时计算相似度, 而且可以分析用户关联性。因此, 下面将用巴氏系数来改进相似度。

2.2 相似度计算方法改进

由 2.1 节分析可知, 当共同评分很少或者没有时, 传统相似度计算性能较差, 因此该文利用巴氏系数优化皮尔逊相关系数, 并与 Jaccard 相似性进行融合得到最终用户相似度。

首先, 用皮尔逊相关系数计算用户局部相似度, 当数据较少时计算的性能较差, 所以用巴氏系数分析用户相关性, 将作为影响因子加入到局部相似度中得出用户新的局部相似度。公式如下:

$$\text{Sim}_{\text{loc}}(u, v) = \text{Sim}_{\text{pcc}}(u, v) \text{BC}(u, v) \quad (11)$$

其次, 用 Jaccard 相似性计算全局相似度, 将优化的局部相似度和全局相似度进行融合, 设置参数 α , 参数 α 用来凸显不同相似度在最终相似度中的权重。公式如下:

$$\text{Sim}(u, v) = \alpha \text{Sim}_{\text{jac}}(u, v) + (1 - \alpha) \text{Sim}_{\text{loc}}(u, v) \quad (12)$$

上述两式经处理获得最终的用户相似度, 公式如下:

$$\begin{aligned} \text{Sim}_{\text{pcc}}(u, v) &= \alpha \text{Sim}_{\text{jac}}(u, v) + \\ &= (1 - \alpha) \text{BC}(u, v)^2 \text{Sim}_{\text{pcc}}(u, v) \end{aligned} \quad (13)$$

2.3 预测评分的改进

根据 2.2 节计算出的用户相似度, 先降序排列后根据顺序挑选前 K 个用户放入邻居集合 $S(K)$ 中; 再从邻居集合中筛选出评价过目标项目的邻居用户, 计算出目标项目与邻居用户评过项目的评分偏差 Dev_{ji} ; 最后通过偏差与目标用户的评分用加权 slope one 算法预测, 公式如下:

$$P(u)_j = \frac{\sum_{i \in S(K)} (\text{Dev}_{ji} + u_i) \text{Num}(S_{ji}(X))}{\sum_{i \in S(K)} \text{Num}(S_{ji}(X))} \quad (14)$$

由于式(14)中仅仅考虑了用户评分对预测结果的影响, 未考虑项目本身对其的影响, 因此, 在评分预测时又考虑项目相似度。对于项目相似度计算, 该文

使用巴氏系数。首先计算所有的项目相似度;然后找出邻居集合 $S(K)$ 中前 K 个邻居所评分的项目集合;再计算目标项目 j 与邻居项目 i 之间的相似度 $BC(i, j)$;最后将所得的项目相似度作为权重因子,加入到式(14)中得出最终的预测评分。公式如下:

$$P(u)_j = \frac{\sum_{i \in S(K)} (\text{Dev}_{ji} + u_i) \text{Num}(S_{ji}(X)) BC(i, j)}{\sum_{i \in S(K)} \text{Num}(S_{ji}(X)) BC(i, j)} \quad (15)$$

将项目相似度作为权重因子加入到预测公式中,一方面可以削弱项目相似度低对预测结果的影响,项目相似度高被忽略的现象;另一方面考虑用户和项目相似度,可以使结果更准确。

2.4 算法流程

输入:用户-项目评分矩阵 R , 参数 α 、邻居数 K 。

输出:目标用户 u 对项目 i 的预测评分。

算法步骤如下:

步骤 1:创建用户-项目评分矩阵 R , 并根据给定的数据集填充矩阵 R 。

步骤 2:根据评分矩阵 R , 利用式(6)计算用户局部相似度,式(7)计算用户全局相似度。

步骤 3:利用巴氏系数(式(9))分析用户相关性,计算项目间相似度。

步骤 4:用步骤 3 计算出的巴氏系数优化式(6)的局部相似度,并与式(7)的全局相似度相融合,设置参数 α 线性融合得到最终的用户相似度。

步骤 5:根据步骤 4 得到的最终用户相似度,将相似度进行降序排列再选择前 K 个作为用户邻居,放入邻居集合 $S(K)$ 中。

步骤 6:由步骤 5 的邻居集合 $S(K)$, 根据最近邻居用户对未知项目的评分来预测目标用户对项目的评分,最后再将步骤 3 中的项目相似度加入到式(14)中,最后使用式(15)预测出最终目标用户对项目的评分。

3 实验结果与分析

3.1 数据集

采用 MovieLens 数据集进行测试 (<https://grouplens.org/datasets/movielens/>), 其中用户个数 943, 电影为 1 682 部, 总共评分数为 100 000 条, 评分范围为 1~5 分, 数据的稀疏程度为 0.063。数据集划分比例为 8:2, 用五折交叉法来进行实验。

3.2 评价指标

文中算法的预测质量由平均绝对误差 (mean absolute error, MAE) 和均方根误差 (root mean square error, RMSE) 来衡量, 公式如下:

$$\text{MAE} = \frac{\sum_{i=1}^N |P_i - T_i|}{N} \quad (16)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (P_i - T_i)^2}{N}} \quad (17)$$

其中, P_i 为预测评分, T_i 为测试集中的实际评分, N 为测试集中评分的总数。

3.3 结果分析与比较

3.3.1 确定参数 α 、 K 的值

在改进相似度时, 用巴氏系数优化皮尔逊相关系数获得用户局部相似度, 用 Jaccard 相似性计算用户全局相似度, 最后在相似度融合时, 使用参数 α 来组合优化的用户局部相似度和用户全局相似度。参数 α 用来凸显不同相似度在最终相似度中的权重, 将参数 α 设置为 $[0.1, 1]$, 步长为 0.1, 选择的邻居数 K 为 $[10, 60]$, 步长为 10。为排除实验结果的偶然性, 将 6 组不同邻居下的值取平均, 分析 α 以及 K 的变化对 MAE 和 RMSE 的影响, 实验结果如图 1 和图 2 所示。

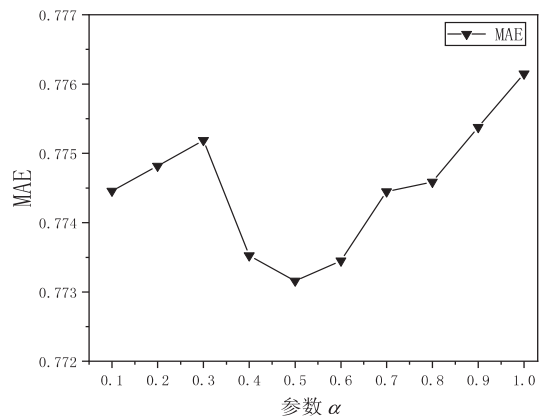


图 1 不同参数 α 对 MAE 的影响

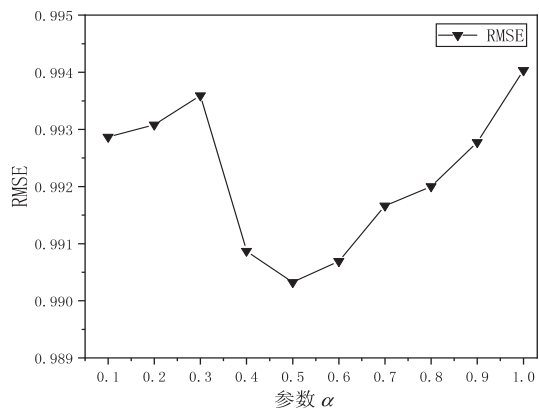


图 2 不同参数 α 对 RMSE 的影响

从图 1 和图 2 可以看出, MAE 和 RMSE 的值随参数 α 增大而变化, 当参数 α 取值为 0.5 时, MAE 和 RMSE 均达到最小值, 说明此时的效果最佳。因此, 在用户相似度改进算法中, 在局部相似度和全局相似度融合时, 将参数 α 设置为 0.5。

在改进的算法中,邻居数量不同也会影响预测结果,因此,该文选取不同的邻居数量进行测试($K=10, 40, 80, 120, 160, 200, 240, 280, 320$),每一个 K 进行5次实验,再对其取平均值作为最终结果。绘制邻居数量 K 与MAE和RMSE的关系图,如图3和图4所示。

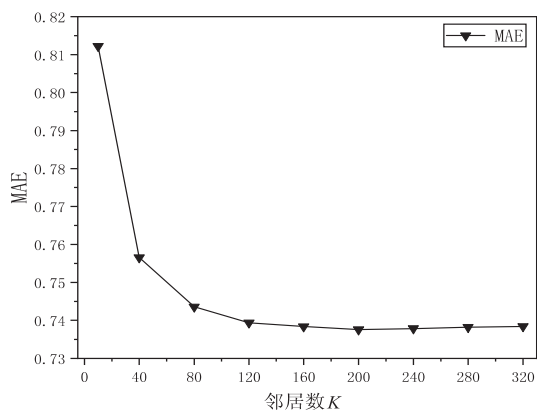


图3 邻居数量 K 与MAE的关系

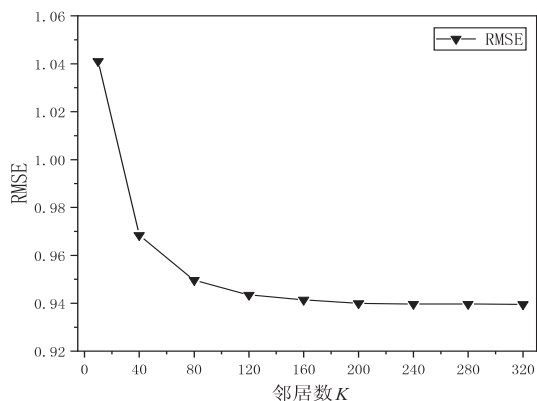


图4 邻居数量 K 与RMSE的关系

从图3和图4可以看出,随着邻居数量 K 的增多,MAE和RMSE均先减小后慢慢趋于稳定。因为在邻居数量较少时可用来分析的数据较少,此时产生的误差就偏大;随着邻居数量的不断增多,可利用的数据越来越多,误差逐渐减小直至最后达到某一数值趋于稳定。从图中可以看出,当邻居数 $K=200$ 时,误差最小,此时预测评分愈加接近真实值。因此,实验选取邻居数量为200进行研究,不仅可以保证良好的预测结果,同时又能降低计算复杂度。

3.3.2 改进预测公式的算法对比

预测公式的不同,实验结果也会存在差异。该文利用巴氏系数计算项目相似度,将其加入到原始加权slope one预测公式中,将改进后的预测公式与未加入权重因子的未改进预测公式作对比,结果如图5和图6所示。

从图5和图6可以明显看出,改进预测公式的MAE和RMSE值均比未改进预测公式的值小,说明用巴氏系数改进预测公式在某种程度上可以提高预测精度。

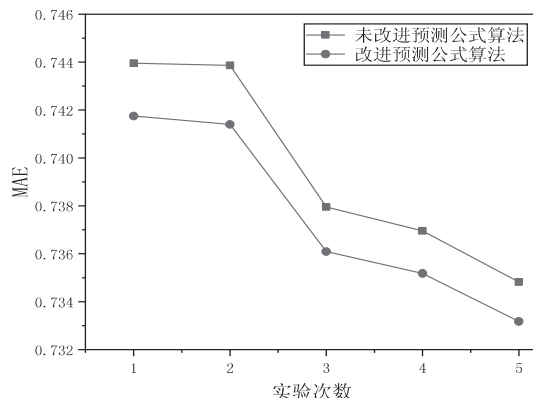


图5 不同预测公式的MAE值

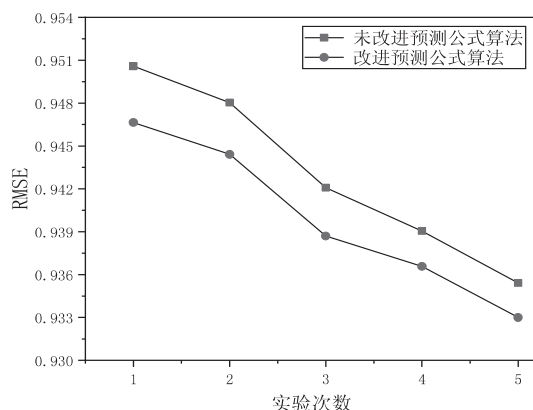


图6 不同预测公式的RMSE值

3.3.3 改进算法与其他算法的对比

利用巴氏系数可以在稀疏数据情况下仍然能够计算相似度,提出用巴氏系数优化皮尔逊相关系数,并与Jaccard相关系数相融合得到一种新的相似度计算方法。通过3.3.1节可知,在相似度融合时,参数 α 取0.5时可以获得最优用户相似度;在邻居数量 K 为200情况下,预测结果愈加接近真实值,误差较小,因此,将参数 α 设置为0.5,邻居数量 K 设置为200进行实验。将改进的算法(BCWSOA)与文献[18]中基于加权slope one和用户协同过滤的推荐算法(CF_WSOA)、Weighted slope one算法、slope one算法和基于用户的协同过滤算法(UCF)进行比较,衡量指标为MAE和RMSE,实验结果如图7和图8所示。

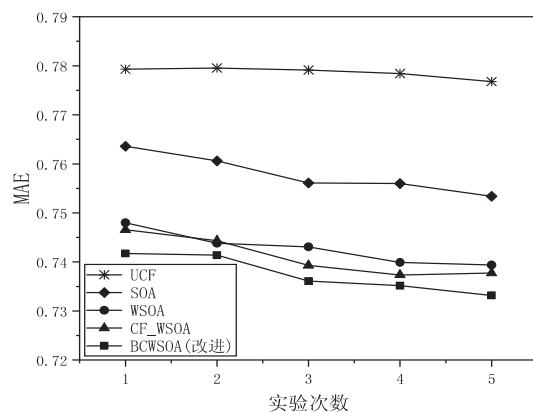


图7 不同算法的MAE值

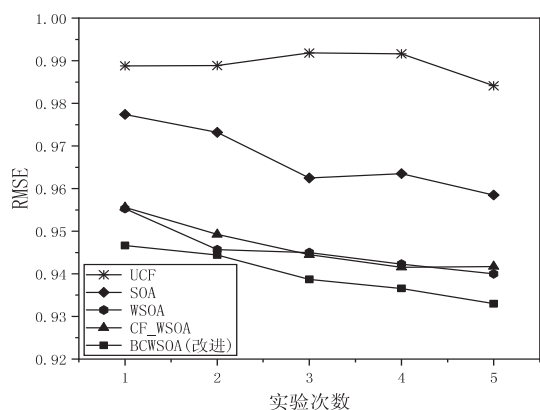


图 8 不同算法的 RMSE 值

从图 7 和图 8 可以看出,当参数 α 为 0.5,邻居数 K 为 200 时,改进算法 BCWSOA 的 MAE 和 RMSE 值相比于其他几种算法均有所减小,预测准确性有所提高。

4 结束语

该文融合巴氏系数改进相似度的计算方法,解决了用户共同评分很少或者没有的问题。考虑了项目相似度对预测的影响,用巴氏系数计算相似度改进加权 slope one 算法。与基于加权 slope one 和用户协同过滤的推荐算法(CF_WSOA)、Weighted slope one 算法、slope one 算法和基于用户的协同过滤算法(UCF)进行比较,融合巴氏系数的加权 slope one 算法在预测精确度上均有所提高。

参考文献:

- [1] 王盼盼. 协同过滤推荐算法及应用研究[D]. 西安:西北大学, 2019.
- [2] LUMAUAG R G, SISON A M, MEDINA R P. An enhanced recommendation algorithm based on modified user-based collaborative filtering[C]//2019 IEEE 4th international conference on computer and communication systems (ICCCS). Singapore: IEEE, 2019: 198-202.
- [3] 杨大鑫, 王荣波, 黄孝喜, 等. 基于最小方差的 K-means 用户聚类推荐算法[J]. 计算机技术与发展, 2018, 28(1): 104-107.
- [4] 薛宇彤, 陈海龙, 葛苗苗. 基于关联规则策略加权的 Slope One 算法改进[J]. 计算机应用研究, 2019, 36(12): 3605-3609.
- [5] 刘 毓, 何 锐. 一种改进 Slope One 的学习推荐算法[J]. 西安邮电大学学报, 2017, 22(4): 105-108.
- [6] 向小东, 邱梓威. 基于 slope-one 算法改进评分矩阵填充的协同过滤算法研究[J]. 计算机应用研究, 2019, 36(4): 1064-1067.
- [7] 王潘潘, 钱 谦, 王 锋. 改进加权 Slope one 协同过滤推荐算法研究[J]. 传感器与微系统, 2017, 36(7): 138-141.
- [8] 李桃迎, 李 墨, 李鹏辉. 基于加权 Slope One 的协同过滤个性化推荐算法[J]. 计算机应用研究, 2017, 34(8): 2264-2268.
- [9] YE G B, ZHAO X Y. The K-Nearest neighbor slope one algorithm based on weighted user similarity and user tag[C]//2018 Chinese control and decision conference (CCDC). Shenyang: [s. n.], 2018: 3665-3670.
- [10] SUN L, LI Y, MICHAEL E I. A slope one algorithm based on item activeness and uncertain neighbors[C]//2017 3rd IEEE international conference on computer and communications (ICCC). Chengdu: IEEE, 2017: 2231-2235.
- [11] 陶志勇, 崔新新. 一种融合用户与项目属性的协同过滤算法的设计与实现[J]. 计算机应用与软件, 2019, 36(2): 12-18.
- [12] BOBADILLA J, ORTEGA F, HERNANDO A. A collaborative filtering similarity measure based on singularities[J]. Information Processing & Management, 2012, 48(2): 204-217.
- [13] ZHAO Z, ZHANG J. Weighted slope one algorithm optimization based on user similarity and item similarity[C]//2018 14th international conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD). Huangshan: [s. n.], 2018: 34-39.
- [14] 郑翠翠, 李 林. 协同过滤算法中的相似性度量方法研究[J]. 计算机工程与应用, 2014, 50(8): 147-149.
- [15] PATRA B K, LAUNONEN R, OLLIKAINEN V, et al. A newsimilarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data[J]. Knowledge-Based Systems, 2015, 82: 163-177.
- [16] 姜少鑫. 基于巴氏系数和聚类的协同过滤技术研究[D]. 北京: 北京工业大学, 2017.
- [17] 武文琪, 王建芳, 张朋飞, 等. 一种巴氏系数改进相似度的协同过滤算法[J]. 计算机应用与软件, 2017, 34(8): 265-269.
- [18] WANG P, QIAN Q, SHANG Z, et al. An recommendation algorithm based on weighted slope one algorithm and user-based collaborative filtering[C]//2016 Chinese control and decision conference (CCDC). Yinchuan: [s. n.], 2016: 2431-2434.