

融合“S”型相似度和关联度的协同过滤算法

余 相 陈 亮 胡亚兰 王 丹
(东华大学 信息科学与技术学院 ,上海 201620)

摘 要: 协同过滤推荐系统是应用最广泛的推荐算法之一,但是其面临严重的稀疏性问题和扩展性问题。针对稀疏的评分矩阵难以准确计算相似度的问题,从推荐算法的流程出发,分离候选集生成和评分预测。针对候选集中存在大量弱或不相关的项目和用户感兴趣比例较低的问题,引入关联度,使用关联矩阵生成候选集;评分预测阶段分析相似度对推荐效果的影响,总结现有相似度的不足,提出一种细粒度划分的“S”型相似度来表述理想增长曲线,并在算法流程中融合候选集生成和评分预测。实验结果表明,减小候选集规模为原来的 1/3,避免了评分时对无效项目的计算,算法层面上提高了可扩展性,改进的“S”型相似度在推荐准确率上较之前提高了 4%,缓解了稀疏性对推荐效果的影响。

关键词: 协同过滤; 相似度; 关联度; 稀疏性

中图分类号: TP311.1

文献标识码: A

文章编号: 1673-629X(2019)03-0035-06

doi: 10.3969/j.issn.1673-629X.2019.03.007

Collaborative Filtering Algorithm Combining “S” Similarity and Relation

YU Xiang ,CHEN Liang ,HU Ya-lan ,WANG Dan

(School of Information Science and Technology ,Donghua University ,Shanghai 201620 ,China)

Abstract: Collaborative filtering recommendation system is one of the most widely used recommendation algorithms ,but it faces serious sparseness and scalability. To solve the problem that the sparse scoring matrix is difficult to accurately calculate the similarity ,the candidate set generation and scoring prediction are separated from the flow of the recommendation algorithm. Aiming at the problem that there are a large number of weak or irrelevant items and low proportion of users' interest in the candidate set ,the correlation degree is introduced and the candidate set is generated by using the correlation matrix. In the prediction stage of scoring ,analysis of the influence of similarity on recommendation effect and summarization of the existing shortcomings of similarity ,we propose a fine-grained “S” type similarity to express the ideal growth curve. The candidate set generation and scoring prediction are fused in the algorithm. The experiment shows that the size of the candidate set is reduced by 1/3 ,which avoids the calculation of invalid items when scoring ,and the scalability is improved at the algorithm level. The improved “S” type similarity is higher than the former in the recommendation accuracy rate. Increased by 4% eased the influence of sparsity on the recommendation effect.

Key words: collaborative filtering; similarity; relation; sparsity

0 引 言

随着互联网的飞速发展和移动端智能手机的普及,人类进入信息时代。据中国互联网络信息中心 2017 年 1 月份发布的互联网络状况统计报告显示,截至 2016 年 12 月,中国网页数量约为 2 360 亿个,比上年增长 11.2%^[1]。从巨大的信息量中挑选出人们满意的项目已经越来越难,从数字化图书、新闻、音乐、影视作品到电商平台都存在这样的问题,用户选择的时间成本越来越高,因此推荐系统应运而生。

目前,主流的推荐系统主要分为 4 类:基于内容的推荐、协同过滤推荐、基于知识的推荐和组合推荐^[2]。作为推荐系统中应用最广泛的算法,协同过滤技术已经在研究上和应用上取得了巨大的成功。然而,其依然有很多问题需要解决^[3]。其中之一便是推荐准确性。

为了使推荐的结果更加符合用户实际需求,学者一直在尝试改进经典协同过滤算法,但是随着用户和产品数量的日益增长,由于用户并不能对所有商品产

收稿日期: 2018-04-13

修回日期: 2018-08-22

网络出版时间: 2018-12-19

基金项目: 上海市自然科学基金资助项目(14ZR1400700)

作者简介: 余 相(1994-) ,男,硕士研究生,通讯作者,研究方向为推荐系统、大数据; 陈 亮,副教授,研究方向为机器学习、图像处理。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20181219.1532.052.html>

生记录,而是其中很小的一部分,从而导致用户-项目矩阵十分稀疏,甚至 99% 以上,用户-项目矩阵的极度稀疏制约着推荐系统的准确性。稀疏性也是协同过滤技术的核心问题,同时用户、项目的高维增长,对算法的效率也提出了挑战,需要实时生成推荐,直接影响着推荐系统的可扩展性。

1 问题分析

1.1 稀疏性问题

协同过滤算法主要是依据用户过往的使用记录来形成推荐,最常见的表现形式是用户-项目评分矩阵。绝大多数用户通过评分操作访问的信息项目数量相对

于系统中信息项目的总量十分有限,一般都达不到信息项目总数的 1%^[3-4],这就导致了评分矩阵的严重稀疏性。稀疏性问题的存在是协同过滤算法中固有的悖论的表现,即 CF 推荐系统本身是用于帮助用户从海量信息空间中选择最符合用户兴趣的信息,而这种需求将导致信息空间的稀疏,进而稀疏性又反过来影响推荐的生成^[5]。

1.2 影响分析

协同过滤算法的唯一数据来源就是评分矩阵,因此算法各不同步骤都是以评分矩阵为基础执行的。稀疏性问题带来的影响会沿着算法步骤一步步扩散到最终的推荐效果中,下面根据图 1 描述这一过程。

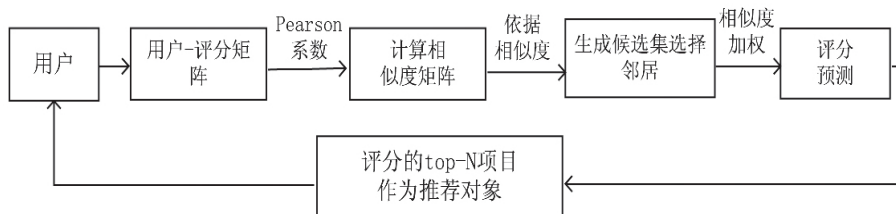


图 1 协同过滤算法流程

(1) 相似度计算: 相似度将直接作为后续邻居选择和评分预测的重要依据。根据相似度的计算方法可知,用户的相似性是对两者评分的交集实现的,当矩阵稀疏率很高时,用户已有的评分数据很少,造成用户间已评分项目的重合率明显下降,导致公共参考评分项目集合规模过小,相似性只能依赖极少数评分进行度量,造成用户间的相似度计算结果非常片面,不能准确反映用户间的真实相关性^[6]。兴趣爱好上原本比较相似的用户,因为少数几个共有项目评分差别较大而变成了非相似用户;事实上,基于小规模共有评分的相似度计算无法体现用户间的真实相似度,继而影响后续推荐步骤。

(2) 候选集生成、邻居选择: 选择目标项目的同类(即邻居)是评分预测的基础,虽然稀疏问题不会直接作用于邻居,但是通过相似度计算间接对邻居选择产生影响,项目的相似度不同选出的邻居集也是不同的,导致真实的邻居集质量下降了,进一步扩散到评分预测中。

(3) 评分预测: 实现推荐依赖于评分预测,实际上是将邻居中对目标项目有评分的用户按相似度加权求和得出预测评分。数据稀疏可能导致相似用户中很多未对目标项目进行过评分,导致只能依赖少数邻居评分实现预测,难以保证评分预测的准确性^[6]。

从上述分析可以看出,在稀疏性问题的影响下,相似用户的评估将大面积失真,相似度计算在评分预测和邻居选择中都扮演着重要角色,前述在相似度误差较大时造成推荐效果差甚至推荐失败。

1.3 扩展性问题

IBCF 算法是以项目相似度为依据筛选候选集,通过相似度加权预测用户对候选集中的项目的评分,根据预测评分高低确定推荐项目。候选集大小、相似度准确程度都将影响推荐精度。此阶段候选集需要对用户 u 的 n 个已评分项目进行操作,每个评分项目平均取 m 个最相似的邻居,这样每个用户的平均候选集大小为 nm ,并从中剔除常数个用户 u 已评分的项目,得到项目候选集 C 。

在 MovieLens 数据集上,当项目的近邻数 m 增加时,IBCF 算法候选集变化曲线如图 2 所示。曲线大致呈线性增长,当近邻数为 60 时,用户的候选集大小约为 200。同样的条件下,观察到候选集大小 k 在急剧上升时,候选集中用户感兴趣的项目占比却没有增长或维持不变,相反用户感兴趣占比在下降,当近邻数 $k =$

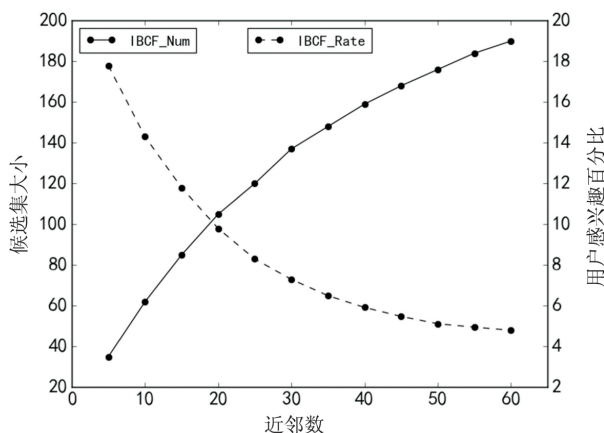


图 2 候选集大小及感兴趣项目比例

60 时,占比只有 4.9% 左右,总体比例非常小。用户的平均候选集大小是 nm , 此时的时间复杂度为 $O(n^2)$, 当用户的评分记录增多时,候选集大小呈二次函数增长,候选集中引入了大量与用户弱相关或不相关的项目,导致推荐准确率下降,评分预测阶段对大量不感兴趣的项目进行计算,算法执行中浪费了大量的 CPU 和内存资源,极大增加了为候选集项目预测评分的时间,当用户和项目呈高维增长时,将大幅度影响算法性能,直接制约着推荐的横向扩展性。

2 相似度和关联度

2.1 现有相似度

传统的皮尔逊相关系数只考虑用户间共有评分的作用,而忽视了非公共评分数量的作用,导致计算用户相似性及其具有片面性^[7]。如表 1 所示,按传统相似度计算,用户 U_1 与 U_2 的相似度为 1, U_2 与 U_3 的相似度要小于 1,这就造成了 U_1 比 U_3 更接近 U_2 的邻居,然而从相似性的可信度来看, U_3 应该比 U_1 更接近于 U_2 的邻居。

表 1 评分矩阵示例

	I_1	I_2	I_3	I_4	I_5	...
U_1	0	4	0	0	2	...
U_2	5	4	4	3	0	...
U_3	4	3	5	3	4	...

当两者间评分过少时,计算得到的相似度并不能反映用户之间的相似性,可通过一定方法对相似度进行修正。Herlocker 等^[8]为了解决此问题,在计算相关性时乘上一个基于共同评分的阈值。考虑两用户共有评分数量和控制共有评分数量的关系,对原来的相似度进行了权重调整,从而能区分不同相似度的可信度。用户 U_i 和 U_j 的修正相似度 $\text{sim}_1(i, j)$ 可以表示为:

$$\text{sim}_1(i, j) = \frac{\text{Min}(I_i \cap I_j, \alpha)}{\alpha} * \text{sim}(i, j) \quad (1)$$

$$\text{sim}(i, j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - \bar{R}_i) (\bar{R}_{j,c} - \bar{R}_j)}{\sqrt{\sum_{c \in I_i} (R_{i,c} - \bar{R}_i)^2} \sqrt{\sum_{c \in I_j} (R_{j,c} - \bar{R}_j)^2}} \quad (2)$$

相似度 $\text{sim}(i, j)$ 采用的是 Pearson 相关系数^[9]。其中, I_{ij} 为用户 i 和 j 共同评分的项目集合; $R_{i,c}$ 和 $R_{j,c}$ 分别表示用户 i 和 j 对项目 c 的评分; \bar{R}_i 和 \bar{R}_j 分别表示用户 i 和 j 的平均评分; $|I_i \cap I_j|$ 是两用户共有评分的数量; α 是控制共有评分项数量的阈值,用户间共有评分数量小于阈值时对传统相似度进行线性衰减修正。根据文献[8]的实验结果表明对推荐效果有所提高但并不显著,文中分析认为不同的用户的评分分布不一

样,采用全局统一的固定阈值无法适用于所有用户,这种修正只对少数用户起作用。

任磊^[10]针对该问题提出了适用不同项目评分数量分布的重合因子,刻画了项目间公共评分的重合程度,强调了共有评分重合度的重要性。其加权相似度表示为:

$$\text{sim}_2(i, j) = \text{sim}(i, j) * c(i, j) \quad (3)$$

$$c(i, j) = \frac{|I_i \cap I_j|^2}{|I_i| |I_j|} \quad (4)$$

其中,重合因子 $c(i, j)$ 中的 I_i 和 I_j 分别是项目 i 和 j 的评分项目数量。与文献[8]的绝对全局阈值不同,式 3 中的相似度不仅与共有评分数目有关,还反比于两用户单独评分数量,这种度量方式是一种全局动态修正,适用于每一个用户。

同时该计算方法的局限也很明显,即两项目间的相似性与共有评分数量的函数关系并非按理想曲线增长。见图 3(a),当共有评分数量由 5 增加到 6 时,增加的重合因子记为 Δc_1 ,评分数量由 13 增加到 14 时,增加的重合因子记为 Δc_2 ,评分数量由 21 增加到 22 时,增加的重合因子记为 Δc_3 ,从图中易知 $\Delta c_3 > \Delta c_2 > \Delta c_1$ 。传统相似度 $\text{sim}(i, j)$ 不变,相应的加权相似度的增量 $\Delta \text{sim}_{23} > \Delta \text{sim}_{22} > \Delta \text{sim}_{21}$,三段增量中对应的共有评分数量都只增加 1,这表明随着用户共有评分数量的增加,对应的加权相似度增长越来越快。当共有评分数量足够大时,任何细微的减少或者增加一个共有数量,重合因子会放大相似度的变化量,使得本来相似度相差较小的两项目现在相似度相差甚远,没有细粒度地度量项目间的相似性,这与理想型的相似度增长曲线相差甚远,理想型相似度增长曲线大致呈“S”型,随着共有数量的增加,相似度增量变化先缓慢再迅速最后接近饱和。

2.2 “S”型改进相似度

根据上述问题的分析,用户间相似性在共有评分数量达到一定数量时接近饱和,而文献[10]中提出的相似度计算方法与这一现象相悖。从实际出发,当两用户各自的独立评分数确定时,随着共有评分数量的增加,相似度的变化率(即增量的导数)应该是先变大后变小,呈“S”型曲线。实际含义即在共有评分数量较少时,两用户相似度比较小,随着共有数量的增加,相似度迅速上升,最后达到饱和,几乎不再发生变化,即两用户已经非常相似,这样才能体现用户间相似度的真实变化过程。

数学中常用“Sigmoid”函数来表示 S 型曲线,其函数特性完全符合相似度的理想增长过程,体现共有评分项增长中相似度变化的特性。其函数表示为:

$$S(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

截取标准 Sigmoid 函数某段区间再经过伸缩变换得到变换后的 Sigmoid 函数。将其引入重合因子中,则改进的相似度计算表示为:

$$\text{sim}_3(i, j) = \text{sim}(i, j) * c(i, j) \quad (6)$$

$$c(i, j) = \frac{1}{1 + e^{-(k * \frac{a-b}{a+m} - m)}} \quad (7)$$

其中, a 和 b 分别为两用户的单独评分数量; c 为共有评分数量; m 为变换的平移系数; k 为伸缩系数, 控制着函数的陡峭程度。

相似度关于共有评分项的函数见图 3(b)。共有评分项数小于 10 时, 相似度增长缓慢, 继而迅速增长, 当达到 20 时, 用户间再增加相同评分数量相似度饱和, 此种情况与相似度理想增长曲线的总体趋势相符。在此基础上, 将上述相似度计算集成到协同过滤算法中, 相似度计算是后续评分预测的基础, 因此将直接提升推荐算法的效果。

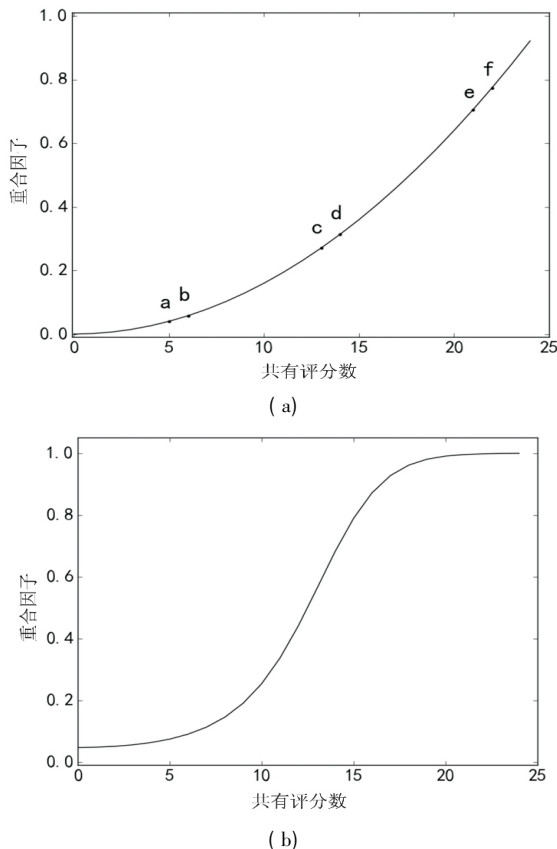


图 3 重合因子增量(a)和“S”型重合因子增量(b)

2.3 关联度

从可扩展性问题可知, 未考虑到用户间的差异, 将用户不感兴趣的项目加入到候选集, 导致候选项目集中大部分项目是无用的, 感兴趣项目占比很低, 故提高感兴趣项目的占比是优化算法效率的关键。在 IBCF 算法中, 寻找项目的近邻项目使用的是项目间的相似

度信息, 相似度信息在很大程度上反映了用户对该项目的评价是否一致, 而较少地考虑项目间依赖或者是顺序关系, 这种依赖的关系是寻找近邻项目的重要因素^[11]。

项目之间的关联关系在实际中大量存在, 这种关系很大程度上左右着用户的选择, 而根据相似度生成候选集无法描述此种关系。如图 4(a) 所示, 用户 1 浏览了项目 A、B、C, 用户 2 的浏览记录有项目 B、C、D, 后面的用户 3 在浏览了项目 B 之后是否想着去浏览 C。肯定的, 当大量用户在浏览了项目 B 之后去浏览项目 C, 可以认为项目 B 和项目 C 之间存在某种意义上的关联关系。文中从该现象出发, 引入 1-频繁项集的思想, 计算项目间的关联度矩阵, 并依据项目的关联度大小来生成候选集。

定义关联度: 关联度用来衡量两项目有联系的强弱程度, 即用户在浏览了某商品后又浏览某一商品的可能性, 大小用 r 表示, r_{mn} 的含义是项目 m 对项目 n 的关联度, 定义为项目 m 与 n 的共同用户数与项目 n 的用户数之比, 实际含义是用户在浏览了项目 n 后转而去浏览 m 的比例, 计算如下:

$$r_{mn} = \frac{C_m \cap C_n}{C_n} \quad (8)$$

其中, C_m 、 C_n 分别是项目 m 和 n 的用户数。

根据关联度的定义, 可以计算所有项目间的关联度并存储为关联矩阵 $R_{k \times k}$, 如图 4(b) 所示。 r_{mn} 表示项目 m 对项目 n 的关联, 主对角线全为 0, 矩阵不是对称的, 即 $r_{mn} \neq r_{nm}$, 两项目按照出现顺序的不同对用户产生的影响不同, 因此项目带给用户的关联度也就不同。

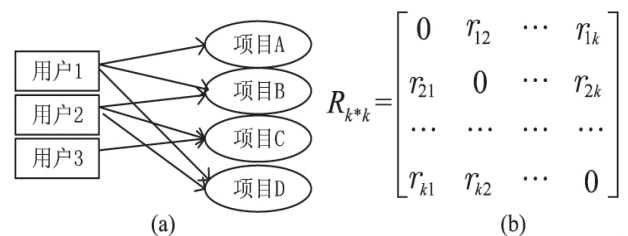


图 4 关联关系(a)和关联矩阵(b)

针对 IBCF 依据相似度生成的项目候选集中感兴趣占比太小的问题, 在候选集选取过程中引入关联度, 使用关联矩阵代替相似度产生候选集。根据用户 u 的已有记录项目集合 $i \in I_u$, 对其中的每一项目计算关联度, 以关联度大小排序得到项目的 k 近邻居集 $N_u = \{i_1, i_2, \dots, i_k\}$, 从中删除所有 I_u 已存在的项目, 得到候选集 C 。这样候选集中的项目很大程度上与已有项目有强关联, 这种关系能够大概率击中用户在未来的需求或兴趣。另一方面, 在评分预测的过程中, 继续使用相似度作为加权评分中的比重, 这样可以保留传统

协同过滤的高准确率,又可以避免出现在候选集中引入太多弱相关的项目而导致的低占比问题。

3 算法评估

3.1 算法流程

如上文所述,文中采用关联度生成项目候选集,对其中的项目使用相似度按式9计算加权平均预测分,然后根据预测的评分按高低排序,最后将top-N项目推荐给用户。

$$P_{i,c} = \bar{R}_i + \frac{\sum_{j \in NN_u} \text{sim}_3(i,j) * (R_{j,c} - \bar{R}_j)}{\sum_{j \in NN_u} |\text{sim}_3(i,j)|} \quad (9)$$

其中, $\text{sim}_3(i,j)$ 表示用户 i 和 j 的相似度; $R_{j,c}$ 表示用户 j 对目标项目 c 的评分; \bar{R}_i 和 \bar{R}_j 分别表示用户 i 和 j 的平均评分; NN_u 表示根据关联矩阵生成的用户 u 的候选项目集。

算法流程如下:

步骤1:在评分矩阵上根据提出的“S”型相似度计算方法和关联度计算方法,得到项目间的相似度矩阵 S 和关联矩阵 R ;

步骤2:在关联矩阵 R 中,对用户 u 的每一项目 I_u 获取 k 最邻居集 $N_u = \{i_1, i_2, \dots, i_k\}$, 合并所有 N_u 得到预选集合 C ;

步骤3:剔除集合 C 中用户 u 已有记录的部分,得到项目候选集 C ;

步骤4:对候选集中的所有项目 $e \in C$, 依据相似度和式9计算项目 e 的加权平均预测评分;

步骤5:对步骤4中所有项目的预测分由高到低排序,将每一用户 u 的 top-N 项目作为用户 u 的推荐对象。

3.2 实验结果及分析

为了验证文中算法的效果,选用 MovieLens^[12] 中 1M 数据集,包括了 6 040 个用户对 3 952 部电影的评分。文中采用 10 折交叉验证^[13] 的方法,将数据集随机划分为 10 个子集,每次选用其中 9 份用作训练,1 份作为测试,重复 10 次,取 10 次结果的平均值作为实验结果。

实验一:比较提出的融合“S”型相似度和关联度的协同过滤算法(CSRCF)和经典的协同过滤(IBCF)的候选集的大小。图5(a)展示的是在 MovieLens 数据集得到的上述两种算法生成的平均候选集大小。开始阶段,两种算法的候选集基本相同,随着近邻数的增长,IBCF 算法候选集会迅速增大,当近邻数达到 60 时,候选集将达到 200 左右;而 CSRCF 算法则增长十分缓慢,当近邻数增加到 50 后,候选集大小几乎收敛

在 75 左右,仅为前者的 1/3。两者对比之下,发现文中提出的依据关联矩阵生成的候选集效果十分显著,大幅降低了候选集的规模,避免在评分阶段计算过多无效的项目。

实验二:上述讨论了两种算法在候选集规模上的区别,接下来讨论在减小候选集规模时,候选集中感兴趣占比会如何变化。由图5(a)可以看出,初始阶段 CSRCF 比 IBCF 有着明显优势,随着近邻数的增加,两者的用户感兴趣占比都在下降,且保持着较大差距,IBCF 仅为不到 5%,最后 CSRCF 的用户感兴趣占比稳定在 15% 左右,比前者高出 10%。

综合上述两个实验,可以认为 IBCF 算法根据相似度选取项目候选集时引入了不相关或弱相关的项目,导致候选集规模过大且用户感兴趣比例较小;而文中提出的 CSRCF 算法,从项目间的关联性出发选取候选集,保证了候选集中的强相关性,在保证不减少用户感兴趣项目数量的前提下,大幅减小了候选集的规模且提升了用户的感兴趣比例,为后续预测阶段减少了不必要的项目评分预测计算。

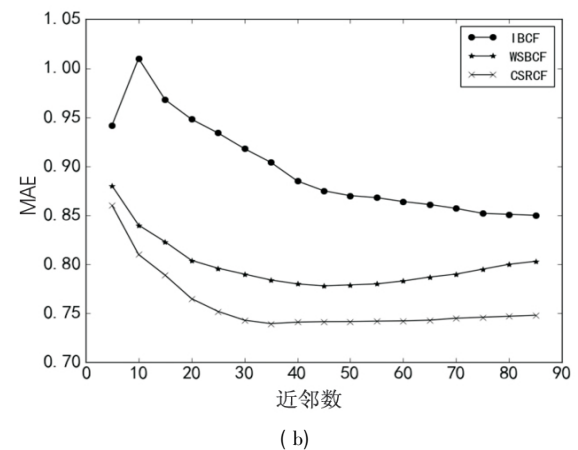
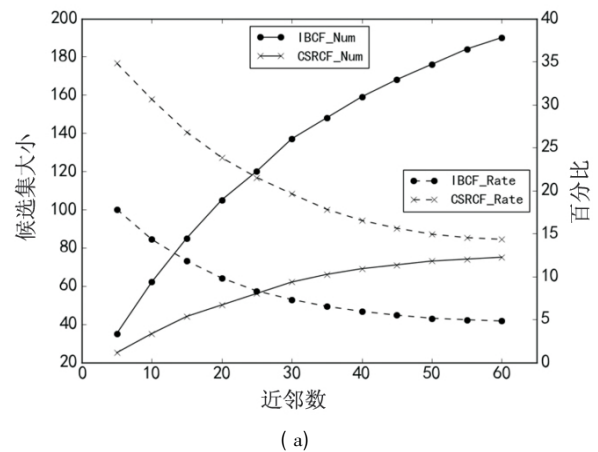


图5 候选集及感兴趣项目占比(a)和 MAE 比较(b)

实验三:为了验证 CSRCF 算法的推荐效果,将其与文献[10]提出的加权相似度协同过滤算法(WSBF)以及传统的协同过滤算法(IBCF)进行比较。在

离线环境下,文中采用 MAE(平均绝对误差)^[14]作为评估度量,三种算法 MAE 准确性结果如图 5(b) 所示。可以看出,随着邻居规模的增长,三种推荐算法的 MAE 都在减小,说明邻居规模的增大一定程度上提高了准确性。传统的 IBCF 随着邻居规模增大 MAE 缓慢减小,最后 MAE 基本稳定在 0.85 左右;WSBCF 算法在开始时推荐准确性显著长,在邻居规模为 [45, 55] 内达到推荐准确峰值,CSRCF 算法在开始阶段与 WSBCF 类似,MAE 迅速下降,下降的速率较前者要快,然后在邻居规模为 35 时到达最优推荐效果,两种算法都在邻居规模继续增大时 MAE 反而会增大,表明过多的邻居规模会导致并不相似的用户参与评分预测,引入不相关的评分噪声导致推荐效果下降。

注意到 CSRCF 不论是在最优推荐效果(MAE 约为 0.74)还是达到最优推荐效果时用户的邻居规模较上述算法都有显著提升,更小规模的邻居用户在大数据量推荐时减少计算负担;较之前两种算法,CSRCF 算法的 MAE 值更小且收敛过程中下降得更快,证明提出的 CSRCF 算法能通过“S”型重合因子准确地描述用户间相似度与共有评分项的真实关系,从而提高后续评分预测的准确性。

4 结束语

大多数推荐系统中,相似度是生成候选集和评分预测的基础,决定着推荐的质量。文中以推荐步骤为出发点,创造性地分离候选集生成和评分预测。针对候选集中存在大量弱或不相关的项目和用户感兴趣比例较低的问题,引入关联度的概念,使用关联矩阵代替相似度矩阵生成候选集来减小候选集规模;评分预测阶段分析相似度对推荐效果的影响,对传统相似度和基于加权相似度无法准确描述相似度的问题进行总结,详细阐述了项目间相似度增长的理想曲线,提出了一种细粒度“S”型相似度来刻画理想相似项目,最后在算法流程中融合关联度和“S”型相似度。实验结果表明,关联度的引入减小了候选集规模,仅为前者的 1/3,避免了预测阶段对无效的项目进行评分,加快了推荐速度,从算法层面提高了可扩展性,改进的“S”型相似度在推荐准确率上较前者提高了 4%。但未考虑在大数据环境下改进算法的时间复杂度,因此下一步将研究如何在分布式环境下提升该推荐算法的可扩展性。

参考文献:

- [1] LI Guojie, CHENG Xueqi. Research status and scientific thinking of big data[J]. Bulletin of Chinese Academy of

Sciences 2017 27(6):647-657.

- [2] 孟祥武,刘树栋,张玉洁,等. 社会化推荐系统研究[J]. 软件学报 2015 26(6):1356-1372.
- [3] SARWAR B, KARYPIS G, KONSTAN J, et al. Analysis of recommendation algorithms for e-commerce[C]//ACM conference on electronic commerce. Minneapolis, USA: ACM 2000:158-167.
- [4] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of 10th international conference on world wide web. Hong Kong: ACM 2001:285-295.
- [5] SARWAR B M, KONSTAN J A, BORCHERS A, et al. Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system[C]//Proceedings of 1998 ACM conference on computer supported cooperative work. Washington: ACM 1998:345-354.
- [6] REN L, GU J, XIA W. A weighted similarity-boosted collaborative filtering approach[J]. Energy Procedia 2011 13:9060-9067.
- [7] 孔小华. 协同过滤系统的稀疏性与冷启动问题研究[D]. 杭州:浙江大学 2005.
- [8] HERLOCKER J, KONSTAN J A, RIEDL J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms[J]. Information Retrieval 2002 5(4):287-310.
- [9] KARAHODZA B, SUPIC H, DONKO D. An approach to design of time-aware recommender system based on changes in group user's preferences[C]//X international symposium on telecommunications. Sarajevo, Bosnia and Herzegovina: IEEE 2014:1-4.
- [10] 任磊. 推荐系统关键技术研究[D]. 上海:华东师范大学 2012.
- [11] KONG W, LIU Q, WANG S, et al. Relation-based collaborative filtering algorithm[J]. Journal of Computational Information Systems 2012 8(15):6257-6265.
- [12] MILLER B N, ALBERT I, LAM S K, et al. MovieLens unplugged: experiences with an occasionally connected recommender system[C]//Proceedings of the 8th international conference on intelligent user interfaces. Miami, Florida, USA: ACM 2003:263-266.
- [13] KOHAVI R. A study of cross-validation and bootstrap for accuracy estimation and model selection[C]//International joint conference on artificial intelligence. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995:1137-1143.
- [14] HERLOCKER J L, KONSTAN J A, TERVEEN LG, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems 2004 22(1):5-53.