

时间衰减制导的协同过滤相似性计算^①

李源鑫, 肖如良, 陈洪涛, 赵 婷, 李宏恩

(福建师范大学 软件学院, 福州 350108)

摘 要: 协同过滤算法中, 相似性计算方法是整个推荐系统的关键, 决定着推荐系统的推荐质量, 为了提高相似性计算的精准性, 提出了一种基于时间衰减的相似性计算方法. 该方法在计算用户相似性时, 考虑目标物品与共同评分物品的相似性, 同时在计算用户与物品相似性时, 考虑时间信息(用户对物品产生行为的时间)对相似性的影响. 实验结果表明, 该方法能够有效地避免传统相似性计算方法的不足, 使推荐系统获得更好的推荐效果.

关键词: 推荐系统; 协同过滤; 相似性计算; 时间信息

Time Decay Guided Similarity Calculation in Collaborative Filtering

LI Yuan-Xin, XIAO Ru-Liang, CHEN Hong-Tao, ZHAO Ting, LI Hong-En

(Faculty of Software, Fujian Normal University, Fuzhou 350108, China)

Abstract: In collaborative filtering algorithm, similarity calculation method is undoubtedly the key of the entire recommender system, and determines the recommended quality of recommender system. In order to improve the precision of similarity calculation, a new similarity calculation method based on time decay is proposed. In this method, the similarity between the objective item and common-scored items is taken into account when calculating users similarity, and the influence of time information is taken into account when calculating users and items similarity. Experimental results shows that this method can effectively avoid the lack of the traditional similarity calculation method, and make the recommender system gain better recommendation result.

Key words: recommender system; collaborative filtering; similarity calculation; time information

1 引言

随着信息技术的发展和电子商务的普及, 人们逐渐进入信息过载的时代, 商品的选择越来越多, 但其结构也越来越复杂, 用户需要花费大量的时间才能找到自己想要的商品^[1], 这使得个性化推荐系统逐渐成为电子商务中的一个重要内容, 一些电子商务系统, 如淘宝、亚马逊、京东等, 推荐系统已成为他们业务增长的重要因素. 个性化推荐系统能够帮助用户立刻找到他们想要的信息.

传统的基于邻域的协同过滤推荐算法可分为两类: 基于用户协同过滤(UserCF)^[2,3]和基于物品协同过滤(ItemCF)^[4,5]. 利用传统的基于用户的协同过滤方法得出推荐通常需要以下几个步骤: 首先, 收集一个目标用户志趣相投的最近邻用户的行为信息; 接着, 计算

这个目标用户和其他用户之间的相似性, 确定这个目标用户的 K 个最近邻居; 最后, 根据目标用户的 K 个最近邻居预测这个目标用户对物品的感兴趣程度, 产生 Top-N 推荐结果(推荐目标用户感兴趣程度最高的前 N 个物品). 由此可见, 相似性的计算是决定推荐性能的关键, 然而相似性计算的精准程度仍然存有很大的提升空间^[6,7].

本文提出了一种基于时间衰减的相似性计算方法, 主要应用在隐反馈数据集的 Top-N 推荐问题上, 旨在提高协同过滤算法中相似性计算的精准性. 下面先介绍协同过滤算法的相关研究工作, 第 3 节提出融入时间衰减参数的相似性计算原理, 并将其应用于协同过滤推荐; 在第 4 节对提出的方法进行实验验证与结果分析; 最后进行了总结.

^① 基金项目:教育部规划基金(11YJA860028);福建省科技计划重大项目(2011H6006)

收稿时间:2013-04-18;收到修改稿时间:2013-05-15

2 相关工作

在基于用户的协同过滤算法中,确定目标用户的最近邻居是决定推荐性能的关键,目标用户的最近邻居是通过计算用户和其他用户之间的相似性决定的,因此,相似性的计算成为了整个推荐算法的关键,选择恰当的相似性计算方法可以明显地提高推荐系统的推荐效率.

传统的相似性计算主要包含用户相似性计算和物品相似性计算,计算公式如下^[8]:

1) 用户之间的相似性表示用户兴趣爱好的相似程度,利用用户 u 和用户 v 的共同评分数据来计算用户 u 和用户 v 之间的相似性 $sim(u,v)$, 表示为:

$$sim(u,v) = \frac{\sum_{j \in N(u) \cap N(v)} 1}{\sqrt{|N(u) \cap N(v)|}} \quad (1)$$

$sim(u,v)$ 表示目标用户 u 与用户 v 之间的相似性, $N(u)$ 与 $N(v)$ 分别表示用户 u 与用户 v 有过行为的物品集合.

2) 物品之间的相似性表示用户对若干物品同时感兴趣的程度,选择用户群体对物品 i 和物品 j 的共同评分数据来计算物品 i 和物品 j 之间的相似性 $sim(i,j)$, 表示为:

$$sim(i,j) = \frac{\sum_{u \in N(i) \cap N(j)} 1}{\sqrt{|N(i) \cap N(j)|}} \quad (2)$$

$sim(i,j)$ 表示物品 i 与物品 j 之间的相似性, $N(i)$ 与 $N(j)$ 分别表示喜欢物品 i 与喜欢物品 v 的用户集合.

相似性的计算是决定推荐性能的关键,国内外已有许多学者在这方面进行了研究:文献[8]提出了时间上下文相关的 UserCF 和 ItemCF 算法,在相似性计算时加入时间上下文信息,使相似性计算具有时效性;文献[9]提出了一种基于共同评分和相似性权重的协同过滤推荐算法,通过相似性权重结合基于用户协同过滤和基于物品协同过滤,优于传统的协同过滤推荐算法;文献[10]提出了一种基于用户兴趣的混合推荐模型,在计算相似性时,将内容过滤和协同过滤的预测值进行加权求和,提高了推荐质量;文献[11]中 Keunho Choi 等人在显性反馈数据集评分预测问题上,计算目标用户与其他用户相似性时,将两个用户共同评分物品与一个目标物品的相似性作为权重带入计算,使目标用户对目标物品的评分更为精准,提高了推荐性能. 以上的方法都从相似性计算的角度提高了推荐质量,然而相似性计算的精准程

度仍然存有很大的提升空间^[6,7],通常情况下,各种方法计算出的相似性值与相似性的真实值之间,总是存在着一定的误差,如果误差越小、计算值越接近真实值,那么最终的推荐结果也就越令人满意.

在隐反馈数据集中,基于用户的协同过滤,是通过整合一个目标用户志趣相投的最近邻用户的行为信息,预测这个目标用户是否会对目标物品做出响应,得出推荐结果.传统方法认为,两个用户的相似性是他们是否对同一个物品有过行为,但在计算目标用户与其他用户相似性时,没有设定一个目标物品,这些被两个用户共同有过行为的物品,被赋予的权重是相等的,从而使得目标用户与其他用户的相似性不会因为物品的不同而有所差异,用户的 K 个最近邻对于所有的物品都是固定的一群人,最后得出目标用户对所有物品的兴趣程度,都是通过同样一群的目标用户 K 个最近邻来计算.也就是说,基于用户的协同过滤方法,忽略了共同评分的物品与目标物品之间的相似性.

与此同时,传统的相似性计算方法没有考虑时间信息(用户对物品产生行为的时间)对相似性的影响.比如,在计算物品相似性时,某购物网站的用户 A 一年前对物品 a 很感兴趣,但是最近一段时间对物品 b 和物品 c 非常感兴趣,在统计意义上,我们会认为物品 b 和物品 c 之间的相似性大于物品 a 与物品 c 之间的相似性,因为用户在相隔很短的时间内喜欢的物品具有更高的相似性,但是传统方法忽略了时间的影响,将用户 A 在一年前的兴趣与他最近的兴趣等同考虑,使得物品 b 和物品 c 之间的相似性等于物品 a 与物品 c 之间的相似性.又比如,在计算用户相似性时,用户 A 在一年前对物品 a 感兴趣,用户 B 和用户 C 在最近这段时间同时对物品 a 感兴趣,在实际情况中,我们会认为用户 B 和用户 C 之间的相似性要大于用户 A 与用户 C 之间的相似性,因为两个用户同时喜欢相同的物品,他们应该具有更高的相似性,然而传统方法会认为用户 B 和用户 C 之间的相似性要等于用户 A 与用户 C 之间的相似性,并没有体现出时间信息的影响.由此可见,传统的相似性计算方法没有考虑到时间对用户兴趣所带来的影响,将用户不同时间的兴趣等同考虑,导致相似性计算缺乏时效性^[12].

另外,评分预测问题一直是推荐系统研究的热点,目前已经有了许多基于评分数据的评分预测研究.典型的有 GroupLens 研究组基于电影评分数据 MovieLens

进行的推荐系统研究, 还有著名的 Netflix 大赛也主要面向评分预测问题, 很多研究人员都将研究精力集中在优化评分预测的评测指标上. 但亚马逊前科学家 Greg Linden 却认为电影推荐的目的是找到用户最有可能感兴趣的电影, 而不是预测用户看了电影后会给电影什么样的评分, Top- N 推荐更符合实际的应用需求.

本文方法着重在隐反馈数据集的 Top- N 推荐问题, 忽略数据集中的评分记录, 该方法旨在提高相似性计算的准确性, 从而提高传统协同过滤算法的推荐质量.

3 基于时间衰减的用户相似性计算原理

利用传统的基于用户的协同过滤方法进行推荐, 需要为每个用户预测出他对每个物品的感兴趣程度, 我们可以把这个过程看成是一个循环计算的过程. 在某次循环中, 当我们要预测某一个用户对某一个物品的感兴趣程度时, 我们就把当前这个用户称为本次循环的“目标用户”, 此时被预测的物品就称为本次循环的“目标物品”.

3.1 考虑目标物品相似性的用户相似性计算

我们将文献[11]中的方法应用到隐性反馈数据集中, 在计算目标用户对目标物品的感兴趣程度之前, 计算目标用户与其他用户相似性的时候, 将这两个用户共同评分物品与目标物品的相似性作为权重, 带入计算中:

$$sim(u, v) = \frac{\sum_{j \in N(u) \cap N(v)} sim(i, j)}{\sqrt{|N(u)| |N(v)|}} \quad (3)$$

u 表示目标用户, i 表示目标物品, $sim(u, v)$ 表示我们改进后的目标用户 u 与用户 v 之间的相似性, $sim(i, j)$ 表示目标物品 i 与被用户 u 与用户 v 共同有过行为的物品 j 之间的相似性, 可以利用公式(2)进行计算.

3.2 加入时间信息的用户相似性计算

在相关工作中, 我们提到了时间信息对相似性的影响, 在此, 我们将时间差因素加入到公式(3)的相似性计算中, 得到以下改进后的用户相似性计算公式:

$$sim(u, v) = \frac{\sum_{j \in N(u) \cap N(v)} sim(i, j) \frac{1}{1+a|t_{uj}-t_{vj}|}}{\sqrt{|N(u)| |N(v)|}} \quad (4)$$

其中,

$$sim(i, j) = \frac{\sum_{x \in N(i) \cap N(j)} \frac{1}{1+b|t_{xi}-t_{xj}|}}{\sqrt{|N(i)| |N(j)|}} \quad (5)$$

u 表示目标用户, i 表示目标物品, $sim(u, v)$ 表示加入时间信息的改进后的用户相似性, $sim(i, j)$ 表示加入时间信息的物品相似性.

上面的公式在分子中引入了和时间有关的衰减项

$$\frac{1}{1+a|t_{uj}-t_{vj}|} \text{ 和 } \frac{1}{1+b|t_{xi}-t_{xj}|},$$

其中 t_{uj} 和 t_{vj} 分别表示用户 u 和用户 v 对物品 j 产生行为的时间, t_{xi} 和 t_{xj} 分别表示用户 x 对物品 i 和物品 j 产生行为的时间, a 和 b 是时间衰减参数.

用户对物品产生行为, 可以指用户在视频网站上对某部电影评分的行为, 或者是用户在购物网站上浏览一个商品页面的行为, 等等. 用户对物品产生行为的时间代表着用户对物品感兴趣的时间.

公式(4)的分子对于用户 u 和用户 v 共同感兴趣的物品增加了一个与时间有关的衰减项. 我们可以找到很多数学衰减项的表示方法, 依照文献[8], 使用

$$\frac{1}{1+a|t_{uj}-t_{vj}|}$$

来表示衰减项. 衰减项的解释是, 用户的兴趣可能会随着时间的变化而变化, 如果用户 u 和用户 v 对物品 i 产生行为的时间越近, 那么这两个用户之间的相似性就越大; 如果用户 u 和用户 v 对物品 i 产生行为的时间越远, 那么这两个用户之间的相似性就越小. 因此可以用时间差来局部影响用户相似性的大小, 将时间差因素引入公式形成衰减项, 其中衰减参数 a 就是用来控制这种影响力的大小.

对于公式(5), 类似地, 在公式分子中加入与时间有关的衰减项, 采用 $\frac{1}{1+b|t_{xi}-t_{xj}|}$ 来表示衰减项. 同样

有这样的解释: 用户的兴趣可能会随着时间的变化而变化, 如果用户 x 对物品 i 和物品 j 产生行为的时间越近, 那么这两个物品之间的相似性就越大; 如果用户 x 对物品 i 和物品 j 产生行为的时间越远, 那么这两个物品之间的相似性就越小. 因此, 可以将时间差因素引入到公式中, 形成衰减项, 衰减参数 b 就是用来控制这种因素对物品相似性计算的影响力大小.

这种时间衰减制导的相似性计算方法, 更适合于用户兴趣变化较快、较多的情景.

3.3 基于相似性计算原理的 Top- N 推荐

利用上文的用户相似性计算方法找到目标用户的 K 个最近邻居用户, 接下来就要产生相应的 Top- N 推

荐结果. 采用如下公式预测目标用户 u 对目标物品 i 的感兴趣程度^[8]:

$$p(u, i) = \sum_{v \in S(u, K)} \text{sim}(u, v) r_{vi} \quad (6)$$

$p(u, i)$ 表示目标用户 u 对目标物品 i 的兴趣程度, $S(u, K)$ 指的是对于目标物品 i , 与用户 u 兴趣最接近的 K 个用户. 因为我们着重于隐反馈数据集的 Top- N 推荐问题, 所以如果用户 v 对物品 i 有过行为, 那么 $r_{vi} = 1$, 否则 $r_{vi} = 0$. 该公式的含义是, 与目标用户越相似的用户如果对目标物品感兴趣, 那么目标用户对目标物品的感兴趣程度就会越高.

在得出目标用户对当前目标物品的兴趣程度后, 继续计算目标用户对下一个目标物品的兴趣程度, 最后选出这个目标用户感兴趣程度最高的 N 个物品, 给出 Top- N 推荐结果.

4 实验结果与分析

4.1 数据集

实验使用的是 Netfilx 数据集, 它包含了 48 万个用户对 1.7 万个电影超过 1 亿条的评分行为记录(5 分制)以及评论日期(精确到天), 这些数据的评论时间都是在 1998 年 10 月到 2005 年 12 月之间.

本文的实验先是从 Netfilx 数据集上随机地提取出 1500 个用户与 1000 个物品的子集; 将抽取出的用户 ID 与电影 ID 重新做映射调整, 用户 ID 对应到 1-1500, 电影 ID 对应到 1-1000.

实验中, 我们将处理好的用户行为子集按照均匀分布随机生成 8 份, 挑选一份作为测试集, 将剩下的 7 份作为训练集. 然后在训练集上建立用户兴趣模型, 并在测试集上对用户行为进行预测, 统计出相应的评测指标.

4.2 评测指标

本次实验最终所给出的是 Top- N 推荐列表, 因此实验采用召回率(Recall)、准确率(Precision)和覆盖率(Coverage)来作为评测指标^[13].

召回率(Recall)描述有多少比例的用户-物品评分记录包含在最终的推荐列表中, 召回率越高, 表明推荐结果越好, 它表示为 0:

$$\text{Recall} = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |T(u)|} \quad (7)$$

其中, $R(u)$ 为推荐给用户 u 的 N 个物品, $T(u)$ 为用户 u 在测试集上喜欢的物品集合.

准确率(Precision)描述有最终的推荐列表中有多少比例是发生过的用户-物品评分记录, 准确率越高, 表明推荐结果越精准, 它表示为:

$$\text{Precision} = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |R(u)|} \quad (8)$$

其中, $R(u)$ 为推荐给用户 u 的 N 个物品, $T(u)$ 为用户 u 在测试集上喜欢的物品集合.

覆盖率(Coverage)描述最终的推荐列表中包含多大比例的物品, 如果所有的物品都被推荐给至少一个用户, 那么覆盖率就是 100%. 它反映了推荐算法发掘长尾的能力, 覆盖率越高, 说明推荐算法越能够将长尾中的物品推荐给用户, 它表示为:

$$\text{Coverage} = \frac{|\bigcup_{u \in U} R(u)|}{|I|} \quad (9)$$

其中, $R(u)$ 为推荐给用户 u 的 N 个物品, I 为训练集上的物品集合.

4.3 时间衰减参数调整

公式(4)和公式(5)中, 参数 a 和参数 b 是时间衰减参数, 它们分别影响的是时间信息相关的 UserCF 和时间信息相关的 ItemCF, 它们的取值在不同系统、不同数据集的情况下都会有所不同, 并没有特定的取值范围, 在实际应用中, 需要先初步了解用户兴趣变化的情况, 然后通过反复试验来取得参数的最优值.

用户相似性的时间衰减参数 a 是用来控制时间差因素对用户相似性计算的影响力的大小: 如果某个数据集的用户兴趣变化很快, 即时间差因素对用户相似性计算的影响较大, 那么即使两个用户同一时间对某样物品感兴趣, 产生行为的时间差很小, 计算出的相似性很大, 这并不能说明这两个用户具有很大的相似性, 这时就需要取较大的 a 来惩罚时间差因素的影响; 如果某个数据集的用户兴趣变化很慢, 即时间差因素对用户相似性计算的影响较小, 那么即使两个用户在相隔很久的时间段对某样物品感兴趣, 产生行为的时间差很大, 计算出的相似性很小, 也不能说明这两个用户的相似性很小, 这时就需要取较小的 a 来惩罚时间差因素的影响.

物品相似性的时间衰减参数 b 是用来控制时间差因素对物品相似性计算的影响力的大小: 如果某个数据集的用户兴趣变化很快, 即时间差因素对物品相似性计算的影响较大, 那么即使某个用户在同一时间

对两样物品感兴趣,产生行为的时间差很小,计算出的相似性很大,这并不能说明这两个物品具有很大的相似性,这时就需要取较大的 b 来惩罚时间差因素的影响;如果某个数据集中的用户兴趣变化很慢,即时间差因素对物品相似性计算的影响较小,那么即使某个用户在相隔很久的时间段对两样物品感兴趣,产生行为的时间差很大,计算出的相似性很小,也不能说明这两个物品的相似性很小,这时就需要取较小的 b 来惩罚时间差因素的影响。

现在,我们将对这两个参数进行调整,由于这两个参数分别影响的是用户相似性与物品相似性,因此可以利用公式(4)和公式(5)分别调整。

4.3.1 用户相似性的时间衰减参数调整

首先,利用公式(4)和公式(6)进行 Top-N 推荐,这时先不考虑物品相似性的影响,令最近邻居数 $K=10$, $sim(i,j)=0$, 得出推荐效果最佳的参数 a 。图 1 表明了随着参数 a 的变化,召回率的变化情况;图 2 表明了随着参数 a 的变化,准确率的变化情况;图 3 表明了随着参数 a 的变化,覆盖率的变化情况。其中,当 $a=0$ 时,是传统的 UserCF 算法的性能。

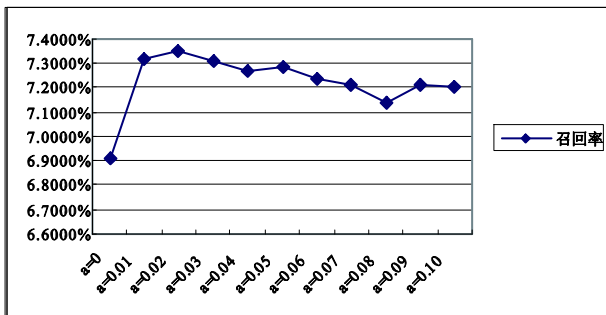


图 1 时间信息相关的 UserCF 中衰减系数 a 与召回率的关系

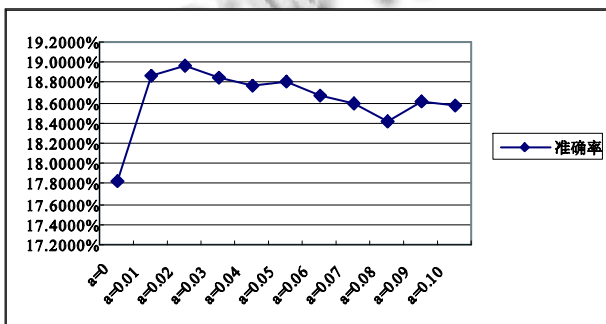


图 2 时间信息相关的 UserCF 中衰减系数 a 与准确率的关系

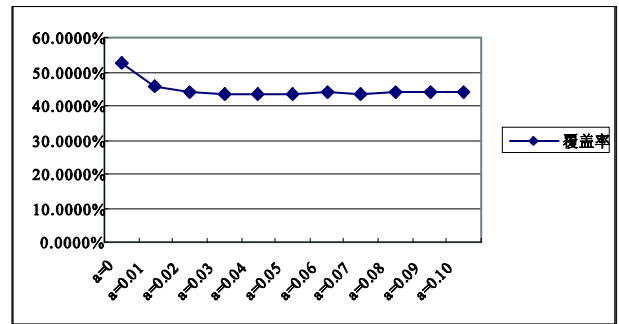


图 3 时间信息相关的 UserCF 中衰减系数 a 与覆盖率的关系

可以看出,在本实验的数据集上,时间信息相关的 UserCF 算法比传统的 UserCF 算法(当 $a=0$ 时)有更高的召回率与准确率,覆盖率有所下降;同时,当 $a=0.02$ 时,时间信息相关的 UserCF 算法的召回率与准确率最高,覆盖率也在可接受的范围内。综合考虑后,我们选择 $a=0.02$ 来进行接下来的实验。

4.3.2 物品相似性的时间衰减参数调整

利用公式(5)与公式(6)进行 Top-N 推荐,令最近邻居数 $K=10$, 得出推荐效果最佳的参数 b 。图 4 表明了随着参数 b 的变化,召回率的变化情况;图 5 表明了随着参数 b 的变化,准确率的变化情况;图 6 表明了随着参数 b 的变化,覆盖率的变化情况。其中,当 $b=0$ 时,是传统的 ItemCF 算法的性能。

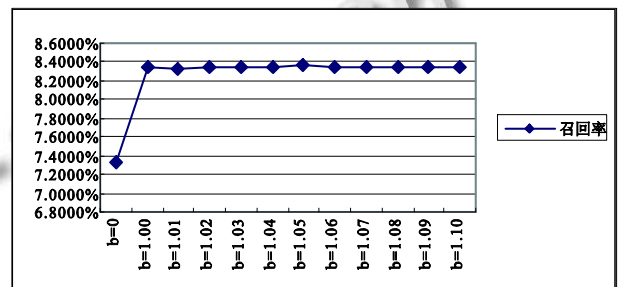


图 4 时间信息相关的 ItemCF 中衰减系数 b 与召回率的关系

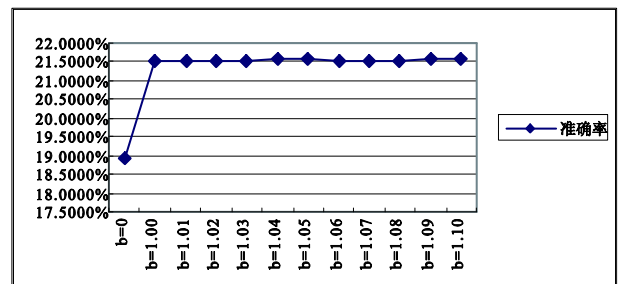


图 5 时间信息相关的 ItemCF 中衰减系数 b 与准确率的关系

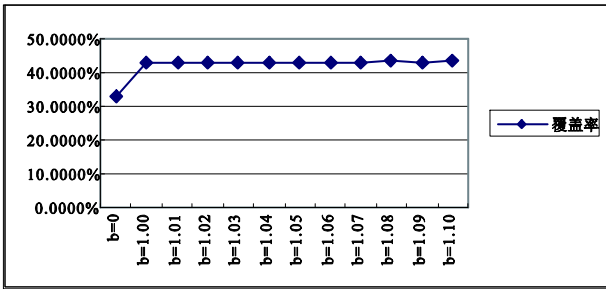


图 6 时间信息相关的 ItemCF 中衰减系数 b 与覆盖率的关系

可以看出, 在本实验的数据集上, 时间信息相关的 ItemCF 算法比传统的 ItemCF 算法(当 $b=0$ 时)能有更高的召回率、准确率与覆盖率; 同时, 当 $b=1.05$ 时, 时间信息相关的 ItemCF 算法的召回率与准确率会稍微高一点, 覆盖率没有明显差异. 综合考虑后, 我们选择 $b=1.05$ 来进行接下来的实验.

4.3.3 推荐质量的比较

为了检验本文提出的相似性计算方法的有效性, 我们将其运用到用户协同过滤推荐中, 并传统协同过滤算法的推荐结果作为比较, 对比它们在不同最近邻居下的结果. 在传统的协同过滤算法中, 分别以基于用户的协同过滤算法与基于物品的协同过滤算法作为参照, 跟利用本文相似性计算方法的用户协同过滤算法(参数 $a=0.02, b=1.05$)作比较. 图 7 是召回率的对比情况, 图 8 是准确率的对比情况, 图 9 是覆盖率的对比情况.

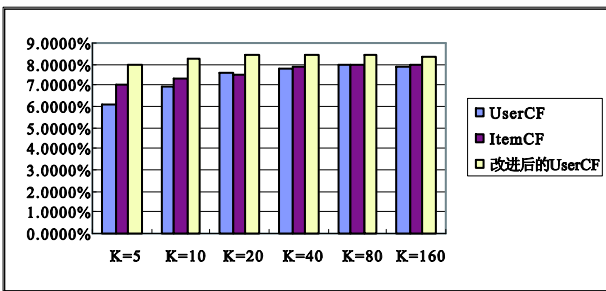


图 7 不同方法的召回率对比

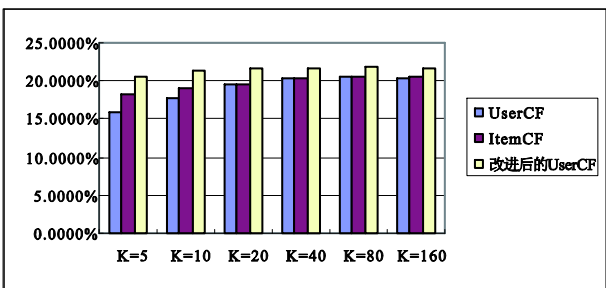


图 8 不同方法的准确率对比

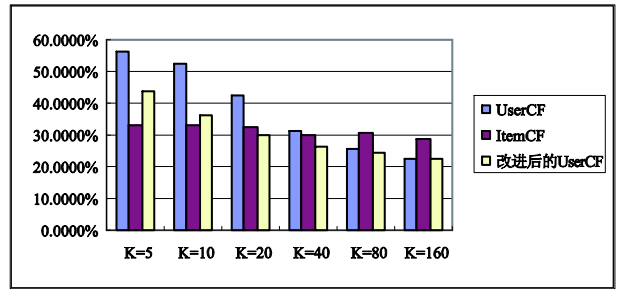


图 9 不同方法的覆盖率对比

从以上评测标准的对比结果可知, 利用本文相似性计算方法改进后的 UserCF 算法, 在召回率与准确率上都能够比传统的 UserCF 与传统的 ItemCF 算法有更好的表现, 当 $K=80$ 时, 改进后的 UserCF 算法能获得最高的召回率与准确率. 这 3 个算法的覆盖率都是随着 K 的增加而逐渐下降的, 虽然改进后的 UserCF 算法在覆盖率上并不总是比传统的 UserCF 和 ItemCF 高, 但从侧重召回率与准确率的角度看, 覆盖率的下降也在可接受的范围内. 由此可见, 利用本文相似性计算原理的协同过滤算法在一定程度上能够比传统的协同过滤算法有更好的性能与效果.

5 总结

本文针对传统相似性计算方法在计算用户相似性时忽略目标物品相似性、忽略用户兴趣时效性的缺陷, 将时间衰减因素融合到相似性计算中来, 提出了改进的相似性计算方法, 并应用改进后的相似性计算方法进行 Top-N 推荐. 本文的方法在计算用户相似性时, 考虑共同评分的物品与一个目标物品的相似性, 同时考虑到时间信息(用户对物品产生行为的时间)对相似性的影响. 实验结果表明, 使用该方法进行推荐在一定程度上能够比传统的用户协同过滤和传统的物品协同过滤推荐获得更好的推荐质量, 为以后研究推荐算法提供了一个新的途径. 下一部工作的重心, 将会研究如何提高本方法推荐结果的覆盖率.

参考文献

- 郭均鹏, 陈莹莹. 改进的基于符号数据的协同过滤推荐算法. 计算机应用, 2011, 31(11): 3060-3062.
- 丛晓琪, 杨怀珍, 刘枚莲. 基于时间加权的协同过滤算法研究. 计算机应用与软件, 2009, 26(8): 120-121.
- 邢春晓, 高凤荣, 战思南. 适应用户兴趣变化的协同过滤推荐

(下转第 158 页)

节点密度较大的情况下, P-ODGC 算法产生的工作节点数目接近于最少工作节点数目, 说明了算法的有效。

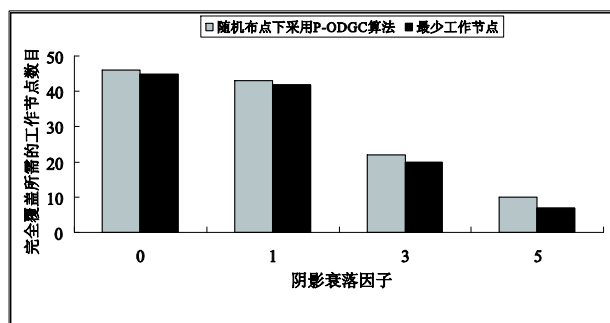


图6 P-ODGC算法与理想情况下工作节点数目的比较

5 结束语

密度控制算法是一种常用的能有效解决能量消耗过大问题的方法, 本文主要研究了在阴影衰落环境下, 采用基于概率的协作感知覆盖模型, 对 OGDC 算法^[4]进行扩展和改进, 提出了一种分布式覆盖控制算法——P-OGDC 算法。该算法能以尽可能少的工作节点来保证网络的覆盖质量要求, 从而达到了节约能量的目

的。实验结果表明, 当节点密度越大, 所需的工作节点数目接近最少工作节点数目, 说明了算法的有效性。

参考文献

- 1 刘乃安. 无线局域网—原理、技术与应用. 西安: 西安电子科技大学出版社, 2004: 32–39.
- 2 Sheu JP, Lin HF. Probabilistic coverage preserving protocol with energy efficiency in wireless sensor networks. Wireless Communications and Networking Conference, Kowloon, 11–15 March 2007. 2007. 2633–2638.
- 3 刘益. 阴影衰落环境下无线传感网络的概率覆盖研究. 电子技术应用, 2011, 37(8): 98–104.
- 4 Zhang H, Hou JC. Maintaining sensing coverage and connectivity in large sensor networks. Ad Hoc & Sensor Networks. 2005, Old City, 2005, 1: 89–124.
- 5 宗大伟. Voronoi 图及其应用研究[博士学位论文]. 南京: 南京航空航天大学, 2009.
- 6 Wang B, Wang W, Srinivasan V. Information coverage for wireless sensor networks. Communications Letters, 2005, 9(11): 967–969.
- 8 项亮, 陈义, 王益. 推荐系统实践. 人民邮电出版社, 2012: 45–53, 131–133.
- 9 汪静, 印鉴, 郑利荣等. 基于共同评分和相似性权重的协同过滤推荐算法. 计算机科学, 2010, 37(2): 99–104.
- 10 曹毅, 贺卫红. 基于用户兴趣的混合推荐模型. 系统工程, 2009, 27(6): 68–72.
- 11 Choi K, Suh Y. A new similarity function for selecting neighbors for each target item in collaborative filtering. Knowledge-Based Systems, 2013, 37(1): 146–153.
- 12 彭德巍, 胡斌. 一种基于用户特征和时间的协同过滤算法. 武汉理工大学学报, 2009, 31(3): 24–28.
- 13 刘建国, 周涛, 郭强等. 个性化推荐系统评价方法综述. 复杂系统与复杂性科学, 2009, 6(3): 1–10.

(上接第 134 页)

算法. 计算机研究与发展, 2007, 44(2): 296–301.

- 4 Chen G, Wang F, Zhang C. Collaborative filtering using orthogonal nonnegative matrix trifactorization. Information Processing & Management, 2009, 45(3): 368–379.
- 5 王岚, 翟正军. 基于时间加权的协同过滤算法. 计算机应用, 2007, 27(9): 2302–2303.
- 6 Bobadilla J, Ortega F, Hernando A. Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowledge-Based Systems, 2011, 24(8): 1310–1316.
- 7 Bobadilla J, Serradilla F, Bernal J. A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems, 2010, 23(6): 520–528.