时间加权不确定近邻协同过滤算法

郑志高[1] 刘 京[2] 王 平[3] 孙圣力[1]
(北京大学软件与微电子学院 北京 100260)[1] (北京大学信息科学技术学院 北京 100871)[2]
(北京大学软件工程国家工程研究中心 北京 100871)[3]

摘 要 围绕传统的协同过滤推荐算法存在的局限性展开研究,提出一种时间加权不确定近邻协同过滤推荐算法TWUNCF。根据推荐系统应用的实际情况,首先对用户和产品相似度进行时间加权以保证数据有效性,在此基础上改进相似度的计算方法。同时引入近邻因子在产品群和用户群中自适应地选择预测目标的近邻对象作为推荐群,计算推荐群中推荐概率较高的信任子群,最后通过不确定近邻的动态度量方法来对预测结果进行平衡的推荐。实验结果表明,该算法考虑了数据的时间有效性,同时平衡不同群体对推荐结果的影响,避免由于数据稀疏带来的推荐结果不准确和计算难度大的问题。理论分析和模拟实验证明,该算法在一定程度上提高了系统的准确性和推荐效率。

关键词 协同过滤算法,时间权重,不确定近邻,信任子群,推荐系统
中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.08.002

Time-weighted Uncertain Nearest Neighbor Collaborative Filtering Algorithm

ZHENG Zhi-gao[1] LIU Jing[2] WANG Ping[3] SUN Sheng-li[1]
(School of Software and Microelectronics, Peking University, Beijing 100260, China)[1]
(School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)[2]
(National Engineering Research Center of Software Engineering, Peking University, Beijing 100871, China)[3]

Abstract To overcome the limitations of the traditional collaborative filtering recommendation algorithm, this paper proposed a Time-Weighted Uncertain Nearest Neighbor Collaborative Filtering Algorithm (TWUNCF). According to the actual application situation of recommendation system, the author weighted the product similarity and user similarity to ensure the data validity firstly, and then improved the calculation method of the similarity. And then the author introduced the neighbor factor to select the trusted neighbors of the recommendation object adaptively. Based on these, balanced the prediction result by using dynamic metrics of uncertain nearest neighbors. Experimental results show that the algorithm can be used to improve data validity according to the time attribute, and balance the impact of the different groups on the recommendation result, and avoid the problems caused by the data sparseness. Theoretical analysis and simulation experiment show that the algorithm this paper proposed outperforms existing algorithms in recommendation quality, and improves the system's accuracy and recommendation efficiency.

Keywords Collaborative filtering, Time-weighted, Uncertain neighbors, Trustworthy subset, Recommendation system

1 引言

伴随着电子商务的普及和发展,众多的科研人员和学者为了更大程度地挖掘潜在的客户,针对推荐系统的推荐效率和准确度等方面都做了相关研究。众多的学者提出了多种不同的推荐算法,其中应用最为广泛的就是协同过滤算法。

目前基于协同过滤技术的研究主要分为基于用户的协同过滤算法(User Based Collaborative Filtering, UBCF)和基于产品的协同算法(Item Based Collaborative Filtering, IBCF)两种。无论是用户还是产品,个体之间都存在一定的差异性,从而导致推荐结果有所不同。在基于用户的推荐中,由于用户之间固有的差异性以及预测场景的不确定性和预测产品的可变性,导致传统的基于用户或产品的协同过滤算法存在一定的局限性,主要表现在以下3个方面:(1)很多学者通常采用kNN方法[1-3]为预测的目标选择推荐的对象,由于kNN方法是通过一定的相似性的比较来选择k个最近邻居,其在一定程度上可以代表预测目标的特性。但是k值是一个普遍的参数,通常不具有特殊性,使得这种方法在某些场景中不一定可用。例如,一个极端的情况就是该对象真正的邻居数量小于k时,kNN方法会将一些与该对象差异性很大的个体当作该

到稿日期:2013-10-21 返修日期:2013-12-11 本文受江苏省自然科学基金项目(BK2010139)资助。
郑志高(1988-),男,硕士生,CCF学生会员,主要研究方向为云计算、大数据和信息物理融合系统,E-mail:zhengzhigao@pku.edu.cn;刘 京(1981-),男,博士,讲师,主要研究方向为物联网工程、信息物理融合系统;王 平(1961-),男,博士,教授,博士生导师,主要研究方向为QoS、DiffServ,ATM、IP网络及交换机系统,数字家庭;孙圣力(1979-),男,博士,副教授,CCF会员,主要研究方向为服务计算、数据管理和数据挖掘等,E-mail:slsun@ss.pku.edu.cn(通信作者)。

对象的邻居进行推荐，使得推荐的结果不可信。(2)当前的推荐算法往往只针对用户或者产品中的某一个单独的群体进行推荐，忽视了另一群体的影响[3-6]。由于我们在进行推荐之前并没有对推荐对象各个维度的信任子群进行深刻的了解和认识，也没有研究这些维度的信任子群对推荐结果的影响，导致推荐质量在一定程度上无法满足人们在各个方面的需求。(3)传统的推荐算法在进行推荐时并没有考虑数据的价值会随着时间不断降低的现实，将不同时期的数据同等对待，从而在一定程度上影响了推荐的准确性和可行性。

为了解决以上3个方面的问题，本文在已有研究的基础上，采用动态选择的思想，在不同场景和不同要求下自适应地选取推荐对象在不同维度的信任子群作为推荐的候选集，并提出一种时间加权不确定近邻协同过滤算法(Time Weighted Uncertain Nearest Neighbor Collaborative Filtering, TWUNCF)。算法中应用logistic函数对评分进行时间加权，区分出不同时期的评分数据，充分考虑数据的价值随着时间的流逝不断减小的事实，保证数据对于时间的有效性。对于具有相同时间属性的数据则根据基于用户和产品的相似性进行计算，同时在两个推荐池中选择出预测目标的信任子群作为推荐的候选集，自适应地选择预测目标的近邻。实验表明本文设计的时间加权不确定近邻协同过滤算法可以有效平衡不同群体对推荐结果的影响，在解决用户评分数据稀疏问题上有良好表现，并且在计算过程中综合考虑了时间属性价值递减的特性对推荐结果的影响，本文提出的算法在一定程度上提高了推荐的质量并且具有较好的稳定性。本文第2节介绍了相关工作并对本文需要解决的问题进行了定义；第3节详细介绍了本文提出的时间加权不确定近邻协同过滤算法；第4节设计多组实验，对本文提出的算法进行验证，并进行简单的分析；最后对全文进行小结。

2 相关工作及问题定义

2.1 相关工作

当前推荐系统中普遍存在数据稀疏[7]、冷启动[8]和可扩展[9] 3大问题。针对这些问题，众多的研究人员做了大量的研究，希望通过设计新的算法来解决这些问题。Park等将协同过滤算法和搜索引擎工具相结合，创建一个新的搜索原型MAD6，并将其应用于Yahoo[4]。他们基于用户基本资料，在用户开始搜索前先使用协同过滤算法生成一个针对特定用户的搜索池以减小搜索的规模，同时还利用此原型对电影搜索的排序结果进行优化，在一定程度上提高了搜索引擎结果排序的效率，并且算法同样适用于音乐搜索、旅行资料搜索和电子商务搜索系统。但是该搜索原型仅仅使用协同过滤算法在线下产生一个搜索池来减少搜索范围，并未在核心业务中使用协同过滤算法改进系统性能。Tomoharu等在传统协同过滤算法的基础上应用最大熵原理对用户感兴趣的商品做出预测[5]，同时利用日志分析技术对推荐系统的推荐效果进行评价，最后将其成功地应用于电子商务系统。文献[5]的方法清晰地展现推荐系统对于不同人群的效果，系统维护人员可以根据这一结果调整推荐系统的参数，尽最大努力满足不同用户的需求，在一定程度上实现了个性化的推荐。但是系统并没有根据评价指标进行自适应的调整，后期的研究中可以在此基础上结合机器学习相关领域知识对系统进行更加深入的研究。Chen[2]等使用双协同过滤的方法，在推荐的结果集中再次使用协同过滤算法进行二次推荐，寻找目标用户可能感兴趣的商品，同时算法中加入了收益因子，在对顾客进行推荐的过程中充分照顾到商家的利益，具有一定的实用性，但是算法在推荐的过程中仍然只考虑到用户群和产品群中的一个群体。顾亦然等提出了一种基于时间加权的推荐算法，算法在推荐过程中适当考虑了时间属性[11]，但是采用添加时间维度的方式将推荐过程由二部图迁移到三部图上进行处理，这在一定程度上增加了算法的复杂度，并且文中并没有指明对于时间属性缺失的数据的处理方法。黄创光等提出一种综合考虑用户和产品的不同群体的不确定近邻推荐算法，实现了用户群和产品群的平衡[12]，但是算法中需要用户来主动设置产品群和用户群的阈值，阈值的设置对推荐结果影响较大，在先验知识相对匮乏的情况下，算法的准确度将受制于用户。陈健等通过建立k最近邻及其影响集的方式提高资源评价密度，并定义一种新的推荐机制来计算预测的评分[13]，有效缓解了数据稀疏性的问题，提高了推荐的质量，但是该算法仅仅基于项目一个群体进行推荐，并没有考虑其他对推荐结果有影响的群体。Liu等在信任子群的基础上采用Beta分布对用户的相似度进行预测，在推荐过程中对比实际相似度和预测相似度，以过滤掉部分噪音数据，这在一定程度上改善了推荐的结果[14]。Jamali为了改进推荐的质量，在用户之间的信任关系上进行了深层次的挖掘，通过数据挖掘的相关方法寻找深层次的用户相似性进行推荐[15]，以提高系统的推荐效率，但是信任模型和时间相关性相对较大，同时在推荐的过程中并没有考虑数据的时间属性。

本文首先使用logistic函数对评分进行时间加权，从时间方面对评分进行区分，改进了相似度的计算方式。同时引入了近邻因子，在产品群和用户群两个群体中动态选择推荐对象的近邻，避免了因为邻居选择不合理导致推荐结果不准确的问题。算法具有下列优点：

1. 充分考虑数据的时间有效性，避免了无效数据的干扰；

2. 充分考虑产品和用户两个群体的影响，避免了照顾一个群体而忽视另一个群体的弊端；同时，考虑了推荐对象实际邻居数量的不确定性，避免了设置固定的近邻数量导致推荐结果精确度不高的问题。

2.2 问题定义

传统的协同过滤算法使用用户群或者产品群内部的个体之间的相互影响来查找对当前个体影响最大的k个邻居从而对当前个体的属性做出预测。但是，推荐系统的应用范围越来越广，面对的环境越来越复杂多变，这种截取k个邻居的方法往往是具有片面性的。这种片面性主要表现在两个方面：1)单独考虑某一个群体的影响而忽视另一个群体的影响本身就是不合理的；2)包含该个体的群体数据稀疏时，单纯基于用户或物品的推荐可能会出现该类簇中的邻居个数小于基本推荐算法kNN中的k值，这种情况下不得不将相似性低的用户或物品加入到训练集中，这会导致算法的精确度急剧下降，这种情况下得到的结果往往是不准确的，甚至是错误的。

例如要预测某一个用户对某一件商品的喜爱程度，目前常用的方法就是UBCF和IBCF。虽然这些方法在一定程度

上已经能够进行推荐，但是由于现实条件的制约使得推荐的准确性还不尽人意。例如当某一个用户对某件商品感兴趣，而很少有其他用户对该商品感兴趣也没有做出评价时，采用基于用户的协同过滤算法进行推荐的准确性相对较低。由于现实情况的复杂性导致我们在进行推荐时必须考虑用户和产品的各个环节，在用户群和产品群中进行自适应的选择。可以根据现实情况的需求动态选择近邻因子和最近邻居，在用户群和产品群中合理选择邻居作为当前对象进行推荐，而不是单纯考虑某一方面的影响而忽视另一方面的影响。

在用户或物品类簇中用户或物品个数满足 kNN 算法中 k 值的最低要求时，也存在时间不一致的情况。由于用户的兴趣会随着时间的变化也会有一些变化，每个用户在不同时间对待同一个项目的评分可能会有所不同。而传统的算法在寻找用户的最近邻居时将用户的所有评分都是同等对待的，没有考虑到用户的兴趣随着时间会发生一定的变化，导致计算得出的邻居不一定是用户真正感兴趣的邻居的集合。而基于 kNN 的协同过滤算法准确性的关键就取决于选择的邻居与目标用户的匹配程度，这也是传统算法在准确性方面有待提升的一个重要原因。例如，某用户 A 在过去的某一段时间对动作电影产生兴趣，对此类电影的评分很高，另一个用户 B 当前对动作电影也产生了浓厚的兴趣，对同类电影给予了很高的评分。从相似性的计算方法来看，这两个用户应该是对方的邻居。但是，在实际中利用不同时间两个用户的兴趣进行推荐显然是不合理的，首先用户 A 当前的兴趣与用户 B 不一定相同；其次 A 当时喜欢的电影由于社会流行度的影响，在当前时间不一定受用户 B 的喜爱。通常，应该在同一段时间或者相近的时间内寻找不同用户对相同物品的评分之间的相似性，这样就可以保证寻找到的邻居是有时效的。我们通过表 1 给出的示例来说明这一问题。

表 1　模板图像的性质示例

用户	项目					
	I_1	I_2	I_3	I_4	I_5	I_6
$u_1(T_1)$	4	3	4	3	3	4
$u_2(T_4)$	3	4	2	3	4	2
$u_3(T_1)$	3	4	4	3	3	4
$u_4(T_2)$	4	3	2	4	3	2

表 1 给出了 4 个用户在 3 个时间段内对于 6 个项目的评分。其中 T_1 和 T_2 是相近时间段，T_1 和 T_4 是相隔较远时间段。

根据上述假设，在使用传统的协同过滤算法对只有 3 个邻居用户进行推荐时，u_2、u_3 和 u_4 构成了 u_1 的最近邻居集，并且相似性满足条件 $\mathrm{sim}(u_1,u_2)>\mathrm{sim}(u_1,u_3)>\mathrm{sim}(u_1,u_4)$。但是，从表中给出的评分数据来看，用户 u_1 和 u_2 对相同项目评分的时间相差较大，根据前文的分析，使用用户 u_1 在过去某个时间段内的评分信息对用户 u_2 的兴趣进行预测并对其进行推荐是不合理的。如果加入时间属性，那么用户 u_1 的邻居只有 u_3 和 u_4。这样的结果将更加符合现实情况。

3　时间加权不确定近邻协同过滤算法

3.1　时间加权的最近邻居

如果在选择用户或物品的 k 个邻居时不对用户或物品评分数据的时间段加以限制，则容易出现把过期数据当作邻居来考虑的情况（例如根据该用户 20 年前的兴趣进行新产品推荐），结果往往不太理想。在传统的协同过滤算法中，并没有对数据的时间有效性进行区分，在一定程度上影响了结果的准确性。综合考虑时间属性对目标群体的影响，采用时间加权的方法对相似性进行修正，可以有效避免由于时间不一致导致的推荐结果准确性降低的问题。

因此本文提出了一种时间加权的最近邻居选择方法（Time Weighted Selected Neighbor），即在进行邻居选择之前先将用户和物品的评分分别应用 logistic 函数进行时间加权修正。本文使用 Pearson 相关性来计算相似性，首先使用 logistic 函数对不同用户以及不同时间段的评分进行加权处理，用 $r_{u,c}\times logistic(r_{u,c})$ 代替 $r_{u,c}$，使得每个评分都获得一个关于时间的权重。由于距离当前时间最近的评分最能代表用户的兴趣，因此，当前权值应该大于过去一段时间评分的权值，对不同时间的评分进行区分。

logistic 函数为：

$$logistic(t_{u,j})=\frac{1}{1+e^{-t_{u,j}}} \tag{1}$$

其中，$-1\leqslant t\leqslant 1$，$0<logistic(t_{u,j})<1$。$logistic(t_{u,j})$ 是单调递增函数，加权值随着时间 t 的增加而增加，并始终保持在 (0,1) 的范围内。本文首先使用标准化转换方法将时间 t 的变化范围映射到 $[-1,1]$。这样就使得通过 logistic 的加权值随着时间的变化几乎呈线性趋势[16]，从而很直观地检测用户兴趣的转移。

协同过滤算法最重要的步骤是目标用户邻居的形成，本文采用 Pearson 相关性计算用户的最近邻居。基于时间加权的 Pearson 相似性可表示为式(2)。

$$\mathrm{sim}(U_a,U_b)=\frac{\sum\limits_{c\in I_U}(r_{U_{a,c}}\times logistic(t_{U_{a,c}})-\overline{r_{U_a}})(r_{U_{b,c}}\times logistic(t_{U_{b,c}})-\overline{r_{U_b}})}{\sqrt{\sum\limits_{c\in I_U}(r_{U_{a,c}}\times logistic(t_{U_{a,c}}))^2}\sqrt{\sum\limits_{c\in I_{Ui}}(r_{U_{b,c}}\times logistic(t_{U_{b,c}}))^2}} \tag{2}$$

式中，$\mathrm{sim}(U_a,U_b)$ 是目标用户 U_a 与其最近邻居 U_b 的相似性，I_U 是指用户 U_a 和用户 U_b 共同评分的项目集合，用户 U_a 对项目 c 的评分为 $r(U_{ac})$，用户 U_a 和用户 U_b 对项目的平均评分则分别为 $\overline{r(U_a)}$ 和 $\overline{r(U_b)}$。假设目标用户为 U_a，那么我们可以根据 U_b 对于任意项 $j(j\notin I_U)$ 的评分来预测 U_a 的评分，计算方法为：

$$P_{u,j}=\overline{r_{U_b}}+\frac{\sum\limits_{i=1}^{n}[\mathrm{sim}(U_a,U_b)(r_{U_{a,c}}-\overline{r_{U_a}})]}{\sum\limits_{i=1}^{n}[\mathrm{sim}(U_a,U_b)]} \tag{3}$$

我们通过时间加权的 Pearson 相关函数来计算相似性，并且针对用户和项目两个方面同时计算相似性，并预测用户对其他项目的评分值，然后取不属于 I_U 的项目按降序进行排序作为推荐集，最终从推荐集中选出合适的项目进行推荐。

3.2　改进相似性计算方式

用户之间的相似性是根据不同用户对相同产品或者项目的评分值来进行衡量的，如果用户之间共同评分项产品或项目太少，无法满足最少邻居数目，则容易导致推荐结果准确性下降。为了克服这种偶然性带来的影响，研究者已经做了一定的研究。Herlocker 等通过增加关联权重的方法来改善相

似性的计算。Ma 等在文献[17]中指出了权重的具体设置方法。本文则通过设置阈值的方法进行控制，设 $I'=U_aU_b$ 表示用户 U_a 和 U_b 共同评分的产品，我们通过加入不同用户在同一时间或相似时间内共同评分的比例来改善用户之间的相似性。

$$\mathrm{sim}'(U_a,U_b)=\frac{\min(|I'|,\gamma)}{\gamma}\times\mathrm{sim}(U_a,U_b) \tag{4}$$

其中，γ 为设置的阈值，根据定义可知阈值的最大值为用户共同评分的项目的数量，因此有 $\frac{\min(|I'|,\gamma)}{\gamma}\leqslant 1$，据此，改善之后的用户相似度 $\mathrm{sim}'(U_a,U_b)$ 的值域仍然落在区间[0,1]上。当用户共同评分项目较多，超过设置的阈值时，就有 $\mathrm{sim}'(U_a,U_b)=\mathrm{sim}(U_a,U_b)$，同理，如果用户之间共同评分项目很少，要改善这种状况对推荐准确度的影响，就应该降低用户之间的相似性。

3.3 不确定近邻的信任子群

目前推荐系统中常用的算法就是基于用户或者产品的协同过滤算法，但是由于用户对推荐质量的要求和现实状况的多变性导致仅能使用某一种方法，尤其是在数据稀疏的情况下，因此推荐的结果往往都达不到用户的要求。如何综合考虑多个因素并且自动对不同的因素进行平衡是我们需要考虑的一个问题。针对推荐系统根据最近邻居产生推荐的特点，本文对推荐集的选取进行优化，并采用自适应的方式在用户和产品之间进行权衡，避免了过多的人为参与致使推荐系统灵活性降低的缺点。

文献[12]提出设置两个相似度阈值 μ 和 ν 分别对用户和产品的相似度进行平衡的方法，在进行近邻对象选择之前首先采用动态选择的方法在用户群和产品群中动态选择适合推荐目标的近邻(Dynamically Selected Neighbor，DSN)。这种方法虽然可以在一定程度上克服传统 kNN 方法的缺点，进行近邻的动态选择，但是算法中需要设置两个阈值进行平衡。由于阈值的设置对近邻的选择存在一定的影响，而且计算相对复杂，阈值设置过大容易导致推荐集的数量变少，推荐结果有失一般性，设置过小，容易带入相似性很低的邻居，干扰了推荐的准确性。因此，在缺乏先验知识的情况下，这种方法依然不能很好地选取推荐对象的近邻。为了克服文献[12]的这一缺点，本文引入一个调和参数对基于产品和用户的方法进行平衡。

在此，我们记用户 U_a 对产品 I_j 的预测评分的集合为 $S'(U_a)=\{U_{a_1},U_{a_2},\cdots,U_{a_{m'}}\}$，并且记集合中元素的个数为 $|S'(U_a)|=m'$。产品 I_j 的近邻产品中用户 U_a 评价过的产品集记为 $S(I_j)=\{I_{j_1},I_{j_2},\cdots,I_{j_{n'}}\}$，并且 $|S(I_j)|=n'$。m' 和 n' 的计算相对容易，并且可以在线下完成。本文引入近邻因子 λ 和 $1-\lambda$ 分别作为用户群和产品群的平衡因子，根据调和参数调节近邻因子的值。

$$\lambda=\begin{cases}\dfrac{\phi\times m'}{\phi\times m'+n'}, & m'+n'>0\\ \dfrac{\phi\times m}{\phi\times m+n}, & m+n>0\\ 0.5, & \text{其他}\end{cases}$$

$$1-\lambda=\begin{cases}\dfrac{\phi\times n'}{\phi\times m'+n'}, & m'+n'>0\\ \dfrac{\phi\times n}{\phi\times m+n}, & m+n>0\\ 0.5, & \text{其他}\end{cases}$$

其中，ϕ 是本文定义的调和参数。下面我们以 λ 为例，分析调和参数 ϕ 对于近邻因子的调节方法。当 $m'+n'>0$ 时，λ 的取值存在以下 4 种可能：

(1) 若 $m'=0$ 同时 $n'>0$，表示采用基于产品的协同过滤方法进行推荐，此时 $\lambda=0$，与调和参数 ϕ 的取值无关。

(2)当 $\phi=\frac{n'}{m'}$ 时，$\lambda=1-\lambda=0.5$，这种情况表示推荐集来自基于产品和基于用户推荐的结果集中各一半。

(3) 当 $\phi\in(\frac{n'}{m'},\infty)$ 时，λ 为递增函数，值域为(0.5,1)。$1-\lambda$ 为递减函数，值域为(0,0.5)。表示随着调和参数 ϕ 的递增，推荐的结果集中的项目更多来自用户群。一种极端的情况是 $n'=0$，此时 $\lambda=1$，表示基于用户的协同过滤方法。

(4)当 $\phi\in[0,\frac{n'}{m'})$ 时，λ 为递减函数，值域为(0,0.5)，表示推荐结果集逐渐倾向于产品群。当 $\lambda=0$，表示采用基于产品的协同过滤方法进行推荐。

本文为了平衡用户群和产品群在不同维度上的影响，引入调和参数 ϕ 来动态调整推荐结果集的来源，避免了单群体推荐造成推荐质量不高的缺点。本文改进的方法通过减少用户输入阈值的数量与难度，降低了用户对算法的干预。本文所使用的阈值相比文献[12]更符合用户使用习惯，降低了用户操作的复杂性。文献[12]中 μ 和 ν 对于邻居的选择是敏感的，从而对推荐结果产生的影响较大；而本文则通过引入调和参数的方法对阈值进行控制，降低了邻居选择的敏感度，保证了推荐结果的准确性。另一方面，由于本文阈值设置相对简单，因此可以通过多次试验的方式来获取最佳阈值，以保证算法准确性，降低人工干预。

3.4 时间加权不确定近邻协同过滤算法

根据上述分析，本文充分考虑评分值的时间属性，在此基础上对基于用户和产品的评分进行平衡。综合多方面的考虑，提出时间加权不确定近邻协同过滤算法(Time Weighted Uncertain Nearest Neighbor Collaborative Filtering，TWUNCF)来对用户的评分进行预测。如果用户 U_a，U_x 对产品所有评分的均值分别表示为 $\overline{R_a}$，$\overline{R_x}$，已知用户对于产品 I_j，I_y 评分的平均值分别表示为 $\overline{R_j}$，$\overline{R_y}$，那么用户 U_a 对产品 I_j 的评分 $R_{a,j}$ 可以表示为式(5)，式中近邻因子 λ 和 $1-\lambda$ 分别作用于用户群对产品评分的均值和产品群得到的所有评分的均值，考虑了两个群体的共同作用。同时，在计算相似性的过程中考虑了评分矩阵中元素的时间有效性。

$$R_{a,j}=\lambda\left(\overline{R_a}+\frac{\sum_{U_x\in S(U_a)}\mathrm{sim}'(U_a,U_x)\times(R_{x,j},\overline{R_x})}{\sum_{U_x\in S(U_a)}\mathrm{sim}'(U_a,U_x)}\right)+$$
$$(1-\lambda)\left(\overline{R_j}+\frac{\sum_{I_y\in S(I_j)}\mathrm{sim}'(I_j,I_y)\times(R_{a,y},\overline{R_y})}{\sum_{I_y\in S(I_j)}\mathrm{sim}'(I_j,I_y)}\right) \tag{5}$$

算法 TWUNCF 是根据不确定的场景，首先通过 logistic 函数引入时间变量对用户群和产品群中的相似度进行时间加权以区分相似度的时间有效性。在此基础上对用户和产品的相似度进行综合考虑，通过调和函数间接控制不同群体的近邻因子，进而综合考虑用户和产品的影响，最终产生推荐的结果集。

因此，算法应该分为两个步骤，第一步是选取推荐目标的

信任子群，第二步是根据选择的推荐集进行推荐，得到最终的推荐结果。算法的描述如图 1 所示。

算法：TWUNCF 算法

输入：目标用户 U_a，待评分产品 I_j，调和参数 ϕ

输出：预测用户 U_a 对产品 I_j 的评分 $R_{a,j}$

Step 1　针对评分矩阵 R(s×t)分别计算用户的相似度矩阵和产品的相似度矩阵，并分别保存在矩阵 Arr_usrSim(s,s)和 Arr_ItemSim(t,t)中；

Step 2　应用 logistic 函数对时间进行加权；

Step 3　判断时间有效性；

Step 4　根据用户和产品评分值计算$|S'(U_a)|=m'$和$|S'(I_j)|=n'$，并计算 $S'(I_j)$的信任子群；

Step 5　选择合适的调和参数 ϕ；

Step 6　计算 λ 的值；

Step 7　计算用户 U_a 对产品 I_j 的预计评分 $R_{a,j}$。

图 1　TWUNCF 算法

在 TWUNCF 算法中，首先是要确定用户和产品的评分矩阵，最坏情况下的时间复杂度为 $O(s^2+t^2)$，这一步骤可以在离线状态下进行计算，以降低算法的时间复杂度。由于$|S'(U_a)|=m'$和$|S'(I_j)|=n'$均为常数，因此，计算近邻和信任子群的时间复杂度为 $O(m+n+m'+n')=O(1)$。于是算法时间复杂度最好的情况为 O(1)，可以有效避免因为数据稀疏和数据积累造成的计算困难的问题。并且，对比文献[12]中提出的 DSN 算法，本文提出的算法参数更少，减少了人工操作对寻找邻居的影响，算法更加简洁易懂。

4　模拟实验及其分析

下面我们通过模拟实验的方法来验证本文所提出的算法的效率与准确性。探索性地考察本文提出的算法在不同数据规模上相比 *k*NN 方法的适应情况，以验证本文提出的动态选择近邻思想的正确性；同时尝试分析本文提出的调和参数的设置是否可以在信任子群中产生更好的推荐结果。这是本节实验需要验证的两方面的问题。

与其他推荐算法测试方法类似，本文实验中使用由 Grouplens 提供的 MovieLens 数据集。数据集包含了 10 万条记录，共有 943 名用户对 1682 部电影进行 5 个等级的评分，评分数值为 1—5。其中 1 表示“poor”，5 表示“perfect”，其他数据则表示中间值，他们代表了用户对电影兴趣的不同程度。实验的硬件环境为 Intel(R) Core(TM) i5-25200 四核 64 位 2.5GHz 的 CPU 和 4GB 的内存，软件环境为 Windows 7-64bit(professional) 操作系统，所有代码均用 Java(64bit JDK)和 Matlab2012 实现。

由于用户和产品的评分矩阵密度为$\frac{100000}{943\times1682}=1.63\%$，因此该评分矩阵为稀疏矩阵。我们在数据集的 943 名用户中随机抽取 3 个用户组进行对比试验(3 个用户组分别有 100、200 和 300 个用户)，并在整体数据集中选取 70%的样本数据作为训练集，30%的数据作为测试集，进行对比实验。

4.1　动态选择法的比较

本文的实验需要分别验证选取信任子群的方法和推荐的方法。首先我们将选择信任子群的 DSN 方法与 *k*NN 方法进行对比，测试 DSN 方法是否能够顺利选取相对较好的近邻对象，以为后面的推荐做好准备。我们以 k 值作为横坐标，对比不同近邻下两种方法的性能，其变化范围为 1,2,4,8,10,…,60，以平均绝对偏差 MAE(Mean Absolute Error)作为衡量标准。实验结果分别如图 2—图 4 所示。

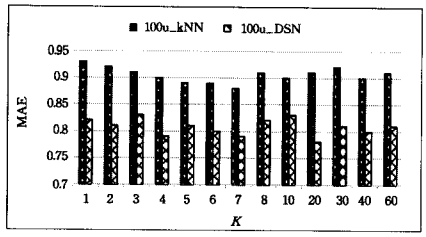

图 2　*k*NN 方法与动态选择推荐对象的比较(100 个用户)

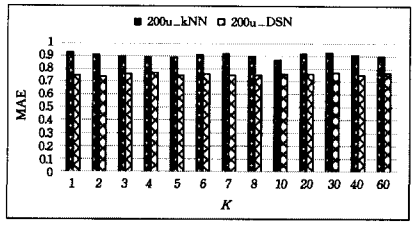

图 3　*k*NN 方法与动态选择推荐对象的比较(200 个用户)

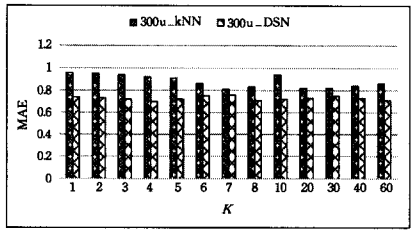

图 4　*k*NN 方法与动态选择推荐对象的比较(300 个用户)

首先对实验结果进行横向的对比，可以看出，在用户数量为 100 时，*k*NN 方法在 $k=7$ 时取得最好的效果，并且当 k 取其他值时，在相同条件下 DSN 方法依然能够取得较 *k*NN 方法更好的性能。对比用户数量在 200 和 300 的条件下的状况，在同样的条件下 DSN 方法都具有较 *k*NN 方法更好的表现。

纵向对比各组的实验结果可以很明显地看出训练集越大对于寻找目标用户的信任子群和进行推荐越有利。通过分析对比可以发现，DSN 方法比同条件下的 *k*NN 方法具有更好的稳定性和准确性。实验说明 DSN 方法对于改善信任子群的选择是有积极意义的。因此在后面的实验中，在使用 UBCF 和 IBCF 进行推荐时，我们均采用 DSN 方法选择推荐对象的信任子群。

4.2　传统推荐算法与本文 TWUNCF 算法的对比实验

该实验在相同条件下对比传统的协同过滤算法 UBCF 和 IBCF 与本文提出的 TWUNCF 算法的性能，实验中的横坐标表示所预测目标产品的近邻数目，纵坐标采用 MAE 作为度量标准。

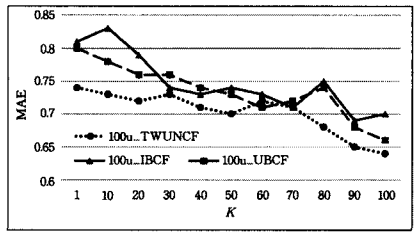

图 5　IBCF、UBCF 与 TWUNCF 算法比较(100 个用户)

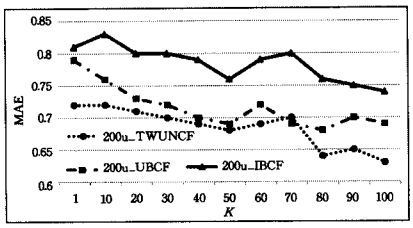

图 6　IBCF、UBCF 与 TWUNCF 算法比较(200 个用户)

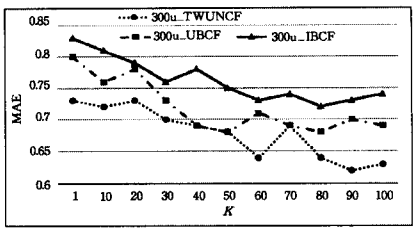

图 7　IBCF、UBCF 与 TWUNCF 算法比较(300 个用户)

对比分析以上 3 组实验,横向对比每一个实验可以发现,在同样的条件下 TWUNCF 方法都能获得比 IBCF 和 UBCF 更小的 MAE 值,具有相对较好的推荐效果。纵向对比 3 组实验,可以很明显地发现推荐的质量随着信任子群的数目的增大而不断提升。综合分析以上各组实验,可以得出结论,在同样的条件下 TWUNCF 都能获得比 UBCF 和 IBCF 更好的性能。

结束语　针对传统协同过滤算法中基于产品和基于用户的偏见以及时间数据有效性的特点,本文提出一种时间加权不确定近邻协同过滤算法,算法利用 logistic 时间函数有效解决了数据时间有效性的问题。在此基础上还提出不确定近邻的思想,在基于用户和基于产品的两种推荐方式中动态选择可信邻居,避免了单纯使用基于产品的推荐和基于用户的推荐导致的不同场合下推荐精度不确定的缺陷。实验和理论分析都证明本文提出的 TWUNCF 算法比 TBCF、IBCF 等传统算法具有更好的准确性和稳定性。

参 考 文 献

[1] Jelenc David. Decision making matters: A better way to evaluate trust models [J]. Knowledge-Based Systems, 2013, 52: 147-164

[2] Chen M-C, Chen L-S, Hsu F-H, et al. HPRS: A profitability based recommender system [C]// Helander M, Xie Min, Jiao R, et al, eds. Proceeding of the IEEE International Conference on Industrial Engineering and Engineering Management. Singapore, IEEE Engineering Management Society Singapore Chapter, 2007(12): 219-223

[3] Xu Xiao-wei, Wang Fu-dong. Trust -Based Collaborative Fil-tering Algorithm [C]// Proceedings of the 2012 Fifth Interna-tional Symposium on Computational Intelligence and Design. Washington DC, USA, 2012: 321-324

[4] Park S-T, Pennock D M. Applying collaborative filtering techniques to movie search for better ranking and browsing [C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007. San Jose, California, USA, August 2007: 834-843

[5] Tomoharu I, Kazumi S, Takeshi Y. Modeling user behavior in recommender system based on maximum entropy [C]// Proceedings of the 16th International Conference on World Wide Web. Canada: World Scientific Publishing Co. Pte. Ltd., 2007: 1281-1282

[6] Pappageorge, Charlotte. Recommended verification approach for human rated systems based on lessons learned on the orion launch abort system [C]// 21st Annual International Symposium of the International Council on Systems Engineering. 2011, 1: 1-13

[7] Kaklauskas A. Recommended biometric stress management system [J]. Expert Systems with Applications, 2011, 38(11): 14011-14025

[8] Massa P, Avesani P. Trust-aware collaborative filtering for recommender systems [J]. Lecture Notes in Computer Science, 2004, 3290: 492-508

[9] Vincent S-Z, Boi Faltings. Using hierarchical clustering for learning the ontologies used in recommendation systems [C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose, California, United States, 2007: 599-608

[10] Yang J-M, Li K-F. An interference-based collaborative fil-tering approach [C]// Proceedings of the 3rd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC). Columbia, MD, United States, 2007: 84-94

[11] 顾亦然,陈敏. 一种三部图网络中标签时间加权的推荐方法[J]. 计算机科学,2012,39(8):96-98,129

[12] 黄创光,印鉴,汪静,等. 不确定近邻的协同过滤推荐算法[J]. 计算机学报,2010,33(8):1369-1377

[13] 陈健,印鉴. 基于影响集的协作过滤推荐算法[J]. 软件学报,2007,18(7):1685-1694

[14] Liu X, Datta A, Rzadca K, et al. Stereo trust: A Group based personalized trust model [C]// Proceedings of the 18th ACM Conference on Information and Knowledge Management. Hong Kong, China, 2009: 7-16

[15] Jamali M, Ester M. Trust Walker: A random walk model for combining trust-based and item-based recommendation [C]// Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France, 2009: 397-406

[16] 楼俊钢,申情,沈张果. 软件可靠性预测的核函数方法[J]. 计算机科学,2012,39(4):145-148

[17] Ma H, King I, Lyu M R. Effective missing data prediction for collaborative filtering[C]// Proceedings of the 30th Annual International ACM SIGIR Conference. Amsterdam, The Netherlands, 2007: 39-46