ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.18, No.7, July 2007, pp.1685−1694
DOI: 10.1360/jos181685

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 基于影响集的协作过滤推荐算法*

陈 健[1+], 印 鉴[2]

[1](华南理工大学 计算机科学与工程学院,广东 广州 510006)

[2](中山大学 计算科学系,广东 广州 510275)

## A Collaborative Filtering Recommendation Algorithm Based on Influence Sets

CHEN Jian[1+], YIN Jian[2]

[1](School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China)

[2](Department of Computer Science, Sun Yat-Set University, Guangzhou 510275, China)

+ Corresponding author: Phn: +86-20-33509119, Fax: +86-20-39380218, E-mail: ellachen@scut.edu.cn, http://www.scut.edu.cn

**Chen J, Yin J. A collaborative filtering recommendation algorithm based on influence sets. *Journal of Software*, 2007,18(7):1685−1694.** http://www.jos.org.cn/1000-9825/18/1685.htm

**Abstract**: The traditional user-based collaborative filtering (CF) algorithms often suffer from two important problems: Scalability and sparsity because of its memory-based *k* nearest neighbor query algorithm. Item-Based CF algorithms have been designed to deal with the scalability problems associated with user-based CF approaches without sacrificing recommendation or prediction accuracy. However, item-based CF algorithms still suffer from the data sparsity problems. This paper presents a CF recommendation algorithm, named CFBIS (collaborative filtering based on influence sets), which is based on the concept of influence set and is a hot topic in information retrieval system. Moreover, it defines a new prediction computation method for this new recommendation mechanism. Experimental results show that the algorithm can achieve better prediction accuracy than traditional item-based CF algorithms. Furthermore, the algorithm can alleviate the dataset sparsity problem.

**Key words**: E-commerce; recommendation system; collaborative filtering; influence set

摘 要: 传统的基于用户的协作过滤推荐系统由于使用了基于内存的最近邻查询算法,因此表现出可扩展性差、缺乏稳定性的缺点.针对可扩展性的问题,提出的基于项目的协作过滤算法,仍然不能解决数据稀疏带来的推荐质量下降的问题(稳定性差).从影响集的概念中得到启发,提出一种新的基于项目的协作过滤推荐算法CFBIS(collaborative filtering based on influence sets),利用当前对象的影响集来提高该资源的评价密度,并为这种新

* Supported by the National Natural Science Foundation of China under Grant Nos.60573097, 60673062 (国家自然科学基金); the Research Foundation of National Science and Technology Plan Project of China under Grant No.2004BA721A02 (国家科技计划项目); the Research Foundation of Disciplines Leading to Doctorate Degree of Chinese Universities under Grant No.20050558017 (高等学校博士学科点专项科研基金); the Natural Science Foundation of Guangdong Province of China under Grant Nos.05200302, 04300462 (广东省自然科学基金); the Research Foundation of Science and Technology Plan Project in Guangdong Province of China under Grant No.2005B10101032 (广东省科技计划项目); the Natural Science Foundation of South China University of Technology under Grant No.B07E5060250 (华南理工大学自然科学基金)

Received 2006-03-20; Accepted 2006-07-05

的推荐机制定义了计算预测评分的方法.实验结果表明,该算法相对于传统的只基于最近邻产生推荐的项目协作过滤算法而言,可有效缓解由数据集稀疏带来的问题,显著提高推荐系统的推荐质量.

关键词: 电子商务;推荐系统;协作过滤;影响集
中图法分类号: TP393 文献标识码: A

Web 个性化通过收集和分析用户信息来学习用户的兴趣和行为,根据用户的兴趣模式将 Web 中一组匹配的对象推荐给当前用户,包括链接、广告、文本、产品或者服务等.Web 个性化服务技术能够充分提高站点的服务质量和访问效率,而高质量的推荐技术对于 Web 个性化则是十分重要的.近来,Web 使用挖掘被提出作为 Web 个性化中的一个主要方法而受到广泛的关注.Web 使用挖掘的目标是捕捉集成在 Web 站点中的用户行为模式和 Web 对象之间的关系.这些模式通常表达为一组页面或者项目的集合,有共同需要或兴趣的用户经常同时访问它们,或是给出相似的评价.这些模式可以用来更好地理解访问者和用户组的行为特征,提高站点的组织结构,以及为访问者提供动态的推荐服务.

目前存在许多不同解决方案实现的推荐系统(recommendation system),研究人员提出了各种产生推荐的思路以及具体的实现方法.根据所采用的技术不同,分别有基于手工决策规则的推荐系统[1]、基于数据挖掘技术的推荐系统[2,3]、基于内容过滤的推荐系统[4]和基于协作过滤的推荐系统等等[5].其中最成功的,也是在实际中得到最广泛应用的是基于协作过滤(collaborative filtering)的推荐思想.

## 1 相关工作

传统意义上的基于协作过滤的推荐系统,通过分析不同用户在相同项目上的评分或浏览网站过程中所体现的偏好,为当前在线用户寻找 $k$ 个最相似的邻居.针对某个特定的、当前用户尚未评分或浏览的项目,根据邻居们感兴趣的程度对其作出预测,或是直接将邻居们最感兴趣的 $N$ 个项目(top $N$ recommendation)推荐给当前用户.这种系统由于主要关注用户与用户兴趣之间的关联性,因而被称作是基于用户的协作过滤(user-based collaborative filtering).这种推荐系统由于提出了一些观点来改进基于规则或基于内容过滤的推荐系统的不足,因此在大多数的电子商务系统中得到了广泛的使用:PocketLens(一个改进的 GroupLens)系统[6]为 Usenet 新闻和电影提供协作过滤推荐的解决方案;TV Scout[7]是一个基于 Web 的个性化电视节目推荐系统;Memior[8]根据用户的兴趣为其寻找同伴;Push!Music[9]为使用无线设备的音乐爱好者共享音乐提供智能代理.基于经典的协作过滤推荐算法,研究人员还发展出很多以这种思想为中心的变形算法:如 Good 等人[10]提出的智能过滤器;Jin 等人[11]提出的基于内容的协作过滤;还有 Sarwar[12]在其上应用了降维技术.

尽管基于用户的协作过滤在电子商务系统中得到了成功的运用,但是,这种模型仍然存在潜在的局限性:

1) 首先是缺乏可扩展性[12],也就是基于内存的 $k$NN 算法需要在线执行,以寻找当前用户的 $k$ 个最近邻居来产生推荐.当数据规模逐渐增大时,就可能导致算法速度急剧下降,无法及时产生推荐.

2) 另一个缺点是由于数据集潜在的稀疏性造成的[13].一是系统使用初期,系统资源还未获得足够多的评价,此时的评分矩阵中获得评分的项目相当少;二是随着数据库中项目数量的增加,每个用户对这些项目相关的评分密度就会减少.由于用户与用户之间在已评分项目上的交集很小,因此由评分项目体现出来的特征相似性就会降低,从而导致系统产生不可靠的推荐结果,而且,此时必须花费很高的代价来维持用户相似性矩阵,计算量大为增加,系统的性能和准确性也都会越来越低.

3) 还有所谓的“新项目问题(new item problem)”[14],即对于新近加入系统中的项目,由于缺乏评价资源,系统不能将新近出现的项目推荐给用户.

针对基于用户的协作过滤中的可扩展性问题,研究人员提出了基于项目的协作过滤(item-based collaborative filtering)的思想[15].这种模型通过在线维护一张项目相似性列表,可在 $O(n)$时间内就为当前用户正在浏览的项目寻找到 $k$ 个最相似的其他项目并产生推荐,其中,$n$ 是数据库中项目的总个数.它比起基于用户的协作过滤而言有以下优点:a) 项目空间远远小于用户空间,因此计算复杂性降低;b) 项目与项目之间的相似性

不易发生变化,而用户的兴趣、爱好、上网目的却时时不同,因此,项目相似性计算可离线完成.

然而,基于项目的协作过滤算法仍然不能解决数据稀疏性所带来的问题,也不能将新近出现的项目推荐给用户[16].受影响集概念的启发,本文提出一种新的基于项目的协作过滤推荐算法 CFBIS(collaborative filtering based on influence sets),同时,结合当前对象的 $k$ 个最近邻和 $k'$ 个逆最近邻来为它产生推荐.CFBIS 利用当前对象的影响集来提高该资源的评价密度,并为这种新的推荐机制定义了计算预测值的方法.多个实验结果表明,该算法相对于传统的只基于最近邻产生推荐的项目协作过滤算法而言,可有效缓解数据集极度稀疏带来的问题,显著提高推荐系统的推荐质量.

## 2 问题陈述和基本定义

在推荐系统中,用户的事务数据库中包含 $m$ 个用户的集合 $U=\{u_1,u_2,...,u_m\}$和 $n$ 个项目的集合 $I=\{i_1,i_2,...,i_n\}$.用户事务集可用一个 $m\times n$ 阶矩阵 $A(m,n)$表示,见表 1.

**Table 1** $A(m\times n)$ user-item ratings matrix

表 **1** 用户-项目事务矩阵 $A(m,n)$

| | $i_1$ | … | $i_k$ | ... | $i_n$ |
|---|---|---|---|---|---|
| $u_1$ | $W_{1,1}$ | … | $W_{1,k}$ | … | / |
| … | … | … | … | … | … |
| $u_j$ | $W_{j,1}$ | … | / | … | $W_{j,n}$ |
| … | … | … | … | … | … |
| $u_m$ | / | … | $W_{m,k}$ | … | $W_{m,n}$ |

其中,矩阵共有 $m$ 行代表 $m$ 个用户,$n$ 列代表 $n$ 个项目.每一个用户 $u_j$ 都对应一个已评分项目集 $I_{u_j}\subseteq I$, $W_{j,k}\in I_{u_j}$ 代表用户 $u_j$ 对项目 $k$ 的赋予的权重,这个权值体现了用户 $u_j$ 对项目的兴趣和偏好.

### 2.1 项目的相似性度量方法

对于项目 $i_p$ 的权值向量 $W_p=\{W_{1,p},W_{2,p},...,W_{m,p}\}$和项目 $i_q$ 的权值向量 $W_q=\{W_{1,q},W_{2,q},...,W_{m,q}\}$,有以下几种度量 $i_p$ 和 $i_q$ 相似性 $sim(i_p,i_q)$的方法[17]:

1) 标准的余弦相似性.通过向量间的余弦夹角进行度量.

$$sim(i_p,i_q)=\cos(\vec{i_p},\vec{i_q})=\frac{\sum_{k=1}^{m}W_{k,p}\times W_{k,q}}{\sqrt{\sum_{k=1}^{m}(W_{k,p})^2\times\sum_{k=1}^{m}(W_{k,q})^2}},$$

其中,$W_{k,p}$ 是指用户 $u_k$ 对项目 $i_p$ 给出的权值,$W_{k,q}$ 是指用户 $u_k$ 对项目 $i_q$ 给出的权值.

2) 修正的余弦相似性.为了修正不同用户存在不同评分尺度的偏见,修正的余弦相似性度量方法通过减去用户对所有项目的平均评分来改善这一缺陷.我们选取矩阵 $A(m,n)$中对项目 $i_p$ 和 $i_q$ 都有评分的行,也就是对项目 $i_p$ 和 $i_q$ 都有评分的用户集合,将其定义为 $U'$.

$$sim(i_p,i_q)=\frac{\sum_{u_k\in U'}(W_{k,p}-\overline{W_k})(W_{k,q}-\overline{W_k})}{\sqrt{\sum_{u_k\in U'}(W_{k,p}-\overline{W_k})^2}\times\sqrt{\sum_{u_k\in U'}(W_{k,q}-\overline{W_k})^2}},$$

其中,$W_{k,p}$ 和 $W_{k,q}$ 如上定义,$\overline{W_k}$ 是指用户 $u_k$ 对所有项目权值的平均值.

3) 相关相似性.根据 Pearson 提出的相关系数来度量项目之间的相似性.为了计算的公正性,这个相似性度量也应在对两个项目都作出评价的用户集合 $U'$中进行计算.

$$sim(i_p,i_q)=\frac{\sum_{u_k\in U'}(W_{k,p}-\overline{W_p})(W_{k,q}-\overline{W_q})}{\sqrt{\sum_{u_k\in U'}(W_{k,p}-\overline{W_p})^2}\times\sqrt{\sum_{u_k\in U'}(W_{k,q}-\overline{W_q})^2}},$$

其中,$W_{k,p}$ 和 $W_{k,q}$ 如上定义,$\overline{W_p}$,$\overline{W_q}$ 分别是项目 $i_p$ 和 $i_q$ 上权值的平均值.

### 2.2 推荐的产生

对于某个在线目标用户 $u_a$,推荐系统的任务有两个:

(1) 对于用户 $u_a$ 还未浏览或评分的指定项目 $i_t \notin I_{u_a}$, 预测用户对它的评分 $P_{a,i_t}$ 并提供给用户;

(2) 将用户最有可能感兴趣的 $N$ 个项目直接推荐给用户,也就是找到一个大小为 $N$ 的项目集合:

$$I_r \subseteq I, I_r \cap I_{u_a} = \varnothing.$$

通过计算项目之间的相似性,基于项目的协作过滤推荐算法为指定项目 $i_t$ 寻找最相似的 $k$ 个邻居,并为其预测权值:

$$W_{a,t} = \frac{\sum_{j=1}^{k}(W_{a,j} \times sim(i_j, i_t))}{\sum_{j=1}^{k} sim(i_j, i_t)} \tag{1}$$

其中,$W_{a,t}$ 是推荐算法为用户 $u_a$ 在指定项目 $i_t$ 上的评分预测.从该公式可以看出,传统的 CF 算法中只有 $k$ 个最相似的邻居会被用于预测评分.

## 3 基于影响集的协作过滤推荐算法 CFBIS

传统的协作过滤推荐算法本质上是利用了群体内个体与个体之间的相互作用(寻找对当前对象影响力最大的 $k$ 个邻居)来为当前对象的属性作出预测的过程.但是,这个预测是单向的,只考虑了其他个体对当前对象的影响作用,而没有考虑当前对象在群体内也是具有一定影响力的.例如,具有摄像功能手机的出现,必然会使一部分用户对没有摄像功能的手机的评价降低.从信息检索领域的影响集概念中得到启发,本文认为,传统的协作过滤算法是不完整的,有必要同时为当前对象找到受其影响的群体($k'$个逆最近邻),结合两个群体的属性共同为当前对象的属性作出判定.

### 3.1 影响集的概念

$k$-最近邻及其检索算法是计算机科学的主要核心问题之一,尤其是在多维数据库系统的检索和查询方面起着相当重要的作用.近年来,$k$-最近邻的逆问题逐渐得到人们广泛的关注.所谓逆 $k$-最近邻,就是在给定的数据集 $S$ 中将查询点 $q$ 视为其 $k$-最近邻的所有点的集合,可以通过 $k$NN 算法的逆算法 R$k$NN(reverse $k$ nearest neighbor)算法来解决[16].

给定 $d$ 维数据集 $S$,$|S|=n$.$\forall p,q$,$D(p,q)$表示 $p,q$ 两点间的距离.在不同的应用中,“距离”的定义各有差异,例如在空间向量模型中,$D(p,q)$表示二者之间的 Euclidean 距离,而在本文中,$D(p,q)$表示项目 $p$ 和项目 $q$ 之间的相似度,那么

$$RkNN(q)=\{p \in S | q \in kNN(q)\}.$$

值得注意的是,$k$NN 和 R$k$NN 不是对称的,即 $p \in kNN(q) \not\Rightarrow p \in RkNN(q)$,反之亦然.如图 1 所示.

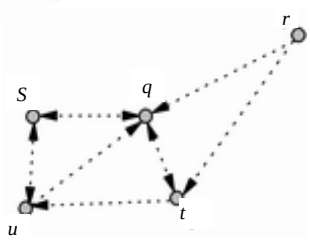


Fig.1 The relationships between $k$NN and R$k$NN

图 1 $k$NN 与 R$k$NN 关系示意图

根据定义,$2NN(q)=\{s,t\}$,$R2NN(q)=\{r,t,s,u\}$.R$k$NN 查询有可能返回 0 个或多个值(与 $k$ 无关).

### 3.2 寻找项目的影响集

由于用户评分数据的极端稀疏性,传统CF算法无法有效地为项目计算预测评分,从而导致协作过滤推荐系统的推荐质量难以保证.针对这个问题,本文提出一种新的基于项目的协作过滤推荐算法CFBIS,同时,结合当前对象的 $k$ 个最近邻和 $k'$个逆最近邻为它产生推荐.算法主要分为两步:首先为指定项目寻找 $k$ 最近邻和影响集,然后计算它的预测评分并产生推荐.

不失一般性,若项目 $i_a$ 以项目 $i_b$ 为 $k$-最近邻,则我们称 $i_a$ 与 $i_b$ 是相关的(related);若项目 $i_a$ 以项目 $i_b$ 为逆 $k$-最近邻,则我们认为 $i_b$ 与 $i_a$ 是相关的.如果二者互为 $k$-最近邻(例如 $i_1$ 和 $i_2$),那么我们认为二者互为关联,相互影响.从图2(其中,$i_a$ 指向 $i_b$ 的箭头表示 $i_b$ 是 $i_a$ 的一个最近邻)中可以看出,项目 $i_2$ 对群体中的其他项目是具有相当影响力的.

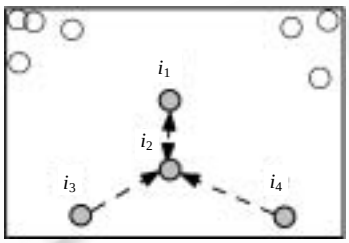


Fig.2 A simple sample of similarity relationships among items

图2 一个简单的项目间相似性关系示意图

CFBIS采用上节所述的相似性距离公式计算矩阵 $A(m,n)$表中列与列之间的相似性,为每一个项目找到 $k$ 个最相似的邻居,并保存在项目相似性列表中,见表2.从表2中可以看到,项目 $i_2$ 的最近邻为 $NN(i_2)=\{i_1\}$,而它的逆最近邻 $RNN(i_2)=\{i_1,i_3,i_4\}$.由此可见,项目的影响集确实可以提高项目的评价密度.而且,假设目标用户 $u_a$ 对某项目(例如 $i_2$)的最近邻都没有评分,公式(1)中的 $sim(i_j,i_t)$均为0,此时传统的只基于最近邻的协作过滤算法无法为这个项目产生预测评分,如表3和图3所示.

**Table 2** Item similarity table

**表2** 项目相似性列表

| | $k$=1 | $k$=2 | $k$=3 |
|---|---|---|---|
| $i_1$ | $i_2$ | $i_3$ | $i_4$ |
| $i_2$ | $i_1$ | $i_3$ | $i_4$ |
| $i_3$ | $i_2$ | $i_1$ | $i_4$ |
| $i_4$ | $i_2$ | $i_1$ | $i_3$ |

**Table 3** If $u_a$ has no ratings for all $k$ nearest neighbor of $i_2$, then traditional item-based CF approaches cannot produce prediction ratings for $i_2$

**表3** 当 $u_a$ 对 $i_2$ 的最近邻 $i_1$ 没有评分,传统的CF算法无法为 $i_2$ 产生评分

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ |
|---|---|---|---|---|
| $u_2$ | … | ? | 4 | 5 |

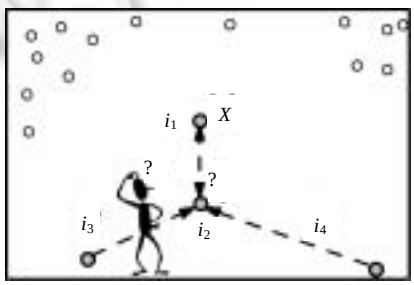


Fig.3 $i_2$ lacks necessary ratings for its neighbors

图3 $i_2$ 缺乏最近邻评分的情况

但是,如果结合了项目 $i_2$ 的影响集,考虑从它的逆最近邻获得信息,则系统可以从 $i_3$ 和 $i_4$ 中获得与 $i_2$ 相关的评分,从而提高项目的评价密度,获得足够的评价资源.

### 3.3 产生推荐

基于以上分析,对于目标在线用户 $u_a$ 和某个他还未浏览或评分的指定项目 $i_t \notin I_{u_a}$,基于影响集的协作过滤推荐算法 CFBIS,同时结合项目 $i_t$ 的 $k$ 个最近邻集合 $kNN(i_t)$与 $k'$个逆最近邻集合 $Rk'NN(i_t)$对用户 $u_a$ 在项目 $i_t$ 上的评分 $P_{a,i_t}$ 产生预测.我们为新的推荐机制定义了以下 4 个产生推荐的公式:

$$W_{a,t}=\frac{\sum_{i_j\in kNN(i_t)}(W_{a,j}\times sim(i_j,i_t))+\sum_{i_{j'}\in Rk'NN(i_t)}(W_{a,j'}\times sim(i_{j'},i_t))}{\sum_{i_j\in kNN(i_t)}sim(i_j,i_t)+\sum_{i_{j'}\in Rk'NN(i_t)}sim(i_{j'},i_t)} \tag{1}$$

其中,$W_{a,t}$是推荐算法为用户 $u_a$ 在指定项目 $i_t$ 上的评分预测,$sim(i_j,i_t)$表示项目 $i_j$ 与 $i_t$ 之间的相似性,$k$ 个最近邻集合 $kNN(i_t)$与 $k'$个逆最近邻集合 $Rk'NN(i_t)$被用于预测.

$$W_{a,t}=\overline{W_t}+\frac{\sum_{i_j\in kNN(i_t)}((W_{a,j}-\overline{W_j})\times sim(i_j,i_t))}{\sum_{i_j\in kNN(i_t)}sim(i_j,i_t)+\sum_{i_{j'}\in Rk'NN(i_t)}sim(i_{j'},i_t)}+\frac{\sum_{i_{j'}\in Rk'NN(i_t)}((W_{a,j'}-\overline{W_{j'}})\times sim(i_{j'},i_t))}{\sum_{i_j\in kNN(i_t)}sim(i_j,i_t)+\sum_{i_{j'}\in Rk'NN(i_t)}sim(i_{j'},i_t)} \tag{2}$$

其中,$\overline{W_t}$,$\overline{W_j}$ 和 $\overline{W_{j'}}$ 分别是项目 $i_t$,$i_j$ 和 $i_{j'}$权值的平均值.

$$W_{a,t}=\alpha\times\frac{\sum_{i_j\in kNN(i_t)}(W_{a,j}\times sim(i_j,i_t))}{\sum_{i_j\in kNN(i_t)}sim(i_j,i_t)+\sum_{i_{j'}\in Rk'NN(i_t)}sim(i_{j'},i_t)}+(1-\alpha)\times\frac{\sum_{i_{j'}\in Rk'NN(i_t)}(W_{a,j'}\times sim(i_{j'},i_t))}{\sum_{i_j\in kNN(i_t)}sim(i_j,i_t)+\sum_{i_{j'}\in Rk'NN(i_t)}sim(i_{j'},i_t)} \tag{3}$$

其中,$\alpha$是控制因子,用来控制 $k$ 个最近邻和 $k'$个逆最近邻在推荐预测计算中所起的作用.当 $\alpha$=1 时,推荐完全根据 $k$ 个最近邻来计算,也就是传统的基于项目的协作过滤算法;当 $\alpha$=0 时,推荐完全由 $k'$个逆最近邻产生,$\alpha$在[0,1]区间滑动.

$$W_{a,t}=\alpha\times\frac{\sum_{i_j\in kNN(i_t)}((W_{a,j}-\overline{W_j})\times sim(i_j,i_t))}{\sum_{i_j\in kNN(i_t)}sim(i_j,i_t)+\sum_{i_{j'}\in Rk'NN(i_t)}sim(i_{j'},i_t)}+(1-\alpha)\times\frac{\sum_{i_{j'}\in Rk'NN(i_t)}((W_{a,j'}-\overline{W_{j'}})\times sim(i_{j'},i_t))}{\sum_{i_j\in kNN(i_t)}sim(i_j,i_t)+\sum_{i_{j'}\in Rk'NN(i_t)}sim(i_{j'},i_t)} \tag{4}$$

### 3.4 算法及性能分析

算法 **1**. 相似性列表计算算法 Find_SimList.

输入:用户-项目评分矩阵 $A(m,n)$;

输出:最近邻-相似性列表 $T_{kNN}$ 和逆最近邻-相似性列表 $T_{RkNN}$.

(1) 在 $A(m,n)$中计算项目的相似度,并存入距离矩阵;

(2) 对于每个项目 $i\in I$,根据距离矩阵,找到 $i$ 的最近邻序列,并按相似度从高到低进行排序,得到如表 2 所示的最近邻-相似性列表 $T_{kNN}$ 并保存;

(3) 扫描 $T_{kNN}$,为每个项目 $i\in I$ 寻求逆最近邻序列,同样按相似度从高到低进行排序,得到逆最近邻-相似性列表 $T_{RkNN}$ 并保存.

在 Find_SimList 中,第(1)步和第(2)步也是传统的 CF 算法的必备步骤,该算法增加了第(3)步,用于计算逆最近邻-相似性列表,其计算复杂度与第(2)步是一样的.假设传统的 CF 算法的计算复杂度为 $O(n^2)$(见文献[15]中的第 3.3 节),则算法 Find_SimList 的计算复杂度最坏情况下为 $2\times O(n^2)$.在实际应用中,项目性质比用户兴趣要稳定,并且项目的增长速度相对于用户历史记录的增长速度而言要慢很多,因此,最近邻-相似性列表和逆最近邻-相似性列表只需要定期离线计算一次并保留下来即可,并不影响在线推荐产生的速度.

算法 **2**. 基于影响集的协作过滤算法 CFBIS.

输入:目标用户 $u_a$,待评分项目 $i_t$,$k$,$k'$,$\alpha$;

输出:$u_a$ 在项目 $i_t$ 上的预测评分 $P_{a,t}$.

(1) 在最近邻-相似性列表 $T_{kNN}$ 中找到 $i_t$ 所对应的行,顺序取出前 $k$ 个项目$\{i_1,i_2,...,i_k\}$;

(2) 在逆最近邻-相似性列表 $T_{RkNN}$ 中找到 $i_t$ 所对应的行中,顺序取出前 $k'$ 个项目 $\{i'_1,i'_2,...,i'_{k'}\}$;

(3) 根据推荐产生式(1)~(4),选择适当的参数值,计算 $u_a$ 在项目 $i_t$ 上的预测评分 $P_{a,t}$ 并输出.

在 CFBIS 中,第(1)步和第(3)步也是传统的 CF 算法的必备步骤,该算法增加了第(2)步,用于得到 R$k'$NN,它与第(1)步具有相同的时间复杂度.假设传统的 CF 算法的计算复杂度为 $O(1)$(顺序取出已有表中的前 $k$ 个项目),则该算法的计算复杂度为 $2\times O(1)$.

## 4 实验结果及分析

我们的数值实验平台是 PC(Pentium 4,CPU 2.4GHz,内存 512M),操作系统是 Windows XP,使用的数据类型是双精度浮点型.算法使用 Java 语言编写.

### 4.1 数据集

本实验采用的数据集是目前在衡量推荐算法质量中比较常用的 MovieLens 数据集.这个数据集由美国 Minnesota 大学的 GroupLens 研究小组创建并维护.目前,该 Web 站点的用户已经超过 43 000 人,用户评分的电影超过 3 500 部.我们选取的是其中公开的一部分数据,包括 943 个用户在 1 682 部电影上的 100 000 条评分记录,其中,每个用户至少对 20 部电影进行了评分.评分的范围是 1~5,5 表示“perfect”,而“1”表示“bad”,用户通过对不同电影上的不同评分表达了自己的兴趣.

本实验首先将数据集进一步划分为训练集和测试集,我们引入变量 $x$,表示训练集占整个数据集的百分比.例如,$x$=0.8 表示数据集中的 80%都将用作训练集,剩下的 20%作为测试集.在本文的实验中,均采用 $x$=0.8 作为实验基础.

另一方面,为了度量整个数据集的稀疏性,我们引入稀疏等级的概念,稀疏等级 $\psi$ 的含义是用户评分矩阵中未评分项目在整体数据集中所占的比例.那么,我们选择的电影数据集的稀疏等级为

$$\psi = 1-\frac{100000}{943\times 1682} = 0.93695 .$$

由此可见,所选数据集的评分矩阵是相当稀疏的.

### 4.2 推荐质量的评估标准

本文实验中采用平均绝对偏差 MAE(mean absolute error)作为度量算法优劣的标准.MAE 通过计算预测的用户评分与实际的用户评分之间的偏差来度量预测的准确性.MAE 为推荐质量提供了直观的度量方法,是最常用的一种推荐质量度量方法[17].推荐算法整体的 MAE 越小,意味着推荐的质量越高.假设算法对 $N$ 个项目预测的评分向量表示为$\{p_1,p_2,...,p_N\}$,对应的实际用户评分集合为$\{r_1,r_2,...,r_N\}$,则算法的 MAE 表示为

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

### 4.3 相似性度量标准比较

本实验首先对各种不同的相似性度量标准进行了实验.该实验使用的是传统的基于 $k$ 最近邻的项目协作过滤推荐算法,目的是为了选择最佳的相似性度量标准,作为下一步实验的基础.我们从 10 个最近邻开始,递增到 50,100,150,200,250 和 300 个最近邻,实验结果如图 4 所示.

从图 4 中可以看出,在各种条件下,标准的余弦和修正的余弦相似性度量方式都取得了比相关相似性度量方式要低的 MAE.因此,在以下的实验中,我们将采用余弦相似性作为 CFBIS 与相应的比较算法中项目相似性的度量方法.

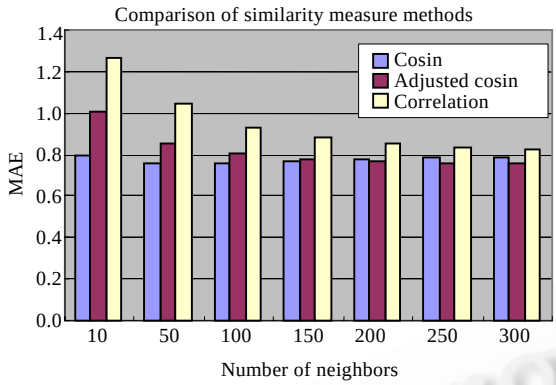


Fig.4 Comparison of similarity measure methods

图 4 相似性度量标准比较

### 4.4 *k*最近邻与*k*′逆最近邻的比率实验

这个实验的目的是为了比较 $k$ 个最近邻和 $k'$个逆最近邻在推荐预测中所起的作用.图 5 表示了分别在标准余弦和修正余弦的相似性度量方式下,只使用 $k$ 个最近邻和只使用 $k'$个逆最近邻来产生推荐预测的结果.

从图 5 中可以看出,在使用标准余弦相似性度量时,当邻居个数大于 40 之后,只使用 $k'$个逆最近邻来产生推荐的预测准确率比只使用 $k$ 个最近邻来产生推荐的预测准确率要高;而在使用修正余弦相似性度量时,只使用 $k'$个逆最近邻来产生推荐的预测准确率一直比只使用 $k$ 个最近邻来产生推荐的预测准确率要高.

因此,我们在式(3)中引入了 $\alpha$因子,以控制 $k$ 个最近邻和 $k'$个逆最近邻在推荐预测计算中所起的作用.当 $\alpha$=1 时,推荐完全根据 $k$ 个最近邻来计算,也就是传统的基于项目的协作过滤算法(文献[15]中的算法);当 $\alpha$=0 时,推荐完全由 $k'$个逆最近邻产生.$\alpha$就在[0,1]区间滑动,每次滑动的间隔为 0.1.在实验 1 中,我们观察在传统的基于项目的协作过滤推荐算法中(只使用 $k$ 个最近邻来产生推荐),当使用标准余弦相似性度量时,使 MAE 最低的 $k$=70;当使用修正余弦相似性度量时,使 MAE 最低的 $k$=300.分别固定 $k'$=$k$ 不变,我们变换$\alpha$的取值,随着 $k'$个逆最近邻在推荐计算所起的作用慢慢增加,我们观察影响集对预测值的效果.

从图 6 中我们可以看出,影响集($k'$个逆最近邻)对推荐计算是起着积极作用的.无论使用标准的余弦相似性度量还是使用修正余弦相似性度量,加入影响集后的 MAE 始终比只使用 $k$ 个最近邻的情况要低($\alpha$=1 时 MAE 最大).但是,逆最近邻的个数也不是越多越好.从图 6 中我们可以看出,当最近邻和逆最近邻的个数达到一定比例的时候($\alpha$=0.5),MAE 降到最低.

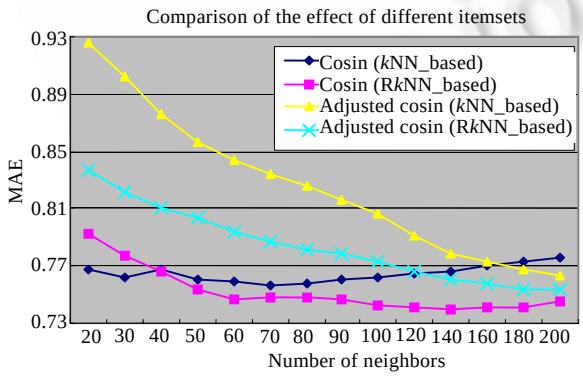


Fig.5 Comparison of the effect of different algorithms which totally based *k*NN or R*k*NN (Formula (1))

图 5 只使用最近邻或逆最近邻的结果比较(式(1))

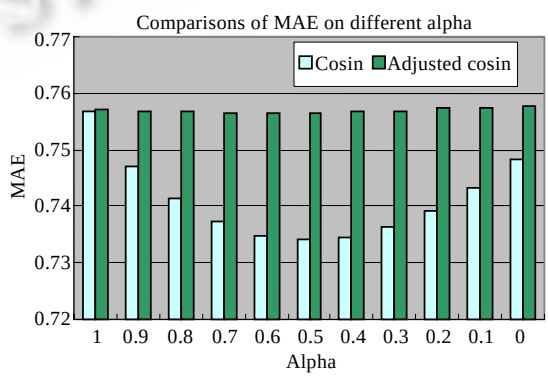


Fig.6 Comparison of MAE on different alpha (Formula (3))

图 6 在不同的 alpha 值上的 MAE 比较(式(3))

### 4.5 CFBIS与传统的项目协作过滤算法比较

本实验的目的是对基于影响集的协作过滤推荐算法 CFBIS 和传统的基于项目的协作过滤算法的推荐质量进行比较.本实验分别采用标准余弦相似性(如图 7 所示)和修正的余弦相似性度量方式(如图 8 所示).在传统的基于项目的协作过滤算法中,采用式(1)来产生推荐,称为 *k*NN_based;CFBIS 采用式(1)~(4)来产生推荐,分别称为 CFBIS1, CFBIS2,CFBIS3 和 CFBIS4.最近邻的个数 $k$ 和逆最近邻的个数 $k'$相同取值,在式(3)和式(4)中,$\alpha$=0.5.

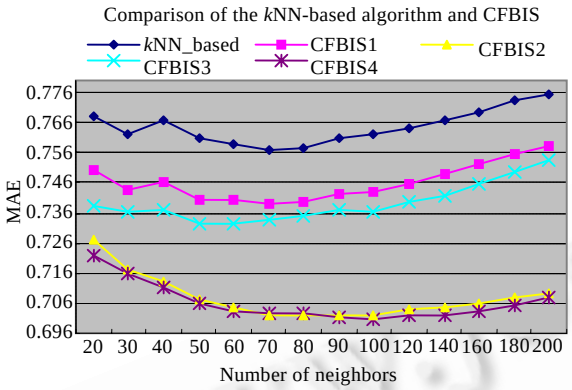


Fig.7 Comparison of the *k*NN-based algorithm and CFBISs (standard cosine similarity measure)

图 7 CFBIS 与传统的推荐算法的比较(标准余弦)

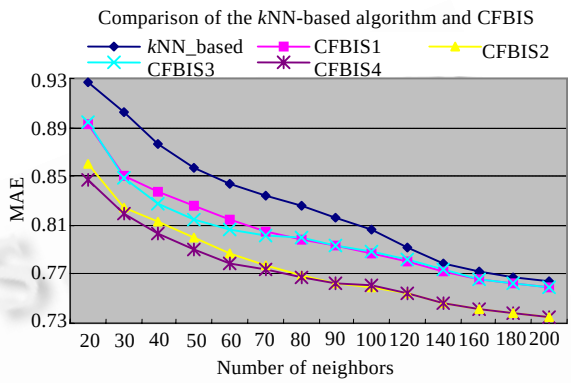


Fig.8 Comparison of the *k*NN-based algorithm and CFBISs (adjusted cosine similarity measure)

图 8 CFBIS 与传统的推荐算法的比较(修正余弦)

从图 7 和图 8 中可以看出,同时结合了项目的最近邻集合和影响集的协作过滤算法 CFBIS 比传统的只使用最近邻集合的协作过滤算法取得了较低的 MAE,因此推荐效果较好.尤其是公式 2 和公式 4,进一步对预测评分进行了修正,因此取得了更好的推荐效果.本实验证明,适当地结合项目的影响集来产生推荐,确实能在一定程度上提高项目的评分密度,从而提高推荐的预测准确率.

## 5 结论与未来工作

推荐系统是 Web 使用挖掘中的一个热点课题.在数据集稀疏的情况下,传统的只使用 $k$ 个最近邻来产生的协作过滤推荐算法由于缺乏足够的评价信息而导致产生的预测不准确甚至不能产生预测的情况.本文从检索技术中的影响集概念中得到启发,在深入分析了基于项目的协作过滤算法之后,提出了一种基于影响集的协作过滤算法 CFBIS,指出同时结合目标对象的 $k$ 个最近邻和 $k'$个逆最近邻来对目标对象进行评价.这种方法可以有效地解决现有的基于 CF 的推荐算法的不足,使得系统对目标对象的预测较为准确.实验结果表明,基于影响集的协作过滤算法 CFBIS 可以在数据集稀疏的情况下,使用影响集来提高目标对象的评价密度,从而对系统产生更准确的推荐起到了积极的作用.另外,CFBIS 还继承了基于项目的协作过滤算法的其他优点,例如,项目相似性的计算与为用户产生推荐的过程是彼此独立的,因此,当某个项目缺乏评分或浏览记录时,系统仍然可以从其他信息源获取关于该项目的知识,如结构化的语义信息等,将其作为项目相似性计算的补充,从而避免新加入项目因为缺乏评价而造成无法产生推荐的问题.

因为推荐系统中的用户-评分矩阵是一个具有相当高维度的数据集合体,目前只能通过扫描矩阵和保存项目相似性列表来进行最近邻和逆最近邻的查询.未来的工作包括为用户-评分矩阵建立多维的动态索引结构[18],同时对最近邻和逆最近邻进行高效的查询,从而提高推荐系统的时间性能.

**References**:

[1] Broadvision. http://www.broadvision.com

[2] Nanopoulos A, Katsaros D, Manolopoulos Y. A data mining algorithm for generalized web prefetching. IEEE Trans. on Knowledge and Data Engineering, 2003,15(5):1155−1169.

[3] Wang S, Gao W, Li JT. Real time persona1ization based on classification. Chinese Journal of Computers, 2002,25(8):845−852 (in Chinese with English abstract).

[4] Jin X, Zhou Y, Mobasher B. A unified approach to personalization based on probabilistic latent semantic models of Web usage and content. In: Proc. of the AAAI 2004 Workshop on Semantic Web Personalization (SWP 2004). San Jose: AAAI, 2004. 26−34. http://maya.cs.depaul.edu/~mobasher/cgi-bin/view-pubs.pl?CID=WUM

[5] Herlocker J, Konstan J, Riedl J. Explaining collaborative filtering recommendations. In: Proc. of the ACM 2000 Conf. on Computer Supported Cooperative Work. 2000. 241−250. http://portal.acm.org/citation.cfm?doid=358916.358995

[6] Miller B, Konstan J, Terveen L, Riedl J. PocketLens: Towards a personal recommender system. ACM Trans. on Information Systems, 2004,22(3):437−476.

[7] Baudisch P, Brueckner L. TV scout: Guiding users from printed TV program guides to personalized TV recommendation. In: Proc. of the 2nd Workshop on Personalization in Future TV. Malaga, 2002. 157−166. http://www.patrickbaudisch.com/publications/2002-Baudisch-TV02-TVScoutGuidingUsers.pdf

[8] DeRoure D, Hall W, Reich S, Hill G, Pikrakis A, Stairmand M. MEMOIR—An open framework for enhanced navigation of distributed information. Information Processing and Management Journal (Elsevier Science), 2001,37(1):53−74.

[9] Holmquist LE, Jacobsson M, Rost M. When media gets wise: Collaborative filtering with mobile media agents. In: Proc. of the IUI 2006, the 10th Int'l Conf. on Intelligent User Interfaces. Sydney, 2006. http://portal.acm.org/

[10] Good N, Schafer JB, Konstan JA, Borchers A, Sarwar BM, Herlocker J, Riedl JT. Combining collaborative filtering with personal Agents for better recommendations. In: Proc. of the 16th National Conf. on Artificial Intelligence (AAAI'99). Menlo Park: American Association for Artificial Intelligence, 1999. 439−446. http://portal.acm.org/citation.cfm?id=315149.315352&coll=&dl=&CFID= 15151515&CFTOKEN=6184618

[11] Jin X, Zhou YZ, Mobasher B. A maximum entropy Web recommendation system: Combining collaborative and content features. In: Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD 2005). Chicago, 2005. 612−617. http://portal.acm.org/citation.cfm?id=1081945&dl=&coll=&CFID=15151515&CFTOKEN=6184618

[12] Sarwar B, Karypis G, Konstan J, Riedl J. Application of dimensionality reduction in recommender systems—A case study. In: Proc. of the WebKDD 2000 Workshop at the ACM-SIGKDD Conf. on Knowledge Discovery in Databases (KDD 2000). 2000. http://citeseer.ist.psu.edu/sarwar00application.html

[13] Deng AL, Zhu YY, Shi BL. A collaborative filtering recommendation algorithm based on item rating prediction. Journal of Software, 2003,14(9):1621−1628 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1621.htm

[14] Mobasher B, Jin X, Zhou YZ. Semantically enhanced collaborative filtering on the Web. In: Berendt B, *et al*., eds. Web Mining: From Web to Semantic Web. LNAI 3209, Springer-Verlag, 2004. 57−76.

[15] Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based collaborative filtering recommendation algorithms. In: Proc. of the 10th Int'l World Wide Web Conf. New York: ACM Press, 2001. 285−295.

[16] Korn F, Muthukrishnan S. Influence sets based on reverse nearest neighbor queries. In: Naughton JF, Bernstein PA, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2000. 201−212.

[17] Herlocker J, Konstan J, Terveen L, Riedl J. Evaluating collaborative filtering recommender systems. ACM Trans. on Information Systems (TOIS), 2004,22(1):5−53.

[18] Chen J, Yin J, Chen L. Research on influence sets and its dynamic indexing structure and query algorithm based on multi-dimensional vectors. Journal of Computer Research and Development, 2004,41(Suppl.):90−95.

附中文参考文献:

[3] 王实,高文,李锦涛.基于分类方法的 Web 站点实时个性化推荐.计算机学报,2002,25(8):845−852.

[13] 邓爱林,朱扬勇,施伯乐.基于项目评分预测的协同过滤推荐算法.软件学报,2003,14(9):1621−1628. http://www.jos.org.cn/1000-9825/14/1621.htm

[18] 陈健,印鉴,陈玲.影响集问题及其多维向量动态索引结构的研究.计算机研究与发展,2004,41(增刊):90−95.

陈健(1977－),女,广西柳州人,博士,讲师,主要研究领域为 Web 挖掘,模式识别,信息处理.

印鉴(1968－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据挖掘,人工智能.