文章编号: 1000-6788(2012)07-1537-09 中图分类号: C931.9; TP18 文献标志码: A

基于 n 序访问解析逻辑的协同过滤冷启动消除方法

李 聪 [1], 梁昌勇 [2]

(1. 四川师范大学 计算机科学学院, 成都 610066; 2. 合肥工业大学 管理学院, 合肥 230009)

摘 要 协同过滤是目前个性化推荐系统中广泛使用和最成功的推荐算法, 但在用户评分极端稀疏的情况下将面临冷启动问题, 具体包括新用户问题和新项目问题. 针对新用户问题, 提出了一种基于 n 序访问解析逻辑的冷启动消除方法, 首先通过 Web 日志来获取用户访问项序, 进而定义了 n 序访问解析逻辑将其分解为用户访问子序集; 在此基础上设计了用户访问项序的相似性计算方法来搜寻新用户的最近邻集合, 进而提出了改进最频繁项提取算法 IMIEA (improved most-frequent items extracting algorithm) 来生成面向新用户的 top-N 推荐. 实验结果表明, 本文提出的新方法能够有效实现面向新用户的个性化推荐, 消除了协同过滤冷启动中的新用户问题.

关键词 推荐系统; 协同过滤; 冷启动; n 序访问解析逻辑

Cold-start eliminating method of collaborative filtering based on n-sequence access analytic logic

LI Cong[1], LIANG Chang-yong[2]

(1. College of Computer Science, Sichuan Normal University, Chengdu 610066, China;
2. School of Management, Hefei University of Technology, Hefei 230009, China)

Abstract Collaborative filtering is the most successful and widely used recommendation technology in personalized recommender systems. However, collaborative filtering faces cold-start problem, which includes new user problem and new item problem, when user ratings are extremely sparse. To solve the new user problem, a cold-start eliminating method was proposed. Firstly, the items access by user was obtained via web logs; secondly, n-sequence access analytic logic was defined to decompose user's access item sequence to user access sub-sequence set; thirdly, a similarity measure for user access item sequence was proposed to search target user's nearest neighborhood; fourthly, improved most-frequent item extracting algorithm, which called IMIEA, was proposed to obtain the top-N recommendation for the new user. The experimental results show that the proposed method can efficiently eliminate new user problem and obtain better top-N recommendation quality.

Keywords recommender systems; collaborative filtering; cold-start; n-sequence access analytic logic

1 引言

随着 Internet 的迅猛发展, 个性化推荐系统迅速在电子商务、数字图书馆、在线影音等诸多领域得到普及应用. 例如在电子商务领域, 推荐系统 [1] 被电子商务网站用作虚拟店员 (virtual salespeople) 向客户提供商品信息和建议, 帮助用户决定应该购买何种商品. 电子商务推荐系统作为一种强大的新兴技术, 能够帮助用户找到他们喜爱的商品, 反过来也提高了电子商务网站的销售额, 因此很快成为电子商务网站一种至关重要的工具, 其作用主要表现在三个方面: 将电子商务网站浏览者转变为购买者、提高电子商务网站交叉销售能力、建立客户忠诚度. 协同过滤 (collaborative filtering) 作为目前个性化推荐系统中广泛使用的、最成功的推荐算法 [2], 完全依赖于用户评分来构建用户 - 项目评分矩阵 (user-item ratings matrix), 使用统计技

收稿日期: 2010-04-26
资助项目: 四川师范大学 "251 重点人才培养工程"
作者简介: 李聪 (1978–), 男, 四川西充人, 博士, 副教授, 研究方向: 电子商务, 商务智能; 梁昌勇 (1965–), 安徽肥西人, 博士, 教授, 博士生导师, 研究方向: 智能决策支持系统.

术寻找与目标用户有相同或相似兴趣偏好 (例如对不同商品的评分相似或所购商品相似) 的邻居用户, 再根据邻居用户对商品项的评分来预测目标用户对其未评分项的评分值, 进而选择预测评分最高的前 N 项商品作为推荐集反馈给用户 (top-N 推荐), 其基本思想是用户会对邻居用户所喜欢的商品产生兴趣, 即基于用户 (user-based) 的协同过滤. 因此, 用户评分数据收集越多, 协同过滤算法的推荐质量越高. 但是, 由于电子商务网站用户及商品项的数量庞大且不断增加, 使得用户 - 项目评分矩阵成为高维矩阵; 同时用户给予评分的商品项很少, 通常在 1% 以下 [3], 导致用户 - 项目评分矩阵中的数据极端稀疏. 协同过滤数据稀疏性 (sparsity) 问题 [4] 由此产生, 并严重影响推荐质量. 这种现象在数字图书馆、在线影音等使用个性化推荐系统的领域也普遍存在.

冷启动 (cold-start)[5] 问题是稀疏性问题的极端情况, 也称为第一评价人问题 (first-rater problem)[6] 或早期评价人问题 (early-rater problem)[7], 具体包括新用户问题和新项目问题, 即当新用户 (新项目) 进入推荐系统后, 由于还未提供 (接受) 任何项目 (用户) 的评分, 导致系统无法向新用户推荐其可能喜欢的项目 (或将新项目推荐给可能喜欢它的用户). 在协同过滤推荐系统刚投入运行时, 每个用户在每个项目上都将面临冷启动问题. "然而对一个电子商务站点来说, 在用户与站点交互的早期阶段就能够提供有效的个性化服务对提高客户保留度和客户购买率具有重要作用"[8], 因此新用户问题相对于新项目问题显得尤为重要.

针对上述问题, 本文首先通过 Web 日志来获取用户访问项序, 进而定义了 n 序访问解析逻辑将其分解为用户访问子序集; 在此基础上设计了用户访问项序的相似性计算方法来搜寻新用户的最近邻集合, 使得最近邻用户与新用户之间具有相同或相似的用户访问项序, 进而提出改进的最频繁项提取算法 IMIEA (improved most-frequent items extracting algorithm), 以获取最近邻用户的最频繁项并得到面向新用户的 top-N 推荐. 通过在美国电子商务网站 www.gazelle.com 提供的用户访问日志数据集上的实验结果表明, 本文提出的新方法能够有效实现面向新用户的个性化推荐, 消除了协同过滤冷启动中的新用户问题.

本文第 2 节对相关工作进行了简要分析; 第 3 节提出了 n 序访问解析逻辑、用户访问项序相似性计算方法、最频繁项提取算法 IMIEA 及面向新用户的 top-N 推荐方法; 第 4 节在 Gazelle 数据集上进行实验并进行结果分析; 第 5 节给出全文结论.

2 相关工作分析

考虑有一位新用户上线访问某资源站点 (例如电子商务网站、数字图书馆等). 不失一般性, 这里我们假设他进入了一家电子商务网站. 由于这位新用户没有在该网站提供过任何针对网站商品的评分数据, 也没有任何购买记录, 因此传统的协同过滤算法无法向其提供推荐服务. 但是, 新用户必然会在网站上浏览一定数量的商品页面, 这些页面实质上也就体现了该用户的兴趣偏好. 对此, 一些研究者采用 k-means 聚类 [9]、模糊聚类 [10] 等方法对用户访问路径进行研究, 但他们都没有考虑用户对 Web 页面的访问顺序.

在研究用户访问路径的过程中不难发现, 我们不仅要考虑用户访问了哪些 Web 页面, 也要考虑用户访问 Web 页面的顺序, 即用户访问的有序性, 因为用户访问页面的先后顺序反映了用户兴趣的动态变化. 例如, 用户 u_1、u_2、u_3 各自的访问页面按时间先后顺序排列如下:

$$u_1 : P_1 \to P_3 \to P_4 \to P_2 \to P_5 \to P_3$$

$$u_2 : P_2 \to P_4 \to P_3 \to P_5 \to P_4$$

$$u_3 : P_3 \to P_4 \to P_2 \to P_5 \to P_3$$

从以上访问内容来看, 用户 u_1、u_2、u_3 均访问了页面集合 $\{P_2, P_3, P_4, P_5\}$. 但是从访问路径来看, u_3 和 u_1 的访问顺序保持一致, 而 u_2 与 u_1 的访问顺序大相径庭, 因此用户 u_1、u_3 之间的相似性大于 u_1 与 u_2 的相似性. 为了充分体现用户在访问路径上的有序性, 王实等学者 [11] 提出了一种 K-paths 聚类方法来对用户访问路径进行聚类, 但该方法是基于 k-means 聚类算法设计的, 导致 "经常中止于一个局部最优解".

除了有序性之外, 对于电子商务个性化推荐服务而言, 用户的访问路径还具有一个特殊性, 就是由于电子商务网站为了更好地进行用户导航, 为用户提供了多种不同的到达某个具体商品项页面的链接方法, 使得用户可以通过不同的 Web 页面链接转到该商品页面. 例如, 用户既可以通过网站的商品分类导航页一步步进入直至到达特定商品所在的页面; 也可以直接通过网站首页的 "站内搜索" 功能直接输入该商品的关键字进行查找, 从而在查找结果列表中点击该商品的链接跳转到相应的商品页面. 这就使得用户访问路径中可能

有相当一部分页面是该用户在寻找其感兴趣商品项的过程中经过的导航页和分类页. 因此, 对于同样的一个用户访问同一项商品, 由于采用了不同的访问方法, 将会导致访问路径差别很大. 如果使用传统的访问路径比较方法, 例如上述 K-paths 聚类, 将会判定两条访问路径不相似, 从而将这两条路径归入不同的聚类.

但是, 对于电子商务网站经营者而言, 他最关心的是用户究竟访问了哪些商品, 以便据此判断用户的兴趣偏好. 而用户通过哪些网页转到这些商品项页面的问题, 则显得相对次要. 因此, 与传统的单纯基于用户访问路径进行用户相似性分析的思路不同, 本文以用户访问项序作为用户相似性的计算依据, 这不仅可以大幅度减少用户访问路径长度、突出用户访问商品项的有序性, 并且可以使用户相似性计算更为合理和准确.

根据上述思路, 本文需要解决以下两个子问题:

1) 如何得到用户的商品项访问序列 (即用户访问项序)? 对此本文提出了 n 序访问解析逻辑来解决.

2) 如何基于用户访问项序给出面向新用户的 top-N 推荐? 对此本文提出了用户访问项序相似性计算方法和最频繁项提取算法 IMIEA 来解决.

3　n 序访问解析逻辑

3.1　用户访问项序的获取

定义 1 (Web 站点的有向图模型)　对于任意 Web 站点 G, 其拓扑结构可表示为一个有向图 $G = \langle P, E\rangle$. 其中, P 是站点 G 所有页面的 URL 集合; E 是各个页面间的超链接集合, 即 G 中有向边的集合.

定义 2 (用户访问项序)　设用户 u 访问的所有商品项集合为 $I(u)$ 且 $I(u)$ 的大小为 $|I(u)| = n$, 则 u 的用户访问项序 $S(u)$ 为一个单向动态增长序列, 可将 $S(u)$ 表示为一个五元组:

$$\langle u.ip, u.uid, \{I_g.iid, I_g.url, I_g.time\}^{(n)}\rangle$$

其中, $u.ip$、$u.uid$ 分别表示用户的 IP 地址和 ID 标识符; $\{I_g.iid, I_g.url, I_g.time\}(1 \le g \le n \le N)$ 表示用户 u 访问的第 g 个商品项的 ID 标识符、URL 地址及访问时间, $\{Ig.url\}^{(n)} \in P$; n 称为用户访问项序的长度, 等于 $S(u)$ 中的商品项总数; N 表示该网站的商品项总数. 通常情况下, 有 $n << N$, 即用户根据其兴趣偏好通常只会访问网站很小一部分商品. 随着用户访问网站次数和时间的增加, 该用户的用户访问项序也会不断增长. 下面给出用户访问项序的具体获取步骤.

Step 1　对网站的 Web 日志进行预处理, 找出用户访问记录集合.

电子商务网站用户在网站上的访问历史数据是由站点 Web 服务器自动记录在日志中 [12]. 在提取用户访问记录的过程中, 最重要的工作就是新用户的识别. 广义上的新用户包括: ①已注册但未作出评分/购买行为的用户; ②未注册的匿名用户. 对于前者而言, 通过用户 ID 识别他们非常容易; 对于后者而言, 由于本地缓存技术和代理服务器的广泛使用, 要想准确识别每一个匿名用户并不容易, 因此对匿名用户的识别需要遵循一定的准则. 本文给出以下两条匿名用户识别准则:

准则 1　在 Web 访问日志中, 若两条访问记录的 IP 地址相同, 但代理日志显示两条访问记录所使用的操作系统或浏览器类型不同, 则上述两条访问记录分别属于两个不同的匿名用户.

准则 2　在 Web 访问日志中, 若两条访问记录的 IP 地址相同, 但用户当前请求的页面同用户已浏览的页面之间无任何超链接, 则上述两条访问记录分别属于两个不同的匿名用户.

Step 2　对于用户访问记录, 根据用户 IP、用户 ID 及商品项 ID 识别和提取用户对商品项的访问记录, 并按照访问时间进行排序, 形成该用户对商品项的有序访问记录集合, 从而得到我们所需要的用户访问项序. 用户访问项序是本文进行面向新用户的 top-N 推荐的依据.

3.2　n 序访问解析逻辑

一个用户访问项序可以解析为多个不同长度的访问序列. 下面首先给出用户访问子序的概念.

定义 3 (用户访问子序)　用户 u 的一个用户访问子序 $S(u(k))$ 是指 u 的长度为 n 的用户访问项序 $S(u)$ 中任意一个长度为 $k(1 \le k \le n)$ 的访问序列. 随 k 值的不同可得到不同长度的 $S(u(k))$, 所有 $S(u(k))$ 则构成用户 u 的用户访问子序集 $\bigcup\limits_{1\le k\le n}\{S(u(k))\}$. $S(u(k))$ 是 $S(u)$ 中从第 I_{n-k+1} 个商品项开始取连续 k 个商品项的访问序列, 即

$$S(u(k)) = \langle u.ip, u.uid, \{I_{n-k+i}.iid, I_{n-k+i}.url, I_{n-k+i}.time\} : 1 \le i \le k\rangle \tag{1}$$

式 (1) 可简化表示为

$$S(u(k)) = I_{n-k+1}.iid \to I_{n-k+2}.iid \to \cdots \to I_{n-k+k}.iid \tag{2}$$

由式 (2) 可知, 当 $k=1$ 时, $S(u(1))$ 表示用户 u 访问的某一个商品项, $\bigcup\limits_{k=1}\{S(u(k))\}$ 即为 u 访问的 n 个商品项的集合; 当 $k=n$ 时, $S(u(n))$ 即为用户 u 的用户访问项序 $S(u)$. 下面给出不同 k 值时的 $\bigcup\limits_{1\le k\le n}\{S(u(k))\}$ 规模:

$$\left|\bigcup_{1\le k\le n}\{S(u(k))\}\right|=n-k+1,\qquad 1\le k\le n \tag{3}$$

例如, 设用户 u 的用户访问项序 $S(u)=I_1.32\rightarrow I_2.65\rightarrow I_3.27\rightarrow I_4.28$, 则其用户访问子序集 $\bigcup\limits_{1\le k\le n}\{S(u(k))\}$ 如表 1 所示.

表 1 用户访问子序集示例

$S(u(1))$	$S(u(2))$	$S(u(3))$	$S(u(4))$
$I_1.32$	$I_1.32\rightarrow I_2.65$	$I_1.32\rightarrow I_2.65\rightarrow I_3.27$	$I_1.32\rightarrow I_2.65\rightarrow I_3.27\rightarrow I_4.28$
$I_2.65$	$I_2.65\rightarrow I_3.27$	$I_2.65\rightarrow I_3.27\rightarrow I_4.28$	
$I_3.27$	$I_3.27\rightarrow I_4.28$		
$I_4.28$			

下面讨论如何从用户 u 的用户访问项序 $S(u)$ 得到 u 的用户访问子序集 $\bigcup\limits_{1\le k\le n}\{S(u(k))\}$. 对此, 本文提出基于循环遍历思想的 n 序访问解析逻辑来得到所有长度的 $S(u(k))$.

定义 4 (n 序访问解析逻辑) n 序访问解析逻辑 $ADL^{(n)}$ 是一个四元组:

$$ADL^{(n)}=\left\langle\overrightarrow{\bigcup},\Sigma,\Psi,l^{(n)}\right\rangle \tag{4}$$

式 (4) 中,

1) $\overrightarrow{\bigcup}=\left\{\bigcup\limits_{1\le k\le n}\{S(u(k))\}\right\}$, 表示一个有限的有向序列集合;

2) $\Sigma=\{I_1.iid,I_2.iid,\cdots,I_k.iid\}$, 表示 $\overrightarrow{\bigcup}$ 中所有序列的节点集合, $I_k.iid$ 表示用户在有向序列上所访问的商品项 (k 是根据用户访问时间顺序的编号赋值), 且有 $\overrightarrow{\bigcup}\bigcap\Sigma=\bigcup\limits_{k=1}\{S(u(k))\}$;

3) Ψ 是序列起始符且满足 $\Psi\bigcap\Sigma\neq\varnothing$;

4) $l^{(n)}$ 是具有 $l^{(n)}$: $(I_i.iid\rightarrow I_j.iid)_{1\le i\le j\le n}$ 形式的有限导出规则集, 同时满足且必须满足如下性质:

① 存在性, 即

$$\{I_i.iid,\forall I_g.iid,I_j.iid\}\in\Sigma,\qquad i<g<j.$$

② 完备性, 即

$$\left|\bigcup_{i<g<j}I_g.iid\right|=j-i-1.$$

③ 有序性, 即

$$\forall I_{g-1}.iid\neq\varnothing,\quad\forall I_{g+1}.iid\neq\varnothing,\qquad i<g<j.$$

3.3 用户访问项序的相似性计算方法

对于两个不同的用户, 在计算用户访问项序的相似性时不能仅仅对其作直接比较, 而是应该在两个用户各自的用户访问子序集上进行比较. 也就是说, 既要考虑两个用户共同访问过的商品项数量, 也要考虑两个用户的用户访问子序的交集大小 (这反映了用户访问商品项的动态性和有序性). 设用户 u、v ($u\neq v$) 各自的用户访问项序分别为 $S(u)$、$S(v)$, $|S(u)|=n$, $|S(v)|=m$, 则用户访问项序的相似性 $sim(S(u),S(v))$ 可以分解为基于 k 序的项序相似性和基于 $\max(n,m)$ 序的项序相似性这两个部分分别计算, 然后进行融合处理得到相似性的唯一值.

3.3.1 基于 k 序的项序相似性

令 $sim(u,v)_{S(u(k)),S(v(k))}$ 表示在长度为 $k(1\le k\le \min(n,m))$ 的用户访问子序集 $\bigcup\limits_{k}\{S(u(k))\}$ 上的相似性, 则对于用户 u、v 的用户访问子序集的并集 $S_k=\left\{\bigcup\limits_{1\le k\le n}\{S(u(k))\}\right\}\bigcup\left\{\bigcup\limits_{1\le k\le m}\{S(v(k))\}\right\}$, 在计算 $sim(u,v)_{S(u(k)),S(v(k))}$ 时需要比较相同子序出现的次数.

令 $|S_k|=h$, $S_{k,i}$ 表示 S_k 中的第 $i(1\le i\le h)$ 个用户访问子序, $S_{k,i}(u)$、$S_{k,i}(v)$ 分别表示 $S_{k,i}$ 在 $\bigcup\limits_{k}\{S(u(k))\}$、$\bigcup\limits_{k}\{S(v(k))\}$ 中出现的次数, 则可用 $h\times 2$ 阶矩阵 $M_{u,v}(k,2)$ 来表示用户 u、v 在长度为 $k(1\le k\le \min(n,m))$ 的用户访问子序集 $\bigcup\limits_{k}\{S(u(k))\}$ 上的相似性. 根据 $M_{u,v}(k,2)$ 我们可以将 $\bigcup\limits_{k}\{S(u(k))\}$、$\bigcup\limits_{k}\{S(v(k))\}$ 表示为向量 $\overrightarrow{S}_{u,k}$、$\overrightarrow{S}_{v,k}$:

$$\overrightarrow{S}_{u,k}=(S_{k,1}(u),S_{k,2}(u),\cdots,S_{k,h}(u)) \tag{5}$$

$$\overrightarrow{S}_{v,k}=(S_{k,1}(v),S_{k,2}(v),\cdots,S_{k,h}(v)) \tag{6}$$

从而, 我们可以采用向量相似性计算方法来计算 $\overrightarrow{S}_{u,k}$、$\overrightarrow{S}_{v,k}$ 之间的相似性 $sim\left(\overrightarrow{S}_{u,k},\overrightarrow{S}_{v,k}\right)$:

$$\begin{aligned} sim\left(\overrightarrow{S}_{u,k},\overrightarrow{S}_{v,k}\right) &= \frac{\overrightarrow{S}_{u,k}\cdot\overrightarrow{S}_{v,k}}{\|\overrightarrow{S}_{u,k}\|_2\cdot\|\overrightarrow{S}_{v,k}\|_2}\\ &= \frac{\sum\limits_{i=1}^{h}S_{k,i}(u,j)\cdot S_{k,i}(v,j)}{\sqrt{\sum\limits_{i=1}^{h}S_{k,i}^2(u,j)}\cdot\sqrt{\sum\limits_{i=1}^{h}S_{k,i}^2(v,j)}}\\ &= sim(u,v)_{S(u(k)),S(v(k))} \end{aligned} \tag{7}$$

从而, 用户 u、v 在各个 k 长度的用户访问子序集上的相似性 $sim(u,v)_{S(u(k)),S(v(k)),1\le k\le\min(n,m)}$ 为:

$$\begin{aligned} sim(u,v)_{S(u(k)),S(v(k)),1\le k\le\min(n,m)} &= \frac{\sum\limits_{k=1}^{\min(n,m)}\left(\lambda_k\cdot sim(u,v)_{S(u(k)),S(v(k))}\right)}{\min(n,m)}\\ &= \frac{\sum\limits_{k=1}^{\min(n,m)}\left(\lambda_k\cdot\frac{\sum\limits_{i=1}^{h}S_{k,i}(u,j)\cdot S_{k,i}(v,j)}{\sqrt{\sum\limits_{i=1}^{h}S_{k,i}^2(u,j)}\cdot\sqrt{\sum\limits_{i=1}^{h}S_{k,i}^2(v,j)}}\right)}{\min(n,m)} \end{aligned} \tag{8}$$

式 (8) 中, λ_k 为 $sim(u,v)_{S(u(k)),S(v(k))}$ 在 $sim(u,v)_{S(u(k)),S(v(k)),1\le k\le\min(n,m)}$ 中的权重, 且需要满足

$$\sum_{k=1}^{\min(n,m)}\lambda_k=1 \tag{9}$$

k 值越大, 则相应的权重 λ_k 越大. 也就是说, 两个用户之间的共同访问子序越长, 则这两个子序之间的相似性越大, 表示两个用户的兴趣偏好越相近. 现在的问题是, 如何赋予合理的权重 λ_k?

首先令 $\lambda_k=kx$, 其中 k 值为用户 u、v 的共同访问子序长度, x 则为权重的动态下限值 (即当前两位用户共同访问子序长度为 1 时的子序相似性), 从而用户子序之间的相似性将随着共同访问子序长度的增加而增大, 且

$$\sum_{k=1}^{\min(n,m)}kx=1 \tag{10}$$

则有

$$\begin{aligned} \sum_{k=1}^{\min(n,m)}kx &= [1+2+\cdots+k+\cdots+\min(n,m)]\cdot x\\ &= \frac{\min(n,m)\cdot[\min(n,m)+1]}{2}\cdot x\\ &= 1 \end{aligned} \tag{11}$$

由式 (11) 可求得

$$x = \frac{2}{\min(n,m)\cdot[\min(n,m)+1]} \tag{12}$$

因此权重 λ_k 为

$$\lambda_k = k\cdot x = \frac{2k}{\min(n,m)\cdot[\min(n,m)+1]} \tag{13}$$

从而有 $\sum_{k=1}^{\min(n,m)}\lambda_k = 1$, 故 $sim(u,v)_{S(u(k)),S(v(k)),1\le k\le\min(n,m)}$ 被改写为

$$\begin{aligned} & sim(u,v)_{S(u(k)),S(v(k)),1\le k\le\min(n,m)} \\ = & \frac{\sum\limits_{k=1}^{\min(n,m)}\left(\lambda_k\cdot sim(u,v)_{S(u(k)),S(v(k))}\right)}{\min(n,m)} \\ = & \frac{\sum\limits_{k=1}^{\min(n,m)}\left(\frac{2k}{\min(n,m)\cdot[\min(n,m)+1]}\cdot\frac{\sum\limits_{i=1}^{h}S_{k,i}(u,j)\cdot S_{k,i}(v,j)}{\sqrt{\sum\limits_{i=1}^{h}S_{k,i}^2(u,j)}\cdot\sqrt{\sum\limits_{i=1}^{h}S_{k,i}^2(v,j)}}\right)}{\min(n,m)} \end{aligned} . \tag{14}$$

3.3.2 基于 $\max(n,m)$ 序的项序相似性

对于两个用户 u、v 的用户访问项序 $S(u)$、$S(v)$, 由于很多时候 $|S(u)|\neq|S(v)|$(等同于 $n\neq m$), 即 u、v 的用户访问项序长度不同, 因此传统的向量相似性以及绝对值距离 (Manhattan)、欧式距离 (Euclidean)、麦考斯基距离等相似性计算方法 [13] 均无法直接应用于比较 $S(u)$ 与 $S(v)$ 在 $\max(n,m)$ 序上的相似性. 对此, 本文借鉴自然语言处理领域广泛应用的 Levenshtein 距离 [14] (Levenshtein distance, 也称 edit distance, 用于计算两个多维向量之间的相似性) 的思想来计算基于 $\max(n,m)$ 序的项序相似性 $sim(u,v)_{S(u(n)),S(v(m))}$.

1) 首先将 $S(u)$、$S(v)$(即 $S(u(n))$、$S(v(m))$) 分别表示为向量 $\overrightarrow{S}_{u,n}$、$\overrightarrow{S}_{v,m}$, 令 $S_{u,n}^i$、$S_{v,m}^j$ 分别表示 $\overrightarrow{S}_{u,n}$、$\overrightarrow{S}_{v,m}$ 中的任意一个商品项, 其中 $1\le i\le n, 1\le j\le m$;

2) 基于 $\max(n,m)$ 序的项序相似性 $sim(u,v)_{S(u(n)),S(v(m))}$ 将通过由向量 $\overrightarrow{S}_{u,n}$ 转化为 $\overrightarrow{S}_{v,m}$ 所需要进行的增加、删除、替换三种操作的次数来体现.

基于 $\max(n,m)$ 序的项序相似性算法描述如下:

1) 定义一个 $n\times m$ 阶矩阵, 以存储距离值;

2) 初始化 $(n+1)\times(m+1)$ 阶矩阵 $M_{n+1,m+1}$, 并让第一行和列的值从 0 开始增长. 扫描向量 $\overrightarrow{S}_{u,n}$、$\overrightarrow{S}_{v,m}$, 若 $S_{u,n}^i = S_{v,m}^j$, 用变量 $temp$ 记为 0; 否则 $temp$ 记为 1. 然后将矩阵的 $M_{n+1,m+1}[i,j]$ 赋值为

$$M_{n+1,m+1}[i,j] = \min\left\{\begin{array}{l}(M_{n+1,m+1}[i-1,j]+1)\\(M_{n+1,m+1}[i,j-1]+1)\\(M_{n+1,m+1}[i-1,j-1]+temp)\end{array}\right\} \tag{15}$$

3) 扫描结束后, 返回矩阵的最后一个值即 $M_{n+1,m+1}[n,m]$, 即为向量 $\overrightarrow{S}_{u,n}$、$\overrightarrow{S}_{v,m}$ 之间的距离. 由于 $\max(M_{n+1,m+1}[n,m]) = \max(n,m)$, 则基于 $\max(n,m)$ 序的项序相似性定义为:

$$sim(u,v)_{S(u(n)),S(v(m))} = 1 - \frac{M_{n+1,m+1}[n,m]}{\max(n,m)} \tag{16}$$

3.3.3 用户访问项序相似性 $sim(S(u),S(v))$

用户访问项序相似性 $sim(S(u),S(v))$ 是将基于 k 序的项序相似性 $sim(u,v)_{S(u(k)),S(v(k)),1\le k\le\min(n,m)}$ 和基于 $\max(n,m)$ 序的项序相似性 $sim(u,v)_{S(u(n)),S(v(m))}$ 进行融合处理得到:

$$\begin{aligned} & sim(S(u),S(v)) \\ = & \sqrt{sim(u,v)_{S(u(k)),S(v(k)),1\le k\le\min(n,m)}\times sim(u,v)_{S(u(n)),S(v(m))}} \\ = & \sqrt{\frac{\sum\limits_{k=1}^{\min(n,m)}\left(\frac{2k}{\min(n,m)\cdot[\min(n,m)+1]}\cdot\frac{\sum\limits_{i=1}^{h}S_{k,i}(u,j)\cdot S_{k,i}(v,j)}{\sqrt{\sum\limits_{i=1}^{h}S_{k,i}^2(u,j)}\cdot\sqrt{\sum\limits_{i=1}^{h}S_{k,i}^2(v,j)}}\right)}{\min(n,m)}\cdot\left(1-\frac{M_{n+1,m+1}[n,m]}{\max(n,m)}\right)} \end{aligned} \tag{17}$$

3.4 基于改进最频繁项提取算法的 top-N 推荐

在完成对新用户 u 与用户访问记录中其他用户的相似性计算后, 我们就可以抽取 $sim(S(u), S(v))$ 值最大的前 r 个用户组成新用户 u 的最近邻集合, 然后通过最频繁项提取算法向新用户作出 top-N 推荐. 传统方法是: 扫描最近邻集合的用户访问项序, 统计这些最近邻用户访问的各个商品的次数, 然后将访问次数最高且新用户 u 还未访问过的前 N (N 值通常取 10) 个商品项作为 top-N 推荐项集反馈给 u.

但是, 上述传统最频繁项提取算法只考虑了最近邻用户对商品项的访问次数, 而未考虑各个最近邻用户与新用户的用户访问项序相似性大小对 top-N 推荐的影响. 从理论上讲, 与新用户之间具有最大用户访问项序相似性的最近邻用户所访问的商品项相对于其他用户访问的商品项更具有推荐价值, 即便这些商品项被访问的次数相同. 因此, 本文综合考虑商品项访问次数及访问用户相对于新用户的重要性, 提出了一个改进的最频繁项提取算法 IMIEA (improved most-frequent items extracting algorithm).

IMIEA 算法的基本描述如下:

Step 1 对于最近邻用户集合 NUS 已访问但新用户 u 未访问的任意商品项 I_k, 即 $\forall I_k \in \bigcup(1 - S(u_r))$, 其中 u_r 为最近邻集合 NUS 中访问过 I_k 的任意用户, 设 I_k 被最近邻用户访问的次数为 $C(I_k)$, 则 I_k 的推荐值 $v(I_k)$ 采用式 (18) 计算:

$$v(I_k) = \sum_{u_r \in NUS} C(I_k) \cdot (sim(S(u), S(v))) \tag{18}$$

Step 2 选择 $v(I_k)$ 最大的前 N 个商品项作为 top-N 推荐项集 I_{rec} 反馈给新用户:

$$I_{rec} = \{I_k | \arg\max(v(I_k))\} \tag{19}$$

在生成 I_{rec} 的过程中, 最近邻数量 r 的大小关系到是否能够完成 N 个最频繁项的搜寻, 也就是说 r 过小的话可能会得不到足够的 N 个最频繁项. 根据 Herlocker 等人 [15] 的研究结果, 在真实环境中最近邻用户数量设为 20–50 比较合理, 即 $r \in [20, 50]$. 若基于这 20–50 个最近邻的用户访问项序并不能得到 N 个最频繁项, 则可将 r 的值进一步放大, 直到完成 top-N 最频繁项集的搜寻.

4 实验结果及分析

4.1 实验环境、数据集及评价标准

实验所用 PC 机的配置为 Intel Pentium 4 2.66GHz CPU、1GB RAM, 操作系统是 Windows XP, 算法程序采用 PowerBuilder 9.0 实现, 数据库为 Access 2003.

实验基于 Gazelle Web 日志数据集完成. 通过对日志数据进行预处理, 提取了 1426 位访问量很少的用户对 207 种商品的 10053 条有效访问记录作为实验数据集, 时间跨度为 2 个月, 数据集稀疏度为 3.4%, 符合算法应用背景. 每条访问记录表示为一个三元组〈用户 ID, 商品 ID, 访问时间〉. 然后, 进一步将实验数据集分为训练集 (training set) 和测试集 (test set) 两个部分, 划分方法是将实验数据集中每个用户最后 10 天的访问记录隐藏起来作为测试集, 其余的访问记录作为训练集. 训练集用于建立用户访问项序并进行相似性计算, 以寻找新用户的最近邻集合并生成 top-N 推荐集 (top-N set); 测试集则用于测量 top-N 推荐质量, 即如果 top-N 推荐集中某个商品项出现在该用户测试集中, 则表示生成了一个正确推荐.

实验采用 F-measure[16] 作为 top-N 推荐质量的评价标准. F-measure 由信息检索领域广泛使用的召回率 ($recall$) 和准确率 ($precision$) 组成, 这是由于召回率和准确率实际上是相互矛盾的, 例如增加推荐集数目 N 能提高召回率, 但降低了准确率. 召回率、准确率、F-measure 的计算方法如分别如式 (20)、(21)、(22) 所示 (式中 N 表示 top-N 推荐总数):

$$recall = \frac{|test \bigcap \{top\text{-}N\}|}{|test|} \tag{20}$$

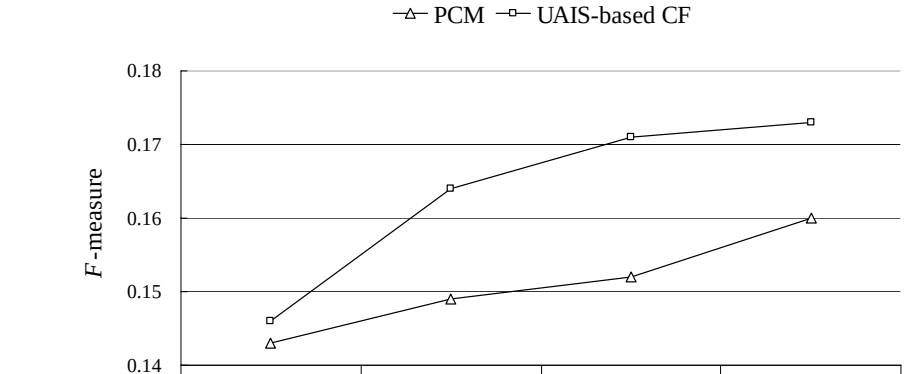
$$precision = \frac{|test \bigcap \{top\text{-}N\}|}{N} \tag{21}$$

$$F\text{-measure} = \frac{2 \times recall \times precision}{recall + precision} \tag{22}$$

4.2 实验结果及分析

实验对以下 2 种方法进行比较: ①基于路径聚类的 top-N 推荐方法 (记为 PCM), 即采用文献 [11] 提出的路径聚类方法对实验数据集进行处理并得到相应的 top-N 推荐结果; ②本文提出的基于 n 序访问解析逻

辑的 top-N 推荐方法 (记为 UAIS-based CF). 实验的一个重要参数是新用户的最近邻用户集合的大小, 根据 Herlocker 等 [15] 的研究结果, 在真实环境中最近邻用户数量设为 20–50 比较合理, 即 $r \in [20, 50]$. 因此, 本文实验分别在最近邻数量为 20、30、40、50 的情况下对 PCM、UAIS-based CF 进行 top-N 推荐生成和 F-measure 值计算, 得到的实验结果如图 1 所示.



	20	30	40	50
PCM	0.143	0.149	0.152	0.160
UAIS-based CF	0.146	0.164	0.171	0.173

Number of neighbors

图 1 top-N 推荐质量比较

上述实验结果表明, 本文提出的 UAIS-based CF 相对于 PCM 在 F-measure 指标上平均提高了 1.25%, 因此 UAIS-based CF 在 top-N 推荐质量上优于 PCM. UAIS-based CF 之所以能够提高 top-N 推荐质量, 是因为传统的基于用户浏览路径来进行 Web 用户分类的方法过于强调用户在路径上的一致性, 而忽略了用户 "行走" 这些路径想要到达的 "目的地" 页面, 从而导致了 "过匹配" 的现象, 使得本来浏览 "目的地" 页面相同的用户之间由于经过的路径不同 (这种不同正如前文所述, 是由于用户自身对网站的熟悉程度不一致、浏览方法不同等多种因素造成的), 将会被聚类到不同的用户组, 这样也就使得最终的 top-N 推荐质量不高. 本文提出的 n 序访问解析逻辑避免了传统思路所存在的 "过匹配" 缺陷, 同时也使得完成推荐服务的时间得到缩减, 有利于提高推荐实时性和产生更好的用户体验.

5 结论

随着电子商务站点用户和商品项数量的不断增长, 如何消除冷启动问题已成为电子商务推荐系统进一步发展所亟待解决的瓶颈问题之一. 本文的贡献在于, 针对协同过滤冷启动中最主要的新用户问题提出了一种基于 n 序访问解析逻辑的消除方法, 结合改进的最频繁项提取算法 IMIEA 来生成面向新用户的 top-N 推荐. 实验结果表明本文提出的新算法有效消除了新用户问题. 此外, 还有两点需要特别指出:

1) 新算法的拓展应用. 本文提出的新算法源于解决电子商务网站冷启动问题, 但实际上该算法完全可以稍作变化甚至不作变化而应用于数字图书馆、在线影音娱乐等热点领域的个性化推荐服务, 例如针对科研人员的文献推荐、针对学生的图书推荐、针对网络用户的影音推荐, 等等. 我们也正在尝试将新算法与所在高校的数字图书馆系统进行结合, 冀在实现面向学生的馆藏图书借阅个性化推荐功能.

2) 下一步研究内容. 目前, 本文的新算法已集成到我们的内部实验网站, 该实验网站的潜在商业用途是提供一个体育类图书影像资料的在线二手交易平台. 通过模拟交易测试, 一方面新算法表现出令人满意的推荐性能; 另一方面我们也发现新算法在用户访问项序的提取上还可进一步设计自动增量更新模型, 否则完全通过人工整理 Web 日志数据来形成用户访问项序的工作量过大, 会影响商业应用效果. 因此, 下一步的主要工作就是如何实现与新算法相匹配的用户访问项序自动增量更新模型. 此外, 能否在消除新用户问题的同时也解决冷启动中的新项目问题, 也是下一步工作中需要考虑的内容.

参考文献

[1] Schafer J B, Konstan J A, Riedl J. E-commerce recommendation applications[J]. Data Mining and Knowledge

Discovery, 2001, 5(1/2): 115–153.
[2] Deshpande M, Karypis G. Item-based top-N recommendation algorithms[J]. ACM Transactions on Information Systems, 2004, 22(1): 143–177.
[3] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]// 10th International Conference on World Wide Web, Hong Kong, 2001: 285–295.
[4] 李聪, 梁昌勇, 杨善林. 电子商务协同过滤稀疏性研究: 一个分类视角 [J]. 管理工程学报, 2011, 25(1): 94–101.
Li C, Liang C Y, Yang S L. Sparsity problem in collaborative filtering: A classification[J]. Journal of Industrial Engineering and Engineering Management, 2011, 25(1): 94–101.
[5] Park S T, Pennock D, Madani O, et al. Naïve filterbots for robust cold-start recommendations[C]// 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, 2006: 699–705.
[6] Good N, Schafer J B, Konstan J A, et al. Combining collaborative filtering with personal agents for better recommendations[C]// 16th National Conference on Artificial Intelligence, Orlando, 1999: 439–446.
[7] Claypool M, Gokhale A, Miranda T. Combining content-based and collaborative filters in an online newspaper[C]// ACM SIGIR Workshop on Recommender Systems, Berkeley, 1999.
[8] 吴丽花, 刘鲁. 个性化推荐系统用户建模技术综述 [J]. 情报学报, 2006, 25(1): 55–62.
Wu L H, Liu L. User profiling for personalized recommending systems — A review[J]. Journal of the China Society for Scientific and Technical Information, 2006, 25(1): 55–62.
[9] Mobasher B, Cooley R, Srivastava J. Creating adaptive web sites through usage–based clustering of URLs[C]// The 1999 Workshop on Knowledge and Data Engineering Exchange, Chicago, 1999: 19–25.
[10] Nasraoui O, Frigui H, Joshi A, et al. Mining web access logs using relational competitive fuzzy clustering[C]// 8th International Fuzzy Systems Association World Congress, Taipei, 1999.
[11] 王实, 高文, 李锦涛, 等. 路径聚类: 在 Web 站点中的知识发现 [J]. 计算机研究与发展, 2001, 38(4): 482–486.
Wang S, Gao W, Li J T, et al. Path clustering: Discovering the knowledge in the web site[J]. Journal of Computer Research and Development, 2001, 38(4): 482–486.
[12] Büchner A G, Mulvenna M D. Discovering internet marketing intelligence through online analytical web usage mining[J]. ACM SIGMOD Record, 1998, 27(4): 54–61.
[13] 史忠植. 知识发现 [M]. 北京: 清华大学出版社, 2002.
Shi Z Z. Knowledge Discovery[M]. Beijing: Tsinghua University Press, 2002.
[14] Levenshtein V I. Binary codes capable of correcting deletions, insertions, and reversals[J]. Soviet Physics-Doklady, 1966, 10(8): 707–710.
[15] 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.
[16] Liu D R, Lai C H, Lee W J. A hybrid of sequential rules and collaborative filtering for product recommendation[J]. Information Sciences, 2009, 179(20): 3505–3519.