

# 一种基于协同推荐的网页排序算法

陈 兰<sup>1</sup> 金远平<sup>2</sup>

(<sup>1</sup>东南大学计算机科学与工程学院, 南京 210096)  
(<sup>2</sup>东南大学软件学院, 南京 210096)

**摘要:** 针对目前搜索结果个性化排序算法中的用户兴趣模型构建难、相关度计算不精确等问题, 提出了一种结合用户兴趣模型和协同推荐算法的个性化排序方法. 该方法从用户的搜索历史, 包括提交查询、点击相关网页等反馈信息来训练用户的兴趣模型, 然后采用协同推荐算法获取具有共同兴趣的邻居用户, 根据这些邻居对网页的推荐程度和网页与用户的相关程度来排序搜索结果. 实验结果表明: 该排序算法的平均最小精确度比一般排序算法提高了约 0.1, 且随着用户邻居数目的增长, 最小精确度随之增长. 与其他排序算法相比, 采用协同推荐算法有助于提高网页与用户兴趣关联程度计算的精确度, 从而提高排序的效率, 有助于改善用户的搜索体验.

**关键词:** 个性化; 排序算法; 协同推荐; 用户模型

**中图分类号:** TP31      **文献标志码:** A      **文章编号:** 1001-0505(2009)03-0453-06

## Web page ranking algorithm based on collaborative recommendation

Chen Lan      Jin Yuanping

(<sup>1</sup>School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)  
(<sup>2</sup>College of Software Engineering, Southeast University, Nanjing 210096, China)

**Abstract** To cope with the limitations in current personalized ranking algorithms for Web search results such as an effective user profile is difficult to build and relevance computing is not precise, a new personalized rank method based on both user profile and a collaborative recommendation algorithm is proposed. First, user profiles are built from the search history including the queries committed and the pages users clicked, and then the neighbors with similar interests to those of the user are acquired. The resulting pages are re-ranked according to the user's interests in the pages and the recommendations of the page from the neighborhood. The evaluation experiment results show that the method can increase the MAE (minimum accuracy) by 0.1, and the bigger the size of the neighborhood, the higher the accuracy. Compared with other ranking algorithms, collaborative ranking can improve the relevance precision, providing an improvement in ranking efficiency and user's search experience.

**Key words:** personalization; rank algorithm; collaborative recommendation; user profile

随着网络信息的快速增长, 在大量信息中快速、有效地找到想要的内容已经成为互联网技术一个非常重要的研究课题. 目前, 各类搜索引擎已经被广泛地用于为用户提供高效的搜索服务.

然而, 网络信息的剧增, 搜索引擎动辄返回成千上万的结果, 而其中只有很少的信息是与用户需求紧密相关的, 很难满足用户高效的、个性化的搜索需求. 因此, 适应用户的兴趣和背景的个性化搜

索引擎逐渐成为搜索引擎发展方向之一.

目前, 对搜索引擎个性化方法的研究主要分为 2 个方向: ① 查询扩展, 通过对查询的个性化来达到个性化的需求; ② 个性化排序, 通过对搜索结果的过滤来体现用户的个性化. 而其中个性化排序越来越受到人们的关注. 起初, 一种个性化的排序算法的思想是根据专家对网页的相关性进行判断, 而这种方法需要人工的参与, 效率低且对个性化的

支持也不好.文献[1]使用了个性化的主题搜索,但是该方法没有考虑到用户的背景,因此对用户的搜索需求的描述不清晰.文献[2]使用了本体的相关技术来表示用户的兴趣,本体可以较好地描述用户的兴趣,然而本体的建立本身仍是一个未解决的问题.最近,通过对用户的隐式反馈来训练和学习排序函数或算法的研究吸引了很多人的关注<sup>[3]</sup>,基于隐式反馈虽然可以方便地获得用户的兴趣信息,但其精确性不高,因此排序的效果并不理想.

现有的个性化排序算法存在着各自的缺点,本文在隐式反馈的基础上结合协同推荐的思想来改进排序算法.一方面,隐式反馈获取用户兴趣很方便且代价很小;另一方面,结合协同推荐可以提高排序的精确度和效率.所谓协同推荐是指通过相同或相近兴趣的用户对资源的评价向用户推荐信息的方式.如果在搜索引擎中用户也使用类似的方式来共享信息,那么这将为用户搜索感兴趣的网页带来便利<sup>[4]</sup>.这是因为具有相似兴趣的用户对同一网页的兴趣很可能也是相似的,那么其他用户对网页的兴趣度一定程度上反映了活动用户对网页的兴趣度.根据这些共同兴趣的用户对网页的推荐或预测来排列当前用户搜索到的网页就可以将与用户相关的网页排在前面,减少用户再搜索的次数,缩短用户浏览的时间.

1 协同推荐

协同推荐技术是推荐系统中应用最为广泛的技术之一.协同推荐算法可分为两大类:基于模型的算法<sup>[5]</sup>和基于内存的算法.

1)基于模型的算法 首先利用一些机器学习算法(如贝叶斯网、聚类和基于规则的方法)建立用户评价模型,然后利用该模型进行预测<sup>[6]</sup>.这种算法的优点是能提高预测时的性能,缺点是模型训练比较费时,所以该算法不太适合于数据更新频繁的系统,在实际应用中使用并不广泛.

2)基于内存的算法 也称为最近邻推荐,可分为 User-based和 Item-based两种推荐算法. User-based推荐算法的核心思想是通过比较当前用户与其他用户兴趣文件的相似度计算出用户间的相似度,生成与当前用户行为兴趣最相近的用户集,将其最感兴趣的项作为当前用户的推荐结果. User-based的推荐是与项的内容无关的. Item-based推荐算法<sup>[7]</sup>是根据用户对相似项的评分,预测该用户对目标项的评分.如果大部分用户对一些项的评分比较相似,则当前用户对这些项的评分也比较

相似.采用统计技术找到目标项的若干最近邻居,根据当前用户对最近邻居的评分预测当前用户对目标项的评分,据此产生推荐.

基于模型的推荐算法难以适应信息更新频繁的互联网;而 Item-based协同推荐并没有解决稀疏性的问题,研究表明 Item-based的推荐在项目稳定的情况下,其性能的优势才会体现出来,因此传统的 Item-based在解决可扩展性问题上的优势也并不明显.随着研究的深入,更多的研究倾向于利用 Item的其他信息如属性信息来提高性能.然而在搜索引擎的背景下,Item对应的是兴趣关键词,而对于关键词并无相关属性信息可以利用.与电子商务的项相比较,兴趣的关键词的变化较大,不利于使用 Item-based推荐.综合权衡利弊,本文采用 User-based的协同推荐技术.

1.1 基于最近邻的协同推荐

最近邻协同推荐算法分为构建用户档案、寻找最近邻、根据预测算法产生推荐 3个步骤.

① 构建用户档案.收集用户的评分、评价行为等,并进行数据清理、转换和录入,最终形成用户对各种项目的评价矩阵.分数越高,用户对该项目的认可度越高.

② 寻找最近邻.在这一阶段,计算目标用户与数据库内各个用户的相似度,寻找相似度最高的作为最近邻居集.一般来说,可采用 Pearson相关度公式、Cosine相关度公式、修正的 Pearson公式等,其中 Cosine相关度公式为

$$S_{ij} = P_a \cdot P_j = \frac{\sum_{i=1}^n w_{ia} w_{ij}}{\sqrt{\sum_{i=1}^n w_{ia}^2} \sqrt{\sum_{i=1}^n w_{ij}^2}} \quad (1)$$

式中,  $S_{ij}$ 为用户  $a$ 和用户  $j$ 的相似度;  $P_a$ 为用户  $a$ 的兴趣模板;  $P_j$ 为用户  $j$ 的兴趣模板;  $w_{ia}$ 为用户  $a$ 的兴趣模板  $P_a$ 的第  $i$ 个权重;  $w_{ij}$ 为用户  $j$ 的兴趣模板  $P_j$ 的第  $i$ 个权重;  $n$ 为兴趣模板向量大小.

③ 预测阶段.通过最近邻居集产生推荐,一般来说有权相加算法和回归方法.其中用得比较多的是权相加法,即

$$R_{ij} = \frac{\sum_{u \in N} (S_{ui} R_{uj})}{\sum_{u \in N} (|S_{ui}|)} \quad (2)$$

式中,  $a$ 为活动用户;  $R_{ij}$ 为用户对项目  $j$ 的评分;  $S_{ui}$ 为用户  $i$ 和用户  $u$ 之间的相似度,如步骤②所述;  $N$ 为邻居数;  $R_{uj}$ 为用户  $u$ 对项目  $j$ 的评分.从式(2)中可以看出,与当前用户越相似的用户对资源的评

价影响越大.

### 1.2 存在的问题及解决的方案

虽然基于最近邻的协同推荐算法得到了广泛的应用,但在实际的应用中常会遇到如下问题:

1)稀疏性 系统中用户和资源数目非常多,而每个用户评价过的资源数目有限.尤其是在系统使用初期,由于系统资源还未获得足够多的评价,评分矩阵会非常稀疏,因此该方法很难利用这些评价来发现相似的用户.

2)可扩展性 随着系统用户和资源的增多,该方法计算的代价会越来越大,性能会越来越低.

3)新用户(资源)问题 新用户未对任何资源评价,无法找到该新用户的邻居,因而无法预测其对资源的评价,不能生成对该用户的推荐.同样,没有用户对新资源进行过评价,因此无法对任何用户推荐该资源,只有等到有用户自己对该资源评价之后才能推荐给与该用户相似的用户.

目前针对这些问题的研究有很多,对稀疏性主要通过对没有评分的项目赋予默认值或用户评分平均值,来获得一个稠密矩阵的方法,通过神经网络、贝叶斯等方法以及潜在语义索引相关技术来改善稀疏性.

而对可扩展性,主要包括矩阵降维、用户或项目聚类等方法来减小矩阵的计算量.对于新用户可采用结合基于内容的推荐,即利用浏览过的内容来预测用户的评分信息.

## 2 基于协同推荐的排序算法

### 2.1 文档和用户模型的表示

为了利用协同推荐来重排序搜索结果,以达到提高相关度的目标,算法必须考虑协同推荐中的关键数据的表示,如用户和文档等.只有清晰地表达出用户的兴趣和文档所包含的信息,才可以根据用户对文档的兴趣度来排列文档,因此一个合适的数据结构来表示这些信息对排序算法至关重要.

#### 2.1.1 文档的表示

目前能够较好地描述页面文档的模型有布尔模型、向量空间模型和概率模型等.其中向量空间模型(vector space model)应用最为广泛.传统的向量空间模型文档表示方法的步骤为:

- ① 选取一组适用于文档的关键词  $\{k_1, k_2, \dots, k_n\}$ ;
- ② 根据关键词  $k_i$  在文档中的重要程度求出其对应的权值  $w_i, i=1, 2, \dots, n$ ;
- ③ 将文档用关键词向量  $\{w_1, w_2, \dots, w_n\}$  表

示.

因此,每个文档可由加权关键词向量组  $D = \{(k_1, w_1), (k_2, w_2), \dots, (k_n, w_n)\}$  表示.其中  $D$  为某页面文档,  $k$  表示  $D$  中出现的某个关键词.当  $w_i$  越大时,表示关键词  $k_i$  的重要程度越高.

显然,在页面文档的特征提取中,关键词权值  $w_i$  的确定是重点<sup>[8]</sup>.目前,使用最广泛获得页面特征的方法是词频法,即根据不同关键词在文档中出现的频率来确定其权重,本文采用的是 TF-IDF 关键词权重的量化方法.

#### 2.1.2 用户模型的表示

每个用户都会有其个性化的兴趣(user Profile).只有充分理解用户的个性化兴趣才能更好地为每个用户服务.用户兴趣通常也可以由网页文档中得到的关键词集合和权值的集合组成,在传统的用户兴趣表示模型中,用户兴趣通常可以表示为  $P = \{(k_1, w_1), (k_2, w_2), \dots, (k_n, w_n)\}$ ,其中  $P$  表示用户兴趣,  $k$  表示用户感兴趣的某个关键词.关键词的权值  $w_i$  通常通过用户所点击网页的权值来确定,而网页的权值则是通过 TF-IDF 等方式获得的<sup>[9]</sup>.在获得所有用户的模型后,将这些模型做规范化处理以方便后面的计算.

然而,用户的兴趣是多样的,一个兴趣模板不能表达一个人不同的兴趣<sup>[10]</sup>.因此,在此基础上,将用户模型表示为多个子兴趣模板,而兴趣的分类是根据一个约定的分类标准进行.即一个用户模型可以表示为  $U = \{P_1, P_2, \dots, P_n\}$ ,其中  $n$  为兴趣分类的个数.每一个子兴趣模板都是根据属于这一类兴趣的查询和相关的文档等按照上述的方法获得.这里使用 SVM 来训练获得不同的兴趣.

### 2.2 协同推荐的个性化排序

与协同推荐的过程类似,本文把整个排序过程分为构建用户兴趣档案、最近邻居用户生成和结果排序 3 个阶段.

#### 1) 构建用户兴趣档案

如表 1 所示,每一个子兴趣类别分别建立一个 User Item 矩阵.从所有拥有该兴趣类别的用户的子兴趣模板中(假设共有  $m$  个用户)选择  $n$  个兴趣关键词,建立 User Item 矩阵  $R_{m \times n}$ .在矩阵  $R_{m \times n}$  中,每一行向量  $R[i, j]$  表示用户的兴趣,而每一列  $R[:, j]$  则表示不同的用户对同一个 Item <sub>$j$</sub>  的兴趣,元素  $R_i$  表示用户  $i$  对 Item <sub>$j$</sub>  的兴趣程度.

如前所述,矩阵建立初期可能会出现稀疏性、可扩展性及冷启动问题.由于采用用户隐式反馈的方法来建立用户的兴趣模型,不会存在用户对所有

表 1 用户兴趣档案

User	Item			
	1	2	...	n
1	3			2
2		4		
⋮				
a	2 5			3 5
⋮				
n		4		

关键词的兴趣度都为空的情况,因此可以忽略冷启动的问题.对可扩展性的问题,通过事先对用户的兴趣进行分类,对每一类兴趣建立矩阵来降低矩阵的维数来缓解.随着分类的细化,降维的效果会更好.而矩阵的稀疏性方面,采用奇异值分解(SVD)<sup>[11]</sup>来预测用户对未评分关键词的评分.

SVD是一种矩阵分解技术,它是一种有效的代数特征提取方法.它可将一个  $m \times n$  (假设  $m \geq n$ ) 的矩阵  $R$  分解为 3 个矩阵,即  $R = U \times S \times V'$ .其中,  $U$  为一个  $m \times m$  的正交矩阵,  $V$  为一个  $n \times n$  的正交矩阵 ( $VV' = I$ ),  $S$  为一个  $n \times n$  矩阵,其非对角线上的元素全为 0 而对角线上的元素满足  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$  称为奇异值,奇异值可以表示一个给定矩阵与比其秩低的矩阵的接近程度.

根据矩阵的单值分解原理,  $S$  可以被降维到  $k$  阶 ( $k < n$ ) 得到  $S_k$  同理可以得到  $U_k, V_k$  于是得到与矩阵  $R$  相似的低维矩阵  $R_k$  即有  $R_k = U_k \times S_k \times V_k$

根据文献[11],用户  $a$  对未评分的关键词的评分预测采用如下式来计算:

$$R_{a,i} = R_a + U_k \cdot \sqrt{S_k}(a) \cdot \sqrt{S_k}V_k'(i) \tag{3}$$

式中,  $R_a$  为用户  $a$  的评分的平均值;  $U_k \cdot \sqrt{S_k}(a)$  为  $U_k \cdot \sqrt{S_k}$  的第  $a$  行;而  $\sqrt{S_k}V_k'(i)$  为  $\sqrt{S_k}V_k'$  的第  $i$  列.

Sarwar等<sup>[11]</sup>的试验结果表明奇异值分解方法应用于协同过滤时对于稀疏的评分矩阵效果比较好,而且有一定的可扩展性.

2)最近邻居用户生成及推荐生成

用户的邻居按照下面的方法计算:① 将用户的查询映射到一个子类中;② 取出该子类的 User-Item 矩阵,然后对矩阵中的所有用户(除自身以外)计算其与目标用户的相似程度即信任度.相似度的计算可以参考协同推荐系统的计算方法.本文采用余弦算法.

用  $S_{j,c}$  表示子类  $c$  中的用户  $j$  和  $a$  的信任度,并采用修正的 Pearson 相关系数度量来计算,即

$$S_{j,c} = \frac{\sum_{i \in I_{ij}} (R_{i,n} - R_i)(R_{i,n} - R_j)}{\sqrt{\sum_{i \in I_{ij}} (R_{i,n} - R_i)^2} \sqrt{\sum_{i \in I_{ij}} (R_{i,n} - R_j)^2}} \tag{4}$$

式中,  $I_{ij}$  为用户  $j, i$  共同评分的关键词;  $R_i$  和  $R_j$  为用户  $j, i$  的评分的平均值.最后将相似度存储下来,然后选择 TOP-N 用户,作为活动用户的最近邻居.

3)结果排序

对文档 Rank 值的计算是根据用户对该文档的兴趣度,以及最近邻用户对该文档的评分来计算的.当前用户对该文档的兴趣度是根据用户的模板和文档的余弦来计算.最近邻评分加权求和的计算方法参考协同推荐的预测阶段的算法.最后的计算如下式所示:

$$R_{a,d,c} = \alpha (P_{a,c} D) + (1 - \alpha) \cdot \frac{\sum_{u \in TOP-K} S_{u,c} (P_{u,c} D)}{\sum_{u \in TOP-K} |S_{u,c}|} \tag{5}$$

式中,  $R_{a,d,c}$  为活动用户  $a$  对属于子兴趣类别  $c$  的文档  $D$  的排序值;  $P_{a,c}, P_{u,c}$  为用户的子兴趣模板;  $D$  为待排序文档,  $D = \{w_1, w_2, \dots, w_n\}$ ;  $\alpha$  ( $0 \leq \alpha \leq 1$ ) 为调节因子,可以根据用户对邻居用户推荐的接受程度来调节,当  $\alpha = 0$  时,表示用户完全根据邻居的推荐来排序,  $\alpha = 1$  时表示完全不接受邻居用户的推荐;  $S_{u,c}$  为用户  $a$  和用户  $u$  的相似程度,即信任度.

在邻居用户的推荐效果不佳时,更多用户可能偏向于传统的排序算法.考虑到这种情况,本文在传统排序算法上融入协同推荐排序,即

$$Rank(a,d,c) = g \times PageRank(a,d) + h R_{a,d,c} \tag{6}$$

式中,  $PageRank(a,d) = (1 - d) + \sum_{i=1}^n \frac{PR(T_i)}{C(T_i)}$ ;

$d$  为阻尼系数,一般取 0.85,  $g, h$  为调节系数;  $PageRank$  为著名的 Google 提出的排序算法.

计算所有搜索到文档的  $Rank(a,d,c)$  的值,然后从大到小的顺序排列.

2.3 算法描述

由于本文的算法不同于其他方法,主要体现在用户对网页兴趣度的计算上,因此给出该方法的核心部分,而忽略查询匹配的过程.本算法是对搜索引擎的搜索结果的重排序,因此输入的待排序网页集合是带有  $PageRank$  值的网页集合.

输入: 用户  $a$  User-Item 矩阵、待排序的网页集合、用户相似度矩阵.

输出: 排好序的网页集合.

步骤:

① 对网页集合中的各个网页, 按照 2.1 节中提出的文档表示方法, 给出各个网页的权重表示, 得到一个新的集合  $D$

② 若集合  $D$  还有未计算  $Rank$  值的文档, 那么从网页集合  $D$  中取出一个网页  $d$  根据余弦公式计算文档与用户的相似程度  $P_{dc}$ .  $D$  否则至步骤⑥.

③ 读取用户相似度矩阵, 计算出与用户最相近的  $N$  个邻居用户. 若用户相似度矩阵为空值, 那么根据式 (4) 计算相似度, 得到最相近的  $N$  个邻居, 并将相似度存入用户相似度矩阵.

④ 对最近邻居集合中的用户, 计算其对文档  $d$  的推荐度, 并按照式 (5) 计算所有用户对  $d$  的推荐度, 其中的  $P_{dc}$  在②中已经获得.

⑤ 按照式 (6) 计算文档  $d$  的最后  $Rank$  值, 将集合  $D$  的  $d$  标为已计算  $Rank$  值. 式 (6) 中的  $PageRank$  作为已知条件给出.

⑥ 按照  $Rank$  值从大到小排列文档, 输出结果.

### 3 实验及分析

本文没有对查询和相关文档分类, 而是手动选取了其中某一类别的用户及相关信息. 这是因为对某一查询的结果页面排序只受某一类查询及相关页面的影响, 实验的目的在于测试协同推荐对排序的准确性的影响, 因此可以将实验简单推广到各个分类中.

#### 3.1 测试数据

训练阶段, 实验模拟每个用户从互联网上提交了约 300 个 `java` 编程相关查询, 每一个查询收集了 TOP-50 范围内的网页. 针对每个用户, 取出查询相关的网页, 建立一个用户兴趣模型. 重复 50 次模拟过程, 从而建立一个对 `java` 编程感兴趣的小型社区. 然后根据文中提出的方法建立 `User-Item` 矩阵, 根据矩阵预先算出一部分用户之间的相似度. 测试阶段, 采用类似训练阶段的方法采集数据, 并任选用户为活动用户, 利用 Google 的 `PageRank` 查询接口, 得出搜索结果中各个页面的 `PageRank` 值, 并根据所描述的核心算法计算本文算法的排序结果.

#### 3.2 分析

采用最小精确度<sup>[11]</sup>来衡量本文的排序算法, 最小精确度衡量的是返回结果中至少含有一个相关文档的能力. 用搜索结果的前 20 个结果来计算

最小精确度. 比较结果如图 1 所示, 从中可以看出, 基于协同推荐的排序算法可以很好地改善排序, 从而提高用户的搜索体验. 同时, 实验考虑了最近邻用户的个数对协同排序的影响, 从图 2 可以看出, 随着用户数的增加, 协同推荐的精确度将越来越好.

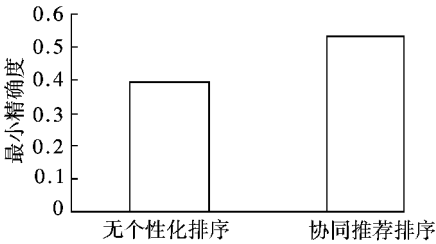


图 1 排序算法比较

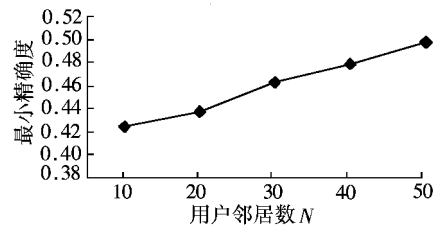


图 2 最近邻用户数的影响

为了更好地分析和比较本文提出的个性化排序算法与其他算法排序效果, 这里采用了 `Average Rank` 作为衡量的标准. 对于一个查询  $s$ , 其 `AvgRank` 的定义为  $\sum_{i \in P_s} R(i) \frac{1}{|P_s|}$ , 其中  $P_s$  是查询对应的页面集合,  $R(i)$  为页面  $i$  的 `Rank` 值. 最终的 `AvgRank` 为  $\sum_{s \in S} \text{AvgRank}_s \frac{1}{|S|}$ . 无个性化排序的 `Web` 算法、`PClick`<sup>[12]</sup> 算法和本文的算法在不同的查询集合比较的结果如图 3 所示.

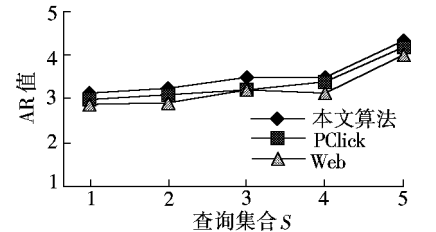


图 3 各算法效率比较

### 4 结语

本文在协同推荐算法的基础上, 提出了一种结合用户兴趣的个性化排序算法. 实验表明算法可以提高搜索结果的排序效果, 改善用户的搜索体验. 然而算法初期, 可能存在的稀疏性、可扩展性等问题影响了算法的效率. 虽然本文提出了改进的方法, 但是这些方法还不是很完善, 不能兼顾所有问

题的解决, 因此一个更加有效的解决稀疏性和可扩展性等问题的推荐算法是进一步研究的方向.

参考文献 (References)

[ 1 ] Haveliwala T H. Topic sensitive pagerank [ C ] //Proceedings of the 11 th International Conference on World WideWeb. Hawaii, USA: ACM Press, 2002: 517-526.

[ 2 ] Tsai W ang-Tien. Building personal ontology to assist Web usage mining using personalized ranking as an example [ D ]. Taiwan, China: Department of Information Management, Chaoyang University of Technology, 2007.

[ 3 ] Agichtein E, Brill E, Dumais S, et al. Learning user interaction models for predicting Web search result preferences [ C ] //Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR). Washington, USA: ACM Press, 2006: 3-10.

[ 4 ] Chidlovskii B, Gance N S, Grasso A. Collaborative ranking of search results [ C ] //Proceedings of AAAI 2000 Workshop on AI for Web Search. Austin, USA, 2000: 1-5.

[ 5 ] Adomavicius G, Tuzhilin A. Towards the next generation of recommender systems: a survey of the state of the art and possible extensions [ J ]. IEEE Trans Knowl Data Eng, 2005, 17(6): 734-749.

[ 6 ] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering [ C ] //Proceedings of the 14 th Conference on Uncertainty in

Artificial Intelligence. Madison, Morgan Kaufmann Publisher, 1998: 43-52.

[ 7 ] Deshpande M, Karypis G. Item-based top-n recommendation algorithms [ J ]. ACM Trans Information Systems, 2004, 22(1): 143-177.

[ 8 ] Rohini U, Ambati V. A collaborative filtering based ranking strategy for search in digital libraries [ C ] //Proceedings of 8 th CADL, Bangkok, Thailand, 2005: 194-203.

[ 9 ] Radlinski F, Joachims T. Query chains: learning to rank from implicit feedback [ C ] //Proceedings of the 11 th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, USA, 2005: 239-248.

[ 10 ] Ziegler C N, McNeel S M, Konstan J A, et al. Improving recommendation lists through topic diversification [ C ] //Proceedings of the 14 th International Conference on World WideWeb, Chiba, Japan, 2005: 22-32.

[ 11 ] Sarwar B M, Karypis G. Application of dimensionality reduction in recommender systems: a case study [ C ] //Proceedings of ACM Web KDD Workshop on Web Mining for E-Commerce, New York: ACM Press, 2000: 114-121.

[ 12 ] Dou Z, Song R, Wen J R. A large scale evaluation and analysis of personalized search strategies [ C ] //Proceeding of the 16 th International Conference on World WideWeb, Banff, Canada: ACM Press, 2007: 581-590.