

# 基于 WWW 缓冲的用户实时二维兴趣模型

张卫丰<sup>1),2),3)</sup> 徐宝文<sup>1),3),4)</sup>

<sup>1)</sup>(东南大学计算机科学与工程系 南京 210096)

<sup>2)</sup>(南京邮电学院计算机科学与技术系 南京 210003)

<sup>3)</sup>(江苏省软件质量研究所 南京 210096)

<sup>4)</sup>(武汉大学软件工程国家重点实验室 武汉 430072)

**摘要** WWW 缓冲技术通过将受欢迎的网页放到与客户较近的地方来提高用户存取这些网页的速度。如何有效充分地利用 WWW 缓冲中的信息,其关键是建立一个合适的用户兴趣模型和构造合适兴趣挖掘算法。简单兴趣模型通过(词条,权重)来刻画兴趣。它没有深入挖掘这些兴趣之间的关联关系,因而在表达用户兴趣的时候,不能实现兴趣之间的关联。该文在充分分析 WWW 缓冲模型的基础上提出了实时二维兴趣模型。该模型的实时性可以保证挖掘出来的用户兴趣更能反映当前用户的兴趣状态;该模型引入的二维概念充分地考虑了用户兴趣之间的递推关系。该模型不是简单兴趣模型的简单扩充,而是模型和相关算法的全面改进。文章给出了二维兴趣模型的存储、二维兴趣的有效计算和二维兴趣的实时更新的相关方法。

**关键词** WWW; 互联网; 兴趣模型; 数据挖掘; 高速缓存

**中图法分类号** TP311

## WWW Cache Based Model of Users' Real Time Two-Dimensions Interest

ZHANG Wei-Feng<sup>1),2),3)</sup> XU Bao-Wen<sup>1),3),4)</sup>

<sup>1)</sup>(Department of Computer Science and Engineering, Southeast University, Nanjing 210096)

<sup>2)</sup>(Department of Computer Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003)

<sup>3)</sup>(Jiangsu Institute of Software Quality, Nanjing 210096)

<sup>4)</sup>(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)

**Abstract** The popular WWW pages are stored in the users' places. By this WWW Cache technology, the browsers can fetch these pages more rapidly. The information in the WWW Cache shows the users' recent interest. The users' interest can be widely used, for example, customizing the WWW pages, filtering the information, pre-fetching the information, and so on. How to use the information in the WWW Cache effectively lies in how to build an adaptive user interest model and how to construct an adaptive algorithm for interest mining. In simple interest model, the interest can be specialized by a tuple (term, weight), and the association relations are not mined, so the interest cannot be associated when expressing the users' interest. Based on analyzing the WWW Cache model, we bring forward a real time two-dimensional interest model. The property of real time in this model can show the users' current interest states. And the inferential relations between interests are well considered in the model. This model is not the simple extension of the simple interest model, but the round improvement of the model and its related algorithm. In this model, we use rough set method to store the data more effectively, and we use in-

收稿日期: 2001-11-16; 修改稿收到日期: 2003-05-28. 本课题得到国家自然科学基金(60073012)、国家“九七三”重点基础研究发展计划项目基金(2002CB312000)、国家预研基金、江苏省自然科学基金(BK2001004)、江苏省科技攻关项目基金(BE2001025)、教育部跨世纪优秀人才基金、教育部博士点基金、江苏省三三三人才基金、高等学校重点实验室访问学者基金、武汉大学软件工程国家重点实验室开放基金、南京大学软件新技术国家重点实验室基金、苏州大学江苏省计算机信息处理技术重点实验室基金等资助。张卫丰,男,1975 年生,博士,主要从事程序设计语言、软件体系结构、搜索引擎、网络语言等方面的研究。E-mail: wfbreezee@hotmail.com. 徐宝文,男,1961 年生,博士,教授,博士生导师,主要从事程序设计语言、软件工程、知识与信息获取技术等方向的教学与科研工作。

cremental algorithm to compute the interest effectively and to update the interest in real time.

**Keywords** WWW; Internet; Interest model; data mining; cache

## 1 引言

随着 WWW 应用的迅速发展,网上信息越来越多。权威资料表明,WWW 上静态网页的数量正以每个月大约 15% 的速度增长。当今 WWW 用户常常受到网络拥塞和服务器过载的困扰。虽然主干网的速度以每年平均 60% 的速度增长,但是随着其它应用系统不断地向 WWW 应用转化以及 WWW 应用用户群的不断增加,用户对带宽的需求还是远远超过了网络带宽的增长速度<sup>[1]</sup>。若没有相应解决措施,WWW 将出现大量拥塞,以致用户的请求大量丢失。为此,早在 1990 年代初,就有许多学者致力于 WWW 性能的研究<sup>[1~4]</sup>。其中 WWW 缓冲(cache)技术就是在一定带宽限制下提高 WWW 性能的一种重要手段,WWW 缓冲技术通过将受欢迎的网页放到与客户较近的地方来提高用户存取这些对象的速度,目前该技术一般应用在浏览器和 WWW 代理中。WWW 缓冲中存放的是用户访问过的歷史信息,而且传统的 WWW 缓冲调度方法(如 LRU 和 LFU<sup>[5]</sup>等)都是将用户最近访问而且使用频率较高的网页存放在 WWW 缓存中<sup>[5]</sup>。这些缓存中的历史信息代表了用户最近的兴趣状况。用户的兴趣信息可以被广泛应用于定制网页、信息过滤和信息预取等<sup>[2,3,6,7]</sup>。关于如何挖掘网页的特征信息,并生成用户的兴趣,已经有很多这方面的研究<sup>[8]</sup>,这些兴趣模型主要通过(词条, 权重)来刻划某个兴趣<sup>[8]</sup>,我们称之为简单的兴趣模型,它没有深入挖掘这些兴趣之间的关联关系,因而在表达用户兴趣的时候,不能实现兴趣之间的联想。然而,根据用户知识的构成特点,知识之间应该是相互关联的。因此简单兴趣模型不能充分贴切地表示用户的兴趣。

为了解决简单兴趣模型在表达用户兴趣方面存在的问题,本文提出了实时二维兴趣模型。该模型不是简单兴趣模型的简单扩充,而是对简单兴趣模型和相关算法的全面改进,所引入的二维概念充分地考虑了用户兴趣之间的递推关系,其实时性可以保证所挖掘出来的用户兴趣更能反映用户当前的兴趣状态。和简单兴趣模型相比,实时二维兴趣模型主要要解决二维兴趣的有效存储、计算和实时更新等几

个方面的问题。

本文将首先介绍简单兴趣模型及其挖掘方法,然后分两个部分分别分析和讨论实时二维兴趣模型的存储问题以及用户实时兴趣的有效计算方法,最后,针对缓冲调度算法的特点,给出缓冲中网页内容调整所涉及到的兴趣更新的兴趣修整算法。

## 2 简单兴趣模型及其数据挖掘

简单兴趣模型是一种描述用户兴趣的简单方法,它将用户兴趣用二元组(兴趣词条, 兴趣权重)表示。若干个兴趣的集合构成了兴趣集,所有兴趣的集合构成兴趣全集(词典);兴趣全集 T 表示为  $\{t_1, t_2, \dots, t_m\}$ ,其中  $t_1, t_2, \dots, t_m$  分别表示兴趣(词条), $m$  为词典的大小。在兴趣集的基础上,为了便于下文讨论,这里首先给出一些基本概念。

**定义 1.** 兴趣结点是二元组  $(t, weight)$ ,简记为  $Node(t)$ ,其中  $t \in T$ ,  $weight$  为兴趣词条  $t$  的权重。

WWW 缓冲的网页反映了用户最近的兴趣状态,这些兴趣可以按照一定的方式转换成兴趣结点,在介绍如何将 WWW 缓冲的用户兴趣信息转换成兴趣结点之前有必要对 WWW 缓存中的网页进行一定的描述。WWW 缓存中的文本集合表示为  $\{d_1, d_2, \dots, d_n\}$ ,记为  $D$ 。对于词典中的词条  $t_i$ ,它在文本  $d_j$  中的出现频率(简称为词频)记为  $tf_{ij}$ ;它在整个  $D$  中出现的频率(在一个文本中不管出现多少次均记为 1)次数记为  $df_i$ (简称为文本频率), $df_i$  的倒数称为反转文本频率,记为  $idf_i$ 。

对简单兴趣模型而言,构造该模型的重要一环就是计算兴趣结点的权重。简单兴趣模型的数据挖掘主要是基于文本的数据挖掘<sup>[6,9]</sup>。在其兴趣计算方法中,将  $D$  中的所有文本看成超文本,将该词在该超文本中的词频作为该兴趣的权重<sup>[6,7,9~11]</sup>。这种方法比较简单,因而不可避免地存在着一些问题,比如,如果一个兴趣词条的文本频率比较大的话,那么该词条就不能明确地区分用户的兴趣。因此,我们认为,对于简单兴趣模型,在计算兴趣权重时应该综合考虑词条频率和文本频率,即对于兴趣结点  $Node(t_i)$ ,其兴趣权重应用下式计算:

$$\text{Node}(t_i).weight = idf_i \sum_{j=1}^n tf_{ij},$$

其中,  $idf_i$  为  $t_i$  在  $D$  中的反转文本频率,  $tf_{ij}$  为词条  $t_i$  在文本  $d_j$  中的词频,  $n$  为  $D$  中文本个数.

对简单兴趣模型挖掘算法的进一步改进是将缓冲中的网页看成结构化文本, 即对出现在网页不同位置、用不同标记来标记的信息赋予不同的权重. 因此对于词条  $t_i$ , 在文本  $d_j$  的位置  $place$ , 标记  $tag$  中出现的权重我们用权重函数  $ptw(t_i, d_j, place, d_j, tag)$  表示, 则词条  $t_i$  在文本  $d_j$  中的权重记为  $stf_{ij} = \sum_{t_i \in d_j} ptw(t_i, place, t_i, tag)$ . 这样对于兴趣结点  $\text{Node}(t_i)$  来说, 它的兴趣权重为

$$\text{Node}(t_i).weight = idf_i \sum_{j=1}^n stf_{ij},$$

其中,  $idf_i$  为  $t_i$  在  $D$  中的反转文本频率,  $n$  为  $D$  中文本的个数.

通过以上对简单兴趣模型及其兴趣权重算法的分析比较, 可以看出简单兴趣模型具有简单、存储空间小、算法执行效率高等优点. 该模型可以广泛应用于个性化定制信息、信息的预取和信息过滤等诸多领域. 但是随着计算机性能的不断提高, 存储空间、运行速度都得到了显著提高, WWW 缓存的容量也越来越大<sup>[1,4]</sup>, 这就为从另一个角度考察用户兴趣提供了信息保障. 怎样充分挖掘缓冲中用户的潜在兴趣成为当今学术界研究的重点<sup>[1~5]</sup>. 我们在对 WWW 模型和 WWW 缓冲结构模型充分分析研究的基础上, 认为应该综合利用 WWW 内容挖掘<sup>[7,9]</sup>、WWW 结构挖掘<sup>[9]</sup>和 WWW 使用记录挖掘<sup>[9]</sup>方法才能从 WWW 缓冲中尽可能地挖掘出用户的兴趣, 为此我们专门提出了实时二维兴趣模型, 用于描述从 WWW 缓冲中挖掘出来的用户兴趣.

### 3 实时二维兴趣模型

将缓冲中的信息看作普通的文本信息的统计, 这样只能得到简单的兴趣模型, 简单兴趣模型的信息量小, 计算简单, 因而得到广泛应用; 我们提出利用文本间的超链接关系的一步统计方法<sup>[12, 13]</sup>, 能将用户兴趣模型扩充到二维, 即它不仅仅考虑兴趣的权重, 而且考虑兴趣间的递推关系. 这种二维兴趣模型大大地丰富了兴趣的内涵, 使得兴趣之间不再相互独立, 但是我们也看到由于二维关系的引入使得需要存放该模型的信息量飞速膨胀, 二维实时兴趣

模型不是简单地将简单兴趣模型进行扩充, 而是需要在存储空间和运行效率方面综合考虑以后建立的模型.

**定义 2.** 兴趣结点间的联系称为兴趣关联规则, 用三元组  $(\text{Node}(t_s), weight, \text{Node}(t_t))$  表示, 简记为  $\text{Rule}(\text{Node}(t_s), \text{Node}(t_t))$ , 其中  $weight$  表示由兴趣结点  $\text{Node}(t_s)$ ,  $t_s$  转到兴趣结点  $\text{Node}(t_t)$ ,  $t_t$  的可能性,  $0 < weight \leq 1$ .

**定义 3.** 兴趣关联规则的集合构成兴趣关联知识库, 用集合  $\{\text{Rule}(\text{Node}(t_s), \text{Node}(t_t)) | \text{Rule}(\text{Node}(t_s), \text{Node}(t_t)) \text{ 为兴趣关联规则}\}$  表示, 简记为 RULE, 兴趣关联知识库中的兴趣关联规则应满足

$$\sum_{P(\text{Node}(t_s))} \text{Rule}(\text{Node}(t_s), \text{Node}(t_t)).weight = 1,$$

其中

$$P(\text{Node}(t_s)) = \{t_t \mid t_t \in T, \text{Rule}(\text{Node}(t_s), \text{Node}(t_t)) \in \text{RULE}\}.$$

定义 2 给出了实时二维兴趣模型中的兴趣关联规则, 这些兴趣关联规则的集合构成了兴趣关联知识库. 用户可以根据该知识库中的知识进行信息过滤、信息预取和个性化检索等<sup>[7,9~11,14~21]</sup>.

### 4 实时二维兴趣模型的存储

利用实时二维兴趣模型可以更加充分地从 WWW 缓冲中挖掘用户的兴趣, 因为该模型充分考虑了兴趣之间的关联关系. 上文中只是对该模型进行了描述, 而没有对它的实际可行性进行讨论. 如果要求兴趣之间的关联关系是完备的, 那么对于字典中的词条个数  $|T|$  来说, 所有兴趣结点的最大可能个数为  $|T|$ , 根据定义 2 中的兴趣关联规则的定义, 兴趣关联规则的最大可能个数(即兴趣关联知识库的最大容量)为  $|T| \times |T|$  个, 也即兴趣关联知识库的空间容量应该为  $|T| \times |T| \times a$  ( $a$  为每条规则所占的存储空间大小). 对于  $|T| = 1M$ ,  $a = 2\text{Byte}$  来说, 兴趣关联知识库的存储空间大小应该为 2000GB. 尽管可以通过对每条规则中的兴趣进行编码来缩小每条规则的存储空间, 即便这样,  $|\text{RULE}|$  的大小也是在 TB 级的. 这种空间需求对于普通的客户端是难于接受的. 虽然通过兴趣节点可以精确描述兴趣关联规则, 而且在理论计算上带来很大的方便, 但是在实际应用中, 如此精确的描述是不可行的. 这就需要引入某种机制对兴趣的描述粒度进行放大, 即粗化. Pawlak 在 1982 年提出的粗糙集理论<sup>[22]</sup> 是标准

集合理论的扩展,支持决策过程中的近似决策。它的基本思想是在论域的等价关系的基础上,通过一对近似操作算子(下近似和上近似)来刻画论域中的集合。许多需要近似计算的应用系统都可以利用这些算子<sup>[22]</sup>。我们将在同义词关系的基础上构建用户兴趣粗糙算子。

设  $R$  为字典  $T$  上的同义词关系,一个非空的对象领域构造了一个近似空间  $apr = (T, R)$ 。对  $T$  的同义词划分可以记为  $T/R = \{C_1, C_2, \dots, C_l\}$ , 其中  $C_i$  为  $R$  的一个等价类(即一组同义词)。对于  $T$  的任意子集  $S$ :

$S$  的下近似  $lower\_apr(S) = \{x \in C_i | C_i \subseteq S\}$ ,

$S$  的上近似  $upper\_apr(S) = x \in C_i | C_i \cap S \neq \emptyset\}$ .

$S$  的两种近似实际上是在近似空间  $(T, R)$  中对集合  $S$  的一种近似描述。

为了简约兴趣关联知识库,将兴趣关联规则建立在  $R$  的等价类的基础上。

**定义 4.** 粗糙兴趣结点是二元组  $(C, weight)$ , 简记为  $Rough\_Node(C)$ , 其中  $C \in T/R$ ,  $weight$  为兴趣结点  $Rough\_Node(C)$  的权重。

**定义 5.** 粗糙兴趣结点间的联系称为粗糙兴趣关联规则,用三元组  $(Rough\_Node(C_s), weight, Rough\_Node(C_t))$  表示,简记为  $Rough\_Rule(Rough\_Node(C_s), Rough\_Node(C_t))$ , 其中  $weight$  表示由粗糙兴趣结点  $Rough\_Node(C_s)$  转到粗糙兴趣结点  $Rough\_Node(C_t)$  的可能性,  $0 < weight \leq 1$ 。

**定义 6.** 粗糙兴趣关联规则的集合构成粗糙兴趣关联知识库,用集合  $\{Rough\_Rule(Rough\_Node(C_s), Rough\_Node(C_t)) | Rough\_Rule(Rough\_Node(C_s), Rough\_Node(C_t)) \text{ 为粗糙兴趣关联规则}\}$  表示,简记为  $Rough\_RULE$ ,粗糙兴趣关联知识库中的粗糙兴趣关联规则应满足

$$\sum_{P(Rough\_Node(C_s))} Rough\_Rule(Rough\_Node(C_s), Rough\_Node(C_t)). weight = 1,$$

其中,

$$P(Rough\_Node(C_s)) = \{C_t | C_t \in T/R, Rough\_Rule(Rough\_Node(C_s), Rough\_Node(C_t)) \in Rough\_RULE\}.$$

利用粗糙集理论可以大大缩小兴趣关联知识库的存储空间。图 1 中为表明兴趣关联知识库的存储空间与所选用的字典大小的关系(为了显示直观,横坐标和纵坐标均采用对数形式。横坐标表示字典大

小,纵坐标表示信息库的存储空间的大小)。从图 1 中可以看出,由于采用粗集中的等价类作为兴趣结点,与原来基于词条的兴趣关联规则库相比,等价关系粒度的关联规则库在字典大小一样的情况下,存储空间大大缩小了。

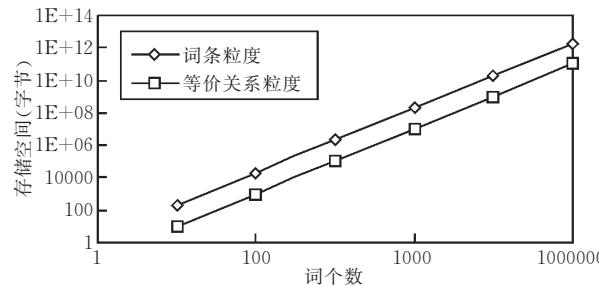


图 1 关联规则信息库与字典大小关系图

为了描述方便,下文中如不特别说明,兴趣结点指粗糙兴趣结点,兴趣关联规则指粗糙兴趣关联规则,兴趣关联知识库指粗糙兴趣关联知识库。

## 5 二维兴趣的数据挖掘

有了基于粗糙集的实时二维兴趣模型,就可以在该模型的基础上提出具体的数据挖掘算法。目前常用的 WWW 挖掘算法有:基于内容的 WWW 内容挖掘、基于结构的 WWW 结构挖掘和基于日志的 WWW 使用记录的挖掘<sup>[9]</sup>。在上文中的简单兴趣模型的挖掘算法中,最多只是考虑到了网页内部的结构,而没有考虑到网页之间的相互关系。网页之间的超链关系包含了大量人类潜在的语义,有助于自动分析出用户的兴趣。当一个 WWW 页面的作者在他的页面中建立指向其它页面的超链接时,可以看作是作者对该页面的注解<sup>[9]</sup>。通过分析不同作者对同一网页的不同的评价,可以综合评判该网页的重要性。目前在用户兴趣的提取过程中是将每个网页对等看待的,这不能反映用户浏览网页过程中对网页感兴趣的程度。实际上一个用户在浏览网页的时候,更倾向于访问权威一些的网页。因此为了更能体现用户的兴趣,在从 WWW 缓冲中挖掘用户兴趣的时候,应该对不同的网页作不同的处理。

在从 WWW 缓冲中挖掘二维兴趣模型描述的兴趣时,要获取两个方面的信息:各个兴趣的权重与兴趣之间的转移权重。WWW 缓冲中的页面可用一个有向图来表示, $G = (V, E)$ :页面抽象为图中的节点, $V$  是缓冲中页面节点的集合,页面间的超链接构成图  $G$  中的有向边, $E$  是这些有向边的集合。 $G$  中节

点的入边表示其它网页对该节点的引用,出边表示该节点对其它网页的引用.因此 WWW 缓冲中网页之间的超链接揭示了缓冲中网页的结构. PageRank<sup>[8]</sup>的基本思想是:一个页面被多次引用,则这个页面可能很重要;一个页面尽管没有被很多的网页所引用,但是它被某个重要网页所引用了,则该页面也可能很重要;一个页面的重要性被均分并被传递到它所引用的页面.在 PageRank 的算法中,页面的重要程度可以通过公式(1)计算.

设在图  $G$  中,页面节点  $A$  被节点  $T_0, T_1, \dots, T_n$  所引用( $A, T_0, T_1, \dots, T_n \in V$ ),记页面节点  $A$  所引用的页面节点的个数为  $C(A)$ ,则页面节点  $A$  的引用相关性  $RR$  为

$$RR(A) = (1 - d) + d(RR(T_0)/C(T_0) + \dots + RR(T_n)/C(T_n)) \quad (1)$$

其中, $d$  为阻尼常数, $0 < d < 1$ , $d$  由系统设置(一般取值为 0.85).  $RR(A)$  就是由于网页之间的引用相关性而产生的权重. 引用分析在信息获取领域已经进行了长时间的研究,对于引用相关性还有 Hub/authority 方法<sup>[9]</sup>,它主要认为并不是所有的引用都有相同的影响. 关于它的具体思想在此不再赘述,这部分的主要问题是如何将网页的相对重要性分布到用户的兴趣中去.

对于粗糙兴趣结点  $Rough\_Node(C)$ ,它的权重可以通过公式(2)计算.

$$Rough\_node(C). weight = \sum_Q (idf_i \sum_{j=1}^n (RR(d_j) stf_{ij})) \quad (2)$$

其中, $Q$  为谓词  $t_i \in Rough\_Node(C)$ .

通过公式(2)计算出的粗糙兴趣结点的兴趣权重充分考虑了词条在网页中出现的位置、在 WWW 缓冲中的出现情况和网页的重要程度,因而计算出的兴趣权重更具客观性和相对性. 二维兴趣模型不仅计算兴趣的重要程度,还要计算兴趣之间的转移关系.

兴趣之间的转移关系反映了用户在某一兴趣状态下的下一步的走向. 我们先来分析在时间  $t$ ,用户处于状态  $S_i$ (用户正在浏览网页  $S_i$ ),用户在下一时

间  $t+1$  可能做出的行为选择:

①继续浏览网页  $S_i$ ;

②点击网页中的某个链接转到另一个新的网页;

③按浏览器上的“返回”按钮返回到上次访问的网页;

④在浏览器的地址栏中重新输入新的网页地址.

对于以上 4 种用户的浏览趋势,我们分别以  $\alpha, \beta, \gamma$  和  $\delta$  表示<sup>[23]</sup>,它们满足  $0 < \alpha, \beta, \gamma, \delta < 1, \alpha + \beta + \gamma + \delta = 1$ . 这样用户的访问趋势可以通过趋势矩阵  $Q$  表示

$$Q = (q_{ij})_{n \times n}, q_{ij} = \begin{cases} \alpha, & i = j \\ \beta, & (v_i, v_j) \in E \\ \gamma, & (v_j, v_i) \in E \\ \delta, & \text{其它} \end{cases},$$

对矩阵  $Q$  归一化后得转移概率矩阵  $P$ ,  $P = (p_{ij})_{n \times n}$ , 其中

$$p_{ij} = \frac{q_{ij}}{\sum_{j=1}^n q_{ij}}.$$

矩阵  $P$  反映了 WWW 缓冲中网页间的转移情况,要得出二维用户兴趣模型中兴趣间的转移情况还需按照某种方式进行转换. 算法 1 给出了基于转移概率矩阵的兴趣关联规则挖掘算法.

### 算法 1. 粗糙兴趣关联规则挖掘.

置粗糙兴趣关联知识库中的所有关联规则的  $weight$  为 0;

```

for i=1 to n do /* for1 */
  for j=1 to n do /* for2 */
    for 每一  $v_i$  中的词条 t1 do /* for3 */
      for 每一  $v_j$  中的词条 t2 do /* for4 */
         $C_s = upper\_apr(\{t1\})$ ;
         $C_t = upper\_apr(\{t2\})$ ;
         $Rough\_Rule(Rough\_Node(C_s),$ 
           $Rough\_Node(C_t)). weight$ 
         $= Rough\_Rule(Rough\_Node(C_s),$ 
           $Rough\_Node(C_t)). weight + p_{ij}$ ;
        end for /* for4 */
      end for /* for3 */
    end for /* for2 */
  end for /* for1 */

```

将粗糙兴趣关联规则的权重单位化,即

$$Rough\_Rule(Rough\_Node(C_i), Rough\_Node(C_j)). weight = \frac{Rough\_Rule(Rough\_Node(C_i), Rough\_Node(C_j)). weight}{\sum_{Q(Rough\_Node(C_i))} Rough\_Rule(Rough\_Node(C_i), Rough\_Node(C_j)). weight},$$

其中, $Q(Rough\_Node(C_i)) = \{C_j \mid C_j \in T/R,$

$Rough\_Rule(Rough\_Node(C_i), Rough\_Node(C_j))$

$\in Rough\_RULE\}$ .

算法 1 计算得到的关联规则中将很多偶然情况所导致的兴趣关联都寻找出来了, 而实际上有些关联权重非常小的关联规则在实际应用中完全可以忽略不计。而且考虑到系统的存储空间的限制, 有必要裁减某些不必要的关联规则。这可以通过设置某个阈值来实现, 凡是关联权重小于该阈值的关联规则全部裁减掉。这样可以大大缩小兴趣关联规则的存储空间, 而且可以大大提高调用和调整兴趣关联规则库中规则的速度。

## 6 兴趣关联知识库的实时更新

兴趣关联知识库中的兴趣关联规则指出了从某一等价类(兴趣)转向其它等价类(兴趣)的可能性。这些兴趣关联规则建立在对大量历史数据进行分析的基础上, 它可以每隔一段时间重新构造一次。用户在访问页面时, 一般是连续访问多个页面。这些页面实际上表明了用户当前的兴趣状况, 它们相对那些用于构造兴趣关联知识库的历史数据来说, 对反映用户的行为更有价值, 即它们的新鲜度更高。考虑到用户访问 WWW 页面时响应速度等问题, 不可能在用户每次访问一个页面后就完全重新构造兴趣关联知识库, 但是又要考虑到用户当前所访问的页面轨迹, 我们首先对文章<sup>[7]</sup>中的兴趣关联知识库调整增量算法进行一定的扩充, 用逐步修整兴趣关联知识库的方法来避免兴趣知识库的数据断层。

### 6.1 兴趣关联知识库的增量修整法

用户当前访问的页面轨迹是由用户当前访问的页面组成的序列。如果每次访问第  $n+1$  个页面对前面的  $n$  个页面都要进行处理的话, 在  $n$  较大时用户的响应速度也得不到保障, 因此采用增量算法, 即用户每访问一个页面就对兴趣关联知识库修整一次(如图 2 所示)。

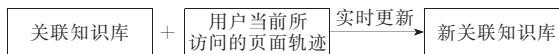


图 2 利用知识库预测用户链接次序

### 算法 2. 粗糙兴趣关联知识库调整增量算法。

设用户当前访问到第  $n$  个页面  $Y_n$ , 可采用如下步骤对兴趣关联知识库进行修整:

1. 统计词典  $T$  中的词条出现在页面  $Y_n$  中的频度(采用了分词技术与词干提取及同义词归类技术<sup>[12]</sup>), 得集合  $K(Y_n)=\{(C'_i, f_i) | t \in C'_i, f_i \text{ 为 } C'_i \text{ 中 } t \text{ 出现在页面 } Y_n \text{ 中的频度之和}, f_i > 0, i \in N\}$ 。

2. 对于兴趣关联知识库中的兴趣结点  $Rough\_Node(t)$ , 进行如下调整。

行如下调整。

如果  $Rough\_Node(C), C=C'_i$  且  $(C'_i, f_i) \in K(Y_n)$ , 那么  
 $Rough\_Node(C).weight = Rough\_Node(C).weight + f_i \times F(n)$ ,  
其中,  $F(n)$  表示新鲜度, 是一单调非递减函数。

易证明, 新鲜度函数的存在可以保证越是最近访问过的页面, 对用户当前的兴趣的作用越大。 $F(n)$  可以取  $n, n^2$  等。由于这是一个根据用户的访问次序对兴趣关联知识库中兴趣结点权重逐步递加的过程, 因此用户的页面访问轨迹的时序已经体现在兴趣关联知识库中(也即兴趣关联知识库在用户使用的过程中不断地更新一部分信息)。

算法 2 可以保证对兴趣关联知识库的实时更新, 但是只修整了兴趣关联知识库中的兴趣结点的权重, 这对于二维兴趣模型来说, 显然是不够的。为此我们提出了兴趣关联知识库的预测修整算法。

### 6.2 兴趣关联知识库的预测修整算法

兴趣关联知识库可以看成一个不确定自动机, 用户当前访问的网页作为该自动机的输入, 对可能的用户兴趣进行强化。在上文的方法中利用新鲜度函数来保证越是最近访问的页面对用户当前的兴趣影响越大。该方法只是更新了兴趣节点的权重, 而没有对兴趣关联规则所关联的兴趣作对应的修改。兴趣关联知识库的预测调整算法是在粗糙兴趣关联知识库调整增量算法的基础上利用兴趣关联规则对兴趣关联知识库进行调整的方法。用户每访问一个页面就对兴趣关联知识库更新一次(如图 3 所示)。



图 3 关联知识库的更新

算法 3 在用户访问网页的过程中, 不但更新了知识库中对应的兴趣结点, 而且更新了相关的兴趣关联规则。这种更新方式保证了用户兴趣及用户兴趣迁移的及时更新。在适当的时候(在机器空闲的时候或者按照一定的周期)对知识库进行重新计算。

### 算法 3. 粗糙兴趣关联知识库预测修整算法。

设用户当前访问到第  $n$  个页面  $Y_n$ , 可采用如下步骤对兴趣关联知识库进行修整:

1. 统计词典  $T$  中的词条出现在页面  $Y_n$  中的频度(采用了分词技术与词干提取及同义词归类技术<sup>[12]</sup>), 得集合  $K(Y_n)=\{(C'_i, f_i) | t \in C'_i, f_i \text{ 为 } C'_i \text{ 中 } t \text{ 出现在页面 } Y_n \text{ 中的频度之和}, f_i > 0, i \in N\}$ 。

2. 对于兴趣关联知识库中的兴趣结点  $Rough\_Node(t)$ , 进行如下调整。

if  $Rough\_Node(C), C=C'_i$  且  $(C'_i, f_i) \in K(Y_n)$  then

$Rough\_Node(C).weight =$

```

Rough_Node(C).weight +  $f_i \times F(n)$ 
for Rough_Node(C) 的每个后续节点 C' do
    Rough_Node(C').weight =
        Rough_Node(C').weight +  $f_i \times F(n) \times$ 
        Rough_Rule(Rough_Node(C),
        Rough_Node(C')).weight
    end for
end if

```

### 6.3 算法分析

为了分析算法 1,2 与 3 的性能指标,我们在奔腾 366,256M 内存、Window NT4.0 中文操作系统和 CERNET 环境下随机采集了 9 个网页,并对这些网页中的词条进行了统计(如表 1 所示). 兴趣结点的更新率表明了该算法在处理过程中对兴趣信息库中兴趣结点的更新情况(兴趣结点个数为 1000). 图 4 中通过不同颜色的柱状图表明了算法 1,2,3 对于表 1 中的不同网页所导致的兴趣结点的更新情况. 从该图我们可以清楚地看出,算法 1 由于要重新构造兴趣信息库,所以 100% 地更新了兴趣结点; 算

法 2 只是更新了当前网页中出现的词条所对应的兴趣结点,因而相对于算法 3 中利用关联规则更新兴趣结点的方法来说,具有相对小的兴趣结点更新率. 兴趣结点更新得越多,就越能及时跟踪用户的兴趣,但是像算法 1 中全部更新兴趣结点的方法不能满足系统实时性的要求,因此只有采用实时更新算法在一定的时间范围内(用户的延迟忍耐程度)对兴趣关联知识库进行一定的更新. 图 5 中给出了算法 2 和算法 3 在访问表 1 中的网页后,其对兴趣关联知识库的更新速度. 可以看出,算法 3 相对于算法 2 所用的时间要相对长一些,算法 2 和算法 3 对这些网页的处理都在 0.025s 之内. 图 6 中的相对更新度反映了更新兴趣结点的个数与网页中不同词条的个数相对程度. 由于采用了粗糙集理论对兴趣词条进行了粗糙化,所以图 6 中下面的曲线不是直线; 算法 3 中由于在更新兴趣结点的时候同时利用关联规则对相关兴趣结点进行了更新,因而它的相对更新度大于算法 2.

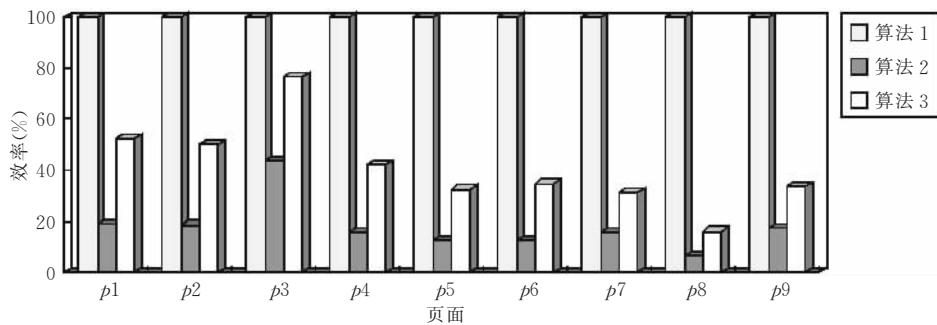


图 4 相对所有兴趣结点的更新率

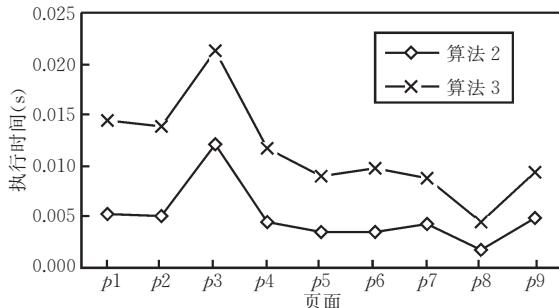


图 5 算法 2 与算法 3 执行时间比较

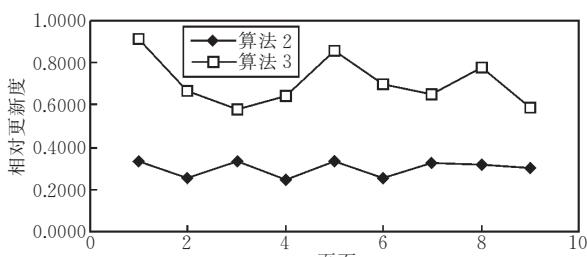


图 6 相对更新度(更新的结点/不同词条个数)

表 1 网页统计信息

页面	字符数	词数	不同词的个数	涉及等价类的个数
P1	1494	650	572	191
P2	2014	876	743	186
P3	3488	1516	1324	440
P4	1667	725	657	160
P5	971	422	379	126
P6	1382	601	498	125
P7	1267	551	481	156
P8	577	251	207	65
P9	1572	683	575	174

## 7 WWW 缓存兴趣挖掘原型系统

为了更加明确用户兴趣的挖掘过程,我们给出了图 7 中所示的基于 Agent 的 WWW 缓存兴趣挖掘原型系统. 该原型系统反映了 WWW 缓存挖掘的基本过程. 与普通的浏览器相比,图 7 中增加了兴趣

挖掘代理、兴趣更新代理和兴趣关联知识库。兴趣挖掘代理运行在客户端,它定时将兴趣关联知识库中的兴趣关联规则进行更新。在我们设计的原型系统中,我们利用算法 1 将 WWW 缓存中的数据转换为兴趣规则存放在兴趣关联知识库中。考虑到系统运行的效率及灵活性,用户可以设定自己的更新时间表。时间表包括更新的频率(可以是一次、每分钟、每小时、每天、每周……)、更新的日期(具体的日期……)、更新的时间(用户可以具体到小时、分钟和秒)。图 7 中的兴趣更新代理实时跟踪用户访问网页的过程,可以利用算法 2 或者算法 3 计算用户的最新兴趣情况,然后更新到兴趣关联知识库中去。兴趣挖掘代理灵活的时间调度能力和兴趣更新代理对用户兴趣的实时更新能力既保证了系统的运行效率又保证了兴趣关联知识库中的信息与用户行为的同步。兴趣关联知识库、兴趣挖掘代理与兴趣更新代理的存在对用户是透明的。用户仍像平时一样使用浏览器。

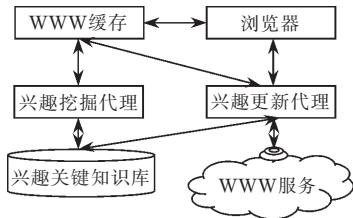


图 7 基于 Agent 的 WWW 缓存兴趣挖掘框架

现在已有许多用于 Web 预取的商品化软件,例如,CNET 网络公司的 NetSonic 浏览器加速软件<sup>①</sup>的 IntelliFetch 技术会预先读入由目前浏览网页所连结出去网页的文字部分以节省时间,但它可能获取大量对用户无用的页面,这对于整个网络系统以及用户的花费是个沉重的负担(有些网络是按访问流量计费的);Noviscope 公司的 Nociscope 软件<sup>②</sup>是一种站内快速检索引擎,当进入一个站点,它会将该站的结构图显示出来,以便快速查到所需内容,并显示下传和上传信息。它也需要获取大量无用的 Web 页面,而且使用起来对用户并不透明。基于 Agent 的 Web 预送系统由于利用了数据采掘的方法,并且充分考虑了用户当前的兴趣状况,所以可以根据用户的兴趣习惯很好地预测用户即将发生的行为,进而预取从目前浏览器连结出去的最有价值(用户最感兴趣)的几个网页。

## 8 结束语

关于用户兴趣的表示已经有较多这方面的研

究,对用户兴趣的描述分别提出了一元、二元和三元的描述方法,将用户兴趣分为长期兴趣和短期兴趣。实时二维兴趣模型是针对 WWW 缓存的兴趣挖掘而提出的,利用该模型可以有效地对 WWW 缓存中的信息进行挖掘,而且可以实时地对用户的兴趣进行更新,从而可以有效地反映用户的当前兴趣。本文在对 WWW 缓存的兴趣挖掘过程中没有考虑多个用户有可能使用同一台机器,如果多个用户对应同一个 WWW 缓存,那么 WWW 缓存中的信息不只反映一个用户的兴趣。现有的一些操作系统如 Windows NT、Windows 2000 根据不同的用户来分别管理对应的 WWW 缓存,这样可以解决这个问题。我们将对 WWW 缓存的兴趣模型和数据挖掘方法作进一步的研究,例如怎样实现多用户共同兴趣的挖掘,对于多用户的系统怎样更加合理地管理存储空间(比如可以将一些图片、声音资源放在多个用户可以共享的 WWW 缓存中)。

## 参 考 文 献

- 1 Jia Wang. A survey of WWW caching schemes for the internet. ACM Computer Communication Review, 1999, 29(5):36~46
  - 2 Cunha C., Jaccoud C. F. B.. Determining www user's next access and its application to pre-fetching. In: Proceedings of ISCC'97, the 2nd IEEE Symposium on Computers and Communications, Alexandria, Egypt, 1997, 6~11
  - 3 Bestavros A., Cunha C.. A prefetching protocol using client speculation for the WWW. Boston University, Department of Computer Science, Boston, MA 02215: Technical Report: TR-95-011, 1995
  - 4 Kroeger T. M., Long D. D., Mogul J. C.. Exploring the bounds of WWW latency reduction from caching and prefetching. In: Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS), Monterey, CA, 1997, 13~22
  - 5 Barish G., Obraczka K.. World wide WWW caching: trends and techniques. IEEE Communications Magazine Internet Technology Series, 2000, 38(5):178~184
  - 6 Xu Bao-Wen, Zhang Wei-Feng, Chu W. C., Yang Hong-Ji. Application of data mining in WWW pre-fetching. In: Proceedings of IEEE MSE2000, TaiWan, 2000, 372~377
  - 7 Xu Bao-Wen, Zhang Wei-Feng. Research on WWW pre-fetching by data mining. Chinese Journal of Computers, 2001, 24(4):430~436(in Chinese)
- (徐宝文, 张卫丰. 数据挖掘技术在 WWW 预取中的应用研

<sup>①</sup> <http://www.netsonic.com/netsonic>

<sup>②</sup> <http://www.naviscope.com/>

- 究. 计算机学报, 2001, 24(4):430~436)
- 8 Brin S., Page L.. The anatomy of a large-scale hypertextual WWW search engine. In: Proceedings of 7th world wide WWW Conference (WWW'98), Brisbane, Australia, 1998, 107~117
- 9 Han Jia-Hui, Men Xiao-Feng *et al.*. Research on WWW mining. Journal of Computer Research and Development, 2001, 38(4):405~414(in Chinese)  
(韩家炜, 孟小峰, 王 静, 李盛恩. WWW 挖掘研究. 计算机研究与发展, 2001, 38(4):405~414)
- 10 Zhang Wei-Feng, Xu Bao-Wen, Chu W. C., Yang Hong-Ji. Data mining algorithms for WWW pre-fetching. In: Proceedings of the 1st International Conference on WWW Information Systems Engineering (WISE'2000), Hong Kong, China, 2000, 34~38
- 11 Zhang Wei-Feng, Xu Bao-Wen, Song W., Yang Hong-Ji. Pre-fetching WWW pages through data mining based prediction. Journal of Applied System Studies, Cambridge International Science Publishing, England, 2002, 3(2):384~398
- 12 Leggett J. *et al.*. Special issues on hypertext. Communication of ACM, 1994, 37(2):26~108
- 13 Chakrabarti S., Dom B., Raghavan P., Rajagopalan S., Gibson D., Kleinberg J.. Automatic resource compilation by analyzing hyperlinkage structure and associated text. In: Proceedings of 17th International World Wide Web Conference, 1998, 65~74
- 14 Zhang Wei-Feng, Xu Bao-Wen, Yang Hong-Ji, Chu W. C.. A genetic algorithm based general search engine. In: Proceedings of IEEE Multimedia Software Engineering'2000 (MSE2000), TaiWan, 2000, 366~371
- 15 Zhang Wei-Feng, Xu Bao-Wen, Xu Lei *et al.*. Personalizing search result using agent. Mini-Micro Systems, 2001, 22(6): 724~727(in Chinese)  
(张卫丰, 徐宝文, 许 蕾等. 利用 Agent 个性化搜索结果. 小型微型计算机系统, 2001, 22(6):724~727)
- 16 Zhang Wei-Feng, Xu Bao-Wen. Research on framework sup-
- porting web search engine. Journal of Computer Research & Development 2000, 37(3):376~378(in Chinese)  
(张卫丰, 徐宝文. Web 搜索引擎框架研究. 计算机研究与发展, 2000, 37(3): 376~378)
- 17 Zhang Wei-Feng, Xu Bao-Wen, Zhou Xiao-Yu. Counting techniques in web pages. Mini-Micro Systems, 2000, 21(10):1096~1099(in Chinese)  
(张卫丰, 徐宝文, 周晓宇. Web 页面中的计数器研究. 小型微型计算机, 2000, 21(10):1096~1099)
- 18 Zhang Wei-Feng, Xu Bao-Wen, Zhou Xiao-Yu. Web page techniques for interacting between elements. Computer Engineering, 2000, 26(8):62~64(in Chinese)  
(张卫丰, 徐宝文, 周晓宇. WWW 页面中元素间交互技术研究. 计算机工程, 2000, 26(8):62~64)
- 19 Zhou Tao *et al.*. Information mining technologies and realization on WWW. Journal of Computer Research and Development, 1999, 36(8):1021~1024(in Chinese)  
(邹 涛等. WWW 上的信息挖掘技术及实现. 计算机研究与发展, 1999, 36(8):1021~1024)
- 20 Zhang Wei-Feng, Xu Bao-Wen, Zhou Xiao-Yu. An improved relativity technology in reference search. Journal of Software, 2001, 12(supplement): 317~322(in Chinese)  
(张卫丰, 徐宝文, 周晓宇. 一种改进的参考文献检索中的相关性技术. 软件学报, 2001, 12(增刊): 317~322)
- 21 Xu Bao-Wen, Zhang Wei-Feng. Research on the improved reference search model. Journal of Computer Research and Development, 2002, 39(5): 599~606(in Chinese)  
(徐宝文, 张卫丰. 一种改进的参考文献搜索模型及相关性技术研究. 计算机研究与发展, 2002, 39(5): 599~606)
- 22 Pawlak Z.. Rough sets. International Journal of Computer and Information Science, 1982, 11(5): 341~356
- 23 Zhang Dell, Dong Yi-Sheng. An efficient algorithm to rank Web resources. In: Proceedings of the 9th International World Wide Web Conference, Amsterdam, Netherlands, 2000, 449~458



**ZHANG Wei-Feng**, born in 1975, Ph. D.. His research interests include program designing language, software architecture, network language, search engine and data mining techniques.

## Background

This topic is related to personalizing the Meta search engine. Meta Search engine is one type of search tool, which realizes its search functions by invoking other search engines. Different search engines may have different emphases in col-

**XU Bao-Wen**, born in 1961, Ph. D., professor and Ph. D. supervisor. His research interests include programming language, software engineering, concurrent and Internet software and Web-based techniques.

lecting information. If the multi search engines can be combined, the coverage of information can be improved. However, normally users only have interests in some specific items, and it is difficult for them to select what they really need sim-

ply by browsing. Thus the search accuracy must be considered. The coverage and the accuracy are the two most important technology guidelines. On the one hand, the Meta search engine improves the coverage by invoking multi search engines; on the other hand, the Meta search engine must take some measures to filter the users' search results, and only the information in which the users have interest is presented to the users. This paper gives a model to describe the users' interest effectively. By this model, the search results from the search engines can be personalized.

The main research work in the Meta search includes:

(1) After analyzing the WWW cache model comprehensively, a real-time two dimensions interest model is introduced in this paper. The real time property of this model is able to ensure that the users' current interest can be represented through the underneath interest. Based on the two dimensions concepts, the inferential relations between the user's interests are fully considered. This model is not a simple extension of the simple interest model, but a full improvement to the model and the related algorithms. The storage method, the effective computing and the real time updating method of the two dimensions interest model are given in our research.

(2) The client-side based personalizing method is given. It includes personalizing the search results by agent and web pre-fetching by data mining. The web pre-fetching technolo-

gy can quicken the speed of retrieving the web pages. In this method, the data in the browsers' cache is represented by the simple WWW data model, and based on the model, the interest association rules are pulled up by the data mining technology. The interest association rules are stored in the interest association repository and they are the basis of prefetching the users' actions.

(3) The server-side based personalizing method is given and the self-adaptive search engine model is brought forward. The self-adaptive search engine produces the feedback signals by collecting the users' accessing serials to the search results. The feedback signals can be used to affect producing the search results. By this method, most users' interest can be considered by the search engines.

(4) The new method of scheduling the search engines by genetic algorithm is given. In this method, the combinations of the actual search engines can be optimized dynamically and the whole performance of the Meta search engine can be improved.

In this paper, the authors present a real time two-dimension interest model by which the search result of the meta search engine can be personalized. In this model, the rough set method is used to store the data more effectively, and the incremental algorithm is used to compute the interest effectively and to update the interest in real time.

填写前三行基本信息即可获赠 2004 年《电信科学》，征订全年《通信学报》免邮费并赠送 2003 年增刊。

#### 2004 年《通信学报》订阅单

《通信学报》月刊 16 开 20 元/本 2004 年全年订价 240 元					
联系人		职业		职务	
单位名称				区号-电话	
单位地址				邮政编码	
订刊金额	¥	订阅数量	份	汇款方式	<input type="checkbox"/> 银行 <input type="checkbox"/> 邮局
备注					

2004 年《通信学报》改为自办发行，全年或破季订阅可随时与发行部联系：010-68373455。

为确保您按时收到杂志并与您保持联系，请务必完整填写表格，传真至 010-84226302 或邮寄到编辑部。

编辑部地址：北京市东城区和平里滨河路 1 号航天信息大楼 9 层 邮政编码：100013。