基于闭相关页面集实现网站自适应

郭 平 陈 婷 李 东
(重庆大学计算学院 重庆 400044)

摘 要 自适应网站能够根据用户需要快速灵活地改变自身,动态适应不断变化的用户需求和应用环境。本文基于图的频繁闭项集从站点一定时期内的日志中挖掘出闭相关页面集,据此提供在线动态推荐为用户导航,改善了传统的合作推荐存在的稀疏性和扩展性问题,在不增大网站服务器负荷的情况下提高对所有用户的信息服务质量。最后,分析了两种测量推荐系统性能的方法并对系统进行评价。

关键词 网站自适应,相关页面集,闭相关页面集,动态推荐

Realizing Adaptive Web Cites Based on Correlative Page Set

GUO Ping CHEN Ting LI Dong
(School of Computer Science, Chongqing University, Chongqing 400044)

Abstract Adaptive web cites can automatically and flexibly improve their organization to fit users' requirements and application environments that are always changing continually. According to concept of frequent-close-item-set based on graph, this paper provides method of on-line dynamic recommendation based on close correlative page set mined from websites' logs of a period. This method improves problems of sparsity and expansibility of traditional collaborative recommendation and enhances quality of service without increasing servers' load. Finally this paper analyzes two methods of measuring performance of recommendation system and evaluates the system.

Keywords Adaptive web cite, Correlative page set, Close correlative page set, Dynamic recommendation

1 网站自适应简述

自适应网站,是指网站通过学习用户访问模式,自发地改善网站自身拓扑结构并且呈现不同的界面给不同的用户[1,2]。网站自适应技术从1996年开始就已经成为人工智能领域的研究热点,近年来伴随着数据挖掘技术的不断成熟,已经取得了一些重要的理论和实践成果。

2003年IBM提出了"随需应变"电子商务的总体框架和概念,其主要特点是:实时响应、灵活可变、聚焦核心、坚固可靠。对于灵活可变,可以理解为网站能够在需要时快速而轻松地改变自身,更好地适应不断变化的商业环境和要求。这其实就是网站自适应思想的体现。目前个性化推荐技术是为每一个用户建模并随着时间推移不断维护更新模型,提供个性化推荐和个性化视图。但随着用户量的增加,用户模型的管理和维护将大大地增加网站服务器的负荷,所以这种方法只适合为个别用户提供服务。而网站自适应技术是一种能为网站所有用户带来方便的方法。在网站如何自适应完善站点拓扑结构的问题上,通常采用的方式是自适应技术提供修改建议而由管理员来最终判断是否根据建议修改网站。

本文采用基于闭相关页面集的方法实现网站自适应,然后通过独立链接实时为用户导航或改变网站视图,提高网站信息服务质量。通过分析网站访问模型论述了方法的可行性,重点介绍本文怎样快速地挖掘闭相关页面集,接下来讲述了如何利用闭相关页面集实现动态推荐,并对实验结果进行分析。最后对全文做总结并讨论需要进一步解决的问题。

2 基于闭相关页面集实现网站自适应的方法

2.1 网站访问模型

Web服务器日志包括访问日志、引用日志和代理日志。对这些日志进行合并和剔除后,我们用 $L=\langle url, SID\rangle$ 的形式来表示Web服务器日志。其中 url 表示一个页面的URL地址,SID是用户会话的标识[5]。

从用户的角度看,用户会话 s 描述了用户的一段时间内的点击流,可以将其表示为如下的 $2*(n+1)$ 元组:

$$S_s=\langle S.sid,\{(u.url,hits)\}^n\rangle n\geqslant 1$$

其中,$hits$ 表示用户会话 s 访问页面 $u.url$ 的次数。

从网页的角度看,网页访问情况是指站点中网页被访问的情况,可以将其表示为如下的 $2*(n+1)$ 元组:

$$U_u=\langle u.url,\{(S.sid,hits)\}^n\rangle n\geqslant 1$$

其中,$hits$ 表示网页 u 被某用户 $S.sid$ 访问的次数。

用户实际访问网站的过程中可能会多次访问同一个网页,而由于本地缓存或智能代理的作用,Web日志往往只记录了用户对该网页的一次请求。但是一次用户会话中访问某些的次数并不影响网页之间的相关关系,所以本文假设每一个用户会话对一个网页只访问一次。

2.2 基于位矢量建立网站访问模型

本文以表示网页访问情况的向量U为行向量构成一个URL-SID矩阵,采用位矢量的结构建立网站访问模型。这样每一个的网页访问情况都有一个 n 位矢量与之对应,矢量的位数 n 等于L中的用户会话总数。URL-SID矩阵的构造步骤如下:首先扫描用户会话数据库1次计算出被频繁请求的

郭 平 副教授,主要研究方向:人工智能、数据挖掘。陈 婷 硕士研究生,主要研究方向:数据挖掘。

网页 URL;然后为每个频繁页面初始化一个 n 位矢量,各位矢量的初始值为 0。然后扫描数据库第 2 遍,对各矢量的每一位进行设置,当某一频繁页面出现在第 k 个用户会话中,则此频繁页面对应的矢量第 k 位置 1。URL-SID 矩阵中,每一个行向量中含 1 的个数就是该网页的访问频繁度。每一个列向量中,为 1 的位表示其所对应的 URL 网页在同一个用户会话中。当某一个 URL 的行向量中"1"的个数大于给定支持度,则称该页面为频繁访问页。

假设从用户会话数据库得到 5 个频繁访问页的网页访问情况,$U_1=10100$,$U_2=01010$,$U_3=10101$,$U_4=10111$,$U_5=01011$,则得到的矩阵如图 1 所示。

$$M_{5\times5}=\begin{bmatrix}1&0&1&0&0\\0&1&0&1&0\\1&0&1&0&1\\1&0&1&1&1\\0&1&0&1&1\end{bmatrix}_{\text{URL-SID}}$$

图 1　网站访问模型

2.3　基于闭相关页面集实现网站自适应方法的提出

2.3.1　闭相关页面集

定义 1　许多用户会话中往往同时包含有一些网页,用 C 表示这些网页构成的集合,C 的支持度 Supp(C)表示包含 C 的用户会话所占用户会话总数的百分比(有时也用包含 C 的事务的频数简化表示。)。例如 $C=\{url_1, url_2,\cdots,url_k\}$,在矩阵 URL-SID 中,SUPP($C$)= $U_1\wedge U_2\wedge\cdots\wedge U_k$ 中 '1' 的数目。如果 C 的支持度不小于给定最小支持度阈值 δ,则称 C 为相关页面集。

由 Apriori 原理,频繁项集的任意子集也是频繁项集,所以要从事务库中挖掘出所有的频繁项集,结果是非常多的。根据闭项集[6]的概念,可以大大地减少挖掘结果中冗余的频繁项集,由此我们提出闭相关页面集。

定义 2　设 C_1 和 C_2 为两个相关页面集,$C_1\subset C_2$,且满足条件:任给用户会话 S 包含 C_1 则 S 也包含 C_2 成立,那么称 C_1 被 C_2 覆盖。记为 $C_1\angle C_2$。

定义 3　设 C 为一相关页面集,如不存在相关页面集 C',$C\angle C'$,即 C 不被任何相关页面集覆盖,则称 C 为闭相关页面集。

本文针对 URL-SID 矩阵挖掘闭相关页面集,算法思想及其正确性证明将在第 3 节给出。

2.3.2　基于闭相关页面集实现网站自适应

闭相关页面集中的这些页面从某种角度上看是围绕同一个主题展开的,往往并没有设置物理链接。可能是因为这些页面而网站设计者的设计初衷中没有考虑到这个角度,自适应网站就可以主动推荐指向这些网页的链接给用户,使这些页面更容易被用户访问到。文[2]提出实现网站自适应的一些问题。比如,站点在什么时机下提供什么样的推荐,推荐链接如何排序,如何调整网站拓扑而不造成网页间链接的混乱。

本文试图就上述的问题提出解决方案。由 URL-SID 矩阵可以方便地计算出每个 URL 页面的访问次数,访问次数的多少也代表了该网页受欢迎的程度,结合用户对某一个闭相关页面集的页面的访问情况,我们提出了一个比较有效的在线推荐策略。

连续考察网站在一段时间内访问模型的变化情况,发现较稳定的访问模式,如闭相关页面集,据此为站点管理员推荐修改网站拓扑的方式。算法直接面向页面间的相关关系,无需考虑资源评价、用户建模、用户分类以及用户真实访问路径识别的问题,所以在降低计算复杂度的同时,也大大提高了自适应的准确性和有效性。

3　挖掘闭相关页面集

本文在 URL-SID 矩阵上递归的挖掘闭相关页面集。因为将不再扫描数据库,对闭相关页面集的挖掘都集中在位矢量上,从而大大提高了速度。

在各频繁访问页的页面访问的位矢量被设定好后,算法利用位矢量在各频繁访问页之间建立有向图。显然,相关页面集$\{url_1, url_2,\cdots,url_k\}$ 的支持度就是 URL-SID 矩阵中,位矢量 $U_1\wedge U_2\wedge\cdots\wedge U_k$ 中"1"的个数,其中"$\wedge$"表示逻辑与操作符。

当位矢量 $U_i\wedge U_j$ 中"1"的个数不小于最小支持度,设 Q 是频繁访问页上的一个序,在 Q 中 $i>j$。则算法在页面 i 和页面 j 之间作一有向边 $i\rightarrow j$,表明项集$\{i,j\}$ 是一个频繁二项集。给定阈值 $\delta=2$,设 $Q=\{1,2,3,4,5\}$,则以图 1 中的矩阵为例建立的有向图如图 2 所示。其中频繁 2 项集为$\{1,3\}$,$\{1,4\}$,$\{3,4\}$,$\{5,2\}$,$\{5,4\}$。下面提出并证明以下命题[10]:

命题 1　$\{i1,i2,\cdots,ik\}$ 是频繁页面集,如在 ik 与 u 之间没有关联边,则$\{i1, i2,\cdots,ik,u\}$ 不可能是频繁页面集。

证明:因为 ik,u 之间没有边,$\{ik,u\}$ 是非频繁页面集。由 Apriori 性质:任何包含非频繁页面集的项集也是非频繁页面集。所以,$\{i1,i2,\cdots,ik,u\}$ 不可能是频繁页面集。

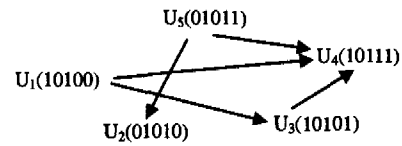

图 2　有向图

因此如果要将频繁 k 项集$\{url_1,url_2,\cdots,url_k\}$扩展为频繁 $k+1$ 项集,只需考虑项 U_k 出发,指向的频繁页面 url_j,如 $U_1\wedge U_2\wedge\cdots\wedge U_k\wedge U_j$ 中 1 的个数不少于支持度阈值,则$\{url_1, url_2,\cdots,url_k,url_j\}$也是频繁页面集。$\{1,3\}$是频繁页面集,且 3,4 之间有边相连,所以检验$U\{1,3,4\}=U_1\wedge U_3\wedge U_4=(10100)$。$U\{1,3,4\}$中 1 的个数为 2,不少于阈值,所以$\{1,3,4\}$也为相关页面集。

定义 4　设 Q 是频繁访问页上的一个序,且所有相关页面集中的页面按 Q 排列,如 $f=\{i1,i2,\cdots,ik\}$ 为一相关页面集,则相关页面集的集合

$$C_f=\{C\mid C=\{i1,i2,\cdots,ik\}\cup\{j1,j2,\cdots,jl\},\ im>j\,n,\ 1\leqslant m\leqslant k,1\leqslant n\leqslant 1\}\quad(1)$$

称为 f 的前缀项集簇。集合

$$C'_f=\{C\mid C=\{i1,i2,\cdots,ik\}\cup\{j\},im>j,1\leqslant m\leqslant k\}\quad(2)$$

称为 f 的一项扩展前缀项集簇.有关系 $C'_f\subseteq C_f$ 成立。

命题 2　设 $I=\{i1,i2,\cdots,ik\}$ 是一个相关页面集,则 I 中所有的相关页面集的集合可表示为 $C=Ci1\cup Ci2\cup\cdots\cup Cik$ 。

证明:当所有相关页面集按 Q 排列,则这些相关页面集可按第 1 个频繁访问页进行分类,每一类即以各频繁访问页为前缀的相关页面集簇。

命题3 设 $f=\{i1,i2,\cdots,ik\}$ 为一相关页面集，则相关页面集簇 C_f 可表示为

$$C_f=\bigcup C(f\cup u),u<in,1\leqslant n\leqslant k \quad (3)$$

证明：同命题2，C_f 也可按 f 后所跟的频繁访问页 u 进行分类，且由于所有相关页面集按 Q 进行排序，有 $u<i_n$，$1\leqslant n\leqslant k$。命题3成立。

有了命题2和命题3，证明算法可以递归地遍历所有的相关页面集。并且可以证明算法能找出所有的闭相关页面集[7]。

4 动态在线推荐的实现

个性化推荐是指根据用户的兴趣特点向用户推荐其感兴趣的信息，其原理是根据用户模型寻找与其匹配的信息，或者寻找具有相近兴趣的用户群而后相互推荐浏览过的信息。个性化推荐技术根据实现的途径不同，可分为基于规则的推荐、基于内容的推荐、合作推荐和混合推荐。

合作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，典型系统有 WebWatcher[10,11]，CiteSeer，LikeMinds(www. macromedia. com)等。这种方法能够动态跟踪发现用户兴趣变化的新信息，但是存在两个难点。一是稀疏性，即在系统使用初期，由于系统资源还未获得足够多的评价，系统很难利用这些评价来发现相似的用户；另一个是扩展性，即随着系统用户和资源的增多，系统的性能会越来越低。

基于闭相关页面集的推荐也是一种合作推荐，由于一个闭相关页面集里的页面具有相关性，所以当用户访问了某个闭相关页面集里的多数页面时，可以推断该用户与访问过闭相关页面集里所有页面的用户具有相同的访问模式，因此把该闭相关页面集里的其他页面推荐给该用户。我们建立推荐图来存储推荐信息，图中每一个结点有一个列表 SL，表示用户访问到该网页时的候选推荐。SL 中的前 Υ 个页面就是对访问过页面 u 的用户的推荐。

定义5 U_i 到 U_j 的推荐支持度，记作 W_{ij}

$$W_{ij}=N_{ij}/\max\{N_i,N_j\} \quad (4)$$

其中 N_{ij} 指同时包含 U_i 和 U_j 的用户会话的数目，N_i 和 N_j 分别是只包含 U_i 或者 U_j 的用户会话数目。

网站中有一些索引页面可方便用户访问，如网页的导航栏里通常都会设置到网站首页的链接。这些索引页面在用户会话中的出现频率比较高，但却因为本身不具备实际内容所以可推荐性不高。用 N_i 和 N_j 两个值的最大值做除数，可以减少索引页面对推荐效果的影响[12]。

4.1 建立推荐树

对于闭相关页面集(闭相关页面集里的页面按访问频繁度降序排序)，以其为关键字做如下逐层分割，首先按首字符不同将它们分为若干子集，然后分别对关键字个数大于1的子集再按其第二个字符不同进行分割。若所得子集的关键字个数多于1个，则还需按其第三个字符不同进行再分割。依次类推，直至每个小子集中只包含一个关键字。首字符为 u_i 的一组闭相关页面集构成与 u_i 相关队列 Qu_i，从 Qu_i 中提取出的各网页构成 u_i 的推荐列表。如表1中的闭相关页面集，第一次处理得到两个集合{ABDFZM，AKIJ，ABD}{ZHG，ZHK}，A 的推荐列表包括 BDFZMKIJ。

表1 闭相关页面集及其支持度

闭项集	支持度
A,B,D,F,Z,M	100
A,K,I,J	105
Z,H,G	100
Z,H,K	125
A,B,F,Z,M	120
…	…

表2 决策支持度

W_{AK}	0.60	W_{BM}	0.23
W_{AB}	0.60	W_{BD}	0.55
W_{AD}	0.40	W_{BF}	0.56
W_{AF}	0.35	W_{ZH}	0.60
W_{AI}	0.33	W_{ZG}	0.50
W_{AJ}	0.23	W_{ZK}	0.50
W_{BZ}	0.23	…	…

推荐树就是根据对闭相关页面集的如此处理建立的。推荐树是度>2的树，树中的每个结点只含有组成闭相关页面集的一个页面，其子树的结点都属于该结点的相关队列中的闭相关页面集。推荐树的初始化过程如下：

(1)闭相关页面集，按访问频繁度对集合里的页面降序排序；
用“∧”表示树 T 的根结点；

(2)for 每一个闭相关页面集 C_i do
if C 的第一个页面 u_i 不在树中 then
新建结点，用 u_i 标志，并作为根的子结点，从 C 中去掉第一个页面 u_i；
建立一个和 u_i 相关的相关队列 Qu_i，C 入队；
else
找到和 u_i 相关的相关队列 Qu_i，从 C 中去掉第一个页面 u_i，C 入队；

(3)for T 的每个叶结点 u_k do
if(相关队列 Qu_k 的结点个数>1)
{初始化一个集合 L 为空；
while(相关队列 Qu_k 至少有一个结点)do
{队首闭相关页面集出队；
if(C 的第一个页面 ui 不是 u_k 的子结点)then
新建结点，用 ui 标志，并作为 uk 的子结点；
建立相关队列 Qui，从 C 中去掉第一个页面 ui，C 入队；
L=C ∪ L
else
找到和 ui 相关的相关队列 Qui，从 C 中去掉第一个页面 ui，C 入队；
}
for each C_k in L do
$Qu_k=C_k\cup Qu_k$
}

(4)对于只有一个闭相关页面集的相关队列 Qul，将这个相关页面集的各页面都作为 ul 的推荐列表。

基于表1和表2的挖掘结果生成推荐树如图3所示，设 $\Upsilon=2$，并且各个闭相关页面集页面已经按访问频繁度降序排列。由于在 B 的 CL 中，W_{BD} 和 W_{BF} 比较大，所以当用户访问 B 时，只推荐 D 和 F；当用户接着访问了 D 或 F，才推荐页面 Z 和 M。

4.2 推荐树生成推荐图

闭相关页面集之间也可能存在相关关系。对于交集不为空的闭相关页面集，通过建立推荐图中相同的页面顶点间的链接，可以解决交集不为空的闭相关页面集的相关关系。对于交集为空的闭相关页面集之间的相关关系，可以通过推荐图的动态更新来发现。假如 W_{MZ} 大于给定阈值，则认为出现了这样一个访问模式，访问 M 的用户都会访问 Z，因此应当把 Z 添加到 M 的 SL 中，如图 3 中的动态更新。系统动态推荐 Z 给访问了 M 的用户。为了确保 SL 的长度不会无限增加，系统需要在一定时间后挖掘日志更新。

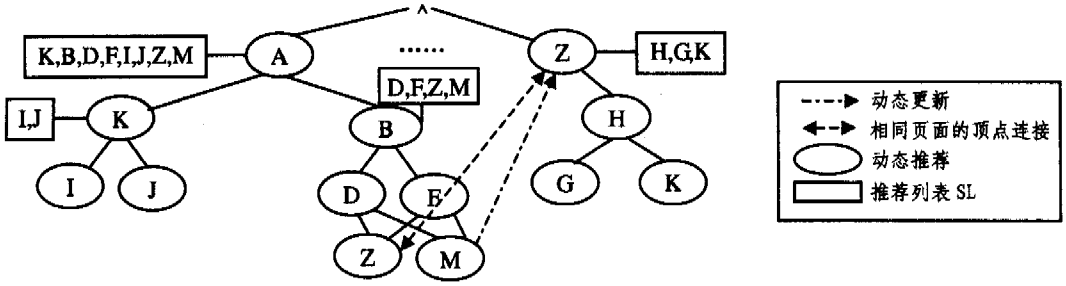

图 3 推荐图

如图 3 中，Z 在 U_A 组支持度不是最高的，但可能在其他组中访问频率比较高，因此可以考虑在推荐图中相同的页面顶点间建立一个链接，在用户访问了 Z 后继续进行推荐。

4.3 推荐图实现网站自适应

系统跟踪用户点击流，将其与推荐图中的某个路径匹配。如果用户请求的网页结点与 T 路径上当前结点的孩子结点匹配，则路径扩展到该孩子结点；否则在当前结点在推荐列表中寻找与请求网页的匹配结点，然后将匹配结点作为当前结点。

系统将首先把当前结点的孩子结点以动态推荐的形式推荐给用户，如果当前结点的孩子只有少于两个，可以考虑从其推荐列表中选择推荐支持度高的页面。动态推荐可以根据独立链接的思想来实现。独立链接就是指链接本身就是网页的基本组成元素，链接的目标地址与描述相分离，在用户访问过程中自动生成。现有的网页制作技术可以方便地实现推荐链接的动态配置。

如果挖掘中发现网站存在这样一个稳定的挖掘模式“ABDZ”，说明很多访问 A 的用户，都是经过了 BD 才去访问 Z。那么 A 中没有必要直接含有到 Z 的链接，因此可以去掉 A 中到 Z 的链接。

5 实验分析

按照本文的方法，我们以某大学网站为平台，根据从一段时间日志（训练数据集）中挖掘出的闭相关页面集为用户（测试数据集）提供动态推荐，并从两个方面对系统的推荐质量和运行效率进行评价分析。经过对原始日志的数据清理后，训练数据集和测试数据集的记录数分别为 6715 条和 1900 条。

推荐链接可以引导用户尽快地找到感兴趣的网页，缩短浏览的时间。为了体现站点推荐对用户访问带来的影响，尽可能准确测量推荐的质量，通过把用户会话划分为两部分，S_{1i} 指用户会话的前一部分，系统根据 S_{1i} 产生推荐 R_{1i}，然后由公式(4)计算 Ω。文[12,13]就是利用 Ω 分别评价了系统 SUGGECT 2.0 以及 SUGGECT 3.0。

$$\Omega=\sum_{i=1}^{Ns}\frac{\sum_{k=1}^{ni}[P_k\in\{S_{i2}\cap R_{i1}\}]f(k)/F}{Ns} \tag{5}$$

其中，F 是对 $f(k)$ 归一化算子，NC 是会话的总数，$[P_k\in\{S_{i2}\cap R_{i1}\}]$ 是一个结果为 0 或 1 的表达式，当页面 p_k 同时属于 S_{2i} 和 R_{1i} 时值为 1 否则为 0。

文[12]认为页面的重要性随着该页面在会话中的位置线性增加。我们认为一个页面对用户的重要性可以根据用户在该页面的停留时间（Store_Time）来决定。实验中，以站点页面的平均停留时间（AVG）为标准，给出 $f(k)$ 的一个分段函数定义。在不同的支持度下，Ω 的变化情况如表 2 所示。

$$f(k)=\begin{cases}2 & store\_time>1.5AVG\\ 1 & 1.5AVG\geqslant store\_time\geqslant 0.5AVG\\ 0.5 & store\_time<0.5AVG\end{cases} \tag{6}$$

推荐链接可以帮助用户提高访问效率，即有推荐链接的用户会话比没有推荐链接的用户会话可以缩短很多。通过测量推荐使用户会话缩短率 α 可以体现出用户访问效率的提高，所以，这也是评价推荐质量的一种方法。其中 $count(S_i)$ 表示未使用推荐时用户会话的长度，$count(RS_i)$ 表示使用推荐链接对用户会话进行约减后的用户会话长度。

$$\alpha=1-count(RSi)/count(Si) \tag{7}$$

从图 4 和图 5 中可以看出 Ω 的值，当闭相关页面集的最小支持度为 10 时最高，而 α 随着闭相关页面集支持度的增加而减小。这是因为随着支持度增加，闭相关页面集的数量减小，因此可推荐的链接数目也减少。当闭相关页面集的最小支持度为 10 时，我们发现约 96% 的用户会话都会因为推荐而约减。

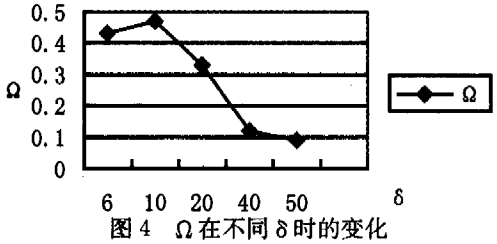

图 4 Ω 在不同 δ 时的变化

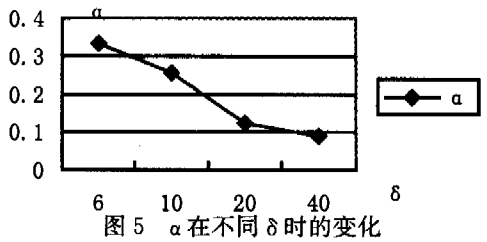

图 5 α 在不同 δ 时的变化

总结 自适应网站利用 Web 挖掘和人工智能的技术，一方面采用各种推荐策略为不同的用户提供不同的站点视图，另一方面通过学习用户访问模式不断改善网站自身拓扑结构，确保网站设计模式和用户访问模式始终保持一致，提高用户访问效率。具有自适应性的网站能够动态适应不断变化的用户需求和应用环境，提高 Web 服务器系统的服务性能并且节省不必要的网络开销。同时，自适应技术还可以使网站降低经营风险，提高开展业务的效率。

目前，关于网站如何自适应地修改自身拓扑仍然是自适应网站技术的一个难点问题。很多研究者倾向于通过站点管理员和网站自适应技术的交互方式来解决这个问题，实践中通常采用的方式是自适应技术提供修改建议而由管理员来最终判断是否根据建议修改网站。论文进一步的工作是分析网站在一段时期内的访问模型，发现比较稳定的访问模式，如闭相关页面集等，然后据此为站点管理员提供完善的站点拓扑修改建议。

参 考 文 献

1 Perkowitz M, Etzioni O. Towards Adaptive Web Sites-Conceptual Framework and Case Study. Artificial Intelligence, 2000, 118: 245～2751
2 Perkowitz M, Etzioni O. Adaptive Web Sites: An AI challenge. In: Proc. of the IJCAI-97, 1997
3 Koutri M, Daskalaki S, Avouris N. Adaptive Interaction with Web Sites: An Overview of Methods and Techniques
4 Sher B M, et al. Automatic personalization based in web usage mining [J]. Communications of the ACM, 2000, 43(8): 142～151
5 Demiriz A, Zaki M J. webSPADE: A Parallel Sequence Mining Algorithm to Analyze the Web Log Data, 2002
6 PaCquier N, BaCtide Y, Taouil R, et al. Discovering frequent closed item sets for association rules [A], In: The 7th Intl. Conf. on Database Theory [C], 1999
7 李力，翟东海，靳蕃. 基于图的频繁闭项集挖掘算法. 西南交通大学学报，2004，39(3)
8 Rucher J, Polanco M J. SiteSeer: personalized navigation for the web [J]. Communications of the ACM, 1997, 40
9 Balabanovicm. An adaptive Web page recommendation service [A]. In: Proc. of the 1st Intl. Conf. on Autonomous Agents [C]. New York: ACM Press, 1997. 378～385
10 Joachim S T, Freita G D, Mitchell T. WebWatch: a tour guide for the World Wide Web [A]. In: Proc. of the Intl. Joint Conf. on Artificial Intelligence [C], 1997. 770～777
11 Freitag D T, Mitchell T. WebWatcher: A Tour Guide for the World Wide Web. In: Proc. of the Intl. Joint Conf. on Artificial Intelligence, 1997
12 CilveCtri F, Baraglia R, Palmerini P, Cerranò M. On-line generation of Suggestions for Web users. In: Proc. of the IEEE Intl. Conf. on Information Technology: Coding and Computing, 2004
13 Baraglia R, CilveCtri F. An Online Recommender System for Large Web Cites. http://citeseer.nj.nec.com

(上接第 67 页)

现消息的编解码。编码函数的实现如下：以 TSPAuthInfoRsp 消息为例来说明。

```
#include(netinet/in.h)
TBool TSPAuthInfoRsp::encode(TCode &code)
{
Code.length = Len_MessageHeader + 3 * sizeof(TInteger) + MaxPasswordLength + ipNum * MaxIPAddressLength + accnoNum * MaxAccessNoLength;//计算码流的长度
NEW_S(code.content,char,code.length); //分配内存
If (code.content == null) //码流长度合法性判断
  {
    Trace("Allocate memory failured in encoding process.\n");
    Retern Wrong;
  }
Memset(code.content,0,code.length); //清空码流内存区
Char * ss=code.content;
TINTEGER tmp = 0;
Tmp = htonl(commandId); //调用函数 htonl()将 32 位主机字符顺序转换成网络字符顺序
Memcpy(ss,&tmp,sizeof(TInteger)); //拷贝字段值到内存指定区域
Ss = ss + sizeof(TInteger); //指针下移，处理该消息的下一个字段
.....
Return Right;
}
```

同时以该消息为例来说明解码函数的实现如下：

```
#include(netinet/in.h)
TBool TSPAuthInfoRsp::decode(const TCode& code)
{
 If((code.length < length) || (code.content == null))
  {
   Return Wrong; //判断码流长度的合法性
  }
Char *p = code.content;
Memcpy(&commandid,p,sizeof(TInteger));
commandid = ntohl(commandId); //调用 ntohl()函数进行转换
p = p + sizeof(TInteger);
memcpy(&sequenceId,p,sizeof(TInteger));
.....
Return Right;
}
```

3.3 共享内存机制技术

由于 SP 信息和业务配置信息被整个系统频繁查询、更新等访问，多个进程又要共用这些数据。于是采用 Unix 的共享内存机制技术。把需要频繁访问的这些数据在系统初始化的时候从数据库中一次性装载读入内存中，各进程可以共享这些数据，这样可以充分提高系统的性能(消息响应时间缩短)。共享内存的 id 可以通过调用 shmget(key_t key,size_t size,int shmflg)函数取得；信号量的 id 可以通过调用 semget (key_t key,int nsems,int semflg)函数取得[6]。在调用这两个函数时使用相同的 key 值，达到共享内存的目的。通过调用 key_t ftok(const char* path,int id)函数来产生 key 值，各进程都用同样的参数来调用此函数，得到相同的 key 值。采取设置一个信号量的方法来控制多个服务进程并发访问这些共享内存，将访问共享内存的服务进程作为临界区来处理。服务进程进入时用 p()操作取得锁，退出时用 v()操作释放锁。

结论 移动数据业务平台(MDSP)是 3G 数据业务网络解决方案中的核心部件，也是 2.5G 数据业务网络发展的关键部件。本文设计的以 MDCC 组件为主的 MDSP 平台体系结构清晰，采用了 3G 的一些标准和原则，具有良好的可扩展性和安全性，高效性。经过各种测试和上网初运行，本平台可以在 3G 数据网络上运行。应用于现有 2G、2.5G 移动数据网络可以从根本上解决运营中出现的各种问题，数据业务可以统一、快速地发布和部署，加强了 SP 和业务的管理，杜绝和防止了 SP 欺诈和骚扰用户的行为，提高了运营商的综合竞争力，同时为向 3G 过渡提供了安全的、可升级的、可过渡的解决方案，保护了运营商和 SP 投资。此平台的一些版本已在某些省级移动，电信公司数据业务网络中应用，产生了巨大的商业价值。本平台是一种很好的 3G、2.5G 移动数据网络解决方案。研究和开发其体系结构和其中采用的各项技术具有重要的意义和前景。

参 考 文 献

1 张欣，曲志峰. 移动数据增值业务运营现状及新一代增值业务平台 OSE[J]. 数据通信，2004，4：59～62
2 何浩，荆涛，李兴华，张思东. 服务提供商统一网关的研究[J]. 计算机工程与设计，2005，26(1)：129～131
3 中国移动通信集团公司互联网短信网关接口协议[S]. CMPP V3.0.0(China Mobile Point to Point)
4 田玉林. NGSP－3G 时代的通信运营平台[J]. 移动通信，2004，10：73～75
5 Merging IP and Wireless Networks. http://www.comsoc.org/livepubs/pci/Public/2003/oct/index.html
6 Richard Stevens W. UNIX 网络编程[M]. 北京：清华大学出版社，1999. 35～39