

基于网站日志挖掘的频繁遍历主干子网发现^①

Mining Web Logs to Discover Major Web Site Structure

鲍 钰 (华东师范大学 软件学院 上海 200062)

摘 要: WWW 上用户的访问路径信息会被记录在 WEB 服务器的日志记录中, 分析这些日志并挖掘出用户的主要行为模式, 可以提取出 WEB 网站被频繁访问的主干部分。本文首先将原始日志信息整理成目标页前向访问路径集 TUPD (Target Pages User Forward Access Path Dataset), 然后在 TUPD 上生成加权网站结构多维树 WWSSMT (Weighted Web Site Structure Multi-Tree), 最后引入决策频度阈值 S , 删除所有 WWSSMT 中 $weight < S$ 的子结点分支, 最终实现了此 WEB 站点的频繁访问主干子网的发现。实验证明, 挖掘获得的主干子网包含了某电子商务网站的最主要热销产品。

关键词: Web 日志 目标页前向访问路径集 加权网站结构多维树 主干子网 访问模式

原始 WEB 日志虽然包含了大量访问者的浏览信息, 但有些日志是随镶嵌在网页中的图片和脚本产生的, 网络搜索引擎使用的 WEB 蜘蛛, 软机器人或智能代理也会产生访问日志, 这些日志对分析用户的访问情况都是无用的, 应该过滤掉。另外, 所有用户的访问日志会零星掺杂在一起, 所以还必须对日志按照用户进行分组, 这便是用户识别^[1]。同一用户对 WEB 站点会有多次的访问, 因此又需要在同一用户的所有访问日志上进行会话分组, 经过会话识别后的一条条记录, 我们称之为网站日志路径 WLP (Web Log Path)。

1 WEB 站点访问模式分析

对于确定的网站层次结构, 如图 1, 访问者的一次旅行路径信息表示如下 TP (True Path)^[2]: 1A, 2B, 3E, 2B, 3F, 2B, 1A, 2D, 7。访问者经过两次回溯 (3E 和 3F) 才最终找到了目标页面 7。这里, 1A 为路径起始点, 3E 和 3F 为回溯点, 7 为路径终止点。因此, 可以将访问者的执行步骤定义如下:

```
While (C is not Target Page Tn) do
  If Exist Ti ? Link(C) and Seem_Under(Tn, Ti)
do Follow(Ti)
  // Link(C) 表示所有从 C 引出的超链,
  Seem_Under(Tn, Ti) 表示 Tn 可能在 Ti 下
```

// Follow(Ti) 表示沿着 Ti 向下寻找

Else BackTrack

但是, 如果页面被浏览器缓存后, 则在 WEB 日志中, 仅记录了如下访问路径 WLP (Web Log Path): 1A, 2B, 3E, 3F, 2D, 7。为了得到完整的访问路径信息 TP (True Path): 1A, 2B, 3E, 2B, 3F, 2B, 1A, 2D, 7 可采用下面两种方法:

① 通过在网页的 <Meta> 标志中设置过期时间, 使缓存失效, 但这种做法显然会增加网站的流量负荷。

② 根据网站的物理链接结构, 采用如下 GTPFWLP (Get True Path From Web Log Path) 算法获得完整的访问路径信息。

GTPFWLP (Get True Path From Web Log Path) 算法主要思想:

① 扫描 WLP (Web Log Path), 对于 WLP 中任何连续两点 $WiPi$ 和 $Wi+1Pi+1$, 若在它们之间无物理链接, 则说明访问者一定在 $WiPi$ 处进行了回溯, 考察 $Wi-1Pi-1$ 和 $WiPi$, 若有 $Wi-1 < Wi$ 则将 $Wi-1Pi-1$ 插入到 $WiPi$ 之后, 否则在 $W1P1 \dots Wi-1Pi-1$ 中寻找 $Wk-1Pk-1$ 和 $WkPk$, 使得 $WkPk = WiPi$ 并且 $Wk-1 < Wk$, 将 $Wk-1Pk-1$ 插入到 $WiPi$ 之后。

② 执行过程①, 直到对于 WLP 中的任何连续的两点 $WiPi$ 和 $Wi+1Pi+1$, 在它们之间均有物理链接。此

① 基金项目: 国家 973 计划项目 (2005CB321904)

收稿时间: 2008-09-24

时 WLP 即被改造为 TP。

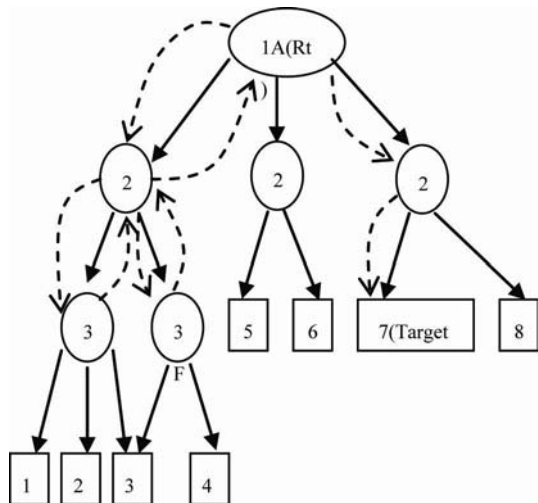


图 1 访问模式图

2 目标页集合TargetSet的确定

对于用户的一次旅行路径 TPi: 1A,2B,3E, 2B, 3F,2B,1A,2D,7。目标页的结点可能为：

- ①路径回溯点
- ②路径终止点。

下面给出一个确定路径回溯点的算法 CBP(Confirm Backtrack Point):

对于集合 TP{P1, P2, ……Pn},Pn 为路径终止点
B: = Ø //B 为一次访问路径 TP 的回溯点集合
For i:=2 to n-2
Begin
If(Pi-1 = Pi+1) add Pi to B // Pi 是回溯点
End

访问者回溯可能是已经浏览到了目标页，然后回到上层目录去寻找其它目标页。也可能是走错路，然后快速回溯。所以，对于回溯点集合 B 中每个结点，还需引入一个时间阈值去判断^[3]，当在此回溯点的停留时间小于此时间阈值时，认为其是真正的回溯点，否则就是一个目标页结点，应将其加入到此条 TP 对应的目标页集合 TargetSet_TPi 中。

同理,当访问者最后在路径终止点 Pn 关闭了浏览器后，并不能表示其成功地找到了目标页面，关闭有可能是放弃寻找的结果。所以也要引入一个时间阈值 T_KEY，对于页面停留时间(STORE_TIME)大于此 T_KEY 的才确定为目标页，并将其加入到此条 TP 的目标页集合 TargetSet_TPi 中。

关于 T_KEY 的设置可以采用不同的方法，例如 T_KEY 可设置为（页面文章字数/最大阅读速度）。当然，T_KEY 大小设置的不同，必然导致最后分析结果有所不同。

对于 TPi: 1A,2B,3E,2B,3F,2B,1A,2D,7，我们假设这里产生的 TargetSet_TPi 为{3F,7}。

3 前向访问路径截取算法TFAP

获得 TP 后，这只是 WEB 用户的一次网上旅行记录，有些途中子路径是有意识主动访问的^[4]，如 1A,2B,3E 以及 1A,2D,7。而有些则是为了路由的需要而进行的被动回溯，如 3F,2B,1A。下面给出一个前向访问路径截取算法 TFAP(Trim Forward Access Path)，从一条 TP 中截取出数条前向访问子路径 FAP(Forward Access Path)，即主动访问子路径，再将其插入到用户前向访问路径集 UPD(User Forward Access Path Dataset)中。

TFAP 算法思想：

```
①Set i=1 and Set SUBPATH=NULL
//SUBPATH STORES FORWARD ACCESS PATH
Add W1P1 To SUBPATH
// W1P1 IS START POINT OF TP
②扫描 TP(True Path)，对于 TP 中任何连续两点
WiPi 和 Wi+1Pi+1
If Wi< Wi+1 Then
Set Flag=1
// FLAG=1 INDICATES A FORWARD TRAVERSAL
Append Wi+1Pi+1 To SUBPATH
Else
If Flag==1 Then Insert SUBPATH Into UDP
Discard all the Items WkPk , with Wk< Wi+1 , in string SUBPATH
Set Flag=0
End If
③If Wi+1Pi+1 is EndPoint of TP Then
Insert SUBPATH Into UDP
Exit
Else
i = i + 1
Goto ②
```

End If

例如对于 TP: 1A,2B,3E,2B,3F,2B,1A,2D,7，算法 TFAP 执行过程如下(表 1):

表 1 算法 TFAP 执行过程

STRING SUBPATH	OUTPUT TO UPD
1A,2B	
1A,2B,3E	
1A,2B	1A,2B,3E
1A,2B,3F	
1A,2B	1A,2B,3F
1A	
1A,2D	
1A,2D,7	1A,2D, 7(END)

对于 TP: 1A,2B,3E,2B,3F,2B,1A,2D,7，通过 TFAP 算法，获得三条前向访问子路径 FAP(Forward Access Path):

- * 1A,2B,3E
- * 1A,2B,3F
- * 1A,2D, 7

根据前面目标页集合 TargetSet_TPi={3F,7}，这里应只保留含有目标页结点的前向访问子路径 TFAP(Target Pages Forward Access Path)。

- * 1A,2B,3F
- * 1A,2D, 7

并将其加入到目标页前向访问路径集 TUPD(Target Pages User Forward Access Path Dataset)中。

4 TUPD上频繁主干子网的挖掘

若对于图 1 的网站层次结构，存在如下的几条 WLP:

- WLP1: 1A,2B,3E,3F,2D,7
- WLP2: 1A,2B,3F, 2D,7
- WLP3: 1A,2C,5

分别对这些 WLP 实行 GTPFWLP 算法，以及目标页集合发现算法^[5]，可得真实访问路径以及每条路径的 TargetSet_TP 如下:

- TP1:1A,2B,3E,2B,3F,2B,1A,2D,7
TargetSet_TP1={3F,7}
- TP2:1A,2B,3F,2B,1A,2D,7 TargetSet_TP2={7}

TP3:1A,2C,5 TargetSet_TP3={5}

三条 TP 分别调用 TFAP 算法执行以后,可得如下事务数据库 UPD，见表 2。

表 2 事务数据库 UPD

TID	TPID	FAP	ISEND
10001	1	1A,2B,3E	NO
10002	1	1A,2B,3F	NO
10003	1	1A,2D,7	YES
10004	2	1A,2B,3F	NO
10005	2	1A,2D,7	YES
10006	3	1A,2C,5	YES

根据每条 TP 的目标页集合 TargetSet_TPi，只保留含有目标页结点的前向访问子路径 TFAP，可得如下表 3 目标页前向访问路径集 TUPD。

表 3 目标页前向访问路径集 TUPD

TID	TPID	TFAP	ISEND
10002	1	1A,2B,3F	NO
10003	1	1A,2D,7	YES
10005	2	1A,2D,7	YES
10006	3	1A,2C,5	YES

之后，在 TUPD 上根据每条目标页前向访问子路径记录 TFAP，利用如下的多维树结点进行 WWSSMT(Weighted Web Site Structure Multi-Tree)加权网站结构多维树的生成:

```
const int M = 1000;  
//每个结点下方最多有 1000 个链接结点  
struct Trie_node {  
    // data members  
    string data; //结点名称  
    int weight; //结点出现次数  
    Trie_node * branch[M];  
    //下方儿子结点链接指针  
    Trie_node(); //constructors  
};
```

WWSSMT 加权网站结构多维树具体生成算法:

对于一条 FAP{P1, P2, ……Pn},其中 Pn 为前向访问子路径终止点，Root 为 WWSSMT 加权网站结构多维树根结点指针，初始为 NULL。

```

Root = new Trie_node(P1);
//路径起点 P1 为 WEB 站点首页
Cur = Root; //Cur 为当前遍历到结点指针
for(i=2; i<=n; i++){
//从结点 P2 依次扫描到 Pn
    Pos= ExistChild(Cur,Pi); /* 当前 Cur 结点下
M 个子结点中有 Pi, Pos 返回此子结点位置索引值。
否则, Pos 返回-1 */
    if(Pos>=0)AddWeight(Cur,Pos);/* 有 Pi 子
结点, 此子结点出现次数 weight 增 1 */
    else Pos=CreateChild(Cur, Pi);/*无 Pi 子结
点, 创建新的子结点, 初始 weight 为 1, Pos 返回此
新子结点指针索引 */
    Cur=Cur->branch[Pos]; /* 将当前 Cur 指
针移到新的位置 */
}

```

5 实验结果分析

根据表 3 中的目标页前向访问路径集 TUPD, 运用此 WWSSMT 加权网站结构多维树生成算法产生如下图 2 的多维树, 树中每个结点里的加权值即为其出现的频度。在某中等规模的电子商务网站事务数据库上, 我们对其数据库服务器上半年内产生的交易记录, 采用文中的方法进行了实验。对目标页前向访问路径集 TUPD 上所有记录, 运用此 WWSSMT 加权网站结构多维树生成算法, 最终生成完整的加权网站结构多维树。引入决策频度阈值 S , 删除所有 $\text{weight} < S$ 的子结点分支, 最终实了此电子商务 WEB 站点的频繁主干子网的发现。频繁遍历主干子网的发现, 不仅能提供精简子网方便 WEB 用户的访问过程, 节省宝贵的网络带宽, 而且此主干子网也基本覆盖了该电子商务网站最主要的热销产品。

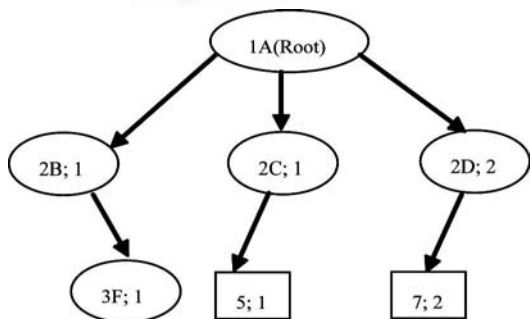


图 2 加权网站结构多维树

6 总结和展望

本文通过对 WEB 日志信息的若干分析挖掘算法, 首先将原始日志信息整理成目标页前向访问路径集 TUPD, 然后在 TUPD 上生成加权网站结构多维树 WWSSMT, 最后引入决策频度阈值 S , 删除所有 WWSSMT 中 $\text{weight} < S$ 的子结点分支, 获得 WEB 站点的频繁主干子网, 从而为生成 3G 应用的 WAP 网站提供参考。我们还在某中等规模的电子商务网站事务数据库上, 采用此方法对其半年的原始数据进行处理, 获得的主干子网包含了本电子商务网站的主要热销产品。实验结果表明, 该方法具有很好的适用推广性。不过, 在实验中我们也发现决策频度阈值 S 对频繁访问主干子网的影响甚大, 文中是根据经验及折中估算的方法来调整参数, 在未来的工作中, 我们希望能够引入更好的方法来解决上述问题, 并进一步提高挖掘算法的运行效率。

参考文献

- 1 Zaki MJ. SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning Journal, 2001, 42(1/2):31 – 60.
- 2 Martinez E, Karamcheti V. A prediction Model for User Access Sequences. WEBKDD Workshop: Web Mining for Usage Patterns and User Profiles, July 2002.
- 3 Pei J, Han J, Mortazavi-asl B, Zhu H. Mining access patterns efficiently from web logs. Proc. of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. April 2000:396 – 407.
- 4 Nanopoulos A, Katsaros D, Manolopoulos Y. Effective prediction of web-user accesses: a data mining approach. Proc. Int. Conf. on Mining Log Data Across All Customer Touchpoints. San Francisco, 2001.
- 5 Pitkow J, Pirolli P. Mining longest repeating subsequences to predict www surfing. Proceedings of the 1999 USENIX Annual Technical Conference, 1999.