第34卷 第10期 计 算 机 工 程 2008年5月
Vol.34 No.10 Computer Engineering May 2008
·软件技术与数据库· 文章编号：1000—3428(2008)10—0070—03 文献标识码：A 中图分类号：TP311

基于非压缩后缀树的在线 PPM 预测模型

班志杰[1,2]，古志民[1]，金 瑜[1]
(1. 北京理工大学计算机科学技术学院，北京 100081；2. 内蒙古大学理工学院电子工程系，呼和浩特 010021)

摘 要：PPM 模型适合预测用户的下一个请求，但已有的 PPM 模型不具备在线性，更新通过重构来实现，不能满足实时更新的要求。该文提出基于非压缩后缀树的在线 PPM 预测模型，采用非压缩后缀树实现增量式在线更新，提高了模型的更新速度。该模型的优点是具备在线性。

关键词：Web 预取；PPM 模型；非压缩后缀树

On-line PPM Prediction Model Based on Non-compact Suffix Tree

BAN Zhi-jie[1,2], GU Zhi-min[1], JIN Yu[1]
(1. School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081;
2. Department of Electronic Engineering, College of Sciences and Technology, Inner Mongolia University, Hohhot 010021)

【Abstract】Prediction by Partial Matching(PPM) models are appropriate for predicting the user's next request, but these models are not on-line and their updates are implemented by rebuilding models which can not meet the real-time update. This paper presents an on-line PPM prediction model based on non-compact suffix tree. The model makes use of non-compact suffix tree to implement the incremental on-line update, and its update speed is improved. This model has the important property of being on-line.

【Key words】Web prefetching; Prediction by Partial Matching(PPM); non-compact suffix tree

1 概述

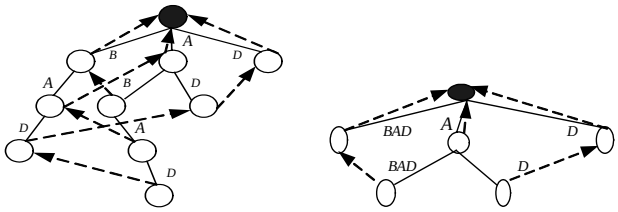
Internet 作为一个全球的、分布的、动态的信息仓库，已成为人们获得日常信息的重要来源。但由于网络用户数的增长，用户的服务质量得不到很好的保证。为减少用户的访问延时，可采用Web缓存和预取 2 种技术。Web缓存利用时间局部性原理[1]，将曾经访问过的文档保存在非原服务器站点。Web预取技术利用空间局部性原理[2]，通过分析用户当前和历史请求，主动预测用户可能浏览的对象，提前将预测的内容取到本地。

Web预取的关键是建立一个有效的预测模型。WWW是一个动态变化的环境，用户的兴趣随着时间的推移而发生变化。因此，Web预测模型应具有在线性，即将用户的最近和当前请求建立进模型，同时将过时的历史访问信息从模型中删除。在线性要求插入新的用户请求和删除过时的信息不能通过重新构造模型来实现，而是在原来训练模型的基础上增量式地进行。研究表明PPM(Prediction by Partial Matching)模型适合建立和预测用户的请求模式[3-5]，但已有的PPM模型不能实时加入新的用户请求和删除过时信息。它们的更新通过隔一段时间重新构造模型来完成，不能充分体现用户浏览模式的改变。通常这些模型采用离线训练方式，模型中存在大量的过时信息，降低了模型的预测准确率。

2 相关知识

2.1 非压缩后缀树

后缀树是传统字符串领域中的一种非常重要的数据结构，由于其高效的索引能力和易于构造的特点而得到广泛的应用。给定长度为n的字符串S，S的后缀树是一棵具有n个叶子节点的树。树中的每一条边上都标识字符串S的一个非空子串。由根节点开始到任何一个叶子节点的路径上所有的标识连接起来构成的字符串对应着字符串S的某一个后缀。由同一节点发出的任何 2 条边上所标识的字符串的首字符不允许相同。非压缩后缀树指每一条边上只标识字符串S的一个字符。为了快速构建后缀树，通常在每个节点中都包含一个后缀指针。一个后缀指针从代表串$\{S_1S_2\cdots S_m\}$的节点指向代表串$\{S_2S_3\cdots S_m\}$的节点。图 1 给出了字符串$\{ABAD\}$的非压缩后缀树和后缀树，带有箭头的虚线表示后缀指针，实心的圆点表示根节点。



(a)$\{ABAD\}$的非压缩后缀树 (b)$\{ABAD\}$的后缀树

图 1 $\{ABAD\}$的非压缩后缀树和后缀树

2.2 PPM 预测模型

PPM预测模型是一种上下文统计模型技术[5]。上下文是处于某个字符之前的一个有限字符序列，这个序列的长度称为上下文的阶。模型要记录所有上下文出现的次数。所有的上下文构成PPM模型。上下文的最大阶称为PPM模型的阶。图 2 是请求序列为$\{ABAC\}$, $\{ABD\}$和$\{ECAC\}$的PPM的预测树结构，其中每个节点中的计数为对应上下文的发生次数。例

基金项目：北京理工大学基础研究基金资助项目(0301F18)
作者简介：班志杰(1976－)，女，讲师、博士研究生，主研方向：Web预取和缓存，数据挖掘；古志民，教授、博士生导师；金 瑜，博士研究生
收稿日期：2007-06-22 **E-mail：**wonderful_beer@bit.edu.cn

如，节点$C/2$代表上下文$\{AC\}$被访问了 2 次。

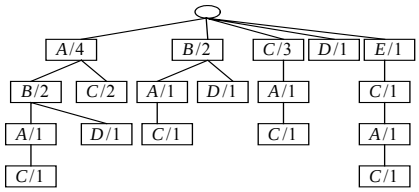


图 2　访问序列为{*ABAC*}, {*ABD*}和{*ECAC*}的 PPM 模型

3　基于非压缩后缀树的在线PPM预测模型

本文提出一种基于非压缩后缀树的在线 PPM(NCST PPM)，其基本思想是将 Web 日志请求看作按时间先后排列的有序序列，并将其划分为多个用户会话，每次加入最新的用户请求，每次删除最早的用户请求。所谓用户会话是在时间间隔小于某个阈值时，由 Web 日志中同一个用户的 IP 发出的连续请求。

基于非压缩后缀树的在线 PPM 预测模型由 3 部分构成：模型的在线添加，模型的在线删除和预测模型。

3.1　模型的在线添加

NCST PPM 模型采用的数据结构是非压缩后缀树。在非压缩后缀树中，每个节点代表了从根到此节点的请求序列，即上下文的长度等于节点的深度，并且每个节点中包含一个后缀指针。为了快速更新和预测，利用额外的一个数据结构记录活动用户会话的最长上下文。所谓活动用户会话指包含当前用户请求的用户会话。一个新的请求 A 按如下方法加入到模型中：

(1)如果 A 属于一个新的用户会话 S，则标识 S 为活动用户会话，并置同一用户原来活动的用户会话为不活动，同时将对应的最长上下文指针指向根节点。

(2)检查A所属的活动用户会话的最长上下文及所有后缀的子节点是否包含 A。

(3)如果存在这样的节点(模型中已存在这样的上下文)，将这个上下文的计数增 1；否则，创建一个新的子节点，将此新节点的计数设置为 1。

(4)改变对应用户会话的最长上下文指针，使其指向新的最长上下文。

(5)若新创建了节点，则使其后缀指针指向相应的后缀，即最长上下文指针指向次长的上下文，以此类推。

算法描述如下：

算法 1　Algorithm ModelAppendOnline(A, T)

```
输入 新的请求 A，预测树 T
输出 预测树 T
Begin
i=Get_Active_Context( A ,T);
//获得 A 所在的活动用户会话
p=Active_Context[i];
//p 指向 A 所在活动用户会话的最长上下文
flag=0;Temp_Suffix=0;
While (p)
Begin
If (p 的某个子节点 q 代表 A)
q 的发生次数加 1；
Else
Begin
构建代表 A 的子节点 q；
节点 q 的发生次数设置为 1；
End
If (flag==0) Active_Context[i]=q;
flag=1;
p=p→Suffix_Pointer;
If(Temp_Suffix!=0) Temp_Suffix→Suffix_Pointer=q;
Temp_Suffix=q;
End
Return T;
End
```

假设请求序列为$\{AABABCBCD\}$，这些请求所属的用户表示为{121321333}，用标号 1, 2, 3 表示请求分别属于用户 1、用户 2 和用户 3。划分用户会话后，用户 1 的会话为$\{ABC\}$、用户 2 的会话为$\{AB\}$和用户 3 的会话为$\{ABCD\}$。图 3 为添加序列$\{AABABCBCD\}$到 NCST PPM 模型的过程。

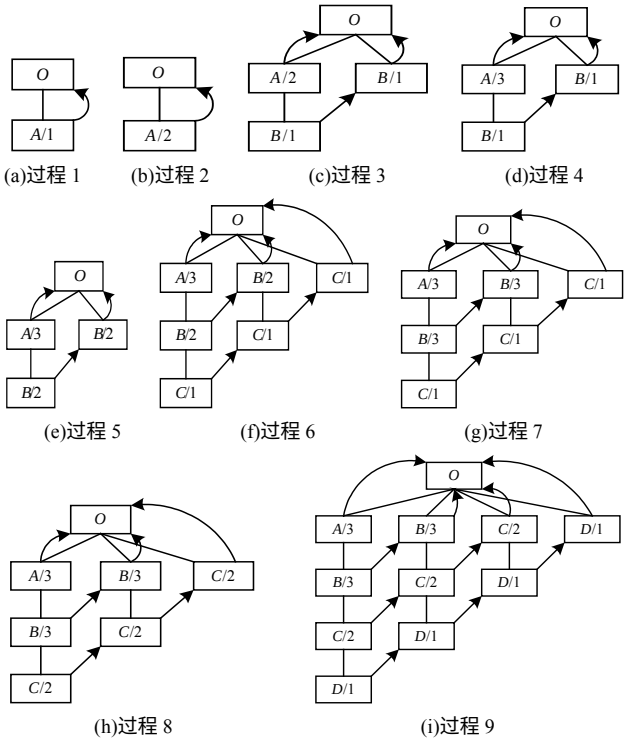


图 3　添加序列{*AABABCBCD*}到 NCST PPM 模型的过程

在图 3 中，带箭头的线表示后缀指针，直线段表示非压缩后缀树本身，每一个步骤代表顺序插入序列$\{AABABCBCD\}$中的一个请求，图 3(h)解释插入的过程。在图 3 (h)中插入序列$\{AABABCBCD\}$的第 8 个访问 C，这个访问属于用户 3，所对应的当前用户会话的活动上文是$\{AB\}$, $\{B\}$和根节点，则将上下文$\{ABC\}$, $\{BC\}$和$\{C\}$所对应节点的计数加 1。

3.2　模型的在线删除

在线删除算法的基本思想是每次删除的总是存在于模型中的最早的用户请求 A。因此，在模型中只存在 A 的下文，而不存在 A 的上文，因为 A 是最早的用户请求。因此，在删除用户请求 A 时，只需从模型中删除 A 所在用户会话的 A 的下文。利用如下方法删除 A：

(1)在根节点下找到A所对应的子节点D_A。

(2)从D_A开始，将A所属用户会话的对应所有下文节点的计数减 1，删除计数值为 0 的节点。

算法描述如下：

算法 2　Algorithm ModelDeleteOnline(A, T)

```
输入 最早的用户请求 A，预测树 T
```

输出 预测树 T

Begin

Begin

S=A$X_1X_2\cdots X_n$;//S代表从A开始到A所在用户会话结束的访问//序列,假设标号从 1 开始

在根节点 T 下找到代表 A 的节点 p；

j=1;

While (p)

Begin

push(p);

在 p 下找代表请求 S[j+1]的节点 q；

If (q!=NULL)

Begin

push(q);

p=q;

End

j++；

End

While(栈不空)

Begin

p=pop();

p→count=p→count-1;

If(p→count==0) delete p;

End

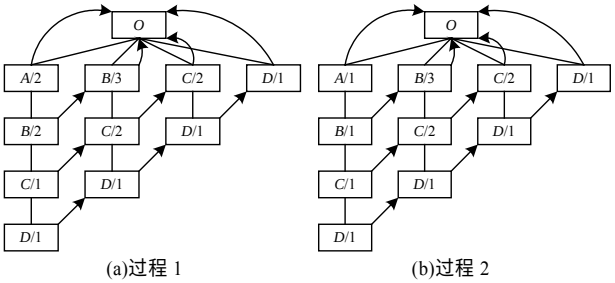
End

Return T；

End

这里应注意在删除的过程中是否删除了那些不是最长上下文的节点，因为如果删除的节点不是最长的上下文，则包含它并比它长的上下文的某个后缀将不在预测树中，这是不允许出现的。下文证明被删除节点对应的总是最长的上下文：假设即将删除节点所对应的上下文是 Y，也就是此节点中的计数为 0。假设 PPM 模型中存在包含 Y 的上下文为 XYZ，下面只须证明 X 和 Z 都是空。从上面的删除算法可以看出实质上每次删除的是叶子节点，则 Z 必为空；每次删除的总是最早的用户请求 A，并且删除节点的计数为 0，这说明从 A 开始的所有用户会话中仅包含一个上下文 Y，同时 A 是没有上文的；并且，其他的用户会话中一定不会包含上下文 Y，因为若包含，则删除节点的计数不能为 0，所以 X 必定为空。

图 4 为从图 3(i)中依次删除序列{*AABABCBCD*}的前 2 个请求的过程。



(a)过程 1　　(b)过程 2

图 4　依次删除序列{*AABABCBCDC*}的前 2 个请求

在图 4(a)中，删除的是序列{*AABABCBCD*}的第 1 个请求 A，这个请求属于用户 1，因此，它的下文是序列{*BC*}，则将上下文{*ABC*}, {*AB*}和{*A*}所对应节点的计数都减 1 ;在图 4(b)中，删除的是序列{*AABABCBCD*}的第 2 个请求 A，这个请求 A 属于用户 2，它的下文是序列{*B*}，则将上下文{*AB*}和{*A*}所对应节点的计数都减 1。

3.3　预测模型

基于 3.1 节和 3.2 节中构造的预测模型，可以对用户的下一个请求进行预测。预测算法描述如下：

算法 3　ModelPrediction(A, Threshold,T)

输入 当前请求 A，预测概率阈值 Threshold，预测树 T

输出 预测结果

Begin

i=Get_Active_Context(A);

p=Active_Context[i];

while(p→Child==NULL&&p!=T) p=p→Suffix_Pointer;

If (p!=T&&p) 在 p 的子节点中，查找预测概率最大，且大于 Threshold 的节点 q;

Else

Return “No Prediction”;

Return q;

End

4　结束语

本文提出基于非压缩后缀树的在线 PPM 预测模型，给出相关实现算法，采用非压缩后缀树实现了 PPM 预测模型的在线性，在原有模型的基础上加入新的用户请求和删除过时的用户请求，满足了实时更新的要求。此模型具有增量式在线更新特性，适用于任何多序列输入的在线 PPM 模型。

参考文献

[1] Aniket M, Anirban M, Williamson C. Locality Characteristics of Web Streams Revisited[C]//Proc. of the SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems. Philadelphia, USA: [s. n.], 2005.

[2] Nanopoulos A, Katsaros D, Manolopoulos Y. A Data Mining Algorithm for Generalized Web Prefetching[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 5(5): 1155-1169.

[3] Levene M, Poulovassilis A. Web Dynamics: Adapting to Change in Content, Size, Topology and Use[M]. Berlin, Germany: Springer, 2004.

[4] Pitkow J, Pirolli P. Mining Longest Repeating Subsequences to Predict World Wide Web Surfing[C]//Proc. of the 2nd USENIX Symposium on Internet Technologies and Systems. Boulder, CO, USA: [s. n.], 1999.

[5] Palpanas T, Mendelzon A. Web Prefetching Using Partial Match Prediction[C]//Proceedings of the 4th Web Caching Workshop. San Diego, California, USA: [s. n.], 1999.