基于决策树分类的智能缓存模型研究

范新灿
(深圳职业技术学院 电信学院，广东 深圳 518055)

摘要：Web 缓存技术是互联网提升访问性能的有效手段。构建了基于决策树的智能缓存数据挖掘模型，通过数据建模、NextAccess 的离散化、构造决策树所用的观察属性集和权重，从而对数据进行缓存处理。实验证明其性能得到了优化。

关键词：Web 缓存；决策树；替换算法

中图分类号：TP311 **文献标识码**：A **文章编号**：1674-7720(2011)04-0011-03

Intelligent model of Web cache based decision tree classification

Fan Xincan
(Electronics &Information Engineering, Shenzhen PolyTechnic, Shenzhen 518055, China)

Abstract: Web cache performance of the Internet to enhance access to effective means, we construct the decision tree based data mining model of intelligent caching, through data modeling, NextAccess discretization of the observation used in constructing a decision tree and the weight set of attributes, so the data Caching. Experiments show that performance has been optimized.

Key words: Web cache; Decision Tree; replacement algorithm

1 Web 缓存技术

互联网已经渗透进人们生活的方方面面，Web 成为获取、发布、加工和处理信息的重要平台，Web 用户的快速增长，导致 Web 流量的爆炸性增长，Internet 带宽产生拥挤，出现访问延迟、通信错误增多，服务器过载等一系列问题，网络带宽的提高已经跟不上用户数量增长的速度，单纯利用增加带宽来解决速度迟缓问题不具有伸缩性，费用也相当昂贵。

获取一个 Web 文档的代价取决于该文档的字节数、传输中链接可获得的带宽以及中间经过的网段个数，若能将文档的复本从原始服务器缓存到离用户较近的机器中，显然可以大大缩短访问的距离，不仅可以减少检索延迟，还可以减少网络负载。Web 缓存技术是一种避免 Web 服务瓶颈、缩减信息流量、提高可伸缩性的手段，是最常用、经济的解决网络拥塞和服务器过载的方法。利用 Cache 技术，复制用户经常访问的内容，将其保存在缓存服务器中，降低了主干网络冗余带宽流量和原始服务器的负载压力，减少文件在网络上的重复传输，可降低网络带宽的浪费，减轻 Web 服务器的负载，最终降低用户的等待时间。

经常被访问的文件被缓存到了临近的代理中，从而避免了从远端的服务器上传输数据，使传输时间最小化。由于网络流量的缩减，没有缓存的文件会相对更快地在网络中传输，因而服务器响应的速度也得到了提高，这些工作负荷被整个互联网上的缓存代理分担了，有效地缩减了对网络带宽的消耗，从而降低了网络的流量，缓解了网络拥塞。Web 缓存技术成为互联网建构中广泛应用的技术。

2 决策树

决策树(decision tree)一般都是自上而下生成的。每个决策或事件(即自然状态)都可能引出两个或多个事件，导致不同的结果，把这种决策分支画成图形很像一棵树的枝干，故称决策树。决策树的构成有四个要素：决策节点、方案枝、状态节点和概率枝。

决策树法的决策程序如下：

(1)绘制树状图，根据已知条件排列出各个方案和每一方案的各种自然状态；

(2)将各状态概率及损益值标于概率枝上；

(3)计算各个方案期望值并将其标于该方案对应的状态节点上；

(4)进行剪枝，比较各个方案的期望值，并标于方案枝上，将期望值小的(即劣等方案剪掉)所剩的最后方案为最佳方案。

决策树算法是一种逼近离散函数值的方法。决策树

算法具有分类精度高、形成的模式简单、对噪声数据有很好的健壮性等优点，因而是目前应用最为广泛的归纳推理算法之一，在数据挖掘中受到研究者的广泛关注。

本文研究的智能缓存模型采用 GATree 决策树算法，该算法是用遗传算法优化产生的决策树算法。采用二叉树结构来表达决策树，每个节点有两个不同节点，每个节点有随机值，选择一个随机的属性，如果其是离散的，则自由选择值；如果这个属性是连续的，则随机选择最小最大值范围之内的一个整数值，这样可以减少搜索空间的范围。

算法的基本形式引入了变异和交叉操作的最小范围，变异操作选择的是期望生成的树的随机节点，替代了节点的具有随机选择值的测试值，当随机节点是叶子节点时，替代了具有新的随机选择类的意境设置好的类。交叉操作选择两个随机节点并且交换这些节点的子树，不会影响决策树的连贯性。

3 缓存模型构建

设计基于决策树的智能缓存模型模拟器，如图 1 所示，模型总体分为构建数据建模、模拟器和缓存输出模块。模拟器是模型的关键，分为 NextAccess 离散化、构造决策树、权重分配和决策树输出 4 个模块。

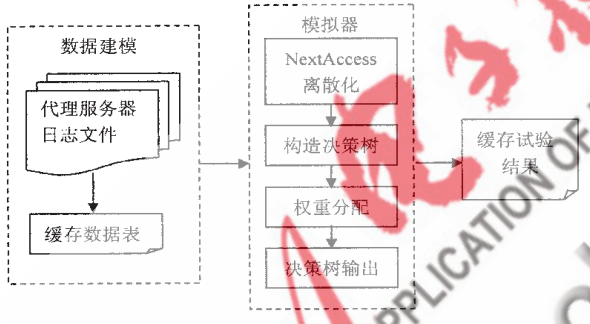

图 1 基于决策树的智能缓存模型

模型首先读取 Web 日志中的记录，并将其进行数据建模。根据 Web 请求序列数据请求对象，将产生的数据存入缓存数据表中。根据数据流的输入，构建决策树智能缓存策略的数据挖掘模型。首先将同一 URL 下次被访问前接受的请求总数（NextAccess）作为分类的目标，进行离散化，构建决策树。权重分配模块采用典型的替换算法 LRU 进行页面替换，当缓存中没有足够的容量来容纳新来的 Web 对象时，调用替换算法做出替换决定。根据当前请求序列的预测结果，更新决策树节点信息，观察属性集，计算预取阈值，进行决策树输出。

3.1 数据建模 缓存数据表

智能模型设计一个 Web 数据表，数据来源是代理服务器日志文件中的数据，产生包括预处理和编码来实现数据选择、清洗和数据转换。设计数据表 tb_cache，关键字段定义如下：

Ndir：URL 的目录层

FirstDir：URL 的第一层目录

NextAccess：同一 URL 下次被访问前接受的请求总数

LastAccess：同一 URL 上次被访问到当前的请求总数

FileExtension：请求的 URL 文件的文件名后缀的代码标识

Size：返回客户端的字节

数据表的每一行数据存储代理服务器的一个事务，但 Web 文档可被缓存必须是 HTTP 协议、是 GET 请求、请求中没有“?”和 HTTP 响应码是 200，这些数据需要在模型中进行过滤清洗，把相应事务导入数据库中。

3.2 构建基于决策树分类的模拟器

数据表字段 NextAccess 存储的是同一 URL 下次被访问前接受的请求总数，为了把决策树作为智能缓存策略的数据挖掘模型，需要将 NextAccess 离散化、构造决策树所用的观察属性集和权重分配算法。下面作几个定义：

ORCLCaches(s)：缓存大小为 s 的使用 ORCL 缓存策略的 Web 缓存系统；

ORCLAvgDSize(s)：Web 缓存中的平均文档的大小；

ORCLTertile(t,s),t{1,2,3}：缓存存储状态为 t*33.3% 时的文档数；

ORCLMax(s)：缓存满时的个体数，等于 ORCLTertiles(3,s)。

3.2.1 NextAccess 的离散化

在决策树模型设计中，将 NextAccess 作为分类的目标，利用决策树作为一个分类器，预测 NextAccess 的值，将值离散化到几个类别中，定义如下：

Class0：NextAccess[1,ORCLTertile(1,s)];

Class1：NextAccess[ORCLTertile(1,s),ORCLTertile(2,s)];

Class2：NextAccess[ORCLTertile(2,s),ORCLTertile(3,s)];

Class3：NextAccess[ORCLTertile(3,s),ORCLMax(s)];

缓存系统在经过 NextAccess 次请求后，可能成功缓存某一资源，这个概率依赖于缓存系统中的实体个数，当 NextAccess 在[1,ORCLTertile(1,s)]之间时，经过 NextAccess 次请求后有 66.66%~100% 的可能性；当 NextAccess 在[ORCLTertile(1,s),ORCLTertile(2,s)]之间时，概率下降到 33.33%~66.66%；界于 [ORCLTertile(2,s),ORCLMax(s)]之间，概率下降到 0-33.33%，因此低类值的要给予高优先权。

3.2.2 权重分配

替换策略的权重分配如下：

$W_{LRU}(E_i)=j$，j 为文档 E 的第 j 次访问。

S3 替换策略的权重分配如下：

$W_{s3}(E_i)=j+a(c)*\mathrm{AvgDsize}(s)/E_i.size;c\in\{0,1,2,3\}$；c 为文档 E_i 根据 GATree 算法所在类别。

$a(3)=\mathrm{Max}(s)$；

$a(c+1)=2a(c)$；

$E_i.size$ 为文档 E_i 的大小。

3.2.3 观察属性集与输出决策树

将 Ndir、FirstDir、LastAccess、FileExt、Hour、Size 作为 GATree 算法的观察属性。GATree 输出决策树如下：

```
……
if Ext=jpg then
  |-if Ndir<=5 then
     ||class=0
    |+-class=2
+-if Firdir=-tito then
   |-class=3
   +-class=0
……
```

4 缓存模型试验结果

实验采用网站的真实访问，获得访问日志数据，进行数据预处理，建立缓存数据表，并采用本文提出的基于决策树算法构建模型。仿真实验选择传统的替换算法 LRU 和本文所建立缓存模型进行比较，从缓存性能的指标命中率（*HR*）、字节命中率（*BHR*）、延迟率（*LR*）三个环节进行分析。*HR* 表示用户从缓存中取到的对象数和所获得的总对象数，*BHR* 表示用户从缓存中获取对象的平均字节数和从网上获取的全部字节数的比值，*LR* 表示从服务器下载对象到客户端缓存的总时间。

从如图 2、图 3、图 4 所示，基于决策树的职能缓存模型比传统的 LRU 替换算法具有较高的命中率和字节命中率，并且延迟率较小，可见本文提出的缓存优化模型较传统的算法减少了缓存文件的冗余度，提高了命中率，改善了系统性能。

图 2 决策树职能模型与 LRU 的命中率比较

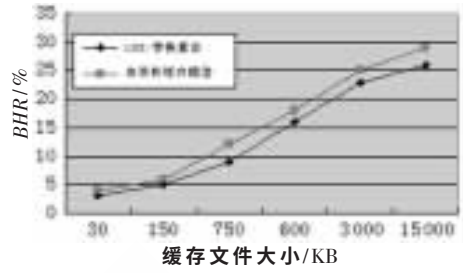

图 3 决策树职能模型与 LRU 的字节命中率比较

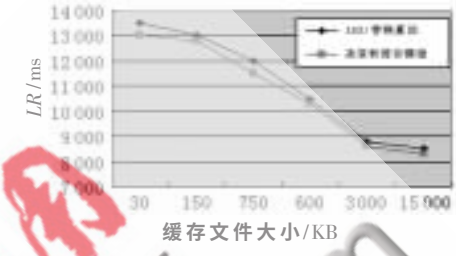

图 4 决策树职能模型与 LRU 的延迟比较

参考文献

[1] 邓磊，陈志刚，黄键，等.基于 AOP 的智能 Web 缓存框架[J]. 计算机工程，2008，34(22)：283-285.

[2] 韩向春，田玉根. 基于预测的 Web 缓存替换算法[J].计算机工程与设计，2010，31(1):110-113.

[3] BALAMASH A，KLUNZ M. An overview ofweb caching replacementalgorithms[J].IEEE Communications Surveys and Tutorials，2004，6(2)：44-56.

[4] Ristenpart T Back to the Future：A framework for automatic malware removal and system repair[C]//Proc.of the Annual Computer Security Applications Conference.Miami，FL，USA：IEEE Computer Society，2006.

（收稿日期：2010-12-21）

作者简介：

范新灿，男，1978 年生，硕士，讲师，主要研究方向：XML，Web 服务。