一种新的基于产生式规则的推理树结构

黄务兰
(江苏工业学院,江苏 常州 213164)

摘 要: 根据正向推理机推理的过程,以及专家系统知识库更新频率极小的特点,描述了一种新的基于产生式规则的推理树结构,这种树结构采用分层分类别的存储方法,大大提高了推理机推理效率。把工作的重心放在推理树的插入和删除算法上,以知识库的更新复杂性来换取专家系统推理机运行的速度。

关键词: 专家系统;推理树结构;产生式规则;正向推理
中图分类号: TP18;TP311.2 **文献标识码**: A **文章编号**: 1000-7180(2007)04-076-03

A New Reasoning Tree Structure Based on Production Rule

HUANG Wu-lan
(Jiangsu Polytechnic University, Changzhou 213164, China)

Abstract: According to the course of the forward reasoning and extremely little frequency update for the knowledge-base of the expert system, a kind of new reasoning tree structure based on production rule is proposed, which adopts hierarchy and classification in rule storing and greatly accelerates the reasoning of the reasoning machine. Our approach focuses on an update phase in insertion and deletion and accelerating the inferring of the reasoning machine in expert system at the expense of update efficiency of knowledge-base.

Key words: expert system; reasoning tree structure; production rule; forward reason

1 引言

专家系统主要是由知识库和推理机组成,因此,专家系统的求解速度就和这两部分密切相关[1]。为了提高推理机推理的效率,必须设计一种好的数据结构,加快推理的速度和准确率。在专家系统中,用户使用专家系统也就频繁在使用推理机推理,而推理的过程就是一个不断规则匹配的过程。而且,专家系统在投入使用后,知识库更新的频率远远少于用户使用它进行推理的频率[2-4]。因此,设计一种实现快速搜索的数据结构是完全合理的,而不必过于考虑知识库更新的开销。

2 新的推理树结构

2.1 树结构设计

对一个专家系统来说,如何提高搜索速度或降低搜索开销对整个专家系统的性能是至关重要的。因此,设计一个合理的搜索树结构是非常有必要的。

根据专家系统的推理策略和逻辑推理过程可知,推理过程是有阶段性的或顺序性的。拿食品装袋的综合系统BAGGER来说,推理过程具有阶段性,分为核对订货、大件物品装袋、中件物品装袋、小件物品装袋四个阶段[5]。对于动物识别系统来说,它的推理过程也具有一定的推理顺序性,首先推理确定大范围(是哺乳动物或是鸟类),之后再逐步确定更细节的判定,最后确定具体的动物名。

2.1.1 树结构描述

相应的,规则也应按照对应的顺序存储,以提高推理速度或降低搜索开销。首先考虑产生式规则的一种分层存储策略。同级范围或阶段的规则安排存储在树的同一级结点上。根结点为空,一级结点存储的是确定最大范围的规则,二级结点其次,依此类推,树的叶结点存储的则是能得出最终推理结果的规则。

针对产生式规则及正向推理特性,可设计这样的树结点及实体结构(如图1所示)。对于同一类别的规则集用一个结点来存储,并共用一个指向父结点的指针和规则集代号,这样可大大节省存储空间并提高搜索效率。结点中的每个实体对应于一条产生式规则,它的结构如图1(b)所示。

2.1.2 新树结点的定义

(1) 根结点为空;

(2) 每个非根结点可能含有:

收稿日期: 2006-04-24

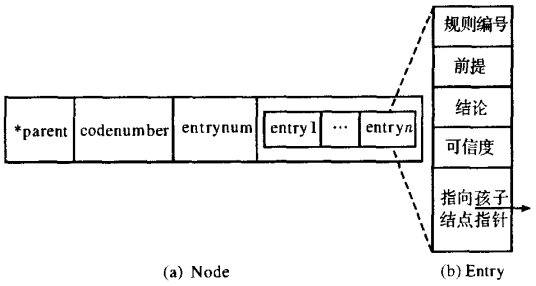

图1　树结点和实体结构

一个指向父结点的指针 D；

一个表示实体集类别的代号 codenumber；

n 个实体 $P_i(1\leqslant i\leqslant n)$。

新的树结构的 C 语言描述如下：

```
#define ENTRYNUM 4;// 结点中最大实体个数，这里暂定为 4
typedef struct NSNode {
    struct NSNode *parent; // 指向父结点的指针
    string codenumber;     // 实体集类别代号
    integer entrynum;      // 结点中实际实体个数
    struct entry p[ENTRYNUM];   // 实体
    struct NSNode *child[ ]; //指向孩子结点的指针，孩子结点可有多个，初始化为 null
    }
```

(3) 其中，每个实体结构在这里表示规则，实体结构定义如下：

```
typedef struct entry
{
  string ruleid;          // 规则编号
  string condition;       // 规则前提
  string conclusion;      // 规则结论
  double CF;              // 可信度
  }
```

假若有如下规则：

R1：如果该动物有毛发，那么它是哺乳动物；

R2：如果该动物能产乳，那么它是哺乳动物；

R3：如果该动物有羽毛，那么它是鸟类动；

R4：如果该动物能飞行，它能生蛋，那么它是鸟类动物；

R5：如果该动物是哺乳动物，它吃肉，那么它是食肉动物；

R6：如果该动物是哺乳动物，它长有爪子，它长有利齿，它眼睛前视，那么它是食肉动物；

R7：如果该动物是哺乳动物，它长有蹄，那么它是有蹄动物；

R8：如果该动物是哺乳动物，它反刍，那么它是有蹄动物，并且是偶蹄动物；

R9：如果动物是食肉动物，它的颜色是黄褐色它有深色的斑点，那么它是猎豹；

R10：如果该动物是食肉动物，它的颜色是黄褐色，它有黑色条纹，那么它是老虎；

R11：如果该动物是有蹄动物，它有长腿，它有长颈，它的颜色是黄褐色，它有深色斑点，那么它是长颈鹿；

R12：如果该动物是有蹄动物，它的颜色是白的它有黑色条纹，那么它是斑马；

R13：如果该动物是鸟类，它不会飞，它有长颈它有长腿，它的颜色是黑色和白色相杂，那么它是鸵鸟；

R14：如果该动物是鸟类，它不能飞行，它能游水，它的颜色是黑色和白色，那么它是企鹅；

R15：如果该动物是鸟类，它善于飞行，那么它是海燕。

对上面的规则进行分层，规则 R1~R4 用于确定生物学分类是哺乳动物或是鸟类，规则 R5~R8 把哺乳动物进一步分类为食肉动物和有蹄动物，规则 R9、R10 对食肉动物进行细分，规则 R11、R12 对有蹄动物进行细分。规则 R13~R15 是对鸟类进行分类的规则。这样可以建立一棵决策树。

为简单起见，假设知识库中有 R1~R15 十五条规则，则该决策树如图 2 所示。

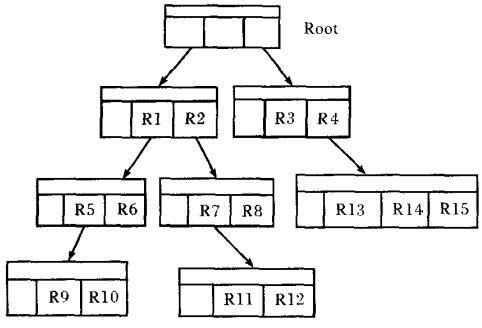

图2　树结构

2.2　插入算法

设计如上的树结构不仅可以减少存储空间，并可以大大加快推理机推理效率，它是以插入或更新代价换来的。当知识库新增一条规则时，如何插入到适当位置是影响以后推理机推理的关键。为此，设计插入算法是核心。在图 1 中，树结点的规则集代号是在这个系统中设计的用于描述规则分类或细化类别代号。它的位数可根据知识库中规则数和

类别数来确定。根据知识库实际类别细化特点,这里每层用两个数字字符来表示这种类别层次,也就是说每层允许有 00~99 共 99 个结点。初始化创建树时兄弟结点之间代号相隔 3 个数字量(考虑到日后插入实体时,结点分裂情况。即使是同一类别层次的实体,由于实体数过多,结点过大,而分裂存储在两个或三个不同的结点上)。并且此代号它具有继承性特点:即是说孩子结点继承父结点的特性,它的代号高位和父结点高位相同,父结点低位用 0 填补。

这里,规则集代号是这种新树结构的索引量,插入实体位置必须以它来衡量。考虑到每个结点允许实体个数,插入后,为避免结点过大,当该结点的实体数超过允许实体数,需对结点进行分裂。

插入算法 insert(r: node, p:,entry,codenumber: string)描述如下:

//插入结点代号为 codenumber 的规则 p 到以 r 为根结点的树中

首先比较待插结点 codenumber(去掉尾部最大偶数个"0") 与树中结点代号 (去掉尾部最大偶数个"0"),找到代号位数相同的结点层,即代插结点与该层结点是同一层次,为兄弟关系,这里分为两种情况:

(1) 如果该代号等于树中已有结点代号,插入实体到该结点中,entrynum 加 1。如果插入后,当该结点的实体数 entrynum 大于结点最大允许实体数 ENTRYNUM,则需对结点进行分裂。i=int(S/2),在原结点中保留(p_0,…,p_{i-1});建新结点,新结点中结点代号增 1 后还原尾部 0,实体集为(p_i,…,p_s);

(2) 如果该代号不等于树中已有结点,则新建一结点。新建结点中指向父结点的指针同该层其它结点,实体属性中指向孩子结点的指针同样同该层其它实体指向孩子结点的指针。

2.3 删除算法

专家系统在投入使用后,需要经常对知识库进行维护,为保证知识库自身的有效性及知识库的规则间的一致性需进行一系列操作。一个不一致或不完备的知识库就可能极大地降低推理效率[6,7]。知识库的维护不仅仅包括添加(插入)规则,还包括删除规则,修改规则这两部分。修改规则操作很简单,只需对原有规则内容进行相应的替换即可。

当一条规则不再适用于专家系统或是错误的,应当即时删除它,以免产生错误推理结果。删除规则的算法描述如下:

delete(r: node,p: entry,codenumber:string)

//在以 r 为根结点的树中, 删除结点代号为 codenumber 的规则 p

(1) 比较待删结点代号 codenumber 与树中结点代号比较,如果找到,跳至(2);如未找到,返回错误信息;

(2) 把规则内容置空并把该结点中 entrynum 减 1;

(3) 比较相邻结点中父结点指针相同的结点(结点存储的规则为同一类别),如果两个结点的实体数相加小于或等于 ENRRYNUM,结点进行合并,后一结点的所有规则移入前一个结点进行存储,并释放后一个结点的空间。合并后的结点结点代号同前一个结点,entrynum 的值为两个结点的实体个数之和。

3 启发式搜索策略

搜索策略有宽度优先搜索和深度优先搜索,它们属于盲目搜索。盲目搜索的效率低,耗费过多的计算空间与时间。启发式搜索搜索效率高,它是通过找到一种方法用于排列待扩展节点的顺序,即选择最有希望的节点加以扩展,在许多情况下,能够通过检测来确定合理的顺序。

在前面数据结构设计中,实际上已经为提供了推理的线索。推理过程是从根结点开始,从上到下的一种层次遍历的过程,最终到达叶子节点即是推理的最终结论。这些启发式信息实际上就是用户输入的初始事实,如果启发信息不够,系统通过用户界面向用户询问获取以确定下一步推理走向。

4 结束语

根据正向推理机推理的过程,以及专家系统知识库更新频率极小的特点,文中设计了一种新的推理树结构,它把规则按推理的顺序分层分类别地进行存储,这样,推理机在进行推理的时候能够依照一定的路线和顺序进行规则匹配,避免了盲目的搜索和匹配过程,大大提高了推理机推理效率,是专家系统设计中一种新的尝试。

参考文献:

[1] Winston P. Artificial intelligence(2nd Ed.). Addison-Wesley, 1984

[2] Joseph M Hellerstein, Jeffrey F Naughton. Generalized

(下转第 81 页)

天的交易量分布是不均匀的。

(6) 在实践中,每个样本只要有每周最高日概率大于 0.259(大于平均分布的 2 倍标准差)就可以说明该日是本周高峰的事实是显著的,在维护和检修中应避开,如没有显著特征,可结合每周分布图和人力资源进行调配。

4 模型部署设计

4.1 流程设计

流程设计图如图 5 所示。

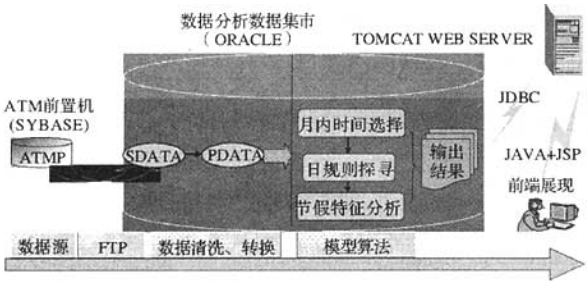

图5 流程设计图

4.2 程序目录结构

与模型相关的 ETL 程序都建立在“ATM_ANY”子系统下,目录结构如图 6 所示。

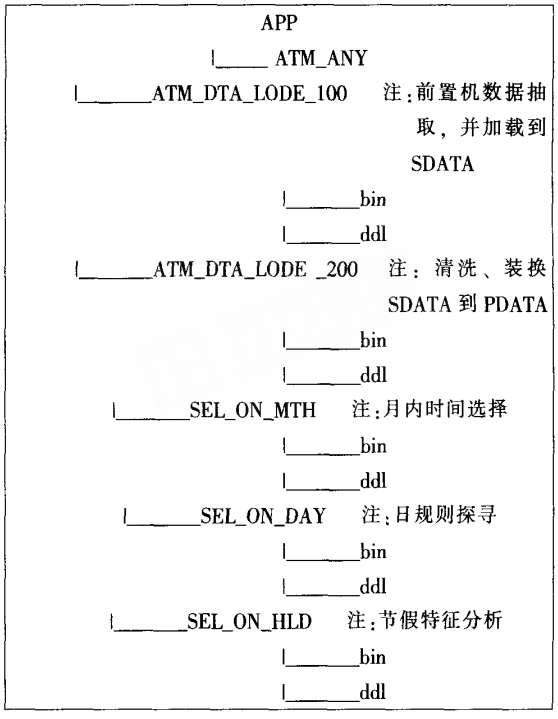

图 6　流程的目录结构

4.3 Web 展现

本模型采用免费软件 TOMCAT 5.0 作为 Web 服务器,服务端 Java BEANS 通过 JDBC 访问 ORACLE 数据库的结果表,生成 HTML 通过 HTTP 协议展现在客户端 Browser。

参考文献:

[1] David S. Moore statistics: concepts and controversies (fifth edition)[M]. W.H.Freeman and Copany, 2001

[2] Philip Hans Franses. Time series models for business and economic forecasting[M]. 中国人民大学出版社, 2001

[3] 张树京, 齐立心. 时间序列分析简明教程 [M]. 清华大学出版社, 2003

[4] 何镇邦, 李桂荣. 概率论与数理统计附随机过程 [M]. 兵器工业出版社, 1992

[5] 聂国星; 徐征, 刘遵雄. 贝叶斯多元线性样条在电力负荷中期预测中的应用 [J]. 微电子学与计算机, 2005, 22(4): 15~18

[6] 张雷, 常天庆, 朱斌, 等. 多步预测方法在故障检测中的应用[J]. 微电子学与计算机, 2005, 22(3): 200~202

作者简介:

周 凯　硕士研究生。研究方向为时间序列及统计学在商业智能中的应用。

(上接第 78 页)

search trees for database systems avi pfeffer Proc. 21st Int. Conf. Very Large Data Bases, VLDB, 1995

[3] Wu L T. Bretschneider comparative analysis of the efficiency of r-tree based indexing strategies for information Retrieval [EB/OL]. http: 1168. 153. 197. 235/CSREA/IKE2016. pdf

[4] 蔡自兴, 徐光祐. 人工智能及其应用(第二版)[M]. 北京: 清华大学出版社, 1996

[5] 吴泉源, 刘江宁. 人工智能与专家系统[M]. 湖南: 国防科技大学版社, 1995

[6] 于会, 李伟华, 陈栋. 专家系统中的知识表示及其实时处理方法研究[J]. 微电子学与计算机, 2005, 22(5): 20~22

[7] 王洪春, 石庆喜, 张勤. 基于因果图的一种推理算法[J]. 微电子学与计算机, 2005, 22(5): 1~4

作者简介:

黄务兰　女,(1979-),硕士研究生。研究方向为人工智能和信息检索技术。