分布式决策树算法研究与实现

戴 南[1,2], 吉根林[1,2]

(1.南京师范大学 数学与计算机科学学院,江苏 南京 210097;
2.苏州大学 江苏省计算机系统信息处理重点实验室,江苏 苏州 215006)

[摘要] 提出了一种基于分布多库环境下的决策树生成算法 DDTA(Distributed Decision Tree Algorithm). 该算法使用基于信息熵增益的思想分割各个分布的、同构训练样本集,各分布站点利用服务器传来的分割属性分割自己的样本集,服务器则通过对所有分布站点传来的信息计算各个属性的信息熵增益得到分割属性. 实验表明 DDTA 算法能对分布同构样本集进行有效决策树挖掘,分布多库环境下生成的决策树是正确的. 与算法 INDUS相比,该算法的通信代价小.

[关键词] 分类,决策树,分布式决策树

[中图分类号]TP311.133.1, [文献标识码]B, [文章编号]1672-1292(2005)04-0046-03

Research and Implementation of ID3 Based on Distributed Database System

DAI Nan[1,2], JI Genlin[1,2]

(1. School of Mathematics and Computer Science, Nanjing Normal University, Jiangsu Nanjing 210097, China;
2. Key Lab of Computer Information Processing of Jiangsu Province, Soochow University, Jiangsu Suzhou 215006, China)

Abstract A new decision tree algorithm DDTA(Distributed Decision Tree Algorithm) based on distributed data repositories is presented in this paper. The algorithm divides each distributed and isomorphic data set uses with the idea of informational entropy increase. Each distributional site divides its own data repository with the dividing properties transmitted by the server, and the server obtains the dividing properties by calculating the informational entropy increase of various properties with information transmitted from all the distributed sites. The experiment shows that DDTA algorithm is effective in excavating distributionally isomorphic data repository with a decision tree, and that the decision tree generated in the environment of distributional multi-data repositories is correct. Compared with the algorithm INDUS, the algorithm has less cost in communication.

Key words classify; decision tree; distributed decision tree

0 引言

随着 Internet的发展与普及,现实中的数据库常常分布存储在网络的各个站点上. 如何对分布存储的数据库挖掘规则与模式成为数据挖掘面临的新问题. 分布式数据挖掘是数据挖掘技术与分布式计算的有机结合,它可以延用传统集中式挖掘算法的思想与技术,但是考虑到系统所处的分布式环境,算法应具备良好的并行性与可伸缩性. 一个好的分布式数据挖掘系统应使得挖掘算法简单、高效、可扩展,生成的模式准确、易理解,结点间的通讯代价最小[1].

分类是数据挖掘研究的重要内容,决策树方法是挖掘分类规则的有效方法. 常用的 ID3算法[2]解决了对集中式数据库的分类规则挖掘问题. 分布式环境下如何生成决策树国内研究较少,文献[3]提出了分布式决策树生成算法 INDUS,它通过传递候选样本集进行分布式决策树挖掘. 本文主要研究分布式决策树生成方法,结合 ID3算法和具有并行性的 SLIQ[4]和 SPRINT[5]算法,提出了一种分布式决策树算法 DDTA(Distributed Decision Tree Algorithm),它应用 ID3算法按属性的信息熵增益分割训练样本集的思想,引入基于类别的属性表 ACL,通过传递基于类别的属性表来分布式生成决策树,

收稿日期: 2005-05-28
基金项目: 江苏省重点实验室开放基金资助项目(KJS03064).
作者简介: 戴 南(1979-),女,助教,主要从事数据挖掘方向的教学与研究. E-mail: dainan@njnu.edu.cn

能对随机分布的具有相同属性结构的同构训练样本集进行分布式挖掘, 在分布多库环境下生成的决策树与对分布多库的并集集中式挖掘生成的决策树完全相同, 与算法 INDUS相比, 通信代价小.

1　相关概念

ID3算法是生成决策树的经典算法, 它的基本思想是根据训练样本集属性的信息熵增益来分割样本集, 具有最大信息熵增益的属性被定义为分割属性, 按照它的取值来划分样本, 由于事先进行了预处理, 每个属性的取值都是离散的. 每次分割得到的一个子样本集对应决策树的一个节点, 分割属性的一种取值成为父节点到子节点的一条边. 按此规则分割所有子样本集, 直至没有子样本集或子样本集中所有样本均属于同一类别.

定义 **1**　训练样本集 S 是一个四元组 $\langle U, A \cup D, V, f\rangle$, 其中 U 是一组样本对象的非空有限集合. 设有 n 个样本, U 可表示为: $U = \{x_1, x_2, \cdots, x_n\}$. $A \cup D$ 为属性的有限集合, $A = \{a_1, a_2, \cdots, a_t\}$ 表示具有 t 个条件属性的条件属性集, D 表示决策属性集, 且 $A \cap D = \phi$, D 只有一个类别属性, $D = \{C\}$; $V = \bigcup_{a \in A \cup D} V_a$, V_a 为属性 a 的值域集. f: $U \times (A \cup D) \rightarrow V$, f 为信息函数, 定义样本的属性值, 即对 $\forall x \in U$, $a \in (A \cup D)$, 有 $f(x, a) \in V_a$.

定义 **2**　对给定的训练样本集 S, V_c 是类别属性 C 的值域集, 则 $V_c = \{c_1, c_2, \cdots, c_m\}$ 表示类别属性有 m 个不同取值. $S_i = \{x \mid x \in U$, 且 $f(x, C) = c_i\}$, 则样本集 S 的信息熵总值 $I(S_1, S_2, \cdots, S_m) = -\sum_{i=1}^{m} p_i \log_2(p_i)$, 其中, p_i 是任意样本属于 c_i 的概率.

定义 **3**　设训练样本集 S 中条件属性 a_i 具有 v 个不同取值, 则 $V_{a_i} = \{a_{i1}, a_{i2}, \cdots, a_{iv}\}$ 是属性 a_i 的值域集. 用属性 a_i 将 S 划分为 v 个子集 $\{S_1, S_2, \cdots, S_v\}$, 则对 S_j 的对象子集 U_j, 有 $S_{ij} = \{x \mid x \in U_j$, 且 $f(x, a_i) = a_{ij}\}$. 设 $S_{ijk} = \{x \mid x \in U_j$, 且 $f(x, a_i) = a_{ij}$, 且 $f(x, C) = c_k\}$ 表示 S_j 中类别为 c_k 的样本子集. 则根据属性 a_i 划分样本的信息熵值 $E(a_i) = \sum_{j=1}^{v} \frac{|S_{ij1}| + \cdots + |S_{ijm}|}{|U|} I(S_{ij1}, \cdots, S_{ijm})$.

定义 **4**　用属性 a_i 划分样本集 S 后所得的信息增益值 $Gain(a_i) = I(s_1, s_2, \cdots, s_m) - E(a_i)$.

2　算法设计

2.1　算法思想

本文提出的分布式决策树生成算法 DDTA 使用了一种基于类别的属性分布表 (简称 ACL), 其结构如表 1所示. ACL存储了计算 S中每个属性信息熵增益所需的全部信息.

表 **1**　属性按类别分布表

a_{11}	S_{111}	S_{112}	...	S_{11m}	S_{11}	...	a_{1v}	S_{1v1}	S_{1v2}	...	S_{1vm}	S_{1v}
a_{21}	S_{211}	S_{212}	...	S_{21m}	S_{21}	...	a_{2v}	S_{2v1}	S_{2v2}	...	S_{2vm}	S_{2v}
...	...	...	...	...	...	...	...	...	...	...	...	...
a_{t1}	S_{t11}	S_{t12}	...	S_{t1m}	S_{t1}	...	a_{tv}	S_{tv1}	S_{tv2}	...	S_{tvm}	S_{tv}

定义 **5**　设 N 个站点分布存放了 N 个样本集 $S_1, S_2, \cdots, S_N$, 则 S_i 基于类别的属性表为 ACL_i.

由表 1所示的 ACL结构可知, 全局 ACL 可以从 ACL_i ($i = 1, 2, \cdots, N$) 中计算得到. DDTA 算法在运行时, 各分布站点向服务器发送局部 ACL_i, 服务器则将所有站点的 ACL_i 相加得到全局 ACL, 再根据全局 ACL计算出具有最大信息熵增益的分割属性, 将其传回各站点, 各站点根据服务器传回的分割属性分割分布样本集.

2.2　算法描述

DDTA的算法分为两个部分: 客户端算法和服务器端算法. 具体描述如下:

算法 1: DDTA客户端算法 ClientTree(S_i)

1) 创建与当前样本集 S_i 相应的树节点

2) 扫描 S_i 获得相应的局部 ACL_i.

3) Send ACL_i; //向服务器发送局部 ACL_i.

4) Receive a; //接受服务器传回的分割属性 a

5) if(a is not null) //如果传回的分割属性不空

6) for all $v \in V_a$ do

7) $\{S'_i = \{x \mid f(x, a) = v\}$;

8) ClientTree(S'_i); }

算法 2: DDTA服务器端算法 ServerTree( )

1) For($i = 1$; $i <= N$; $i++$)

2) Receive ACL_i;

3) $ACL = \sum_{i=1}^{N} ACL_i$ //通过各局部 ACL_i 计算得到全局 ACL

4) 根据 ACL计算总信息熵 I

5) for($j = 1$; $j <= t$; $j++$)

6) 计算用属性 a_j 分割样本集的信息熵增益 $Gain(a_j)$;

7) 计算具有最大信息熵增益的分割属性 a;

8) for ($i = 1$; $i <= N$; $i++$)

9) Send a;

3 实验与结果分析

本文利用 4台 PC机构成分布式环境, 其中一台为服务器, 操作系统为 W indows2000 使用 M icrosoft V isual C++ 6. 0实现了 DDTA 算法. 使用蘑菇数据库作为训练样本集, 该样本集具有 8124条样本, 24个属性值. 利用 ID3算法对整个蘑菇数据库进行决策树挖掘, 生成如图 1所示的决策树. 将蘑菇数据库平均分为 3个子样本集存放于 3个不同的站点上, 利用 DDTA 算法对 3个分布同构样本集进行分布式挖掘. 实验结果显示, 系统生成的全局决策树与图 1所示的决策树相同.

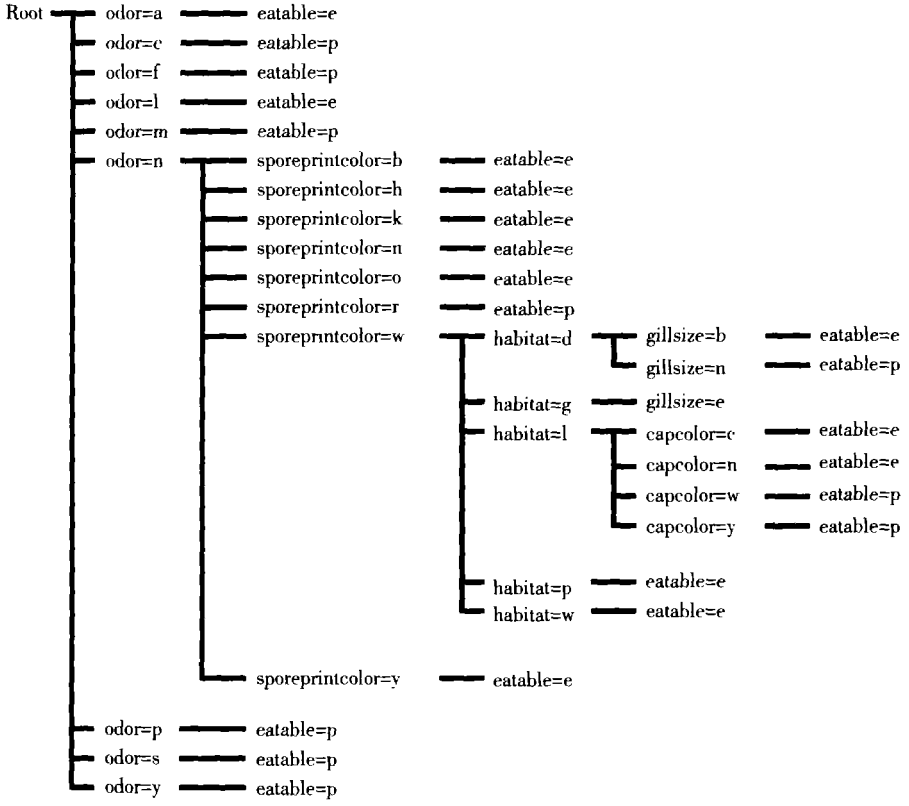

图 1 蘑菇数据库的分类决策树

4 结束语

本文对分布多库环境下的决策树生成进行了研究, 提出了分布式决策树算法 DDTA. 它使用基于信息熵增益的思想分割各个分布同构样本集, 各分布站点向服务器发送的是各站点待分样本集的基于类别的属性表 ACL, 与文献 [3]传递候选样本集相比, 通信代价小. 实验表明 DDTA 算法是正确有效的, 对分布同构样本集进行挖掘, 生成的决策树与对分布同构样本集的并集进行集中式挖掘生成的决策树一样.

[参考文献]

[1] 张敏灵, 陈兆乾, 周志华. 分布式数据挖掘综述 [J]. 计算机科学, 2002, 29(9): 424-429.

[2] Quinlan J R. Induction of decision trees [J]. Machine Learning, 1986(1): 81-106.

[3] Caragea D, Silvescu A, Honavar V. Decision tree induction from distributed heterogeneous autonomous data sources [C] // Proceedings of the Conference on Intelligent Systems Design and Applications. ISDA, 2003.

[4] Manish Mehta, Rakesh Agrawal, Jorma Rissanen. SLIQ: A fast scalable classifier for data mining [C] // EDBT 96. France: Avignon, 1996.

[5] John Shafer, Rakesh Agrawal, Manish Mehta. SPRINT: A scalable parallel classifier for data mining [C] // Proc of the VLDB Conference. India: Bombay, 1996.

[责任编辑: 刘 健]