

基于概念层次树的数据挖掘算法的研究与实现^{*}

Research and Implementation of A Data Mining Algorithm Based on Concept Hierarchy Tree

王大玲 于 戈 鲍玉斌 王国仁 刘斌武

(东北大学信息科学与工程学院 沈阳110006)

Abstract A data mining algorithm merging the processes of establishing concept hierarchy tree, concept ascension and generating rules was introduced in the paper. The algorithm was applied in the analysis and induction of rod production parameter and related rules were obtained from the induction. In real application, the method is more objective than compound query because its result was refined from real data, and more flexible than statistics because it can divide a attribute into different length processed regions according to requirements of users.

Keywords Data mining, Concept hierarchy tree, Attribute-oriented induction

1 引言

随着计算机技术、特别是数据库技术的发展,多年来,在冶金企业中,积累了海量的生产现场参数,它们以各种形式存放在数据库中。如何通过对这些数据的分析和处理,发现其中潜在的、未知的规则和知识,以便从中找到影响产品质量的原因,达到提高产品质量、降低成本等目的,是亟待解决的问题,数据挖掘技术为此提供了有利的支持。

面对数据库中大量的数据,若要从中找到有用的规则和知识,一种方法是首先对数据进行提炼加工,依据某种理论将大量的数据记录浓缩成较少的、能够表示某种规律的若干记录,再将其解释为相应的规则或进行进一步的处理^[1]。概念层次树是一种面向属性进行数据归约的方法,在机器学习、决策支持系统等方面都起到了有益的作用^[2]。

对于面向属性的数据归约问题,一些学者提出了许多有意义的算法^[1~3],在运行速度和执行结果的效率方面不断进行了改进,但这些算法的前提是,有若干棵对应各属性的、由领域专家提供的概念层次树。但在我们开发的面向冶金企业质量控制的数据挖掘系统中,数据量大、属性多、结构复杂,这种概念树是不存在的,也不可能由领域专家预先给定,因此,我们采用“边建立概念层次树、边进行概念提升”的思想,根据数据集的内容,采用与用户交互的方式建立概念层次树,同时进行概念提升。在这个意义上,用户即为领域专家。

2 基于概念层次树的数据挖掘

概念层次树(Concept Hierarchy Tree)是数据库中各属性值和概念依据抽象程度不同而构成的一个层次结构^[1,2]。图1是某数据库中反映“籍贯”属性的概念层次树,其中每个结点成为一个概念,如“沈阳”、“大连”等,而“辽宁”则为它们的父概念。叶结点为第0层,其父概念所在结点为第1层,依次类推。

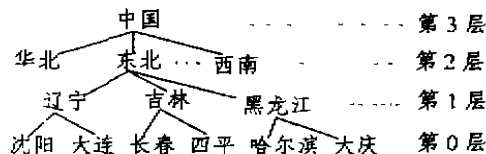


图1 一个概念层次树示例

基于概念层次树的数据挖掘方法是一种归纳方法,是一个元组合并的处理过程,其基本思想是:首先将一个属性较具体的值被该属性的概念层次树中的父概念所替代,然后,对数据库中出现的相同元组进行合并,构成更宏观的元组,并计算元组所覆盖的元组数目,这个过程称为概念提升。若数据库中生成的宏元组数目仍然很大,则需继续进行概念提升,即用该属性概念树中更一般的父概念去替代,直至生成覆盖面更广、数量更少的宏元组,将归纳所得的结果转换成逻辑规则。

例如,“职务”及“年龄”概念层次树如图2所示,某

^{*} 本课题获863/CIMS 主题资助(课题号:863-511-946-004)及霍英东教育基金资助。王大玲 副教授,在职博士生。于 戈 教授,博士生导师。

研究所人员组成如表1所示,根据图2的概念层次树进行概念提升,可得到表2所示的宏元组集合。

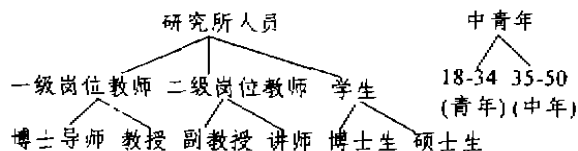


图2 “职务”、“年龄”概念层次树

表1 输入信息

姓名	年龄	职称
王文鼎	38	博士生导师
王心智	34	教授
宋 丹	35	副教授
王慧然	31	讲师
鲍 文	32	讲师
王丽梅	36	副教授
田春晓	27	博士生
刘红刚	23	硕士生

表2 概念提升后的宏元组集合

年龄	职称	覆盖数
中年	一级岗位教师	1
青年	一级岗位教师	1
中年	二级岗位教师	3
青年	二级岗位教师	2
青年	学生	2

由表2可以得到如下规则:

(年龄 = “中年”) & (职称 = “一级岗位教师”) 占 (12.5%)

(年龄 = “青年”) & (职称 = “一级岗位教师”) 占 (12.5%)

(年龄 = “中年”) & (职称 = “二级岗位教师”) 占 (25%)

(年龄 = “青年”) & (职称 = “二级岗位教师”) 占 (25%)

(年龄 = “青年”) & (职称 = “学生”) 占 (25%)

在应用概念层次树对数据集进行概念提升时,需设定一个阈值,规定提升后的宏元组集合中的元组数目,若元组数目大于阈值,则需继续进行高层次的概念提升。如若上例规定阈值为3,则还需继续对“年龄”属性进行提升,将“中年”和“青年”提升至它们的父概念“中青年”,这样,提升后的宏元组数目便降至3。

3 基于概念层次树的数据挖掘算法实现

根据概念层次树的基本思想和方法,我们设计了

一种基于概念层次树的数据挖掘算法。

3.1 算法设计的基本思想

由前面分析可知,基于概念层次树的数据挖掘方法在进行概念提升时需要两个重要前提,一是具有相关属性、元组的预分析数据集,二是包含相关属性的概念层次树。在这里,我们根据分析要求在源数据集中抽取相关的属性和元组构成预分析数据集,同时,根据该数据集的属性值为每个属性构建相关的概念层次树。

在构建概念层次树时,我们采取“自底向上”、与用户交互的方式来构建,即:在为某一个属性构造概念层次树时,首先将该数据集中该属性所有的不同属性值取出来,然后提示用户为这些不同的属性值指定父概念,再将指定的父概念当作普通的属性值,由用户为其指定父概念,直到每一个属性值和概念都有各自的父概念,规定: Any 为根结点的值,只有 Any 没有父概念。

在实际应用中,我们发现,有些数据集的属性值分布比较集中,这样,在进行概念提升时,在第0层、第1层即可使提升得到的宏元组数目满足阈值要求。例如,对于图1的概念层次树,若数据集中四个元组为(张三,沈阳,男)、(李四,沈阳,女)、(王五,大连,男)、(赵六,沈阳,女),则对属性“籍贯”进行概念提升时,至第1层即可满足阈值要求,显然,第2层对于概念提升不起作用。

针对上述情况,我们在构建概念层次树及概念提升时,采取“构建概念层次树与概念提升同步进行,提升结果满足阈值要求时,即可停止”的策略,即为每个属性构建了概念层次树第1层后,便根据该层概念对数据集进行提升,若提升后的宏元组数目不满足阈值要求,再构建概念层次树的第2层,再提升,直至满足阈值要求。采用这样的方法,对于某些属性而言,得到的概念层次树可能不是完整的一棵概念层次树,但我们认为,构建概念层次树及进行概念提升的目的是为了对数据集进行归类,得到有用的规则,因此,只要能够得到满足阈值要求的宏元组集合,便能够获得有用的规则。况且,这样可以节省构建和存储概念层次树所需的时间和空间。

3.2 算法实施步骤

根据我们算法设计的基本思想,我们在算法实施时,按照以下步骤进行:

Step1 对源数据集进行初始化,从中抽取满足预分析条件的数据、属性存放在初始数据库中。用SQL语句表示为:

```
SELECT 选择属性1,选择属性2,...,选择属性n
FROM 源数据集
WHERE 预分析条件;
```

Step2 根据初始数据库中的数据,为其中的每个

属性构建第0层对应的概念层次树。在这里,每个概念结点的结构为:

概念值	覆盖数	父概念	同层后继概念
-----	-----	-----	--------

该结构描述为:

```

CONCEPT = ↑ RECORD
CONCEPT-VALUE STRINGS;
CONCEPT-COUNT INT;
FATHER ↑ CONCEPT;
NEXT ↑ CONCEPT
END

```

其中,第0层各结点对应初始数据库中各属性值,有一个值,便建立一个结点,概念值即为属性值,覆盖数为1,第1层各结点值为一个形如“值1~值2”的区间,覆盖数为其各子概念结点的覆盖数之和,给出此值旨在为构建下一层概念结点的值提供参考依据,每一层结点的 FATHER 指针指向其父概念结点, NEXT 指针指向同层的下一个概念,这样,在第1层概念结点建立后,便得到如图3所示的概念层次树。

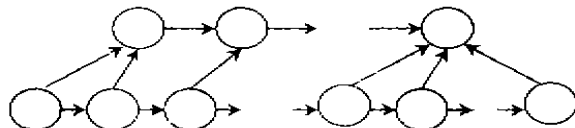


图3 概念层次树示意图

同时,建立一个结构为 CONCEPT、长度为 n (选择属性个数)的线性表,存放当前处理层的第一个属性对应的概念结点,另外,为实现不同属性类型在数据处理时的一致性,在建立第0层概念结点时,对于非数值型的属性赋予数值型编码,以便在建立第1层概念结点时采用前述形式的“区间”作为结点值。

Step3 根据本层概念结点,对数据库进行概念提升、提升时,首先通过对概念层次树的搜索,为数据库中每个元组的每个属性值找到其对应的高一层父概念,找到后,将该属性值替换为其父概念,然后,再将每个属性值均相同的元组合并,并统计覆盖数,得到若干个提升后的宏元组,由它们构成提升后的新数据集^[2]。

为每个元组中的一个属性寻找父概念的过程描述为:

```

选定一个元组 TUPLE;
FOR ALL ATTRIBUTE DO
[GET 当前层对应属性概念头结点 HEAD-ATTR;
P := HEAD-ATTR;
WHILE P ↑ NEXT ≠ NULL AND P-CONCEPT-VALUE
< ATTRIBUTE-VALUE
DO P = P ↑ NEXT
IF P ↑ NEXT = NULL THEN TUPLE-ATTRIBUTE-FA-
THER = "OTHER"
ELSE [TMP = P-CONCEPT-COUNT; P := P ↑ FA-
THER;
TUPLE-ATTRIBUTE-VALUE := P ↑ CON-
CEPT-VALUE;

```

```

P ↑ CONCEPT-COUNT := P ↑ COUNT +
TMP;
];

```

当所有属性处理完毕后,比当前处理层低一层的概念结点已无作用,因此,可将该层结点释放。

Step4 浏览概念提升后的宏元组构成的新数据集,若满足阈值要求,则转 Step6,否则转 Step5,继续构造概念层次树高一层概念结点。

Step5 浏览低一层概念结点值及覆盖数,构造高一层概念结点,得到该层概念结点构成的链表,设某属性当前新建一层概念结点链表的头结点为 HEAD_i,低一层为 HEAD_{i-1},建立概念层次树的过程表示为:

```

P = HEADi-1;
WHILE P ↑ NEXT ≠ NULL DO
[Q = HEADi;
WHILE Q ↑ NEXT ≠ NULL AND P-CONCEPT-VAL-
UE ≤ Q-CONCEPT-VALUE DO
Q = Q ↑ NEXT;
P ↑ FATHER := Q;
];

```

概念结点钩链后,转 Step3进行概念提升。

Step6 根据宏元组的数目及每个宏元组的覆盖数计算各宏元组的覆盖度,剔除覆盖度低的宏元组,将剩余的每个宏元组转换成一条规则。

另外,在实际应用中,由于第0层概念结点数即为对应属性的所有值,数量巨大,在处理时,用户难以给出第1层结点的概念值,选择的属性多时,内存也难以处理如此多结点的若干棵概念层次树,因此,我们在建立第1层概念结点时,将各属性值(第0层结点的概念值)在其最大值 maxV 和最小值 minV 之间均匀地划分成 m 份,每份为 ΔV ,这样,第1层结点概念值为:

$[minV \sim minV + \Delta V], [minV + \Delta V \sim minV + 2\Delta V], \dots, [minV + (m-1)\Delta V \sim maxV]$

而且,不在内存中建立第0层结点,而直接在数据集中进行替换、累加等处理,形成第1层概念结点。从上面给出的实施步骤及相关分析可见,这种方法是完全可行的。

4 实例分析

我们将基于概念层次树的数据挖掘算法应用于宝钢线材产品生产的参数分析中^[4]。该线材产品数据集中包括了生产现场的各种有关参数,如加热温度、均热温度、出炉温度、终轧速度、风机流量、水箱流量及各种时间参数。

首先,我们根据用户要求,在源数据集中抽取日期为1999.12.15~2000.1.20、轧批号为550859的数据,选择属性“加热温度1”、“均热温度1”、“出炉温度”、“终轧速度”进行分析,将满足上述条件的元组中的该四个属性值置于初始数据集中。然后,分别对每个属性按其值在最大值和最小值之间分成20份,按3中介绍的方法

分别为每个属性建立了第1层概念结点,例如,“终轧速度”属性的第1层结点如图4所示。

进行概念提升后,不满足阈值要求,因此继续建立新一层概念结点,第2层结点的建立是与用户交互式进

行的,即由用户输入各结点概念值,再由系统建立本层结点,与低一层结点勾链,处理后,“终轧速度”属性得到的结果如图5所示。

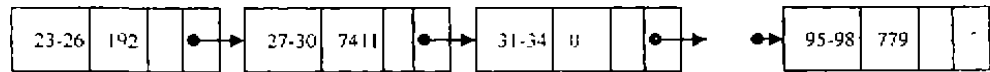


图4 实例中的第1层概念结点

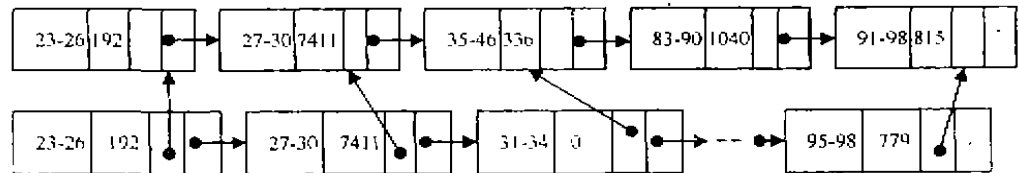


图5 实例中的第2层概念结点

在此基础上,继续对数据集进行概念提升,如此继续下去,直至满足阈值要求,在最后的提升结果中经计算覆盖度、剔除、转换等操作后,得到以下一些规则:

加热温度 $I=979-1044\%$, 均热温度 $I=1093-1119\%$, 出炉温度 $=1107-1162\%$, 终轧速度 $=23-30\rightarrow 9.74\%$;
加热温度 $I=979-1044\%$, 均热温度 $I=1120-1164\%$, 出炉温度 $=1107-1162\%$, 终轧速度 $=23-30\rightarrow 9.34\%$;
加热温度 $I=979-1044\%$, 均热温度 $I=1093-1119\%$, 出炉温度 $=1099-1106\%$, 终轧速度 $=23-30\rightarrow 7.64\%$;
加热温度 $I=902-945\%$, 均热温度 $I=1003-1092\%$, 出炉温度 $=1011-1098\%$, 终轧速度 $=83-98\rightarrow 7.75\%$;
.....

结束语 我们设计了一种基于概念层次树的数据挖掘实现算法,并将其应用于宝钢线材产品生产过程参数的分析和归类中,获得了有益的规则。

同一般的多项查询相比,该方法更具有客观性,因为它的规则的获取,是通过初始数据集不断进行概念提升得到的宏元组转换而成的,是从客观数据中提炼出来的。而一般的多项查询,需要用户假定一个前提,即预先提出查询条件,而面对大量的数据,用户是难以提出诸如“加热温度 $I=979-1044\%$, 均热温度 $I=1093-1119\%$, 出炉速度 $=1107-1162\%$, 终轧速度 $=23-30$ ”这样客观的查询条件的。

同传统的统计分析方法相比,该方法更具有灵活性,因为它的每一层概念结点的建立都是采用与用户交互的方式进行的,其概念结点的数目、每个结点的概

念值均是用户在浏览了低一层概念结点值及其覆盖的元组数目的基础上确定的。而传统的统计分析方法一方面难以同时对多个属性进行处理,另一方面,其处理区间的划分一般是统一的和固定的,缺乏灵活性。

当然,仅仅靠某一种方法获取的规则对于决策支持是远远不够的,但我们相信,作为数据挖掘技术的一种方法,作为决策支持的一类工具,基于概念层次树的数据挖掘方法一定能够发挥其优势,起到积极的、有益的作用。

参考文献

- 1 Han Jiawei, Cai Yandong, Cercone N. Knowledge Discovery in Database: An Attribute-Oriented Approach. In: Proc. of the 18th VLDB Conf. Vancouver, British Columbia, Canada 1992: 547~559
- 2 陈文伟. 智能决策技术. 电子工业出版社, 1998: 6
- 3 周生炳, 张敏, 成栋. 基于规则面向属性的数据库归纳的无回溯算法. 软件学报, 1999, 10(7): 673~678
- 4 Wang D L, Bao Y B. Development of A Data Mining System Supporting Quality Control in Metallurgy Enterprise. In: 16th World Computer Congress 2000 on Intelligent Information Processing, 2000: 578~581