基于概念层次树的数据挖掘算法及应用研究*

肖　娟[1]，叶　枫[2]

(1. 浙江工业大学 之江学院 经贸管理系，浙江 杭州 310024；2. 浙江工业大学 经贸管理学院，浙江 杭州 310014)

摘　要：概念层次树在大规模数据挖掘中已得到广泛应用。在介绍基于概念层次树的数据挖掘算法的基础上，针对已有数值型数据概念提升算法的不足，提出了改进后的算法，并通过数据测试给出两种算法的比较效果和应用实例。

关键词：概念层次；数据挖掘；阈值

中图法分类号：TP391　　文献标识码：A　　文章编号：1001-3695(2005)03-0061-03

Research on Data Mining Algorithm Based on Conception Hierarchy Tree and Its Application

XIAO Juan[1]，YE Feng[2]

(1. *Dept. of Economic Management, Zhejiang College, Zhejiang University of Technology, Hangzhou Zhejiang* 310024, *China*; 2. *College of Economic Management, Zhejiang University of Technology, Hangzhou Zhejiang* 310014, *China*)

Abstract: Conception hierarchy tree classifiers have found the widest applicability in large-scale data mining environments. After introduced a data mining algorithm based on conception hierarchy tree. This paper, in the light of the limitations of existed concept assension algorithm to numerical attributes in database, introduces an improved algorithm. Moreover, the result of two algorithm testing on actual data is provided in this paper.

Key words: Conception Hierarchy; Data Mining; Threshold

目前常用的各类数据挖掘算法，主要用于特征规则、关联规则、分类规则、序贯模式的发现，但将这些算法用于实际的大型数据库进行知识发现，却不能取得很好的效果。概念层次树作为数据分类的方法，可以将大量详细的细节数据总结上升到较高的概念层，为数据挖掘的各个步骤提供背景知识，提高知识的准确性和可理解性。适合用户需要较高层次的、能反映一定关系的规则来支持决策的实际应用，此外可用于对数据预处理得到清洁的元数据及知识表示。

本文试图对基于概念层次树的数据挖掘算法有所改进，如实现层次自动提取与概念提升同步的数据挖掘算法，非均匀分布下数据的概念层次的自动提取等功能。

1　概念层次及概念层次树

所谓概念层次 H 就是部分有序集(h, ；)，其中 h 是有限概念集合，； 是 h 上的部分有序关系。概念层次能够以层次的形式和偏序的关系组织数据和概念。如我们取一般——特殊的关系为； ，可以表示城市、省份的关系，如杭州； 浙江省； 中国。通常我们以层次树来表示一个概念层次，即概念层次树(Concept Hierarchy Tree)，树的节点表示概念，树枝表示偏序。图 1 是客户年龄的概念层次树。

概念层次树可由领域内的专家提供，但在实际评估中，因为数据规模很大，协调专家之间的意见非常困难，人工定义大型的概念层次树亦不合理、不现实，且提供的概念层次树可能是最一般的概念层次树，常包含全部可能的属性值以及它们对应的全部可能的父概念。这种概念树对于特定的数据库显得偏大，并且影响到概念提升的速度，因而缺乏一定的灵活性和针对性。通常，无论是领域专家定义还是自动生成概念层次树，概念层次树的构造有自顶向下和自底向上两种方式[1]。

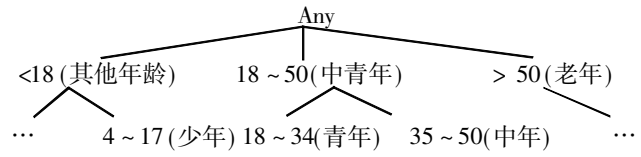

图 1　年龄概念层次树

所以对于数据库中常常存在各种数值型属性的情况，一般采用自动生成数值型概念层次的概念化方法：由用户指定期望的分段数，由机器自学习，将属性值分成若干个区间。该方法可满足大型数据库中特殊挖掘任务的要求，它能针对特殊挖掘任务的要求构造专门的概念层次，反映特殊数据集合中的数据分布。本文正是对面向数值型数据构建概念层次树的数据挖掘方法提供一些建议和改进。

2　基于概念层次树的数据挖掘算法

基于概念层次树的数据挖掘方法的基本思想[2]是：首先，一个属性的较具体的值被该属性和概念层次树中的父概念所代替；然后，对知识基表中出现的相同记录进行合并，构成更宏观的记录，并计算宏记录所覆盖的记录数目，如果数据库中记录生成的宏记录数目仍然很大，那么用这个属性的概念层次树中更一般的父概念去替代，或者根据另一个属性进行概念层次树的提升操作，最终生成覆盖面更广，数量更少的宏记录；最

收稿日期：2004-01-08；修返日期：2004-05-12

基金项目：浙江省自然科学基金资助项目(602096)

后,将所得的结果转换为逻辑规则。

一般的,基于概念层次树的数据挖掘算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。通常在得到完整的概念层次树后,再对初始数据集中的属性进行概念提升。

3 数值型数据自动提取概念层次的算法

对数值型字段数据的概念化,就是将字段中的所有数据进行概念分段,将每一段用一个概念值表示,然后将原字段中的所有数据用它所对应的概念段的概念值来代替,产生概念表。例如,对某一个数值型字段进行概念分段得到如下的概念段:[2, 12K] ,[12, 16K] [16K, 23K] ,[23-90K] },这一字段中的每一个值用对应的概念段或指定的概念段值来代替,就完成了对这一字段的概念化。

一般的数值型概念层次生成算法都是通过将数值型属性的值域区间离散化,形成多个子区间作为概念层次的叶节点,基本方法有等距离区间法和等频率区间法。

其中等频率区间法[3,4]实现简单,且效率较高。这种算法假定数据的取值呈现均匀分布,即各分段内的记录数较为接近。算法先用相同的小间隔 Interval = (High-Low) /($K \times T$) 将数值字段中的数据分段,其中 High 和 Low 表示字段的最大值和最小值,K 表示将数值型区间分段的合理性即分段频度,T 指属性值在数据库中出现的不同值的个数的界限即区段数,接着对落在每一数据段中的数据量进行计数及记录总数 TotalCount;然后利用合并分段算法[5]从第一个数据段开始累计数据的总计数,如果大于期望覆盖数(TotalCount/T) ,则将其合并成一个概念段,对其赋予概念值。用同样的方法直至到达最后一个数据段。将数据段合并成期望(T) 个概念段,其中每个概念段中的数据个数是基本相同的。

从本质上讲,这种算法先用相同的小间隔将数值字段中的数据分段,然后将数据段合并成期望(T) 个概念段,其中每个概念段中的数据个数是基本相同的。而这只是针对数据取值均匀分布的情况,其概念分段的标准只有数据个数,没有考虑数据的分布情况。所以,当数据取值分布杂乱不均匀时,此算法就显得不足。由于用于划分数据段的 Interval 是一个不变的量,显然,这种情况下每一数据段中的数据个数不完全相同,如果字段中的数据分布在某一取值范围内非常集中的话,那么在一些数据段中的数据量就会接近期望覆盖数或者可能会使概念段中的数据量远大于期望覆盖数,并且使实际求出的概念段数可能小于期望的概念段数。当然通过增大值 K,使 Interval 变小,去增加数据段数,可以减少每一数据段的数据个数,但这并不是本质的解决方法。

4 改进的基于概念层次树的数据挖掘算法

通过分析知道,算法的主要缺陷在于用于分割的 Interval 是一个不变的量,不适于数据取值分布不均匀的情况,对于此缺陷,本文提出一种用于面向数值型数据的概念层次自动提取与概念提升同步的数据挖掘算法。

4.1 算法设计的基本思想

(1) 构建概念层次树与概念提升同步。基于对数值型数据自动提取概念层次的算法,在自底向上逐层构建概念层次时,同步进行概念提升和规则挖掘。为每个属性构建概念层次的第一层,便根据该层概念对数据集进行概念提升,若提升后的宏元组数目小于用户指定的阈值要求,则停止概念层次自动建树,获得挖掘规则;若提升后的宏元组数目大于阈值要求,则继续构建上一层概念层次。

这种同步挖掘尤其适于用户以获得规则为目的的情况,即只要能得到满足阈值要求的宏元组集合,便能得到有用的规则。特别是对属性值分布比较集中的情况,无须再向上构建父概念,即可满足概念提升的阈值要求。此时,该算法概念层次停止向上构建,获得所需的宏元组。这样也节省了构建和存储概念层次树所需的空间。

(2) 变间隔分割数值型数据。一般的,由于自动构建概念层次树的最后目的是使每个分段内的记录数基本接近,概念层次自动提取算法假定数值分布均匀,提取出的概念分段的取值是连续的。而对于数据取值分布不均匀的情况,若仍采用这种等间隔分割数值型数据的方法,就只能通过增大数值 K 来解决问题,却使算法的效率降低。针对此,本文的改进算法中引入变间隔分割数值型数据的概念化方法[4]提取概念层次。

该算法基本思想是:考虑概念分段时的数据个数和数据分布。对选定字段中的数据进行分割时,依次判断一个 Interval 间隔内的数据量的多少,采用变间隔进行分割。如判断数据量过大,则缩小 Interval 值并将其作为分割间隔,将数据分成数据段。这种方法提取出的概念分段的取值也是连续的。

(3) 通过方差与用户交互指定参数值。方差是通过计算偏离总均数的平方和再除以 n - 1(样本量减 1) 而得到的。这样,给定 n 值的情况下,方差就是离均差平方和。

由于 k 值的改变可能会影响算法的最终结果。如前所述,若增大 k 值,可使 Interval 值变小,增加初始数据段数,减少每一数据段的数据个数,从而改变合并分段后概念段的取值范围,最终可能改变概念层次提取的结果。此外,由于算法中最后一个概念分段的合并标准是概念节点数目,而不是期望覆盖数,就可能使这个分段的记录数与平均记录数有较大的偏离。所以本改进算法中引入方差计算,采用与用户交互的方式指定 k 值,对得到的记录分布与期望平均记录比较计算方差,通过对方差值的观察,更便于用户在试验中指定合理的 k 值。

4.2 算法流程

输入:训练集、预分析条件和配置信息(提升概念层次的阈值、噪音阈值、区段数、粒度) ;

输出:数据的概念层次。

(1) 准备预分析的初始数值型属性数据集;

(2) 指定约束条件(参数) ;

(3) For 所有属性 do

(4) 利用变间隔分割算法构建上一层概念层次;

(5) For 所有元组 do

(6) For each 属性 do

(7) 概念提升(用父概念替换原属性值) ;

(8) 合并属性值均相同的元组构成宏元组集合;

(9) If 宏元组记录数大于期望分段数 Then 继续构造高层概念节点;

(10) 计算方差。

其中的概念提升算法根据本层概念节点,对初始数据库进行概念提升:为每个元组的每个属性找到其对应的高一层父概念,将该属性替换为其父概念。

```
For each tuple in R;
For all attribute A do
[ For each seg = [ low + h*  interval, low + i*  interval] do
```

```
[ if t[ A] < = low + i* interval then
将属性 A 的值用父概念代替;
]  ]
```

同时, 建立一个 Cust_TotalUp 表, 用于存放当前处理层的概念节点值, 节点值采用区间的形式。且概念值被进一步提升时, 直接在表中对原属性值进行替换。

然后, 合并表中属性值均相同的元组, 计算覆盖数, 由它们构成提升后的宏元组集合(其中增加一个属性——覆盖数) 。

下一步浏览提升后的宏元组集合, 若满足于阈值 Threshold(或泛化阈值) 且没有更高的概念用于提升, 则根据宏元组的数目及每个宏元组的覆盖数计算各宏元组权值 T-weight = 覆盖数/∑覆盖数, 剔除覆盖度低的宏元组(低于剪枝阈值 Noisethreshold) , 将剩余的每个宏元组转换成一个规则。

若不满足阈值要求, 则继续构造高一层的概念节点。

4.3　数据分析

若将上述算法对以 SQL Server 形式的外贸企业业务数据库进行测试, 可看出改进后算法在面对大规模数据库时具有的优势。数据分布结果由直方图作出, 图 2 和图 3 是分别为原算法和改进后算法在 34 条记录上测试得到的结果。概念段的范围由细线分割的区域表示。参数 T 和 K 分别是 4 和 7, 为了便于比较分段的效果, 阈值和噪声阈值为 4 和一较小值, 即不考虑由于阈值控制的进一步概念提升和删减, 而只是对变间隔和等间隔分割的效果进行比较。

实验分析的字段的最大值是 4 730 200, 最小值 Low 是 10 140, 由参数 T 和 K 可以求出平均覆盖度是 8.5, 初始间隔 Interval = (high-low) /($K \times T$) 放大为 143 574。由图 2 可以看出, 由于平均覆盖度和固定初始间隔的影响, 原算法将字段内的数据分为四个概念段内, 最后一个分段的覆盖度仅为 3, 远远小于平均覆盖度, 计算方差为 17.0。

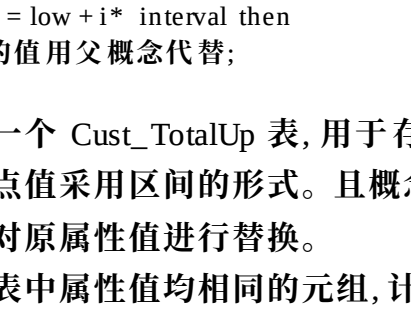

图 2　等间隔分割数据结果

产生这种情况的原因就是因为所测试的数据分布极不均匀, 使得原算法不再适用。由于前两个初始分段间隔内的数据分布过密, 均大于 9, 已远远超出平均覆盖度, 从而使合并分段后的最终分段也失去平衡。而利用改进算法后, 可以得到 38 个经过调整后的初始分段(图 3) 。其中的覆盖数都在 1 ～5 之间, 其分布均比在原算法的初始间隔中的要均匀得多。

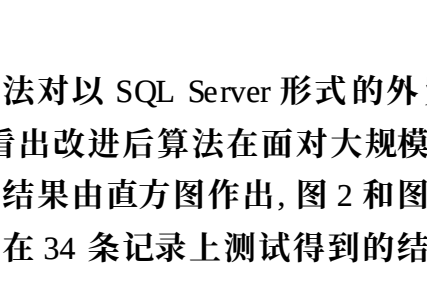

图 3　变间隔分割数据结果

从图 3 可以看出, 改进的算法对数据分布过密的初始间隔进行缩小调整, 得到的各概念段都与平均覆盖度接近, 经计算方差为 3.7。相比原算法对其缺陷给予了改进。如果为了得到更加理想的分段结果, 可以调整 T, K 值。

对数据库中需处理的数据表的所有数值型字段进行概念分段后, 就可以将原数据表中的数据以它的概念值表示, 产生一个新的概念表, 对这个概念就可以运用聚类、关联规则等各种方法进行进一步数据挖掘。

5　外贸企业客户关系管理中的应用实例

本实例以某外贸企业的客户数据库为研究对象, 学习任务是通过改进的数值型数据自动提取概念层次法, 概念化客户合同金额, 在一定高度上发现指定时间内各地区外贸企业客户的外销模式。由于广大企业后台一般以 MS SQL Server 7.0 数据库为基础构建的, 考虑数据库的转换和系统运行环境的一致性, 为避免不必要的麻烦, 开发环境的后台采用相同的数据库。对于前端开发工具的选择, 因为数据挖掘技术本身需要大量的计算, 为节省时间, 采用执行效率高的系统存储过程(Stored Procedure) 作为开发工具, 通过单独设计存储过程以达到专门的目的。用户交互界面设计采用 Power Builder 8.0 为开发工具, 可很好地保证两者的完美结合, 只需写一个语句就可利用参数传递在服务器端调用存储过程, 比在本地编程再传送到服务器端可以减少许多负载。

6　总结

本文分析了原有基于概念层次树的数据挖掘算法的局限性, 提出了一种改进的基于概念层次树的数据挖掘算法, 并将改进算法应用于实际的外贸企业业务数据库上。由于时间原因和客观条件所限, 笔者认为在如下几个方面仍需完善:

(1) 由于所能获得的客户数据库的限制, 用于实验的客户数据不够多, 实例分析过程中选取的属性个数也较少, 因而在充分体现该算法的优势中受到一些限制。

(2) 在该算法中, 仍未能对已有算法中存在的不足之处做全面改进。对于原算法的合并分段算法中, 最后一个概念分段的合并标准是概念节点的数目, 而不是期望覆盖数。本改进算法中引入方差计算, 利用可视化方式辅助用户在试验中指定合理的粒度来缩小可能的误差, 已在一定程度上提高了算法的准确性, 但仍然期待能探讨出从本质上解决问题的方法。

(3) 本文仅对数值型属性自动提取概念层次的方法进行了研究。而对于其他属性(如名词性属性) , 一般由人工定义概念层次, 工作量大, 如何从数据库中提取这些属性的概念层次是个很有意义的研究内容。

参考文献:

[1]　陈文伟, 等. 智能决策技术[M] . 北京: 电子工业出版社, 1998.

[2]　王大玲, 于戈, 等. 基于概念层次树的数据挖掘算法的研究与实现[J] . 计算机科学, 2001, (28) : 88-91.

[3]　Han J, Fu Y. Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases[C] . Proceedings of the KDD '94, Seattle, WA, 1994. 157-168.

[4]　李涛涛, 李世祥. 一种数据库数值型字段概念化算法的介绍及讨论[J] . 微型电脑应用, 1999, (15) : 24-26.

[5]　刘胜军, 杨学兵, 蔡庆生. 关系数据库中概念层次自动提取算法研究[J] . 计算机应用研究, 1999, 16(12) : 15-17.

作者简介:

肖娟(1978-) , 女, 浙江杭州人, 助教, 硕士, 主要研究方向为决策支持系统、管理信息系统; 叶枫(1964-) , 男, 浙江杭州人, 教授, 硕士生导师, 主要研究方向为决策系统支持、地理信息系统和管理信息系统。