

一种新颖的基于量化概念格的属性归纳算法

王德兴, 胡学钢, 刘晓平

(合肥工业大学计算机科学与技术系, 230009, 合肥)

摘要: 为了解决数据挖掘过程中挖掘的知识粒度过粗或过细问题, 并利用概念格的偏序特性, 提出了一种基于量化概念格的属性归纳算法. 首先对概念格的外延进行量化, 得到量化概念格, 再根据概念格的哈斯图, 采用概念的爬升进行相应的泛化, 从而获得基于量化概念格的多层、多属性归纳. 与面向属性归纳(AOI)算法相比较, 结果表明所提算法不仅能实现 AOI 的单一属性归纳, 还能进行多层、多属性的归纳, 其属性泛化的路径不是惟一的, 并且很容易在量化概念格的哈斯图中寻找合适的泛化路径和阈值, 以此得到用户要求的、合理的属性归纳结果.

关键词: 面向属性归纳; 概念格; 数据挖掘

中图分类号: TP181 **文献标识码:** A **文章编号:** 0253-987X(2007)2-0176-04

Novel Attribute Induction Algorithm Based on Quantized Concept Lattice

Wang Dexing, Hu Xuegang, Liu Xiaoping

(Department of Computer Science and Technology, Hefei University of Technology, Hefei 230009, China)

Abstract: In order to deal with the problem of over-coarse or over-fine knowledge granularity in data mining, an attribute induction algorithm based on quantized concept lattice is proposed by using the partial property of the concept lattice. Firstly, the quantized concept lattice is defined by quantifying concept extension of the concept lattice, and then it is generalized using concept ascension according to the Hasse diagram of the concept lattice so as to get the induction with multi-level and multi-attribute based on the quantized concept lattice. Compared with the attribute-oriented induction (AOI) algorithm, the proposed algorithm can not only perform the unitary induction of AOI, but also carry out the induction with multi-level and multi-attribute, and the path of attribute generalization is not unique. Moreover, it is easy to find proper generalized paths and thresholds in Hasse diagram of quantized concept lattice to obtain the reasonable results required by users.

Keywords: attribute-oriented induction; concept lattice; data mining

在知识发现过程中人们感兴趣的往往是一些高层次、适度概括的信息, 而不是数据库中的琐碎细节, 因此数据预处理中的数据简化显得尤为重要. 面向属性归纳(AOI)的算法^[1]是一种数据简化方法, 之后在此基础上的改进算法有 LCHR (Learn Characteristic Rule)算法^[2]、GDBR (Generalize Data Base Relation)算法^[3]和 FIGR (Fast Incremental Generalization and Re-generalization)算法^[4], 而且

与之相关的算法主要包括由属性阈值和记录值同时作用的数量型控制面向属性的量化归纳算法^[5], 以及基于规则的面向属性归纳的无回溯算法^[6].

目前, 这类算法都是基于 AOI 的, 其结构是单一属性的概念树, 每个属性值只有惟一的一条从低层到高层的泛化路径^[7]. 为了解决这个问题, 本文提出了基于量化概念格的属性归纳算法, 它不仅能完成基于 AOI 的单一属性归纳, 还能进行多层、多属

性的归纳,并且泛化的路径也不是惟一的,突破了基于 AOI 的泛化属性必须是单一属性的条件限制。

1 量化概念格

有关概念格的知识可参阅文献[8-12],这里只简要介绍一些基本概念。

定义 1 设给定上下文 $K=(O,D,R)$,在形式概念分析中称之为形式背景,其中 O 是对象集合, D 是属性集合, R 是 O 与 D 之间的一个二元关系,对于一个对象 $o \in O$,属性 $d \in D$,那么 oRd 就表示对象 o 具有属性 d 。

形式背景的对象集 $A \in P(O)$ 、属性集 $B \in P(D)$ 之间按如下关系进行 Galois 连接,即

$$f(A) = \{y \in D \mid \forall x \in A, xRy\}$$

$$g(B) = \{x \in O \mid \forall y \in B, xRy\}$$

则称从形式背景中得到的每一个满足 $A=g(B)$ 、 $B=f(A)$ 的二元组为一个形式概念,其中 A 是对象幂集 $P(O)$ 的元素,称为概念 (A,B) 的外延 $\text{Extent}(C)$, B 是属性幂集 $P(D)$ 的元素,称为概念 (A,B) 的内涵 $\text{Intent}(C)$ 。

定义 2 在给定的上下文中,如果 2 个概念 $C_1=(A_1,B_1)$ 和 $C_2=(A_2,B_2)$ 满足 $A_1 \subseteq A_2$,则称 C_1 为子概念, C_2 为超概念, C_1 和 C_2 有子概念-超概念关系,表示为 $C_1 \leq C_2$ 。在形式背景中,所有概念及其子概念-超概念的偏序关系所诱导出的格称为概念格 (Galois Concept Lattice, GCL)。

定义 3 对概念 $C_1=(A,B)$,称 $C_2=(|A|,B)$ 为 C_1 的量化概念, C_1 为 C_2 的真实概念,其中 $|A|$ 是外延 A 的基数。

定义 4 2 个等价的基本概念 $C_1=(A_1,B_1)$ 和 $C_2=(A_2,B_2)$ 所对应的基本量化概念 $C'_1=(|A_1|,B_1)$ 和 $C'_2=(|A_2|,B_2)$ 是等价的。量化概念间的关系取决于其相对应的真实概念间的关系: $C_1 \leq C_2$ 当且仅当 $C'_1 \leq C'_2$ 。将由量化概念所构成的格称为量化概念格 (QCL)。

概念格的构造算法有渐进式、批处理法等,其中渐进式构造算法适用范围广,是比较好的算法,如文献[9-12]所述。量化概念格只是将概念的外延进行量化,改变概念的外延表示形式,没有改变概念格的结构,其构造算法如文献[9]所述。

2 基于量化概念格的属性归纳

2.1 量化概念格内涵的属性表示

量化概念格是由惟一的顶部概念节点、惟一的

底部概念节点和一组中间概念节点组成的所谓“树”结构。量化概念格有惟一的顶部概念节点,是属性的最一般概念,它作为最高层次概念,没有具体特征。这个最泛化的概念是一个全概念,用“ $(\text{all}, \emptyset)$ ”表示,其中 all 表示全概念的所有外延, $\emptyset$ 表示公共属性为空,即概念的内涵为空。概念格有惟一的底部概念节点,是最低层次、特化的空概念,用“ $(\emptyset, \text{all})$ ”表示,其中 $\emptyset$ 表示概念的外延为空, all 表示概念格中的所有内涵属性。对于非顶、非底部的概念节点,它们有一个或一个以上的子(或超)概念节点,其概念节点的内涵是其子概念节点内涵中一组公共相关的属性值(便于在量化概念格的哈斯图中清晰地表示,一般采用符号标识),每个属性值是数据集中该属性的一般属性值,这些属性值可以是离散的,也可以是连续的。对于连续的值,每个中间值表示属性值的一个范围集合;对于离散的属性值,中间值表示一个或几个属性值集合。

2.2 算法思想

基于量化概念格的属性归纳算法描述如下。

(1)如果关系数据库中的某个属性值是大量的且不相同,根据用户的领域知识要求,应将该属性值离散化,即将其分箱、分类或聚类,并标以类别标识。

(2)对关系数据库中所有的属性,如果量化概念格底部顶点的直接超概念的某概念 C 的内涵 $\text{Intent}(C)$ 存在 $\text{Int} \subseteq \text{Intent}(C)$ (Int 为用户感兴趣的单个或多个属性集),则从 C 开始进行宽度优先的向上遍历(泛化),寻找 C 的所有超概念,直到该概念格为顶部全概念 $(\text{all}, \emptyset)$ 为止,并记录相应的遍历(泛化)路径。

(3)确定属性可以泛化的概念层次和该层次的分支属性,并确定这种情况下的属性归纳颗粒度的大小,这些可以用属性归纳控制阈值来决定。如果泛化路径上的概念外延个数 $|\text{Extent}(C)|$ 大于属性归纳控制阈值,则该概念应作为属性归纳的备选概念节点,并将其记录下来,从而得到相应的泛化路径(一般会有多条泛化路径),当然也可采用泛化路径上的概念内涵长度 $|\text{Intent}(C)|$ 作为属性归纳控制阈值。属性归纳的概念层次越高,泛化概念的外延数越大,则概念的内涵长度就越小。

2.3 属性归纳算法

算法 1 基于量化概念格的属性归纳算法描述如下。

输入:①关系数据库 D ;②用户感兴趣的归纳属性集 Int ;③单个或多个属性集的泛化阈值。

输出:泛化路径上的概念。

(1)将关系数据库 D 中的属性值离散化,并将其分箱、分类或聚类,并标以类别标识,删除不可归纳的属性。数据库的主码是不可删除的,因为它是生成量化概念格的概念外延,是使关系数据库变为决策表形式数据库的关键。

(2)构造量化概念格。

(3)求解概念的泛化路径算法。

算法2 概念的泛化路径算法如图1所示。

```

Void concept-lattice-generalization (QCL, generalization-threshold)
{  $C_0 = (\emptyset, \text{all})$ ;
  Setnull(Q); // * 队列 Q 置空
  FOR each direct-sup-concept  $C_p$  node of  $C_0$  DO
  /*对于底部顶点的直接超概念
  { IF  $\text{Int} \subseteq C_{pi}. \text{intent}$  THEN // * 如果感兴趣的属性 Int 在概念的内涵中
    Enqueue(Q,  $C_{pi}$ ); // * 结点  $C_{pi}$  入队列 Q
    Visited[ $C_{pi}$ ]=TRUE;
    WHILE Not Empty(Q) DO
    {  $L\_node = \text{Outqueue}(Q)$ ; // * 取队列头的节点
       $C\_node = \text{First-direct-sup-concept}(C, L\_node)$ ;
      /*取出  $L\_node$  概念的第一个直接超(父)概念
      WHILE (  $\text{Int} \subseteq C\_node. \text{intent}$ ) DO
      { IF (Not Visited[ $C\_node$ ])
        { Visited[ $C\_node$ ]=TRUE;
          IF  $C\_node. \text{Extent} \geq \text{generalization-threshold}$ 
          /* 如果  $C\_node$  的外延大于泛化阈值
          THEN  $\text{Path}[n++] = C\_node$ ; // * 则记录属性泛化路径
          Enqueue(Q,  $C\_node$ );
           $C\_node = \text{Next-direct-sup-concept}(C, L\_node)$ ; // * 取出  $L\_node$  概念的下一个直接超(父)概念
        } }
  } }

```

图1 概念的泛化路径算法

2.4 算法分析与比较

文献[12]的概念格构造算法的理论时间复杂度应是 $O(2^n)$, 其中 n 是概念格的属性个数。实际应用的时间复杂度不会如此, 在比较好的情况下的时间复杂度应是 $O(n^2)$, 甚至是 $O(n)$, 一般情况下的时间复杂度是 $O(n^3)$ 。由于量化了概念格的外延, 并不会减少概念格的结点数, 所以量化概念格的构造时间复杂度是 $O(n^3)$ 。

文献[2]中的 LCHR 算法的时间复杂度是 $O(n \lg n)$, 而 AOI 算法的时间复杂度是 $O(np)$, 其中 n 是输入关系数据库的元组数, p 是主关系的数量,

它们的空间复杂度均为 $O(n)$; GDBR 和 FIGR 算法的时间复杂度是 $O(n)$, 要比 LCHR 和 AOI 算法的时间复杂度小, 其空间复杂度均为 $O(p)$; 基于量化概念格的属性归纳算法遍历的时间复杂度为 $O(nm)$, 其中 n 是输入关系数据库的元组数, m 是概念格中不重复的属性个数, 也是概念格的高度, 而在空间复杂度上它并没有很大的优势, 其主要的时间开销是构造量化概念格的时间, 其次是在输入的关系数据库中进行属性归纳的时间。如果输入的关系数据库过大, 可抽样选取部分样本数据, 这样构造量化概念格的时间会比较短, 所以这种开销还是值得的。

3 应用实例

以人才招聘和就业信息为实验数据来评估学校毕业生的素质, 并选取一些重要的属性数据来构造如表1所示的人才就业信息(其中删除了不可归纳的属性, 如姓名、年龄等), 并对属性标号, 以此构造出相应的量化概念格, 其哈斯图如图2所示。

表1 人才基本信息表

编号	姓名	年龄	籍贯	户籍	所学专业	毕业学校	综合素质评价
1	兰新	26	红桥(d_1)	天津(t)	s_1	k	C_0
2	柳河	20	西城(d_2)	北京(b)	s_2	g	C_0
3	张双	22	海淀(d_3)	北京(b)	s_1	g	C_1
4	莉莉	22	海淀(d_3)	天津(t)	s_2	g	C_0
5	许军	30	海淀(d_3)	北京(b)	s_1	k	C_1
6	利平	36	东城(d_2)	北京(b)	s_1	g	C_0

从概念格中抽取一定数量的记录, 由基于 AOI 的概念层次背景知识可知, $\{\text{海淀}, \text{东城}, \text{西城}\} \subset \{\text{北京}\}$, 概念 $(2, \{d_3, b, s_1, C_1\})$ 和概念 $(2, \{d_2, b, g, C_0\})$ 含有籍贯属性 d_2 (西城)、 d_3 (海淀), $d_2 \subset b, d_3 \subset b$, 所以籍贯属性可进行属性归纳, 可将 d_2, d_3 的概念爬升归纳为 b (北京), 保留属性 b , 删除属性 d_2, d_3 , 因此基于量化概念格的属性归纳完成了 AOI 属性归纳。

根据基于量化概念格的属性归纳算法, 任意选取一条属性归纳路径, 如从概念 $(2, \{s_2, g, C_0\})$ 开始向上寻找, 则该概念的超概念依次是 $(3, \{g, C_0\})$ 、 $(4, \{g\})$ 和 $(4, \{C_0\})$ 的属性泛化路径有 2 条。下面依次选择第 1 条路径进行不同层次、多属性的归纳。

如果以概念 $(2, \{s_2, g, C_0\})$ 进行属性归纳, 则第 2 和第 4 条记录以属性专业 s_2 、大学 g 、综合素质 C_0 作为归纳对象, 并合并为同一个概念 $(2, \{s_2, g,$

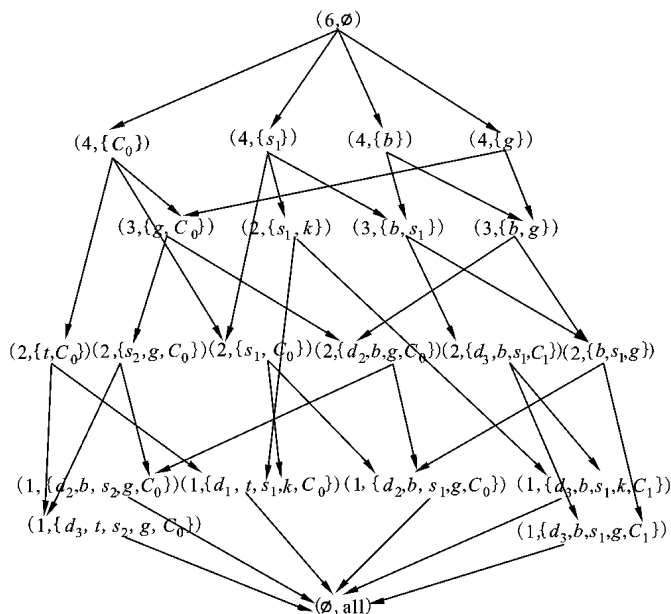


图2 人才信息表对应的量化概念格的哈斯图

C_0)). 该概念的外延个数为2, 内涵长度为3, 属性归纳的层次较低, 粒度较细, 合并的记录数量较少, 需继续向上进行属性归纳。同理, 以概念 $(3, \{g, C_0\})$ 进行属性归纳, 则第2、第4和第6条记录以属性大学 g 、综合素质 C_0 作为归纳对象, 合并为概念 $(3, \{g, C_0\})$, 属性归纳的层次较高, 粒度较粗, 合并的记录数量较多。其余依此类推。

4 结束语

基于量化概念格的属性归纳是在量化概念格的哈斯图中交互地进行属性归纳的, 同AOI相比, 它具有直观、简捷的特点。基于量化概念格的属性归纳算法是多层次、多属性的属性归纳, 每个属性值有多条从低层到高层的泛化路径或多条从高层到低层的特化路径, 即有多条双向泛化或特化路径, 能直观地提供属性归纳的知识粒度, 得到用户要求的属性归纳结果。

参考文献:

- [1] Han Jiawei, Cai Yangdong, Cercone N. Knowledge discovery in databases: an attribute-oriented approach [EB/OL]. [2006-06-06]. <http://citeseer.ist.psu.edu/cache/papers/cs/5021/han92knowledge.pdf>.
- [2] Han Jiawei, Cai Yangdong, Cercone N. Data-driven discovery of quantitative rules in relation databases [J]. IEEE Transactions on Knowledge and Data Engineering, 1993, 5(1):29-40.
- [3] Carter C L, Hamilton H J. Performance evaluation of attribute-oriented algorithms for knowledge discovery from databases [C]//Proceedings of 7th IEEE International Conference Tools with Artificial Intelligence. Los Alamitos, USA: IEEE Computer Society, 1995: 486-489.
- [4] Carter C L, Hamilton H J. Efficient attribute-oriented generalization for knowledge discovery from large databases [J]. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(2):193-208.
- [5] 陈红梅, 王丽珍. 面向属性的量化归纳 [J]. 计算机研究与发展, 2001, 38(2):150-156.
Chen Hongmei, Wang Lizhen. Quantifiable attribute oriented generalization [J]. Journal of Computer Research and Development, 2001, 38(2):150-156.
- [6] 周生炳, 张钊, 成栋. 基于规则面向属性的数据库归纳的无回溯算法 [J]. 软件学报, 1999, 10(7):673-678.
Zhou Shengbing, Zhang Bo, Cheng Dong. A nonbacktracking algorithm for the rule based attribute oriented database induction approach [J]. Journal of Software, 1999, 10(7):673-678.
- [7] 刘明吉, 王秀峰, 李宝林. 基于多层次概念提升的知识发现方法 [J]. 计算机科学, 2001, 28(3):109-111.
Liu Mingji, Wang Xiufeng, Li Baolin. A knowledge discovery method based on multi-level concept generalization [J]. Computer Science, 2001, 28(3):109-111.
- [8] Wille R. Restructuring lattice theory: an approach based on hierarchies on concepts, in ordered sets [M]. Dordrecht, Netherlands: Reidel, 1982:445-470.
- [9] Wang Dexing, Hu Xuegang, Wang Hao. The research on model of mining association rules based on quantitative concept lattice [C]//IEEE Proceedings of the 1st International Conference on Machine Learning and Cybernetics. Piscataway, USA: IEEE, 2002:134-138.
- [10] 强宇, 刘宗田, 林炜, 等. 模糊概念格在知识发现的应用及一种构造算法 [J]. 电子学报, 2005, 33(2):350-353.
Qiang Yu, Liu Zongtian, Lin Wei, et al. Research on fuzzy concept lattice in knowledge discovery and a construction algorithm [J]. Acta Electronica Sinica, 2005, 33(2):350-353.
- [11] Godin R, Missaoui R, Aloui H. Incremental concept formation algorithms based on Galois (concept) lattices [J]. Computational Intelligence, 1995 11(2):246-267.
- [12] Snelting G. Reengineering of configuration based on mathematical concept analysis [J]. ACM Transactions on Software Engineering and Methodology, 1996, 5(2):146-189.

(编辑 苗凌)