文章编号:1001 - 9081(2005)04 - 0827 - 03

基于概念格的数据挖掘方法研究

王旭阳,李　明
(兰州理工大学 计算机与通信学院,甘肃 兰州 730050)
(wangxy_liaj@sina.com)

摘　要:分析了概念格和关联规则之间的关系,提出了将频繁项集及其支持度存储在概念格上,然后在创建好的概念格上提取关联规则的方法,以及数据发生变化以后概念格的维护算法。

关键词:概念格;关联规则;在线挖掘;增量维护
中图分类号: TP311.131　　**文献标识码**:A

Method of data mining based on concept lattice

WANG Xu-yang, LI Ming
(*College of Computer and Communication, Lanzhou University of Technology, Lanzhou Gansu* 730050, *China*)

Abstract: The relation between concept lattice and association rule was analysed, a method to store the frequent item set on concept lattice was expounded, and the rules were abstracted from it. At last, an algorithm of maintenance was proposed when database has been changed.

Key words: concept lattice; association rule; online mining; incremental maintenance

0　引言

关联规则是数据挖掘领域中的一个重要课题,揭示了数据间的相互关系。关联规则的挖掘就是从一组给定的数据项以及交易集合(每一条交易是一个数据项的集合)中,分析出数据项集在交易集合中出现的频度关系。挖掘关联规则的算法已经有很多,比较重要的有 Apriori 算法[3]、挖掘定量关联规则算法和 DIC 算法[9]。

已有的一些挖掘关联规则的算法往往是离线的或者批处理式的,而在文献[4、12]中提出了挖掘关联规则找出数据项频集的在线算法 Car-ma。所谓在线算法,是相对于批处理式算法而言,它有以下特点:1)算法执行过程中即能不断产生部分计算结果,供用户参考;2)在算法执行过程中,用户能根据产生的部分计算结果控制算法如何进行下去;3)算法给出的结果必须是精确的。在线挖掘关联规则的算法允许用户随时调整最小支持度(阈值),以得出合理的结果,如果中间结果已经令人满意,用户也可以随时终止算法的执行。

在知识发现领域,概念格(形式概念分析理论中的一种核心数据结构)可以从关系数据中构造出来,然后从概念格上可以提取各种类型的知识,如蕴含规则、关联规则、分类规则等[5]。

Apriori 算法是最有影响的挖掘布尔型频繁项目集的算法,该算法采用逐层搜索迭代:首先找出频繁 1-项目集 l_1,在 l_1 上找出频繁 2-项目集 l_2,依次类推,直到找不到频繁 k-项目集 l_k 为止。该算法存在如下问题:1)所挖掘的规则存在大量冗余,可能出现组合爆炸;2)挖掘规则的计算量呈指数增加,因而提交给用户的规则数量激增,导致提取频繁项目集之间所蕴涵的用户感兴趣的规则变的更加困难。本文所描述的基于概念格的关联规则挖掘方法能有效地解决上述两个问题。运用概念格技术挖掘关联规则,首先需要将频繁项集及其支持度存储在概念格上,然后在创建好的概念格上提取关联规则,最后还要考虑数据增加以后概念格的维护。

1　基本理论

假设给定形式背景为三元组 $T = (O,D,R)$,其中 O 是事例集合,D 是描述符(属性)集合,R 是 O 和 D 之间的一个二元关系,则存在唯一的一个偏序集合与之对应,并且这个偏序集合产生一种格结构,这种由背景 (O,D,R) 所诱导的格 L 就称为一个概念格[6]。格 L 中的每个结点是一个序偶(称为概念),记为 (Y,X),其中 Y 是幂集 $P(O)$ 中的事例集合,称为概念的外延;$X \in P(D)$ 是 Y 中所有事例的共同描述符的集合,称为概念的内涵。每一个序偶关于关系 R 必须是完备的,即只有最大扩展的序偶才出现在概念格结构中。一个序偶 $(Y,X) \in P(O) \times P(D)$ 关于关系 R 是完备的,当且仅当满足性质:

(1) $Y = \{y \in O \mid x \in X, xRy\}$

(2) $X = \{x \in D \mid y \in Y, xRy\}$

表 1　形式背景例子

	A	B	C	D
1	a1	b1	c1	d1
2	a1	b2	c1	d2
3	a2	b1	c2	d3
4	a3	b3	c1	d4

在概念格结点间能够建立起一种偏序关系,给定 $H1 = (Y1, X1)$ 和 $H2 = (Y2, X2)$,则 $H1 < H2 \Leftrightarrow X1 \subseteq X2$,领先次序意味着 $H1$ 是 $H2$ 的父结点或称直接泛化。实际上,格中集合 $X1$ 和 $X2$ 之间存在着对偶关系,即 $X1 \subseteq X2 \Leftrightarrow Y2 \subseteq Y1$,从

收稿日期:2004 - 09 - 09;**修订日期**:2004 - 11 - 25　　**基金项目**:甘肃省自然科学基金资助项目(ZS022 - A25 - 035)

作者简介:王旭阳(1974 -),女,甘肃陇西人,讲师,硕士研究生,主要研究方向:粗糙集理论、数据挖掘与人工智能;　李明(1959 -),男,甘肃兰州人,副教授,研究生导师,主要研究方向:粗糙集理论、数据挖掘与人工智能.

而$H1 < H2 \Leftrightarrow Y2 \subseteq Y1$。所以，概念格本质上是相互联系的两个格。格的 Hasse 图是根据偏序关系产生的：如果 $H1 < H2$ 并且在格上不存在另一个元素 $H3$ 使得 $H1 < H3 < H2$，则从 $H1$ 到 $H2$ 就存在一条边。Hasse 图揭示了概念的内涵与外延之间的泛化关系和特化关系，可作为数据分析与知识获取的一种工具。

表 1 表示了一个形式背景。其中 $O = \{1,2,3,4\}$，$D = \{a1,a2,a3,b1,b2,b3,c1,c2,d1,d2,d3,d4\}$，$R$ 描述了 O 中元素所拥有的 D 中的属性值集。图 1 是相应概念格的 Hasse 图。

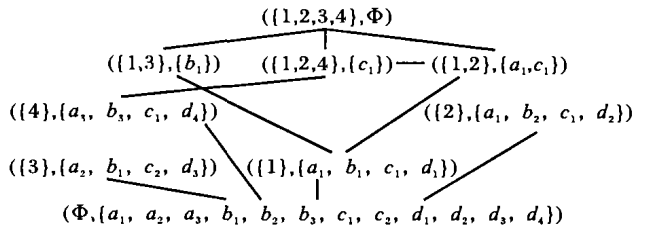

图 1 表 1 对应的概念格的 Hasse 图

从定义可以看出，每个结点（概念）$C_1 = (O_1,D_1)$ 的内涵都是最大化的：对于任意 $D_2 \subseteq D_1$，都有 $g(D_2) \subseteq g(D_1) = O_1$，即对内涵的任意扩充都会使相应的外延缩小。由此，可以定义出概念结点内涵的最小化 —— 内涵缩减。

定义 1 对于概念格中的结点（概念）$C_1 = (O_1,D_1)$，特征子集 D_2 被称为是 (O_1,D_1) 的内涵缩减当且仅当：

(1) $g(D_2) = g(D_1) = O_1$

(2) $D_3 \subseteq D_2, g(D_3) \supseteq g(D_2) = O_1$

其中条件(1) 称为内涵缩减的外延不变性，即结点 (O_1, D_1) 的内涵 D_1 和它的内涵缩减 D_2 具有相同的外延。条件(2) 称为内涵缩减的最小性，即从中任意去除一个属性将会导致外延的增加。C_1 的所有内涵缩减的族集被标记为 $INTRED(C_1)$。

定理 1 对于概念格中的结点（概念）$C = (O_1,D_1)$ 以及特征子集 D_2，有 $g(D_2) = g(D_1) = O_1$，当且仅当 $D_2 \subseteq D_1$ 且对于 C 的任意父结点 $C_p = (O_3,D_3)$ 有 $D_2 \cap (D_1 - D_3) \neq \varnothing$。

2 运用概念格技术挖掘关联规则

事务数据库 TD 可以方便地理解成为一个形式背景 $K = (U,D,R)$，其中 U 为数据库 TD 中事务的集合，D 为数据库中所有可能特征的集合，对于 $x \in U, d \in D$（xRd 当且仅当 d 属于事务 x 的项集）。根据与形式背景 K 对应的概念格，可以计算出所有的频繁项集。

对于关联规则 $A \Rightarrow B$，其支持度 $supp(A \Rightarrow B) = |g(A \cup B)| / |U|$，可信度 $conf(A \Rightarrow B) = |g(A) \cap g(B)| / |g(A)|$。关联规则 $A \Rightarrow B$ 被称为是 (θ,φ)- 关联规则，如果 ①$A \cup B$ 为频繁的，即 $supp(A \Rightarrow B) > \theta$，且 ②$conf(A \Rightarrow B) > \varphi \Leftrightarrow |g(A \cup B)| / |g(A)| > \varphi$，即 $|g(A)| < |g(A \cup B)| / \varphi$。

可以观察到属性值集合（即项目集）之间的关系在概念格之间得到了体现。若把 O 看做是关联规则中的交易集（$TIDs$），D 作为项目集，R 描述它们之间的关系，即哪些交易具有相同的项目集。那么，每个格节点实际上就是一个最大项目集。格中的节点可表示为 (C,X)，其中 C 是事例集合 Y 的基数，X 是它们共有的属性。现在能够很明显地看出，对于每个节点 (C,X)，若 C 大于最小支持度 t，则 X 就是一个最大频繁集，而 C 是频繁集的支持度。任何非最大的频繁项集的支持度都可以通过对格的广度优先遍历来找到第 1 个包含此集合的节点而得到，可以从格中导出蕴含规则。规则 $Q \Rightarrow S$ 成立，当且仅当包含 Q 的最小概念同样包含 S。

在上下文中，如果概念 $C_1 = (A_1,B_1)$ 和 $C_2 = (A_2,B_2)$ 满足 $A_1 \subset A_2$，则称 C_1 和 C_2 有子概念 — 超概念关系，C_1 和 C_2 之间的直接子概念 — 超概念的关系定义为 $C_1 < C_2$：如果 $C_1 < C_2$，并且不存在概念 C 使 $C_1 < C < C_2$ 同时成立。概念 C 的广义超概念 $sup(C)$ 定义如下：

(1) C 的直接超概念是 C 的广义超概念；

(2) C 的直接超概念的广义超概念包含于 C 的广义超概念中。

2.1 概念格的构造算法

可以通过逐个插入初始概念并做调整来创建一个概念格，即纵向实现构造。下面给出该构造算法[1,2,7]。

在概念格 L 中插入概念 $C = (N,B)$ 的算法 $Insert(L,C)$ 如下：

(1) 如果 L 存在 C 的等价概念，则简单的将 C 的内涵合并到其等价概念中即完成了插入操作。

(2) 否则，假设 L 中的 C_0 为 C 的直接超概念，将 C 作为 C_0 的子概念并做如下操作：① 将 C 作为每一个 C_i（$C_i < C_0$，$C_i < C$）的直接超概念。②将由 C 与 L 中的概念相交而生成的概念 $C_k = (N_k,I_k)$ 插入到 L 中。

设上下文 (O,D,R) 中的属性集 $A = \{a_1,a_2,\cdots,a_n\}$，由属性 a_i 所产生的划分表示为 $\{O_{i1}/v_{i1}, O_{i2}/v_{i2}, \cdots, O_{iki}/v_{iki}\}$，其中每个 O_{ij}/v_{ij} 表示属性 a_i 取值为 v_{ij} 的对象集为 O_{ij}。因此，由属性 a_i 可得到一组初始概念 $C(a_i) = \{(N_{i1}, a_i = v_{i1}), (N_{i2}, a_i = v_{i2}), \cdots, (N_{iki}, a_i = v_{iki})\}$。

构造 L 的算法 $Create(L)$ 如下：① 初始化：将全概念 (O, Φ) 和空概念 (Φ, all) 作为仅有的两个概念构造出 L。②for $I := 1$ to n do for $C_i \in C(a_i)$ do $Insert(L,C_i)$。

2.2 关联规则提取算法

给定构建好的概念格，可以按如下方法得到关联规则：

(1) 如果概念 $c_1 = (N_1,B_1), C_2 = (N_2,B_2)$ 满足 $C_2 \in sup(C_1)$，则可得到规则：$B_2 \Rightarrow B_1 - B_2$，其可信度是 N_1/N_2；反之，$B_1 - B_2 \Rightarrow B_2$，其可信度是 100%。

(2) 对任意两个概念 $C_1 = (N_1,B_1), C_2 = (N_2,B_2)$，如果存在非空的最大公共子概念 $C = (N,B)$，则可得 B_1, B_2 之间的关联规则：$B_1 \Rightarrow B_2, B_2 \Rightarrow B_1$，可信度分别是：$N/N_1, N/N_2$。

2.3 示例说明

我们用一个简单的事务数据库（表 2）来说明上述算法的执行情况。

采用上述构造算法所生成的关联规则格如图 2 所示。若设定 $\theta = 0.2, \varphi = 0.8$，则生成的规则有：$\Phi \Rightarrow BE$，$\Phi \Rightarrow C$，$B \Rightarrow CE$，$E \Rightarrow BC$，$C \Rightarrow BE$。

表 2 事务数据库

TID	Itemset
1	A, C
2	B, C, E
3	A, B, C, E
4	B, E
5	B, C, E, F
6	B, C, D, E, F

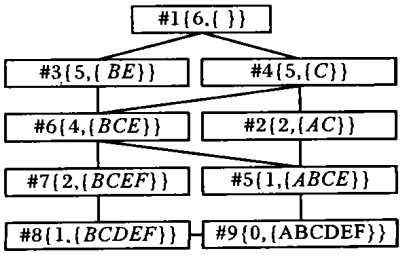

图 2 表 2 对应的概念格

由以上所述得知,基于概念格的挖掘关联规则的方法直观、算法简捷。如果支持度小于支持度阈值,可直接从概念格上删除,同样,如果规则的可信度小于可信度阈值,则忽略其规则,从而避免了无意义的关联规则,提高了挖掘效率,减少了挖掘过程的盲目性。

3　概念格的增量维护

由于数据库是"动态"的,即用户可周期性或者偶然性地递增或递减更新数据,而数据库的更新不仅影响规则的产生,也影响一些现有规则的有效性[10,13]。因此,关联规则的实时维护很重要。本文提出一种改进的基于概念格的递增关联规则挖掘方法。用改进的概念格对项目集间的关系建模,格中的结点用(C,f,X)表示,其中,C和X的含义不变,f记录X在数据库中出现的频率(整数表示),其定义如下:

$$f=\begin{cases}\text{结点出现频率} & X\text{为不变结点}\\ -1 & X\text{为派生结点}\\ 0 & X\text{为删除结点}\end{cases}$$

其中,不变结点的出现次数有助于计算支持度值和可信度值。以下将分数据递增、递减两种情况进行讨论。

3.1　数据递增(加入)

构造 Hasse 图时记录项目集在数据库中出现的频率值f。计算大项目集时,如给定阈值t,则当X的f值不为 -1 时,只需从 $C+f=t$ 的结点入手,而其f值为 -1 时,只需计算 $C=t$ 的结点。下面给出基于概念格的数据递增时大项目集生成方法。

输入:初始概念格 L 为 $N_0*item=(0,0,sup(L))$,$N_0*chd=\Phi$,$N_0*par=\Phi$,欲加入数据为 X,阈值 t;

输出:更新的概念格 L';

方法:

$\forall$ 结点 $N_i\neq NULL(N_i\in L)$,i 为结点个数;

if (IsContained (X,N_i))　　/*判断 X 是否包含 N_i */

{

$X*par=N_i$,$N_i*chd*par=X$,$C(N_i)++$,$f(N_i)++$;

if $(C(N_i)+f(N_i)\geqslant t)$ then $N_i\rightarrow AddMark$;

/* $AddMark$ 为大项目集标志集合 */

if $(N_i=X)$ then {$f(N_i)++$,break};

};

else if ($\exists X_k\in AddMark$,使 $X_k=N_i\cap X$)

{

new 结点 $(C(N_i)+1,0,N_i\cap X)$;

$N_i*par=N_i\cap X$,$X*par=N_i\cap X$,

$(N_i\cap X)*par=N_i*par$,$(N_i\cap X)*par=X*par$;

if $(C(N_i)+1\geqslant t)$ then $N_i\cap X\rightarrow AddMark$;

}

其中,$AddMark$ 存放势与频率之和大于阈值的项目集。计算这些项目集的各阶子集即可得到各大项目集。

3.2　数据递减(删除)

首先,遍历图找到需删除的结点 X,令其频率值减 1。当频率为 0 时,则表明 X 已不存在,一般的方法是把所有与 X 相连的父结点和子结点全部删除。考虑到数据库有加入 X 的可能,因此,不作删除,而是把与 X 相连的父结点和子结点作一标记,并放入 $DeleteMark$ 集合中。待到需加入 X 时,则恢复原来面目,并删除 $DeleteMark$ 中的相应结点和边。也就是说,删除操作完全是在频率值上的演绎。

至于数据同时递增、递减的情况,综合使用 3.1 和 3.2 即可得到所求的大项目集。

以上基于概念格的数据库递增或递减的关联规则挖掘方法,其算法复杂性由递增的新交易数据时所产生的结点数决定,降低了算法复杂度。假定项目集平均由 k 个项目构成,并且假定最坏情况为每个项目集都不完全相同(即频率值均为 1),给定阈值 t,则结点的势最多为 $k+1$,当有新数据加入时,最多会产生 $t*2k/(k+1)$ 个新结点,算法的复杂度最多为 $O(t*2k/(k+1)*\|D\|)$。其中,$\|D\|$ 为数据库的基数,k 和 t 为常数。

4　结语

知识发现是一个比较复杂的过程,关于概念格的深入应用还有许多问题有待研究[11],例如对概念格代数性质的研究;如何把概念格与其他处理不确定性数学工具如粗糙集合理论有效结合以处理不确定性规则提取,以及如何拓广格结点结构以处理更丰富的知识表示模式,容差关系的具体的构造方法等。另外,文中的方法也存在不足,如果属性数很大、项目集很多时,生成概念格的规模会很大,需要的时间和空间较多,对此可以采用动态剪枝等方法来解决,其中的一些算法也需进一步改进和细化,这也是我们下一步努力的方向。

参考文献:

[1] XIE ZP, LIU ZT. A Fast Incremental Algorithm for Building Concept Lattice[J]. Chinese Journal of Computer, 2002, 25(5).

[2] QIAO SY, WEN SP, CHEN CY, *et al*. A Fast Algorithm for Building Concept Lattice[A]. Proceedings of the second International Conference on Machine Learning and Cybernetics[C], Xi'an, 2003.

[3] HAN JW, KAMBER M. Data mining-concepts and techniques[M], San Francisco: Morgan Kaufmann Publishers, 2001. 227 - 236.

[4] Hidber C. Online association rule mining[A], Proceedings of ACM-SIGMOD International Conference on Management of Data[C]. Philadelphia PA: Association for Computing Ma-chinery, 1999, 145 - 156.

[5] Graphviz - Graph Visualization Software[EB/OL]. Http://www.research.att.com/sw/tools/graphviz/, 2002.

[6] BERRY A, SIGAYRET A. Representing a Concept lattice by a graph. Workshop on Discrete Mathematics for Data Mining[A], Proceedings of 2nd SIAM workshop on Data Mining, Arlington(VA)[C], 2002.

[7] 谢志鹏, 刘宗田. 概念格的快速渐进式构造算法[J]. 计算机学报, 2002, 25(5): 490 - 496.

[8] 缑锦, 叶东毅. 基于概念格的求所有绝对属性约简的一个算法[J], 福州大学学报, 2002, 30(3): 298 - 300.

[9] 王琦珍, 简宋全, 胡学钢, 等. 基于相对约简格的关联规则研究[J], 计算机工程, 2001, 27(11): 30 - 32.

[10] 吴刚, 简宋全, 胡学钢, 等. 扩展概念格的维护[J]. 计算机工程与应用, 2002, (38)4: 75 - 77.

[11] 陈世权, 程里春. 模糊概念格[J]. 模糊系统与数学, 2002, 16(4): 12 - 18.

[12] HidberC. Carma: update[EB/OL]. http://www.icsi.berkeley.edu/~hidber/, 1999.

[13] 赵文兵, 简宋全, 王浩等. 约简概念格的纵向维护算法[J]. 计算机工程与应用, 2002, 38(7): 209 - 211.