基于概念格的规则产生集挖掘算法

梁吉业[1,2]　王俊红[2]

[1](中国科学院计算技术研究所智能信息处理重点实验室　北京　100080)

[2](山西大学计算机与信息技术学院　太原　030006)

(ljy @sxu. edu. cn)

摘　要　传统的规则提取算法产生的规则集合相当庞大,其中包含许多冗余的规则. 使用闭项集可以减少规则的数目,而概念格结点间的泛化和例化关系非常适用于规则提取. 基于概念格理论和闭项集的概念,提出了一种新的更有利于规则提取的格结构,给出了相应的基于闭标记的渐进式构造算法和规则提取算法. 最后提供给用户的是直观的、易理解的规则子集,用户可以有选择地从中推导出其他的规则. 实验表明该方法能够高效地挖掘规则产生集.

关键词　概念格;闭项集;规则产生集;规则提取

中图法分类号　TP18

An Algorithm for Extracting Rule-Generating Sets Based on Concept Lattice

LIANG Ji-Ye[1,2] and WANG Jun-Hong[2]

[1](*Key laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)

[2](*School of Computer and Information Technology, Shanxi University, Taiyuan* 030006)

Abstract　The rule sets extracted by traditional algorithm are usually very large, because it includes many redundant rules. The number of rules can be reduced using closed item sets. The relationship of generalization and specialization among concepts of concept lattice is very suitable for extracting rules. A new and more advantageous lattice structure for extracting rules is proposed based on the theory of concept lattice and the concept of closed item set. Then, an incremental algorithm based on closed label for constructing lattice and algorithm for rules extracting are developed. Finally, a visual and easily understandable set of rules is presented to user, who can selectively derive other rules of interest. The example shows that the algorithm used in this paper can efficiently extract rule-generating set.

Key words　concept lattice; closed item set; rule-generating set; rule extracting

1　引　　言

数据库知识发现是当前人工智能研究中较为重要的一个领域[1],而概念格是近年来获得飞速发展的数据分析的有力工具,是由Wille[2]在1982年提出的,是进行数据挖掘和规则提取的一种有效工具. 它的每个结点被称为是一个概念,概念的外延表示为属于这个概念的所有对象的集合,而内涵则表示为所有这些对象所共有的属性的集合. 概念格在本质上描述了对象和属性之间的联系,表明了概念之间的泛化和例化关系,而它的Hasse图则实现了对数据的可视化,作为数据分析和知识处理的形式化工具,概念格理论已被广泛地应用于软件工程、知识工程、数据挖掘、信息检索等领域[3~7].

概念格的每个概念的内涵就是满足一定条件的属性集合,而概念格及其Hasse图又体现了概念的内涵及外延的泛化和例化关系. 因此,概念格可作为挖掘规则的自然平台. Godin等[8]提出了由概念格提取蕴含规则的算法,但得到的规则的数目往往很大. 一些作者[9,10]提出了使用闭项集来进行规则的提取,但目前主要集中在对关联规则频繁项集的

收稿日期:2003-06-09;**修回日期**:2003-09-24

基金项目:国家自然科学基金资助项目(60275019);山西省自然科学基金资助项目(20031036)

挖掘,还没有有效地应用于规则提取.

本文中我们并不是提取所有规则,而是提取规则集中的一个子集,称为规则产生集. 与所有规则的集合相比(因为冗余),它的规模大大减少了,相应的挖掘效率提高了,但是从中仍然可以推导出所有的规则. 这样,我们提供给用户的是较小的而且是容易理解的规则的集合,用户可以根据自己的兴趣有选择地从产生集中推导出其他的规则. 本文通过对概念格的结点进行扩充,提出了一种新的更有利于规则提取的格结构称为闭标记格,在新的格结构上利用闭项集的性质来提取规则产生集.

本文其余部分组织如下:第 2 节介绍概念格和闭项集的基本概念;第 3 节提出一种新的格结构及其渐进式的构造算法;第 4 节给出在已建好的格上提取规则产生集的算法,并通过实例说明算法的思想及其实现过程;第 5 节分析了实验结果;第 6 节总结全文.

2 基本定义

2.1 概念格

假设给定形式背景(context)是一个三元组(U, D, R),其中 U 是对象的集合,D 是属性的集合,R 是 U 和 D 之间的一个二元关系,则存在惟一的一个偏序集合与之对应,并且这种偏序集合产生一种格结构,这种由形式背景(U, D, R)所诱导的格 L 就称为是一个概念格[2]. 格 L 中的每个结点是一个序偶(称为概念),记为(X, Y),其中 $X \in P(U)$ 称为概念的外延,$Y \in P(D)$ 称为概念的内涵. 这里每个序偶关于关系 R 都是完备的,即在 U 的幂集 $P(U)$ 和 D 的幂集 $P(D)$ 之间存在两个映射 f 和 g:

$$\forall X \in P(U), f(X) = \{y \in D \mid \forall x \in X, xRy\};$$
$$\forall Y \in P(D), g(Y) = \{x \in U \mid \forall y \in Y, xRy\}.$$

格 L 中所有概念的集合用 $L(K)$ 来表示. 给定 $L(K)$ 中的两个元素:$H_1 = (X_1, Y_1)$ 和 $H_2 = (X_2, Y_2)$,定义 $H_1 \leq H_2 \Leftrightarrow Y_2 \subseteq Y_1 \Leftrightarrow X_1 \subseteq X_2$,则 $\leq$ 就是 $L(K)$ 中的一个偏序关系. 根据此偏序关系可以生成格的 Hasse 图.

2.2 闭项集

下面简要介绍所需要的闭项集的基本概念,关于闭项集的详细描述可参考文献[9,10].

定义 1[10]. 设(U, D, R)是一个形式背景,(f, g)称为是在 U 的幂集 $P(U)$ 和 D 的幂集 $P(D)$ 之间的一个 Galois 连接,在 $P(U)$ 上的运算 $h = f \circ g$ 和 $P(D)$ 上的运算 $h' = g \circ f$ 称为 Galois 闭运算.

定义 2[10]. 设(U, D, R)是一个形式背景,$C \subseteq D$ 是属性的集合,则 C 是一个闭项集当且仅当 $h(C) = C$.

也就是说,闭项集就是有共同对象集的最大的属性的集合. 那么,由闭项集的定义和概念格的性质可以得出下面的结论:

推论 1. 概念格的每个结点的内涵必是一个闭项集.

推论 2. 概念格的所有结点的内涵的集合就等于一个形式背景可导出的闭项集的集合.

3 闭标记格及其渐进式构造算法

现在我们给出一种新的更有利于规则提取的格,称为闭标记格,它只是对概念格中的每个结点进行扩充,并不改变原来的格结构.

定义 3. 设 C 是概念格中的一个概念,用 $extention(C)$ 表示概念的外延,$intention(C)$ 表示概念的内涵. C 的闭标记记为 $Closedlabel(C)$,$Closedlabel(C)$ 是一个集合,$\forall X \in Closedlabel(C)$ 满足以下条件:

① $h(X) \neq X$;

② $h(X) = intention(C)$;

③ $\forall Y \subset X, h(Y) \subset h(X)$.

从闭标记的定义我们可以看到,一个概念的闭标记就是进行 h 运算后等于该概念的内涵的非闭项集的集合,且这些非闭项集的任一子集都不满足这一条件. 另外要注意的一点就是,对于一些非闭项集,虽然它也满足上述性质,但是由于在闭标记中存在一些元素,这些元素是它的子集,那么也不把它包含到闭标记里面. 也就是说,一个概念的闭标记中的每个元素就代表着这个概念,且只能代表该概念,而且这些元素又具有最简单的形式,含有该概念的信息. 这样,我们就可以用一个概念的闭标记中的每个元素来表示该概念.

定理 1. 对于只有一个父结点 C 的概念 C,它的闭标记即是 $Closedlabel(C) = \{x \mid x \in intention(C) - intention(C)\}$.

证明. 对于只有一个父结点的概念,它的闭标记即是相对于父结点新增属性的集合,因为对新增属性进行 h 运算后就等于该概念的内涵. 而其他的一些非闭项集,虽然进行 h 运算后也等于该概念的内涵,但其含有属性个数都大于 1.

定理 2. 对于有两个父结点 C_1', C_2'的概念 C, 若 $intention(C) - (intention(C_1') \cup intention(C_2')) \neq \varnothing$,则 $Closedlabel(C) = \{x \mid x \in (intention(C) - (intention(C_1') \cup intention(C_2')))\}$. 若 $intention(C) - (intention(C_1') \cup intention(C_2')) = \varnothing$,则 $Closedlabel(C) = \{x_1 x_2 \mid x_1 \in intention(C) - intention(C_1'), x_2 \in intention(C) - intention(C_2')\}$.

证明. 若概念 C 相对于它的父结点的内涵的并有新增属性,则它的闭标记就是所有单个的新增属性,它的形式更简单. 否则,即概念 C 相对于它的父结点的内涵的并集没有新增属性,那么求出进行 h 运算后等于该概念的内涵的非闭项集,用 $x_1 x_2$ 来表示,其中 $x_1 \in intention(C) - intention(C_1')$, $x_2 \in intention(C) - intention(C_2')$,容易看出, $h(x_1 x_2) = intention(C)$. 求出这样的属性对的集合,它们分别单独地代表该概念,而根据定义 3 的第③个条件,其他的形式都是冗余的,是不可取的.

对于有两个以上父结点的情形,可根据定理 2 相应地推出. 可以使用算法 1 来生成格中一个概念的闭标记.

算法 1. 计算概念 N 的闭标记 $Computeclosed(N)$.

输入:结点 N 的父结点的集合 $parents(N)$.

输出:结点 N 的闭标记 $Closedlabel(N)$.

步骤 1. 若 N 的外延为空或内涵为空,则 $Closedlabel(N) := \varnothing$,转步骤 6.

步骤 2. 求出 $parents(N)$ 中所有元素的内涵的并集 L.

步骤 3. 若 $L \subset intention(N)$,即有新增属性,则 $Closedlabel(N) := \{x \mid x \in intention(N) - L\}$,转步骤 6.

步骤 4. 若 $L = intention(N)$,即无新增属性,则对于 N 的 $parents(N)$ 中的任两个元素 M_i, M_j 分别计算: $L_i := intention(N) - intention(M_i)$, $L_j := intention(N) - intention(M_j)$, $Closedlabel(N) := Closedlabel(N) + \{x_i x_j \mid$ 其中 $x_i \in L_i$, $x_j \in L_j$ 且 $x_i x_j \neq intention(M_i)$, $x_i x_j \neq intention(M_j)\}$.

步骤 5. 若 $Closedlabel(N) = \varnothing$,则从 N 的 $parents(N)$ 中任取 3 个元素,依步骤 4 的方法计算,依此类推,直到 $Closedlabel(N) \neq \varnothing$.

步骤 6. 结束,返回 $Closedlabel(N)$.

从算法 1 可以看出,若概念的外延为空,那么它的闭标记也为空. 因为若一个概念的外延为空,说明没有对象满足该概念的内涵,那么由它推导出的规则也没有任何意义,因而是冗余的. 算法 1 中步骤 3 用于求一个概念相对于它的惟一的父结点或多个父结点的集合有新增属性时的闭标记,若有,则根据定理 1 和定理 2,它们就是能代表该概念的最简单的非闭项集,则跳过步骤 4,步骤 5 因为新增属性就可以代表该概念. 另外,除内涵或外延为空的结点外,每个结点的闭标记都不为空,也就是说总有属性集来表示该概念. 通过闭标记的概念我们定义闭标记格.

定义 4. 闭标记格中的每个结点是一个三元组 $(extention(C), Closedlabel(C), intention(C))$,称其为闭标记格的概念.

已经有一些根据二元关系来生成概念格的算法,这些建造算法可分为两类[11]:批处理算法和增量算法. 其中 Godin 等人提出了一种渐进式的构造格及其 Hasse 图的算法[12],谢志鹏等[13]给出了构造关联规则格的算法,都是通过动态地增加对象生成格结构. 下面给出的算法 2 的思想与上述算法相似,但是,由于格结点结构的不同而进行了相应的修改.

算法 2. 基于闭标记的渐进式建格算法.

输入:已有的格 L,要追加的对象为 x,它满足的属性集合用 $f(x)$ 表示,即在已有的格中添加概念 $(x, \varnothing, f(x))$.

输出:更新后的格 L'.

步骤 1. 初始化 $Mark := \varnothing$.

步骤 2. 对格中的每个结点 C 按 $Card(intention(C))$ 升序排列.

步骤 3. 对于结点 C,若 $intention(C) \subset f(x)$,则把 x 加入到 $extention(C)$ 中;将 C 加入到 $Mark$ 中,转步骤 7. 若 $intention(C) = f(x)$,则把 x 加入到 $extention(C)$ 中;将 C 加入到 $Mark$ 中,转步骤 8.

步骤 4. 令 $Int := intention(C) \cap f(x)$,若不存在 C 的祖先结点 $C_i \in Mark$,使得 $intention(C_i) = int$,则增加新结点 N; $intention(N) := int$; $extention(N) := extention(C) \cup x$;增加 N 到 $parents(C)$,并增加 C 到 $children(N)$,否则转步骤 7.

步骤 5. 取出 $Mark$ 中的一个元素 MP.

①若 $intention(MP) \subset intention(N)$,则 $Parent := \text{true}$.

②对每个 $M \in children(MP)$,若 $intention(M) \subset intention(N)$,则 $Parent := \text{false}$,退出②

③若 $Parent$ 为真,则将 N 添加到 $children(MP)$,并将 MP 添加到 $parents(N)$;若 $MP \in parents(C)$,则从 $parents(C)$ 中删除 MP,并从 $children(MP)$ 中

删除 C. 重复步骤5.

步骤6. 调用 $Computeclosed(N)$, 计算新增结点 N 的闭标记, 同时重新计算 N 的每个子结点的闭标记, 即对每个 $Y \in children(N)$, 调用 $Computeclosed(Y)$.

步骤7. 取出下一个结点, 转步骤3.

步骤8. 结束, 输出更新后的格 L'.

4 规则提取

概念格是进行数据挖掘和规则提取的一种有效工具, 可以直接从概念格中根据泛化和例化关系来提取规则, 但提取的规则数目往往很庞大, 会产生很多冗余的规则. 王志海等[14]提出了在概念格上提取规则的一般算法和渐进式算法, 主要依据格结点的直接泛化来产生相应的无冗余规则, 提取的规则与文献[15]具有相同的形式.

在本节中我们使用闭标记格来提取规则产生集, 产生集占有很小的存储空间, 而且其他的规则都可以从产生集中推出. 有时用户感兴趣的只是整个规则集合中的一部分规则, 比如说和某些属性有关的规则, 那么就可以有选择地找出一部分规则, 再推导出其他的规则. 但本文中提取的产生集并不是最小的规则集合, 不同结点之间产生的规则也可能存在冗余.

定理3. 设 X_1, X_2 是概念格中的闭标记, 则规则 $X_1 \to X_2$ 成立当且仅当 $h(X_1) \to h(X_2)$ 是成立的.

证明: 由闭项集的定义, X_1 与 $h(X_1)$ 对应的对象集相同, X_2 与 $h(X_2)$ 对应的对象集相同, 所以定理的成立是显然的.

同时, 对于概念格中的任意的一个闭标记 X, $X \to h(X)$, $h(X) \to X$ 是成立的.

在闭标记格中, 闭标记的闭项集即是概念的内涵, 闭标记是能代表概念的内涵的最简单的属性集, 所以我们在提取规则产生集时只考虑闭标记.

定理4. 设 X_1, X_2, X_3 是概念格中的闭标记, 且 $h(X_1) \subseteq h(X_2) \subseteq h(X_3)$. 若 $X_1 \to X_2$, $X_2 \to X_3$, 则 $X_1 \to X_3$.

证明. 根据概念格结点间的泛化和例化的关系, 上述定理的成立是显然的.

定理4说明在格上提取规则的时候只需考虑邻近的概念, 也就是说, 父子结点, 不相邻结点产生的规则都可以通过传递性得到.

定理5. 若一个结点包含两个以上的新增属性, 则这些属性可以相互推导.

证明. 因为这两个属性是在同一个概念中而且是同时出现的, 所以它们一定对应相同的对象集, 即该概念的外延, 所以它们之间是相互蕴含的. 设 a, b 是两个新增属性, 则 $a \to b$, $b \to a$ 是两条规则.

定理6. 对于只有一个父结点 C' 的概念 C, $\forall X \in Closedlabel(C)$, $\forall Y \in Closedlabel(C')$, 有 $X \to Y$ 成立.

证明. 同定理4.

定理7. 对于有两个父结点 C_1', C_2' 的概念 C, $\forall X \in Closedlabel(C)$, $\forall Y \in Closedlabel(C_1')$ 或 $Y \in Closedlabel(C_2')$, 若 Y 不包含于 X, 则 $X \to Y$ 是一条规则.

证明. 若 Y 不包含于 X, 因为概念格中的一个概念相对于它的每个父结点都是例化的, 所以 $X \to Y$ 是成立的. 若 $Y \subseteq X$, 规则 $X \to Y$ 是显然成立的, 因而是冗余的, 可以去掉这样的规则.

定理8. 从规则产生集可推导出所有的规则.

证明. 因为规则产生集是由闭标记(而不是概念的内涵)产生的, 而闭标记可以代表该概念, 也就是说, 闭标记是所有其他的能表示概念的内涵的属性集的子集, 那么其他所有形式的规则都可以根据定理3～7推导得到.

从规则产生集中推导出规则可以采用以下的方法:

(1) 向规则的前件或后件中添加属性(添加同一闭包中的属性才有意义);

(2) 根据定理4采用递推的方法进行推导;

(3) 合并若干规则, 如 $W \to C$, $D \to C$, 则 $DW \to C$.

算法3. 在闭标记格上提取规则产生集

输入: 已生成的闭标记格 L.

输出: 规则产生集 Σ.

步骤1. 初始化, $\Sigma := \varnothing$

步骤2. 对格 L 中的每个概念 C 按 $Card(intention(C))$ 升序排列并放入 $Mark$ 中, $Mark' := Mark$.

步骤3. 取出 $Mark$ 中的一个概念 C, $Mark' := Mark' - C$. 若 $Closedlabel(C) = \varnothing$, 则转步骤10.

步骤4. $parents'(C) := parents(C)$.

步骤5. 取出 $parents'(C)$ 中的一个概念 N, 令 $parents'(C) := parents'(C) - N$. 若 $Closedlabel(N) = \varnothing$, 则转步骤8.

步骤6. $S := Closedlabel(C)$, 若 $Closedlabel(N)$ 包含两个或两个以上的新增属性, 则从 S 中去

掉包含这些新增属性的元素.

步骤 7. 对每个 S 中元素 X 和 $Closedlabel(N)$ 中的元素 Y,若 C 有 3 个或 3 个以上的父结点,则若 Y 与 X 没有交集,则产生规则 $X \to Y$, $\Sigma := \Sigma \cup \{X \to Y\}$;若 C 有一个或两个父结点,则若 Y 不包含于 X,则产生规则 $X \to \{X - Y\}$, $\Sigma := \Sigma \cup \{X \to \{X - Y\}\}$;若 S 本身包含两个或两个以上的新增属性,则对任意的两个属性 a, b,产生规则 $a \to b$, $b \to a$, $\Sigma := \Sigma \cup \{a \to b, b \to a\}$.

步骤 8. 若 $parents'(C) \neq \varnothing$,则转步骤 5.

步骤 9. 若 S 中存在没有产生规则的元素 X,则产生规则 $X \to \{intention(C) - X\}$.

步骤 10. 若 $Mark' \neq \varnothing$,则转步骤 3.

步骤 11. 结束,输出 Σ.

关于算法 3 要说明一点,若一个概念的父结点的闭标记中有两个新增属性 a, b,而它的子结点的闭标记中有形如 ca, cb 的元素,则产生规则 $ca \to b$, $cb \to a$. 但这两条规则相对于规则 $a \to b$, $b \to a$ 是冗余的,所以可以去掉此类规则. 除新增属性外,每个闭标记的元素都产生且仅产生一条规则.

下面我们用一个简单的形式背景来说明上述算法的思想和执行情况.

例 1. 给定的形式背景如表 1 所示. 图 1 是根据表 1 的形式背景渐进式所建的闭标记格,从格中可以提取到的规则产生集为$\{f \to e, d \to h, h \to d, a \to i, b \to g, b \to f, cd \to e, ed \to c, ch \to e, eh \to c, ca \to g, ga \to ci, gc \to a, gi \to a, ei \to f, fi \to ec, fc \to ei, di \to a, hi \to a\}$,共有 19 条规则. 下面举例说明规则的产生过程. 例如对于概念$(2, \{gc, ci, ga, ca\}, acgi)$,由它所提取出来的规则为$\{ca \to g, ga \to ci, gc \to a, gi \to a\}$,由这几条规则可推导出$\{gca \to i, gia \to c, gci \to a, gi \to ac, gc \to ai, ca \to gi\}$等规则. 所以我们提取出来的规则形式更为简单. 从以上可以看出,规则产生集只是一个较小的规则集合,若要得到所有的规则,还需要从规则集中进行推导.

表 1　形式背景

对象	属性									
	a	b	c	d	e	f	g	h	i	j
0	0	1	0	0	1	1	1	0	0	0
1	0	0	1	1	1	0	0	1	0	0
2	1	0	1	0	0	0	1	0	1	0
3	0	0	1	0	1	1	0	0	1	0
4	1	0	0	1	0	0	0	1	1	0

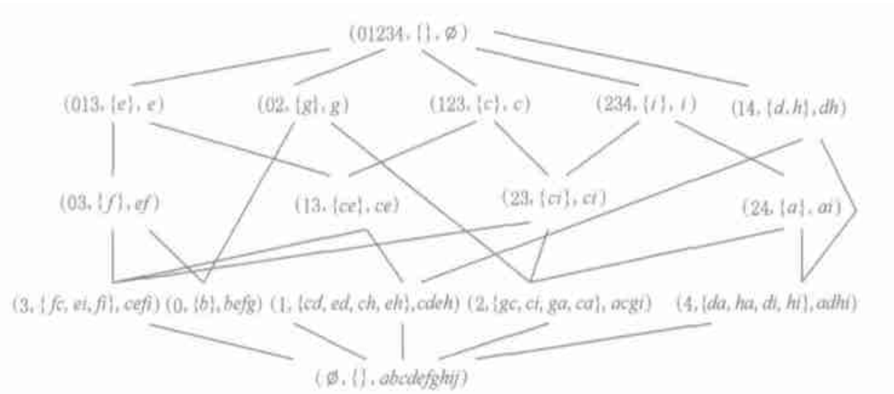

图 1　表 1 的形式背景所建概念格的 Hasse 图

5　实验结果分析

渐进式概念格构造算法的时间复杂性在文献[12]中已有分析,为 $O(2^k \| L \|)$,其中, $\| U \|$ 为对象的数目, k 为每个对象对应的属性集的个数 $\| f(\{x\}) \|$的最大值, $\| L \| = 2^k \| U \|$为格中结点的个数. 本文中闭标记格的构造算法是在渐进式概念格构造算法的基础上加入了闭标记的计算,而对于每个新生结点或产生子的闭标记的计算是通过遍历它的父结点,父结点的个数至多为 $2^k - 1$,所以整个算法的时间复杂度为 $O(\| L \|(2^k + 2^k - 1)) = O(2^k \| L \|)$. 规则提取算法的 for 循环的执行次数为结点的父结点的个数,所以在闭标记格上提取整个产生集的时间复杂度为 $O(\| L \|(2^k - 1)) = O(\| L \| 2^k)$. 然而,实际应用中父结点的个数远少于此,而且在格的构造过程中已经记录了每个结点的父结点,提取产生集时算法的效率会有所提高.

上面的算法我们已在 Windows XP 环境下用 Delphi5.0 实现,并在随机生成的数据集上进行了实验. 我们在选定属性个数为 15,在每个对象具有 5 个属性的数据集上,从闭标记格的生成到规则提取进行了测试,对不同的对象个数进行多次测试的平均时间与对象个数的关系由图 2 来表示,此结果说

明,算法的执行具有一定的稳定性.

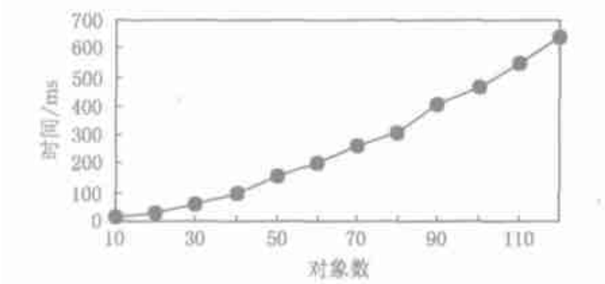

图 2 算法执行时间与对象集的关系

图 3 表明了在上述条件下对象的个数与产生集中规则个数的关系,同时与文献[15]的结果进行比较. 图 3 中上面的曲线对应文献[15]的算法的结果,下面的曲线对应于本文的算法. 从图中可以看出,本文的算法比传统的算法产生的规则要少,而且,随着对象的增加,数据集的增大,规则的个数增加缓慢. 所以当属性个数一定时,随着对象数的增加,本文的算法具有一定的优势.

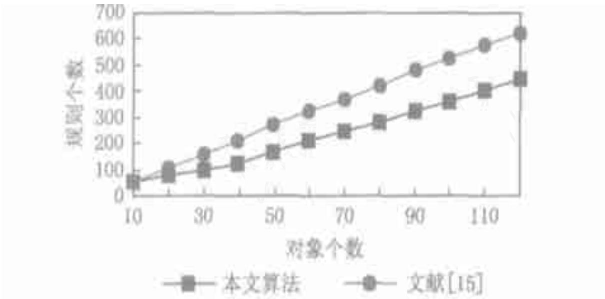

图 3 规则个数与对象集的关系

6 结束语

本文提出了一种有利于规则提取的基于闭项集的概念格方法,给出了基于闭标记格的渐进式建格算法,在此基础上提出了提取规则产生集的算法. 它的最大特点是提供给用户一个较小的、易理解的规则集合,在闭标记格上可以进行关联规则提取.

参 考 文 献

1 史忠植. 知识发现. 北京: 清华大学出版社, 2002
(Shi Zhongzhi. Knowledge Discovery (in Chinese). Beijing: Tsinghua University Press, 2002)

2 R Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In: I Rival ed. Ordered Sets. Dordrecht-Boston: Reidel, 1982. 445～470

3 R Godin, G Mineau, R Missaoui, *et al*. Applying concept formation methods to software reuse. International Journal of Knowledge Engineering and Software Engineering, 1995, 5(1): 119～142

4 G W Mineau, R Godin. Automatic structuring of knowledge bases by conceptual clustering. IEEE Trans on Knowledge and Data Engineering, 1995, 7(5): 824～828

5 C Carpineto, G Romano. A lattice conceptual clustering system and its application to browsing retrieval. Machine Learning, 1996, 24(2): 95～122

6 R Cole, P Eklund. Scalability in formal concept analysis. Computational Intelligence, 1999, 15(1): 11～27

7 R Cole, P Eklund, G Stumme. CEM —A program for visualization and discovery in email. In: The 4th European Conf on Principles and Practice of Knowledge Discovery in Databases. Berlin: Springer-Verlag, 2000

8 R Godin, R Missaoui. An incremental concept formation approach for learning from databases. Theoretical Computer Science, 1994, 133: 387～419

9 N Pasquier, Y Bastide, R Taouil, *et al*. Discovering frequent closed itemsets for association rules. The 7th Int'l Conf on Databases Theory. Jerusalem, Israel, 1999

10 N Pasquier, Y Bastide, R Taouil, *et al*. Efficient mining of association rules using closed itemset lattice. Information Systems, 1999, 24(1): 25～46

11 胡可云, 陆玉昌, 石纯一. 概念格及其应用进展. 清华大学学报, 2000, 40(9): 77～81
(Hu Keyun, Lu Yuchang, Shi Chunyi. Advances in concept lattice and its application. Journal of Tsinghua University (in Chinese), 2000, 40(9): 77～81)

12 R Godin, R Missaoui, H Alaoui. Incremental concept formation algorithms based on Galois (concept) lattices. Computational Intelligence, 1995, 11(2):246～267

13 谢志鹏, 刘宗田. 概念格与关联规则发现. 计算机研究与发展, 2000, 37(12): 1415～1421
(Xie Zhipeng, Liu Zongtian. Concept lattice and association rule discovery. Journal of Computer Research and Development (in Chinese), 2000, 37(12): 1415～1421)

14 王志海, 胡可云, 胡学钢, 等. 概念格上规则提取的一般算法与渐进式算法. 计算机学报, 1999, 22(1): 66～70
(Wang Zhihai, Hu Keyun, Hu Xuegang, *et al*. General and incremental algorithms of rule extraction based on concept lattice. Chinese Journal of Computers (in Chinese), 1999, 22(1): 66～70)

15 R Missaoui, R Godin. Search for concepts and dependencies in databases. In: W P Ziarko ed. Rough Sets and Fuzzy Sets and Knowledge Discovery. London: Springer-Verlag, 1994. 16～23

梁吉业 男,1962 年生,博士后,教授,博士生导师,主要研究方向为粗糙集理论、数据挖掘和人工智能.

王俊红 女,1979 年生,硕士,主要研究方向为概念格、知识发现.