

# 基于量化规则格的关联规则之分布式获取<sup>\*</sup>)

李 云<sup>1</sup> 蔡俊杰<sup>1</sup> 刘宗田<sup>2</sup> 陈 峻<sup>1</sup> 李 拓<sup>1</sup>

(扬州大学信息工程学院 扬州 225009)<sup>1</sup> (上海大学计算机学院 上海 200072)<sup>2</sup>

**摘 要** 在形式概念分析中,提取关联规则是构造概念格的目的之一。量化规则格是为了便于提取规则的扩展概念格。概念格分布处理是先分布构造多个部分概念格,再合并构造出完整概念格,以提高构造格的速度。本文提出一种规则合并技术和相应的算法,由部分量化规则格提取部分关联规则,然后直接进行部分关联规则的合并,得到完整的关联规则,即对关联规则进行分布提取。由于无需构造完整格,并且直接再利用已有的部分关联规则,从而降低了利用概念格提取关联规则的复杂度。最后,给出一个简单的实例,说明算法的正确性,并通过实验验证了算法的有效性。  
**关键词** 量化规则格,关联规则,分布获取

## Distributed Acquisition of Association Rules Based on Quantitative Rule Lattice

LI Yun<sup>1</sup> CAI Jun-Jie<sup>1</sup> LIU Zong-Tian<sup>2</sup> CHEN Ling<sup>1</sup> LI Tuo<sup>1</sup>

(Institute of Information Engineering, Yangzhou University, Yangzhou 225009)<sup>1</sup>

(School of Computer Science, Shanghai University, Shanghai 200072)<sup>2</sup>

**Abstract** In the research of FCA, rule mining is one of the purposes for constructing concept lattice. The quantitative rule lattice is the extended concept lattice for extracting rules conveniently, and distributed treatment of lattice is to construct many part lattices distributively, then obtain the complete lattice by merging these part lattices in order to accelerate the speed of concept lattice construction. This paper presents a method of uniting rules and corresponding algorithms, the part rules are extracted from part quantitative rule lattice, and then the final rules are obtained by direct uniting these part rules. In this paper we propose a method for the rules, we focus on how to unite the rules from a portion of concept lattice. As dispensing with final lattice, and reusing the association rules which have already extracted, the complexity of rules acquisition from concept lattice will be decreased. Finally, a specified example and experiment are introduced to illustrate the correctness and effectiveness of our solution.

**Keywords** Quantitative rule lattice, Association rules, Distributed acquisition

## 1 引言

关联规则挖掘是数据挖掘中的一个重要问题,关联规则反映了数据库中不同属性(字段)之间的联系。比如在一个商品数据库中,一条关联规则可以反映出商品之间搭配出售的内在联系,“牛奶+啤酒→尿布”这条规则能反映出有一定比率的顾客在购买牛奶和啤酒的同时,他还购买了尿布。这种规则对销售人员有重要的指导作用。

最早有 Agrawal 等人<sup>[1]</sup>提出挖掘这种顾客交易数据库中项集间的关联规则问题,引起了很多研究人员的兴趣,并对原有的算法进行了不同的优化,但仍无法满足庞大的动态数据集的计算要求。当形式概念分析<sup>[2,3]</sup>(Formal Concept Analysis)产生以后,以概念格为工具提取关联规则的算法得到了很快的发展<sup>[4~6]</sup>。目前,基于概念格提取关联规则的都是把构格和提取规则分开处理。先构造概念格,然后在构好的概念格上提取规则。由于构格的固有复杂度,从而使利用概念格提取规则的效率不高。为了更有效地在概念格上提取关联规则,一般采用两种方法来加快构造概念格的效率:(1)改进构造概念格算法;(2)分布式构造概念格。

量化规则格<sup>[7]</sup>是为了便于提取最小无冗余关联规则<sup>[8]</sup>的一种概念格的扩展结构。概念格的分布处理<sup>[9]</sup>,就是把形式背景进行拆分,分别构造相应的部分概念格,然后进行部分概

念格的合并,得到完整概念格,再在这个完整概念格上提取相应的关联规则,也就是说这种分布处理的思想只应用于概念格的构造,把构造格和提取关联规则分开处理。而本文提出一种基于量化概念格分布处理的规则合并算法,在部分概念格产生的过程中就产生部分关联规则,然后无需经过完整概念格的构造,直接对部分关联规则进行合并,从而得到最终的关联规则。这实际上是把概念格分布处理进行了延伸,实现了关联规则的分布获取。由于无需构造完整格,并且直接再利用已有的部分关联规则,从而降低了利用概念格提取关联规则的时空复杂度,提高了提取关联规则的效率。

## 2 关联规则与量化规则格的基本概念

### 2.1 关联规则

关联规则问题可描述如下:令  $I = \{i_1, i_2, \dots, i_m\}$  是  $m$  个不同项目(item)的集合,  $D$  是给定的针对  $I$  的交易数据集,它包含  $N$  项交易,每一项交易  $T$  是  $I$  中的一组项目集,  $T \subseteq I$ , 每一项交易都有标示符 TID。

在整个事务数据库  $D$  中出现项集  $X$  的比例(频率)称为项集  $X$  的支持度,记为  $\text{sup}(X)$ 。项集  $X$  是频繁的,是指其支持度达到用户设定的最小阈值  $\text{minsup}$ , 即  $\text{sup}(X) \geq \text{minsup}$ 。

对于一个形如  $X \Rightarrow Y$  的关联规则  $R$  来说,  $X, Y \in I$  且  $X \cap Y = \emptyset$ 。规则  $R$  的支持度  $\text{sup}(R)$  定义为  $\text{sup}(X \cup Y)$ , 规则  $R$  的置信度(可信度)记为  $\text{conf}(R)$ , 可表示为  $\text{sup}(X \cup Y) /$

<sup>\*</sup>国家自然科学基金(60575035);江苏省自然科学基金(BK2004052)。李 云 博士,副教授,研究兴趣为概念格、数据挖掘等;蔡俊杰 硕士研究生,研究兴趣为概念格、数据挖掘、并行计算等;刘宗田 教授,博士生导师,主要研究领域为软件工程、人工智能;陈 峻 教授,博士生导师,主要研究领域为并行计算、系统优化等。

sup(X)。关联规则根据置信度的不同分为精确关联规则和近似关联规则。精确关联规则的置信度为 100%；近似关联规则的置信度小于 100%。在提取的关联规则中，往往存在着大量的冗余规则，其中一定存在着最小的无冗余规则集合。

定义<sup>[6]</sup> 对于关联规则  $R: X \Rightarrow Y$ ，如果不存在关联规则  $R': X' \Rightarrow Y'$ ，满足  $\text{sup}(R) = \text{sup}(R')$ 、 $\text{conf}(R) = \text{conf}(R')$ ，且  $X' \subseteq X, Y \subseteq Y'$ ，那么  $X \Rightarrow Y$  为最小无冗余的关联规则。

### 2.2 量化规则格

在频繁项集中存在一些与交易集 TID 相同的项集，这些项集的集合称为同交易集的频繁项集集合 (the Set of Frequent Itemset with the Same Tidset, 记为 SFIST)，并且每个同交易集的频繁项集中最大的项集对应于概念格中的一个项集概念。

SFIST 中的最小项集集合称为 SLIT (the Set of the Least Itemsets)，每个格节点都对应一个最小项集集合 SLIT。因此，提取最小无冗余规则的关键就是获取节点所对应的最小项集集合 SLIT。为了便于从格中提取最小无冗余关联规则，可以把格节点所对应的 SFIST 中的最小项集 SLIT 与格一起生成，同时格节点的对象集用其势来表示，所形成的概念格称为量化规则格 (Quantitative Rule Lattice)，简记为 QRL<sup>[7]</sup>。

## 3 基于量化规则格分布处理的关联规则合并

### 3.1 量化规则格提取关联规则的方案描述

量化规则格是一种基于概念格的扩展结构，在节点中集成了 SLIT，利用它能快速地提取最小无冗余关联规则。由于在实际应用中，形式背景有规模大和动态变化的特性，因此，对整个形式背景进行量化规则格的构建其效率不是太高，所以我们提出了部分量化规则格的合并技术来提取关联的思想，如图 1 所示。

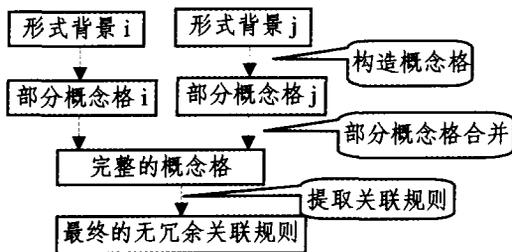


图 1 利用概念格合并提取规则

由部分量化规则格分布处理得到完整的概念格，虽然可以降低构造概念格的复杂度，但最终的规则还是在完整格上提取。考虑到构造完整的量化概念格的性能还是依赖于部分概念格的规模，当一个部分概念格很大或者有很多个部分概念格时，构造完整规则格还是需要很长时间的。因此，本文提出在部分量化规则格的基础上实现部分关联规则的直接合并，省去了构造完整格的时间，其示意图如图 2 所示。

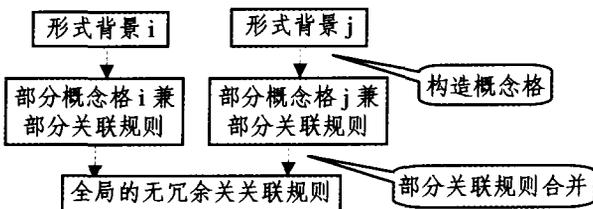


图 2 利用部分关联规则合并提取规则

### 3.2 利用部分概念格的规则合并获取关联规则

通过对形式背景进行拆分，分别构造相应的部分概念格，再从部分概念格中分别提取出部分关联规则，然后直接进行部分关联规则的运算和处理，得到全局关联规则，也是获取知

识的一个更有效、更直接的手段。

#### 3.2.1 候选关联规则的核查与合并

在上述的利用概念格的规则集成获取关联规则的方案中，其核心就是如何对部分概念格所提取的候选关联规则的核查和合并，得到最后的有效关联规则。

对于一个规则来说，其有效性主要是核查它的支持度和置信度。对于从概念格的概念节点对  $(C_1, C_2)$  中提取的规则  $A \Rightarrow B$  来说，其支持度和置信度可定义为

$$\text{sup}(A \Rightarrow B) = |\text{Extent}(C_2)| / |O|$$

$$\text{conf}(A \Rightarrow B) = |\text{Extent}(C_2)| / |\text{Extent}(C_1)|$$

衡量一个候选规则 (Candidate Rule) 是否能够成为最终的有效规则 (Final Rule)，就是核查它的最终支持度 (Final Support) 和最终置信度 (Final Confidence) 是否满足最小支持度  $\text{minsup}$  和最小置信度  $\text{minconf}$  的要求。

对于存在  $n$  个机器节点的系统，一个候选规则 CR 在各机器节点中的支持度和置信度表示为  $\text{sup}(0), \text{sup}(1), \dots, \text{sup}(n-1)$  和  $\text{conf}(0), \text{conf}(1), \dots, \text{conf}(n-1)$ 。各个部分数据库中交易记录的数量表示为  $\text{extcount}(0), \text{extcount}(1), \dots, \text{extcount}(n-1)$ ；我们检索每个部分概念格的概念节点，包含候选规则 CR 前件内涵所对应的概念节点的最大外延数表示为  $\text{antext}(0), \text{antext}(1), \dots, \text{antext}(n-1)$ 。那么，规则所对应的最终支持度和置信度分别定义为

$$\text{final sup}(CR) = \left( \sum_{i=0}^{n-1} \text{sup}(i) \times \text{extcount}(i) \right) / \left( \sum_{i=0}^{n-1} \text{extcount}(i) \right) \quad (i)$$

$$\text{final conf}(CR) = \left( \sum_{i=0}^{n-1} \text{conf}(i) \times \text{antext}(i) \right) / \left( \sum_{i=0}^{n-1} \text{antext}(i) \right)$$

现在的问题就是如何计算一个机器节点的部分候选规则在另一个机器节点所对应的部分概念格下的支持度和置信度。设在机器节点  $m (m=0, 1, \dots, n-1)$  中有一个候选的规则  $CR: A \Rightarrow B$ ，现在要计算该规则在机器节点  $m'$  中的支持度  $\text{sup}'$  和置信度  $\text{conf}'$ 。设在机器节点  $m'$  中交易记录的数量或对象/外延数为  $\text{extcount}$ ，搜索机器节点  $m'$  中所对应的部分概念格  $L(m')$ ，找到概念的内涵包含  $A \cup B$  的最大概念节点，该节点的外延数目 (外延的势) 用  $\text{antext}$  表示，那么支持度  $\text{sup}' = \text{antext} / \text{extcount}$ ；同理，在部分概念格  $L(m')$  中，搜寻概念的内涵包含  $A$  的最大概念节点，该节点的外延数目 (外延的势) 用  $\text{antextcount}$  表示，那么置信度  $\text{conf}' = \text{antext} / \text{antextcount}$ 。

规则合并利用了一个这样的朴素思想：对于对象域数据库的拆分来说，如果某最终规则是有效的，那么它一定在至少一个部分数据库中也是有效的候选规则；如果某个规则在所有部分数据库中无效，那么在整个数据库中也是无效的。

定理 1 对于某事务数据库，如果存在某最终规则  $R: A \Rightarrow B$  是有效的，即， $\text{final sup}(R) \geq \text{minsup}$ ， $\text{final conf}(R) \geq \text{min conf}$ ，那么至少在某个部分数据库中一定存在一个有效的候选规则  $CR: A \Rightarrow B$ 。

证明：设完整数据库为 context，它至少分为  $\text{context}(m)$  和  $\text{context}(m')$  两个部分事务数据库，那么完整事务数据库的事务记录数  $\text{extcount}(\text{context}) = \text{extcount}(\text{context}(m)) + \text{extcount}(\text{context}(m'))$ 。规则  $R: A \Rightarrow B$  在完整数据库中是有效的，那么  $\text{sup}(R) \geq \text{min sup}$ ， $\text{conf}(R) \geq \text{min conf}$ ，这时在部分事务数据库  $\text{context}(m)$  和  $\text{context}(m')$  中必然都存在  $CR: A \Rightarrow B$ ，设其支持度和置信度分别为  $\text{sup}(R_m)$ 、 $\text{sup}(R_{m'})$  和  $\text{conf}(R_m)$ 、 $\text{conf}(R_{m'})$ 。

设  $\text{sup}(R_m) = \text{minsup} + \text{diff\_sup}(R_m)$ ， $\text{sup}(R_{m'}) = \text{minsup} + \text{diff\_sup}(R_{m'})$ ，根据支持度的定义，规则  $R: A \Rightarrow B$  在

完整数据库中的支持度为

$$\begin{aligned} \text{Sup}(R) &= (\text{sup}(R_m) * \text{extcount}(\text{context}(m)) + \text{sup}(R_m') * \text{extcount}(\text{context}(m'))) / \text{extcount}(\text{context}) \\ &= ((\text{minsup} + \text{diff\_sup}(R_m)) * \text{extcount}(\text{context}(m)) + (\text{minsup} + \text{diff\_sup}(R_m')) * \text{extcount}(\text{context}(m'))) / \text{extcount}(\text{context}) \\ &= \text{minsup} + ((\text{diff\_sup}(R_m)) * \text{extcount}(\text{context}(m)) + (\text{diff\_sup}(R_m')) * \text{extcount}(\text{context}(m'))) / \text{extcount}(\text{context}) \end{aligned}$$

这样, Sup(R) 和 minsup 的大小关系, 就由 diff\_all = (diff\_sup(R\_m)) \* extcount(context(m)) + (diff\_sup(R\_m')) \* extcount(context(m')) / extcount(context) 是否大于 0 的状态决定:

① diff\_sup(R\_m) ≥ 0, diff\_sup(R\_m') ≥ 0, 即 sup(R\_m) ≥ minsup, sup(R\_m') ≥ minsup, 那么 diff\_all ≥ 0, 即 sup(R) ≥ minsup;

② diff\_sup(R\_m) ≥ 0, diff\_sup(R\_m') < 0, 即 sup(R\_m) ≥ minsup, sup(R\_m') < minsup, 若 |diff\_sup(R\_m) \* extcount(context(m))| ≥ |(diff\_sup(R\_m')) \* extcount(context(m'))|, 那么 diff\_all ≥ 0, 即 sup(R) ≥ minsup;

③ diff\_sup(R\_m) < 0, diff\_sup(R\_m') ≥ 0, 即 sup(R\_m) < minsup, sup(R\_m') ≥ minsup, 若 |(diff\_sup(R\_m')) \* extcount(context(m'))| ≥ |diff\_sup(R\_m) \* extcount(context(m))|, 那么 diff\_all > 0, 即 sup(R) ≥ minsup;

④ diff\_sup(R\_m) < 0, diff\_sup(R\_m') < 0, 即 sup(R\_m) < minsup, sup(R\_m') < minsup, 那么 diff\_all < 0, 即 sup(R) < minsup.

同理, 可证明规则的置信度 conf(R) 和 conf(R\_m)、conf(R\_m') 也有上述类似的结论。

对于完整的数据库分为多于两个的部分数据库, 其情形也是类似的。

第(1)(2)(3)种情况说明规则 R: A => B 在整个数据库中是有效的, 至少应在一个部分数据库中有效; 第(4)种情形说明若规则 R: A => B 在所有部分数据库中都无效, 那么它在完整数据库中也一定是无效的。得证。

这样, 通过核查单个机器节点中的候选规则在整个分布系统中的有效性, 就能知道哪些候选规则能最终成为全局的有效规则, 即最终有效规则。

### 3.2.2 候选关联规则的有效性算法

下面给出核查单个机器节点中的候选规则在整个分布系统中的有效性算法。把所有机器节点中有效的候选规则合并, 就得到全局的最终有效的关联规则。

Algorithm: 核查单个机器节点中的候选规则在整个分布系统中的有效性

Input: 整个系统的全部节点 ALLNODE, 单个节点 THISNODE, minsup, minconf;

Output: 单个节点中的候选规则将成为最终有效规则的规则集 RULES

Begin

FOR 单个 THISNODE 中的每个候选规则 RULE DO

Finalsup := RULE.support;

Finalconf := RULE.confidence;

计算 THISNODE 的每个概念格中的 antext 和 extcount;

FOR 在 ALLNODE 中其他节点 OTHERNODE, OTHERNODE ≠ THISNODE DO

计算 RULE 在 ORTHERNODE 中的支持度 Tempsup 和置信度 Tempconf;

计算 ORTHERNODE 的概念格中的 antext' 和 extcount';

Finalsup = (Finalsup \* extcount + tempsup \* extcount') / (extcount + extcount');

extcount = extcount + extcount';

Finalconf = (Finalconf \* antext + Tempconf \* antext') / (antext + antext');

antext = antext + antext';

ENDFOR

IF (Finalconf ≥ minsup) 且 (Finalconf ≥ minconf) THEN

RULES = RULES ∪ {RULE};

ENDIF

ENDFOR

END

## 4 简单实例描述和实验验证

设系统存在 2 个机器节点 Node(i)、Node(j), 分别处理 2 个部分数据库 Context(i)、Context(j), 假设用户设定的规则的最小支持度和置信度为 minsup=0.2, minconf=0.8。部分数据库 Context(i)、Context(j) 分别如表 1 和表 2 所示。

表 1 部分数据库 Context(i)

tid	a	b	c	d
1	1	0	0	1
2	1	1	0	1
3	0	0	1	0
4	1	0	0	1
5	0	0	0	0
6	0	1	1	0

表 2 部分数据库 Context(j)

tid	a	b	c	d
1	1	0	0	1
2	0	0	0	0
3	0	1	1	0
4	0	0	1	1

对于上述 2 个部分数据库分别构造概念格。这里, 假定提取最小无冗余规则, 那么构造的概念格为部分量化规则格。2 个部分数据库 Context(i)、Context(j) 所对应的量化规则格如图 3 和图 4 所示。在图中, 每个概念节点只表示了 3 个信息: 交易集(对象或外延)的势、最小项集 SLIT、封闭项集(内涵)。

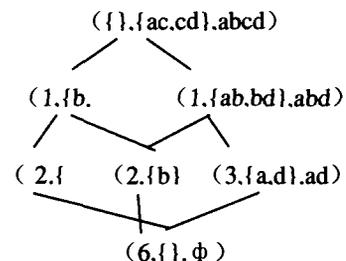


图 3 量化规则格 Lattice(i)

(下转封四)

(上接第 196 页)

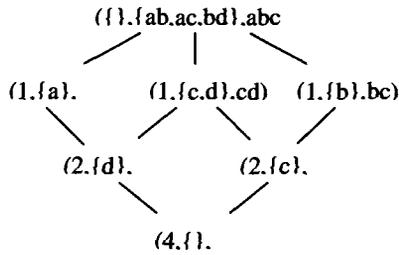


图 4 量化规则格 Lattice(j)

利用提取最小无冗余规则的算法<sup>[10]</sup>,在两个部分量化规则格中很容易提取有效的候选规则集:

Candidate\_rules(i):

$d \Rightarrow a$ , support =  $3/6 = 0.5$ , confidence =  $3/3 = 1$

$a \Rightarrow d$ , support =  $3/6 = 0.5$ , confidence =  $3/3 = 1$

Candidate\_rules(j):

$a \Rightarrow d$ , support =  $1/4 = 0.25$ , confidence =  $1/1 = 1$

$b \Rightarrow c$ , support =  $1/4 = 0.25$ , confidence =  $1/1 = 1$

$c \Rightarrow d$ , support =  $1/4 = 0.25$ , confidence =  $1/1 = 1$

$d \Rightarrow c$ , support =  $1/4 = 0.25$ , confidence =  $1/1 = 1$

应用上述的核查单个机器节点中的候选规则在整个系统中的有效性算法,可以得到满足设定的最小支持度和置信度要求的最终有效规则集为

Final\_rules:

$a \Rightarrow d$ , support =  $((3/6) * 6 + (1/4) * 4) / (6+4) = 4/10 = 0.4$  confidence =  $((3/3) * 3 + (1/1) * 1) / (3+1) = 4/4 = 1$

$d \Rightarrow a$ , support =  $((3/6) * 6 + (1/4) * 4) / (6+4) = 4/10 = 0.4$  confidence =  $((3/3) * 3 + (1/2) * 2) / (3+2) = 4/5 = 0.80$

$b \Rightarrow c$  support =  $((1/4) * 4 + (1/6) * 6) / (6+4) = 2/10 = 0.2$  confidence =  $((1/1) * 1 + (1/2) * 2) / (1+2) = 0.67$

把两个部分数据库 Context(i)、Context(j)合并为一个交易集数目为 10 的完整数据库,然后利用其量化规则格,提取满足 minsup=0.2, minconf=0.8 的规则,可以发现,其结果和上面得到的最终有效规则集是一致的。

我们利用扬州大学高性能计算中心的深腾 1800 集群系统,使用 MPIJAVA 实现了本文的算法,并针对 mushroom 数据集进行了实验。该数据库拥有 8124 条事务集、22 个属性,采用均分原则对形式背景进行拆分,然后分派到 5 至 30 个节点上分别进行实验。对本文提出的图 1 和图 2 所示的方案进行了实验比较,设置用户阈值 minsup=0.4, minconf=0.8,产生相同无冗余规则的时间如图 5 所示,实线代表采用图 1 方案,虚线代表采用图 2 方案。可以看出,采用部分规则的直接合并获取关联规则在时间上具有明显的优势。

结语 本文提出了分布式挖掘关联规则的思想,主要

是从量化规则格出发,把合并格后所要提取规则的处理过程分布到每个挖掘部分规则格的子过程中去。由于无需构造完整格,并且直接再利用已有的部分关联规则,从而降低了利用概念格提取关联规则的时空复杂度,比一般的基于概念格提取规则要快得多。在今后的工作中,我们将致力于对形式背景纵向合并时的规则分布提取,以及在形式背景中删除记录时的分布规则提取研究,最后能满足数据库的各种动态变化,形成一种高效的关联规则分布挖掘算法。

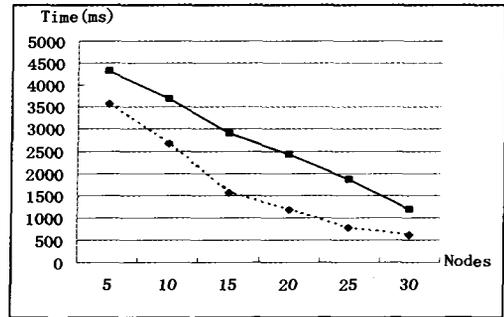


图 5 两种方法的时间图

### 参考文献

- 1 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large database. In: Proceedings of the ACM SIGMOD Conference on Management of Data. New York, NY: ACM press, 1993. 207~216
- 2 Ganter B, Wille R. Formal Concept Analysis: Mathematical Foundations. Berlin: Springer-Verlag, 1999
- 3 Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival I, ed. Ordered Sets, Reidel, Dordrecht, 1982. 445~470
- 4 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
- 5 Pasquier N, Bastide Y, et al. Efficient Mining of Association Rules Using Closed Itemset Lattices. Information Systems, 1999, 24(1): 25~46
- 6 LI Yun, LIU Zongtian, et al. Extracting Minimal Non-Redundant Implication Rules by Using Quantized Closed Itemset Lattice. In: Proceedings of the 16<sup>th</sup> International Conf. on Soft Engineering and Knowledge Engineering (SEKE'04), Banff, Canada, June 2004. 402~405
- 7 李云, 蔡俊杰, 刘宗田, 等. 量化规则格及其渐增式构造. 模式识别与人工智能(已录用)
- 8 Bastide Y, Pasquier N, et al. Mining Minimal Non-redundant Association Rules Using Frequent Closed Itemsets. Lecture Notes in Computer Science, 2000, 1861: 972~986
- 9 李云, 刘宗田, 等. 概念格的分布处理研究. 小型微型计算机系统, 2005, 26(3): 448~451
- 10 LI Yun, LIU Zongtian, et al. Extracting Minimal Non-Redundant Associate Rules from QCIL. In: Proceedings of the 4th International Conf. on Computer and Information Technology(CIT'04), Wuhan, China, September 2004. 986~991

# 计算机科学

(1974年1月创刊)

第34卷第06期(月刊)

2007年6月25日出版

国际标准连续出版物号 ISSN 1002-137X  
国内统一连续出版物号 CN50-1075/TP

定价: 30.00元 国外定价: 5美元

邮发代号: 78-68

发行范围: 国内外公开

主管单位: 国家科学技术部

主办单位: 国家科技部西南信息中心

编辑出版: 《计算机科学》杂志社

重庆市北部新区洪湖西路18号 邮政编码: 401121

电话: (023) 63500828 E-mail: jsjxk@swic.ac.cn

网址: www.jsjx.com

社长: 牟炳林

总编: 彭丹

主编: 朱宗元

主编助理: 徐书令

印刷者: 重庆科情印务有限公司

总发行处: 重庆市邮政局

订购处: 全国各地邮政局

国外总发行: 中国国际图书贸易总公司(北京399信箱)

国外代号: 6210-MO

