第35卷 第22期 计 算 机 工 程 2009年11月
Vol.35 No.22 Computer Engineering November 2009
·软件技术与数据库· 文章编号：1000—3428(2009)22—0074—03 文献标识码：A 中图分类号：TP18

概念格上无冗余关联规则的提取算法 NARG

苗 茹，沈夏炯，胡小华
(河南大学计算机与信息工程学院，开封 475004)

摘 要：在数据挖掘中，关联规则是很有价值的一类规律。普通的挖掘算法会产生大量的规则，尤其是当最小支持度和最小可信度减少时，关联规则的数目急剧上升。如何对规则进行约减而又不丢失数据信息是消除冗余关联规则的关键。根据概念格的理论和冗余关联规则的性质，提出在概念格上提取无冗余关联规则的NARG算法。该算法可以得到最小的无冗余的关联规则集，而且不丢失任何信息，可有效提高关联规则生成的效率。

关键词：形式概念分析；概念格；关联规则挖掘；最小无冗余规则

NARG Algorithm of Extracting Non-redundant Association Rule in Concept Lattice

MIAO Ru, SHEN Xia-jiong, HU Xiao-hua
(School of Computer and Information Engineering, Henan University, Kaifeng 475004)

【Abstract】 Association rules are the very valuable kind of law in data mining. A large number of rules are usually generated from database using ordinary mining algorithms. Especially when the minimal support and minimal confidence are reduced, the number of association rules rise rapidly. The key of eliminating redundant association rules is to reduce rules without losing data information. This paper presents a new algorithm called NARG to extract non-redundant association rules based on concept lattice and properties of redundant association rules. This algorithm can gain the minimal non-redundant set of association rules while effectively improve efficiency of extracting rules without losing any information of data.

【Key words】 formal concept analysis; concept lattice; association rules mining; minimal non-redundant rule

1 概述

在数据库中，关联规则是很有价值的一类规律，文献[1]提出从大型数据库发现关联规则的Apriori方法。而在关联规则的挖掘过程中，通常产生的关联规则的数量很多，这使用户进行分析和利用这些规则变得十分困难，尤其是当最小支持度和最小可信度减少时，关联规则的数目急剧上升。目前，常见的关联规则挖掘算法大多是在该算法的基础上加以改进或扩展的，不能有效减少冗余关联规则的数量。

文献[2]提出由二元关系来建造相应概念格的基本思想。概念格节点反映了概念内涵和外延的统一，因此非常适合作为规则发现的基础性数据结构，利用概念格能够有效地描述关联、蕴涵等规则的提取。本文应用概念格的结构和其中概念的闭包性质，设计了从概念格上提取无冗余关联规则的NARG(Non-redundant Association Rule from Galois)算法。NARG算法得到的是全部关联规则的一个最小完备子集，即其中的关联规则具有最小前件和最大后件的性质，并且其他关联规则均可从该子集中推出，可覆盖数据库的全部信息。

2 相关概念

2.1 概念格

概念格从形式背景(Formal Context)计算而来。一个形式背景是一个三元组 $K=(G, M, I)$，其中 G 是对象的集合，M 是属性的集合，$I \in G \times M$。

对于集合 $X \subseteq G$ 和 $Y \subseteq M$，定义：

$X'=f(X)=\{m \in M | \forall g \in X, (g, m) \in I\}$

$Y'=g(Y)=\{g \in G | \forall m \in Y, (g, m) \in I\}$

一个形式概念是一个二元组 $C=(X, Y)$，其中，$X \subseteq G$, $Y \subseteq M$ 并满足：$X'=Y$ 和 $Y'=X$，集合 X 称为概念的外延，集合 Y 称为概念的内涵。若概念 $C_1=(X_1, Y_1)$和 $C_2=(X_2, Y_2)$，满足 $X_1 \subseteq X_2$ (或 $Y_2 \subseteq Y_1$)，则称(X_1, Y_1)为亚概念，(X_2, Y_2)为超概念，记为：$(X_1, Y_1) \leq (X_2, Y_2)$。这种由形式背景中所有形式概念的亚概念-超概念的偏序关系(也称泛化-特化关系)所诱导出的格称为概念格，记为 $L(K)$。一个数据上下文如表1所示。

表1 一个数据上下文

tid	items				
1	A	B		D	E
2		B	C		E
3	A	B		D	E
4	A	B	C		E
5	A	B	C	D	E
6		B	C	D	

2.2 概念格节点的内涵缩减

对于概念格中的一个概念(X, Y)，它的内涵 Y 是最大化的，也就是说 Y 的任意扩大将导致 $g(Y)$的减小，与此相反 Y 的减小未必会导致 $g(Y)$的增大。由此，可以得到内涵缩减的定义[3]。

定义 1 对于一个给定的概念 $C=(X_1, Y_1)$如果特征集合 $Y_2 \subseteq Y_1$ 满足如下2个条件：

(1)$g(Y_2)=g(Y_1)=X_1$；

作者简介：苗 茹(1979－)，女，讲师、硕士，主研方向：概念格，数据挖掘；沈夏炯、胡小华，教授、博士

收稿日期：2009-03-20 **E-mail：**mr1015@henu.edu.cn

(2)对于任意的 $Y_3 \subset Y_2$ 有 $g(Y_3) \supset g(Y_2)=X_1$；

则特征集合 Y_2 被称为是 C 的一个内涵缩减。

定义 2 概念格中的节点 C 的所有内涵缩减的集合称为 C 的内涵缩减，记为 $INT\_RED(C)$。

例如，给定一个数据上下文，如表 1，若概念 $C=(135, ABDE)$，则 $INT\_RED(C)=\{AD, DE\}$。

概念的内涵缩减体现了特征集的最小化，可作为规则的最小前件。

2.3 关联规则与概念格

关联规则是由文献[4]提出的一个重要 KDD 研究课题，它反映了大量数据中项目集之间有趣的关联或相关联系。

设 $I=\{i_1, i_2, \cdots, i_m\}$ 为一组 m 个不同的特征，一条关联规则可以表示为 $A \Rightarrow B$，其中 $A, B \subset I$ 并且 $A \cap B=\varnothing$。如果事务数据库 TD 中含有该项集的事务比例为 sup，称为该项集的支持度；如果该项集的支持度不小于用户确定的最小支持度 min_sup，则该项集是频繁项集；如果包含 A 也包含 B 的事务占包含 A 的事务的比例为 $conf$，则 $conf$ 是该关联规则的可信度；同样要求关联规则的可信度大于规定的最小可信度 min_conf。

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)。对于关联规则 $A \Rightarrow B$，它唯一地对应于概念格中的节点二元组 (C_1, C_2)，其中，$C_1=(g(A), f(g(A)))$，$C_2=(g(A \cup B), f(g(A \cup B)))$。

3 无冗余关联规则的提取

目前常见的关联规则挖掘算法产生的规则数量太多，用户很难有效地分析和利用。而挖掘出的规则中又存在大量的冗余规则，利用概念格节点性质能够有效地提取出无冗余的关联规则的集合。

3.1 无冗余关联规则

定义 3 对于规则 $A \Rightarrow B$，若 $|g(A \cup B)|/|g(A)|=1$，则称 $A \Rightarrow B$ 为精确关联规则(Exact Rule, ER)；若 $|g(A \cup B)| \neq |g(A)|$ 且 $|g(A \cup B)|/|g(A)| \geq \min\_conf$，则称 $A \Rightarrow B$ 为近似关联规则(Approximate Rule, AR)。

定义 4 对于规则 $A \Rightarrow B$ 和 $C \Rightarrow D$，当且仅当使用某种推理，可由规则 $A \Rightarrow B$ 推出规则 $C \Rightarrow D$ 成立，则称规则 $C \Rightarrow D$ 是 $A \Rightarrow B$ 的冗余规则。

性质 1 如果 A，B，C 是两两不相交的项集，$A \Rightarrow B \cup C$ 是关联规则，则 $A \cup C \Rightarrow B$ 必然也是关联规则。也就是说，由 $A \Rightarrow B \cup C$ 可以推出 $A \cup C \Rightarrow B$。

性质 2 如果 $C \subset B$ 则由 $A \Rightarrow B$ 可以推出 $A \Rightarrow C$。

性质 3 若关联规则 $A \Rightarrow B$ 相对 $X \Rightarrow Y$ 是冗余的，则冗余规则总数为 $(3^{|Y|}-2^{|Y|}-1)$[5]。

定义 5 如果关联规则 $A \Rightarrow B$ 满足：

(1)不存在另外一条规则 $A' \Rightarrow B'$，使通过性质 1 和性质 2 可推出规则 $A \Rightarrow B$

(2)若去掉规则 $A \Rightarrow B$ 则会丢失数据信息，则称规则 $A \Rightarrow B$ 为无冗余关联规则。

3.2 无冗余关联规则提取算法 NARG

定义 6 设概念 $C_1=(O_1, D_1)$, $C_2=(O_2, D_2)$并且 $C_1 \prec C_2$，则 (C_1, C_2)生成的精确关联规则基为 $NER_{(C1, C2)}=\{D \overset{p=1}{\Rightarrow} D_2-D | D \in INT\_RED(C_1)\}$。

定理 1 $NER_{(C1, C2)}$中的关联规则是精确关联规则。

证明：设 $C_1=(O_1, D_1)$为频繁概念，则 C_1 的直接父节点 $C_2=(O_2, D_2)$必定也为频繁概念，即有 $support(D_1)=|O_1|/|U| \geq \min\_sup$, $support(D_2)=|O_2|/|U| > \min\_sup$。令 $r: D \Rightarrow D_2-D$，其中，$r \in NER_{(C1, C2)}$，则 $D \in INT\_RED(C_1)$。因为 D 是 C_1 的内涵缩减，所以 $g(D)=g(D_1)=O_1$，$support(D)=|O_1|/|U| \geq \min\_sup$。因为 $C_1 \prec C_2$，则 $g(D \cup D_2)=O_1$，所以 $conf(r)=|g(D_2 \cup D)|/|g(D)|=|O_1|/|O_1|=1$。

定理 2 $NER_{(C1, C2)}$是(C_1, C_2)生成的最小精确关联规则集，其他精确关联规则均可从中得出。

证明略。

例如，给出一个频繁概念 $C_1=(135, ABDE)$，$C_2=(1345, ABE)$，且有 $C_1 \prec C_2$，$INT\_RED(C_1)=\{AD, DE\}$，则 $NER_{(C1, C2)}=\{AD \Rightarrow BE, DE \Rightarrow AB\}$。

定义 7 所有频繁概念的精确关联规则基所组成的集合是整个数据库的精确关联规则基，记为 NER。

算法 1 构造最小精确关联规则基 NER 的算法 GE

输入 频繁概念集合 $Candnode$

输出 最小精确关联规则基 NER

```
BEGIN
NER = {∅};
FOR Candnode 中的每个频繁节点 C DO
FOR each Cp∈Dirt_parent(C) DO
NER = NER ∪ {C.int_red => C.intent - C.int_red};
END FOR
END FOR
return NER;
END
```

定义 8 概念 $C_1=(O_1, D_1)$, $C_2=(O_2, D_2)$并且 $C_1 \geq C_2$，则(C_1, C_2)生成的近似关联规则基为 $NAR_{(C1, C2)}=\{D \overset{p<1}{\Rightarrow} D_2-D | D \in INT_RED(C_1)$且不存在 $C_2' \leq C_2$，使规则 $D \overset{p<1}{\Rightarrow} D_2'-D$ 成立，其中，$p \geq \min\_conf\}$。

定理 3 $NAR_{(C1, C2)}$是(C_1, C_2)生成的最小近似关联规则集，其他近似关联规则均可从中得出。

证明与定理 1 类似，略。

定义 9 所有频繁概念的近似关联规则基所组成的集合是整个数据库的近似关联规则基，记为 NAR。

算法 2 构造最小近似关联规则基 NAR 的算法 GA

输入 频繁概念集合 $Candnode$，最小可信度 min_conf

输出 最小近似关联规则基 NAR

```
BEGIN
NAR = {∅};
FOR Candnode 中的每个频繁节点 C DO
numcon = C.ext_num * min_conf;
Pairs[C] = ∅;
queue = nil;
push(queue, C);
WHILE notempty(queue) DO
C' = pull(queue);
FOR each children cc of C' DO
IF cc.ext_num >= numcon THEN
push(queue, cc);
END IF
END FOR
Pairs[C] = Pairs[C] ∪ {C'};
```

```
END WHILE
END FOR
FOR Candnode 中的每个频繁节点 C DO
在 Pair[C]中消去具有父子关系的父节点;
FOR each CP in Pair[C] DO
NAR = NAR ∪ {C.int_red => CP.intent - C.int_red};
END FOR
END FOR
return NAR;
END
```

算法 3 从建好的关联规则概念格中提取无冗余的关联规则集

输入 概念格 *L*，最小支持度 min_*sup*，最小可信度 min_*conf*

输出 无冗余关联规则集合 *RULES*

```
BEGIN
Candnode = {∅};
RULES = {∅};
numsup = min_sup * |U|;
FOR L 中每个节点 C 按|Intent(C)|升序 DO
IF C.ext_num≥numsup THEN Candnode=Candnode∪{C};
END FOR
RULES = GE(Candnode) ∪ GA(Candnode, min_conf);
END
```

3.3 算法实验与分析

对于一个给定的事务集，如表 1 所示，设 min_*sup*=50%，min_*conf*= 50%，NARG 算法得到的关联规则集合是最小的。表 2 和表 3 对比了 NARG 算法求出的精确关联规则基和近似关联规则基相对于 Apriori 算法求出的精确关联规则和近似关联规则消除冗余的程度。

表 2 精确关联规则

序号	Apriori	NER	序号	Apriori	NER
1	C⇒B	C⇒B	10	AD⇒BE	AD⇒BE
2	D⇒B	D⇒B	11	AD⇒B	
3	E⇒B	E⇒B	12	AD⇒E	由 10 推出
4	CE⇒B	CE⇒B	13	ABD⇒E	
5	A⇒BE	A⇒BE	14	ADE⇒B	
6	A⇒E		15	DE⇒AB	DE⇒AB
7	A⇒B	由 5 推出	16	DE⇒A	
8	AB⇒E		17	DE⇒B	由 15 推出
9	AE⇒B		18	BDE⇒A	

从中可以看出，NARG 算法有效消除了 Apriori 算法所产生的大量冗余规则，大大减少了关联规则的数量。Apriori 算法共提取出了 60 条关联规则，而 NARG 算法仅生成 14 条关联规则；另外，NARG 算法得到的是最小的关联规则集合，即所生成的规则集不可再约减，否则将会丢失有用信息。

表 3 近似关联规则

序号	Apriori	NAR	序号	Apriori	NAR	序号	Apriori	NAR
1	C⇒BE	C⇒BE	15	AE⇒D		29	BD⇒AE	由 19 推出
2	C⇒E	由 2 推出	16	AB⇒DE	由 10 推出	30	BE⇒AD	
3	BC⇒E		17	AE⇒BD		31	D⇒ABE	D⇒ABE
4	B⇒CE	B⇒CE	18	ABE⇒D		32	D⇒A	
5	B⇒C		19	B⇒ADE	B⇒ADE	33	D⇒E	
6	B⇒E	由 4 推出	20	B⇒A		34	D⇒AB	由 31 推出
7	BE⇒C		21	B⇒D		35	D⇒AE	
8	E⇒BC	E⇒BC	22	B⇒AD		36	D⇒BE	
9	E⇒C	由 8 推出	23	B⇒AE		37	E⇒ABD	E⇒ABD
10	A⇒BDE	A⇒BDE	24	B⇒DE	由 19 推出	38	E⇒A	
11	A⇒D		25	BD⇒A		39	E⇒D	
12	A⇒BD	由 10 推出	26	BE⇒A		40	E⇒AB	由 37 推出
13	A⇒DE		27	BD⇒E		41	E⇒AD	
14	AB⇒D		28	BE⇒D		42	E⇒BD	

4 结束语

针对关联规则挖掘经常产生大量规则的问题，本文提出在概念格上直接提取无冗余的关联规则的 NARG 算法。该算法能从建好的概念格中直接提取出关联规则集，所得到的集合能够涵盖所有的规则，并且不可再约减。算法实现简单、准确，可有效提高关联规则生成的效率。

参考文献

[1] Agrawal R, Imielinski T, Swami A. Mining Association Rules Between Sets of Items in Large Databases[C]//Proc. of the ACM SIGMOD International Conference on Management of Data. Washington D. C., USA: [s. n.], 1993.

[2] Ganter B, Wille R. Formal Concept Analysis: Mathematical Foundations[M]. Berlin, Germany: Springer Verlag, 1999.

[3] 谢志鹏, 刘宗田. 概念格与关联规则发现[J]. 计算机研究与发展, 2000, 37(12): 1415-1421.

[4] Han Jiawei, Kamber M. 数据挖掘: 概念与技术[M]. 范 明, 孟小峰, 译. 北京: 机械工业出版社, 2001.

[5] 韦素云, 吉根林, 区维光. 关联规则的冗余删除与聚类[J]. 小型微型计算机系统, 2006, 27(1): 110-113.

编辑 金胡考

(上接第 73 页)

5 结束语

通过大量测试表明，本文提出的快照具有较好的性能。其最大的特点是把通常在文件层实现的写前拷贝和重定向写技术引进到块设备层，大大增加了系统无关性，应用范围拓广，也提高了效率。此外，元数据和数据的存放和查找方案经过了仔细的研究，与其他方案对比，该方案结构清晰，效率高。下一步将针对如何把此快照融入新型系统结构中并发挥最大效能作相关研究。

参考文献

[1] 李怀阳. 进化存储系统数据组织模式研究[D]. 武汉: 华中科技大学, 2006.

[2] Acharya A, Uysa M. Active Disk[R]. Santa Barbara, CA, USA: University of California, Santa Barbara, Tech. Rep.: TRCS98-06, 1998.

[3] Lienhart R, Pfeiffer S, Effelsberg W. Video Abstraction[J]. Communications of the ACM, 1997, 40(12): 55-62.

[4] 韩德志, 谢长生, 李怀阳. 存储备份技术探析[J]. 计算机应用研究, 2004, 21(6): 1-4.

[5] Daniel P. Bovet, Cesati M. 深入理解 Linux 内核[M]. 陈莉君, 张琼声, 张宏伟, 译. 北京: 中国电力出版社, 2007.

编辑 金胡考