基于概念格的分类和关联规则的集成挖掘方法*

胡可云, 陆玉昌, 石纯一

(清华大学 计算机科学与技术系,北京 100084)

E-mail: hky@s1000e.cs.tsinghua.edu.cn; {lyc,scy}@tsinghua.edu.cn

http://www.tsinghua.edu.cn

摘要: 改进了一个 Bordat 的建格算法,使之适合于集成挖掘的需要,进而提出一个从概念格上提取关联规则和分类规则的算法,实现了关联规则和分类规则的挖掘在概念格框架下的统一.实验证明了算法的有效性.

关键词: 分类;关联规则;概念格;集成挖掘

中图法分类号: TP181 **文献标识码**: A

概念格,也称为 Galois 格,首先由 Wille R 于 1982 年提出[1].概念格的每个节点是一个形式概念,由两部分组成:外延,即概念所覆盖的实例;内涵,即概念的描述,亦即该概念覆盖实例的共同特征.概念格通过 Hasse 图生动而简洁地体现了这些概念之间的泛化和特化关系.因此,概念格被认为是进行数据分析的有力工具.目前,已有一些建造概念格的算法和概念格在信息检索、数字图书馆、软件工程和知识发现等方面的应用[2~5].

分类规则和关联规则挖掘是两个重要的数据挖掘任务.典型的分类规则挖掘方法有决策树、神经网络、AQ 以及粗糙集等.实验评估显示,基于概念格的系统具有和这些典型系统相媲美的分类效果[3,4,6~8].关联规则挖掘是近来数据挖掘的热门研究课题.有些作者说明了概念格是关联规则挖掘的一个自然平台,因为概念格的每个节点实际上就是一个最大项目集[9~11].分类规则和关联规则的集成挖掘可以给用户带来很大便利.虽然有人讨论过集成挖掘这个课题[3],但由于概念格体现了概念之间的关系,更易于理解,因此在概念格的框架下讨论这个问题是有意义的.

本文通过引进支持度的概念和使用计数信息,改进了一个 Bordat 的建格算法[3],使之适合于集成挖掘的需要,进而根据概念格的有关性质提出了一个从概念格上提取关联规则和分类规则的算法,实现了关联规则和分类规则的挖掘在概念格框架下的统一.实验证明了算法的有效性.

1 概念格的基本概念

本节简要介绍所需要的概念格的基本概念,关于概念格的比较详尽的形式化描述可参考文献[1,2].

假设给定形式背景(context)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.格 L 中的每个节点是一个序偶(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:

* 收稿日期: 1999-05-20; 修改日期: 1999-09-14

基金项目: 国家自然科学基金资助项目(79990580);国家重点基础研究发展规划资助项目(G1998030414);国家"九五"攀登计划预研资助项目

作者简介: 胡可云(1970),男,湖北人,博士生,主要研究领域为数据挖掘,粗糙集理论,概念格;陆玉昌(1935-),男,辽宁人,教授,主要研究领域为机器学习,知识发现与数据挖掘,知识工程;石纯一(1935-),男,河北人,教授,博士生导师,主要研究领域为人工智能.

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

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

在概念格节点间能够建立起一种偏序关系.给定 $H_1=(Y_1,X_1)$和 $H_2=(Y_2,X_2)$,则 $H_1<H_2\Leftrightarrow X_1\subset X_2$,领先次序意味着 H_1 是 H_2 的父节点或称直接泛化.实际上,格中集合 X_1 和 X_2 之间存在着对偶关系,即 $X_1\subset X_2\Leftrightarrow Y_2\subset Y_1$.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 $H_1<H_2$ 并且在格上不存在另一个元素 H_3 使得 $H_1<H_3<H_2$,则从 H_1 到 H_2 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 $O=\{1,2,3,4\}$,$D=\{a_1,a_2,a_3,b_1,b_2,b_3,c_1,c_2,d_1,d_2,d_3,d_4\}$,$R$ 描述了 O 中元素所拥有的 D 中的属性值集.图 1 是相应的概念格的 Hasse 图.

Table 1 Example of a context

表 1 形式背景例子

cf	A	B	C	D
1				
1	a_1	b_1	c_1	d_1
2	a_1	b_2	c_1	d_2
3	a_2	b_1	c_2	d_3
4	a_3	b_3	c_1	d_4

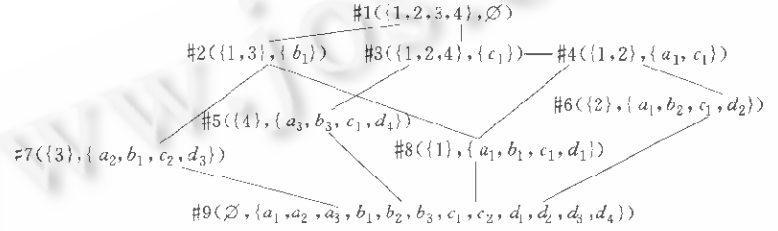

Fig. 1 Hasse diagram of concept lattice corresponding to table 1

图1 表1所对应的概念格的Hasse图

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.由于在本文算法中重要的是它的 Y 的基数.为了保证算法的效率,在算法中使用 Y 的基数代替 Y.基数可用于计数.这样,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 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[8].一种直截了当的寻找节点 $N=(C,X)$中蕴含规则的方法是找到所有形式的 $Q\Rightarrow S$,其中 $Q\subseteq P(X)$,$S=X-Q$,并且 Q 与 N 的父节点 N 是一致的,即不存在父节点 $P=(C',X')$使得 $Q\subseteq X'$.但是,以这种方法生成的蕴含规则是高度冗余的.

本文提出一种快速启发式算法,能从概念格中生成非冗余的蕴含和关联规则.

2 格的建造

为了削减格中节点的数目(在某些情况下会呈指数增长),有必要引进一个支持度 ε.只有在支持度之上的格节点才会生成,从而改进了 Bordat 的算法.

格 L 初始化为最顶端的节点$(O,\varnothing)$,并通过递归地构造其子节点来扩展.这种自顶而下的方式有利于修剪掉不满足支持度的节点.在算法中,使用了一个数组 PX,用来存放所有单个属性值对的第 1 次出现节点的指针.这个数组用来进行关联规则的挖掘.

算法 1. LCA(lattice construction algorithm)

1. $L\leftarrow(\|O\|,\varnothing)$,queue$\leftarrow(\|O\|,\varnothing)$,$A_\varepsilon\leftarrow${所有计数大于 ε 的属性值对},$PX\leftarrow\varnothing$
2. While stack not empty
3. 从 queue 头取出(C_k,X_k)
4. Candset$\leftarrow$FindSubNodes(C_k,X_k,A_ε)

```
5.    For 每个节点(C_ki,X_ki)∈candset
6.      if (C_ki,X_ki)∉L
7.        if C_ki/‖O‖>ε                    //只扩展满足支持度的节点
8.          L←L∪(C_ki,X_ki),更新 PX
9.          加边(C_k,X_k)→(C_ki,X_ki)
10.         将(C_ki,X_ki)推入 queue 尾部
11.       endif
12.     else
13.       加边(C_k,X_k)→(C_ki,X_ki)
14.     endif
15.   endfor
16. endwhile
17.
18. FindSubNodes(C_k,X_k,A_t)
19.   Candattr=A_t−X_k,subnodes←∅
20.   for 在 Candattr 中的每个属性值对 a
21.     Max←true;maxitem={a}
22.     for 在 Candattr 中的每个属性值对 a'≠a
23.       if count(X_k+a)=count(X_k+a+a')        //使用计数代替集合操作
24.         Max←false;break;
25.       endif
26.       if count(X_k+a)=count(X_k+a')          //使用计数代替集合操作
27.         maxitem=maxitem∪{a'}
28.       endif
29.     endfor
30.     if Max
31.       subnodes=subnodes∪(count(X_k∪maxitem),X_k∪maxitem)
32.     end
33.   endfor
34.   return subnodes
```

算法 LCA 中使用计数信息来取代原来算法中的集合包含/相等运算.实验表明,当 ε 设为 0 的情况下,它比原算法平均快了 5 倍.此算法建造的概念格实质上是一种"频繁"格,只包括那些支持度大于 ε 的节点,从而降低了建造整个格的复杂度.

3 格上的规则提取

本节提出一个生成所有给定后件的非冗余规则的算法.当给定规则后件是决策属性的某个类别时,就得到了分类规则.因此,此算法可适用于生成分类及关联规则.首先找到包含该后件的最小概念,然后对该节点的子格进行宽度优先遍历,对每个节点生成所有非冗余规则.对每个节点首先生成蕴含规则(可信度为 1 的规则),然后生成低可信度规则.这是因为高可信度的规则在分类中所起作用较大.

生成蕴含规则的方式依赖于下列事实.

事实 1. 若节点 $H=(C,X)$ 仅有一个父节点 $P=(C',X')$,则

(1) H 所产生的蕴含规则的前件仅由一个属性值组成;

(2) H 所产生的蕴含规则的数目是 $\|X\|-\|X'\|$;

(3) 对于每个属性值 $p\in\{X-X'\}$,存在一条蕴含规则 $p\Rightarrow X-p$.

假定 H 所产生的蕴含规则的前件包含多于一个属性值对.若所有这些属性值对都包含在 X' 中,则该规则应该在其父节点中已经考虑过了,因而是冗余的.若前件中存在一个属性值对 $p\in\{X-X'\}$,则该规则对于 $p\Rightarrow X'-p$ 是冗余的,因为后者更为简单.

事实 2. 若节点 $H=(C,X)$有 d 个父节点 $P_1(C_1,X_1),P_2(C_2,X_2),\ldots,P_d(C_d,X_d)$,则对于每个 $p\in\{X-(X_1\cup X_2\cup\ldots\cup X_d\}$ 存在一条蕴含规则 $p\Rightarrow X-p$.

由于任何属性值对 $p\in\{X-(X_1\cup X_2\cup\ldots\cup X_d\}$都是首次出现在子格中,它的可信度应该为 1.任何前件包含 p 的规则对于 $p\Rightarrow X-p$ 都是冗余的.此种类型的蕴含规则的数目应为 $\|X\|-\|X_1\cup X_2\cup\ldots\cup X_d\|$.

事实 3. 若节点 $H=(C,X)$有两个父节点 $P_1(C_1,X_1)$和 $P_2(C_2,X_2)$,$\forall\ p_1\in\{X_1-X_1\cap X_2\}$和$\forall\ p_2\in\{X_2-X_1\cap X_2\}$,存在一条规则 $p_1p_2\Rightarrow X-p_1p_2$,且此种类型的蕴含规则的数目应为 $\|X_1-X_1\cap X_2\|*\|X_2-X_1\cap X_2\|$.

因为若两个属性值对来自于同一个父节点,它们的关系必然在其父节点被访问时已经生成过.因此,任何前件包含 p_1p_2 的规则相对于 $p_1p_2\Rightarrow X-p_1p_2$ 都是冗余的.

事实 3 可以扩展到任意数目的父节点的情形.注意,当 $\|X\|<k$ 时,生成的规则前件最多由 k 个属性值对组成,并且 k 小于父节点的数目.前件只包括当属性值对的规则除外.容易看到,从一个节点产生规则只依赖于它的父节点.

在算法中采用一种启发式方法,只有那些不能形成规则的属性值对集合才被扩展.若一个属性值对不能形成规则,它就被保存在一个候选集中.在下一个循环中,所有候选集中的元素以一种类似于 Apriori 算法的方法连接[5],形成新的候选集.然后测试这些元素,看它们是否可以生成新的规则.

对于可信度小于 1 的规则,使用数组 PX 来辅助计算可信度.PX 指向前件中每个属性值对在格中的首次出现.对于前件为一个属性值对的规则,这些信息可以直接利用.否则,从它们中最大的概念开始查找前件在格中的首次出现.函数 $PX(lhs)$通过此种方法找到前件的支持度.若规则的前件包含在其他规则的前件中,则规则即被放弃,因为通常情况下长前件的规则有较高的可信度.

规则的计算严重依赖于计算几个元素是否包含在一个父节点中的情况.算法中设计了一种有效的计算方法.基本思想是,对于出现在所有父节点中的属性值对使用一个位向量,记录了该属性值对在所有父节点中的出现情况.例如,假设某节点具有 3 个父节点:$\{a,b,c\},\{b,c,d,e\},\{a,e,f\}$,则生成位向量数组 V:

$$V[a]=101,\ V[b]=110,\ V[c]=110,\ V[d]=010,\ V[e]=011,\ V[f]=001.$$

这样,属性值对的任意组合都可以通过简单而快速的 AND 操作来计算.例如,为了确定 a 和 d 是否同时出现在某个父节点中,只需计算 $V[a]\&V[d]$.若结果为 0,它们就没有包含在同一个父节点中.

算法 2. CLACF(concept lattice based association and classification rule mining framework)

```
1.  GenRule(itemset rhs)
2.    通过广度优先搜索找到第 1 个包含 rhs 的节点 H
3.    queue←H,ruleset←∅,singleset←∅
4.    while queue 非空
5.      从队列头移出 H.将所有 H 的子节点推入队列尾部
6.      if H 未访问
7.        GenRuleFromNode(H)
8.        标记 H 为已访问
9.      endif
10.   endwhile
11.
12. GenRuleFromNode(H=(C,X))
13.   d←H 的父节点个数:(C1,X1)...(Cd,Xd)
14.   if (d==1)
15.     ruleset=ruleset∪{p→rhs,sup=C/‖O‖,conf=1|p∈{X-X1}}
16.     singleset←singleset∪{p|p∈{X-X1}}
17.     return
18.   endif
19.   for H 的每个父节点计算它们的并 S,同时生成数组 V
20.   ruleset=ruleset∪{p→rhs,sup=C/‖O‖,conf=1|p∈{X-S}}
```

```
21.  singleset←singleset∪{p|p∈{X−S}}
22.  ruleset=ruleset∪{p→rhs,sup=C/‖O‖,conf=‖PX(p)‖/‖H‖|p∈S,p 和 rhs 不包含在同
       一个父节点中,p∉singleset}
23.  L←{S1∪S2|S1∈S,S2∈S}
24.  while L 非空
25.    S←∅
26.    for L 中每个元素 K
27.      if K 中的所有属性值对不含在同一父节点中,且没有 R∈ruleset 使 R⊆K
28.        ruleset=ruleset∪{K→rhs,sup=C/‖O‖,conf=1}
29.      else
30.        S←S∪K
31.        if K 和 rhs 不含在同一父节点中
32.          ruleset=ruleset∪{K→rhs,sup=C/‖O‖,conf=‖PX(K)‖/‖H‖}
33.        endif
34.      endif
35.    endfor
36.    L←{S1∪S2|S1∈S,S2∈S,‖S1∪S2‖=‖S1‖+1}
37.  endwhile
38.  return ruleset;
```

第 23～38 行以一种类似于 Apriori 算法[12]的方式生成前件多于一个属性值对的规则.算法是依据事实 1～3 设计的.建格的时候,把类属性值看成是普通属性,一起插入到格中.然后对每个类属性值运行一次 CLACF 算法.生成的规则按可信度从大到小排序.若可信度相同,则高支持度优先.分类是通过将新事例从头至尾与规则集进行匹配来完成的.若找不到相应规则,则使用多数类.

4 实验分析

我们使用 MLC++[13]实现并测试了 CLACF 算法.首先对格建造算法 LCA 进行了初步的测试,发现比 Apriori 算法(只生成频繁项目集)要快很多,格算法只生成所有频繁项目集的一个小子集——最大项目集,因此,这样的比较没有多大意义.

算法 CLACF 用于在分类时与事实上的工业标准 C4.5 的比较,实验使用了从 UCI 机器学习库[14]中的 10 个数据集.最小支持度设为 1%,最小可信度设为 0.6,在运行 Windows 98,具有 64MB 内存,PII 233 处理器的 PC 机上进行实验.实值属性离散化采用 MLC++中的熵方法,并使用缺省参数.实验结果见表 2.

Table 2 Empirical comparison of CLACF and C4.5

表 2 CLACF 和 C4.5 的实验比较

Datasets①	C4.5	CLACF	Rule time②	No. of rules③	Lattice time④
Breast	5.0	3.8	20.1	307	24.6
Diabetes	25.8	27.8	2.14	35	0.9
Glass	31.3	18.9	0.88	49	0.82
Heart	19.2	16.6	3.51	528	11.1
Iris	4.7	4.0	0.0	19	0.0
Led7	26.5	23.7	0.83	278	8.0
Monk 1	19.0	9.0	0.22	127	2.5
Monk 2	30.1	19.9	0.38	169	4.3
Monk 3	8.3	5.1	0.28	113	2.4
Pima	24.5	27.1	1.0	25	0.4
Average⑤	18.5	15.6	2.93	165	5.43

①数据集,②规则提取时间,③规则数目,④建格时间,⑤平均.

第 1 列给出了 10 个来自 UCI 数据集的名字.第 2 和第 3 列分别显示了 C4.5 和 CLACF 的错误率.CLACF 在 10 个数据集中有 8 个识别正确率高于 C4.5.CLACF 的平均错误率是 15.6,低于 C4.5 的 18.5 的平均错误

率,第4和第6列给出了建格算法和规则生成算法的执行时间.第5列显示了规则生成算法产生的规则数目.可以看到,CLACF产生的规则集合要比文献[5]中的小得多,而精确率并无降低.

5 结 语

本文提出了一个基于概念格集成分类和关联规则挖掘的算法.概念格体现了概念之间的关系,更易于理解.本文改进了一个Bordat的建格算法,生成一种"频繁"概念格,并在此基础上提出了一个从概念格上生成分类/关联规则的算法.实验表明,算法是有效的.

References:

[1] Wille, R. Reconstructing lattice theory: an approach based on hierarchies of concepts. In: Rival, I ed. Ordered Sets. Dordrecht-Boston: Reidel, 1982. 445~470

[2] Godin, R. Incremental concept formation algorithm based on Galois (concept) lattices. Computational Intelligence, 1995,11(2):246~267.

[3] Njiwoua, P., Nguifo, E.M. Forwarding the choice of bias LEGAL-F: using feature selection to reduce the complexity of LEGAL. In: Walter, D ed. Proceedings of the BENELEARN-97, ILK and INFOLAB. Netherlands: Tiburg University Press, 1997. 89~98.

[4] Carpineto, C., Romano, G. Galois: an order-theoretic approach to conceptual clustering. In: Utgoff, P ed. Proceedings of the ICML-93. Amhers: Elsevier Science Publishers, 1993. 33~40.

[5] Oosthuizen, G.D. The application of concept lattice to machine learning. Technical Report, University of Pretoria, South Africa, 1996.

[6] Sahami, M. Learning classification rules using lattices. In: Lavran, N, Wrobel, S eds. Proceedings of the ECML-95. Grete: Elsevier Science Publishers, 1995. 343~346.

[7] Wang, Zhi-hai, Hu, Ke-yun, Hu, Xue-gang. General and incremental algorithms of rule extraction based on concept lattice. Chinese Journal of Computers, 1999,22(1):66~70 (in Chinese).

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

[9] Deogun, J.S., Raghavan, V.V., Sever, H. Association mining and formal concept analysis. In: Anita, W., San, C eds. Proceedings of the RSDMGrC'98. Duke: Elsevier Science Publishers, 1998.

[10] Hu, Ke-yun, Lu, Yu-chang, Shi, Chun-yi. Incremental discovering association rules: a concept lattice approach. In: Zhong, N., Zhou, L eds. Proceedings of the PAKDD-99. Beijing, 1999. 109~113.

[11] Zaki, M.J., Ogihara, M. Theoretical foundations of association rules. In: Haas, L., Tiwary, A eds. Proceedings of the SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD). Seattle, WA: ACM Press, 1998,7:1~8.

[12] Agrawal, R., Mannila, H., Srikant, R., *et al*. Fast discovery of association rules. In: Fayyad, M., Piatetsky-Shapiro, G., Smyth, P eds. Advances in Knowledge Discovery and Data Mining. Boston: AAAI/MIT Press, 1996. 307~328.

[13] Kohavi, R., Sommerfield, D., Dougherty, J. Data mining using MLC++: a machine learning library in C++. Tools with Artificial Intelligence. New York: IEEE Computer Society Press, 1996. 234~245.

[14] Merz, C.J., Murphy, P. UCI repository of machine learning database. http://www.cs.uci.edu/~mlearn/MLRepository.html.

[15] Liu, B., Hsu, H., Ma, Y. Integrating classification and association rule mining. In: Agrawal, R., Stolorz, P., Piatetsky-Shapiro, G eds. Proceedings of the KDD-98. New York: Kluwer Academic Publishers Group, 1998. 80~86.

附中文参考文献:

[7] 王志海,胡可云,胡学钢.概念格上规则提取的一般和渐进式算法.计算机学报,1999,22(1):66~70.

An Integrated Mining Approach for Classification and Association Rule Based on Concept Lattice

HU Ke-yun, LU Yu-chang, SHI Chun-yi

(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

E-mail: hky@s1000e.cs.tsinghua.edu.cn; {lyc,scy}@tsinghua.edu.cn

http://www.tsinghua.edu.cn

Received May 20, 1999; accepted September 14, 1999

Abstract: This paper adapts Bordat's lattice construction algorithm to integrated mining, presents a fast algorithm to extract both association and classification rules from concept lattice. It is shown that classification and association rules mining can be unified under concept lattice framework. Experimental evaluation proved the validity of the algorithm.

Key words: classification; association rule; concept lattice; integrated mi...