基于粗糙信息向量的一种决策规则获取算法

桑妍丽,梁吉业
SANG Yan-li,LIANG Ji-ye

山西大学 计算机与信息技术学院,太原 030006
College of Computer & Information Technology,Shanxi University,Taiyuan 030006,China
E-mail:dada_syl@163.com

SANG Yan-li,LIANG Ji-ye.Algorithm of decision rules mining based on rough information vector.Computer Engineering and Applications,2009,45(17):136-138.

Abstract: A decision rule mining algorithm based on rough information vector is proposed in inconsistent information system.The brief decision rules according with the demands of threshold is directly mined from decision tables based on rough information vector.In this process,decision support ability of condition vector to decision vector is used,and the concrete decision support ability of condition attribute value is not losing.The consistent rules set and default rules set are mined in the case of the each simplified level of condition attribute.The feasibility of algorithm is indicated using theoretical analysis and the concrete instance.
Key words: inconsistent information system;decision rules;rough information vector

摘 要:针对不一致信息系统中决策规则获取问题,提出了一种基于粗糙信息向量方法的决策规则挖掘算法。基于粗糙信息向量,利用条件向量对决策向量的决策支持能力,直接从决策表中挖掘出符合阈值要求的尽可能简洁的决策规则,且不损失条件属性值的决策支持能力。利用该算法可以挖掘出决策系统中条件属性在各个简化层次情况下的确定性规则和缺省规则集合。理论分析和实例表明该算法在不一致信息系统中的决策规则获取上是可行的。
关键词:不一致信息系统;决策规则;粗糙信息向量
DOI:10.3778/j.issn.1002-8331.2009.17.041 **文章编号**:1002-8331(2009)17-0136-03 **文献标识码**:A **中图分类号**:TP181

1 前言

决策规则获取是粗糙集理论应用的一个重要领域。从数据挖掘的角度来看,数据库可以分成两大类,一类是一致性的,另一类是不一致性的。

基于粗糙集的规则知识获取研究,在一致性决策表的情况下人们已经提出很多有效方法,解决了一些问题[1-3]。但现实中却存在大量的不一致性。不一致性的存在,缘于很多因素,如选择的描述属性不充分、测量中的差错以及记录过程的失误等。因此,规则知识获取的研究中,考虑不一致性是很重要的,否则可能会阻止发现一些有价值的分类知识,也会使其对待识样本的预测能力大为降低.基于粗糙集理论获取不确定规则的研究已取得了一些成果[4-7]。文献[7]中 Skowron 提出了从不一致性的数据中提取命题缺省规则(default rule)的方法,能有效地解决在不一致性数据中的规则提取问题。

将针对文献[7]中 Skowron 提出的从不一致性的数据中提取命题缺省规则的方法所存在的问题提出一种基于粗糙信息向量方法的决策规则获取算法。在对规则的不一致性分析的基础上,利用条件向量对决策向量的支持能力,直接从决策表中挖掘出符合预先设置的阈值要求的描述长度小的无冗余规则集。

2 相关概念

2.1 决策表与信息向量

定义 1[8] 设决策系统 $S=(U,C\cup D,V,f)$,对象集合 $Q\subseteq U$ 在属性集合 $A\subseteq C\cup D$ 上的不分明关系 $IND_Q(A)=\{(x,y)\in Q\times Q|\forall a[a\in A\rightarrow f_a(x)=f_a(y)]\}$,$IND_U(A)$简记为 A。

定义 2[8] 对象 $x\in U$ 在属性集合 $A\subseteq C\cup D$ 上的信息向量:$inf_A(x)=\{(a,f_a(x))|a\in A\}$

若属性集 A 中有 k 个属性,那么称信息向量 $inf_A(x)$为 k 元信息向量,记为 $inf_A^k(x)$。

若 $A\subseteq C$,则称 $inf_A(x)$为条件向量;若 $A\subseteq D$,则称 $inf_A(x)$为决策向量。

定义 3[8] 对象集合 $Q\subseteq U$ 在属性集合 $A\subseteq C\cup D$ 上的信息向量集:

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60773133,No.70471003);国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z165);高等学校博士学科点专项科研基金(the Foundation of Doctoral Program Research of Ministry of Education of China No.20050108004)。
作者简介:桑妍丽(1978-),女,讲师,主要研究方向:数据挖掘与智能决策;梁吉业(1962-),男,博士生导师,教授,主要研究方向:粗糙集理论,数据挖掘和人工智能。
收稿日期:2008-04-03 **修回日期**:2008-06-20

$INF_A(Q)=\{inf_A(x)|x\in Q\}$

若属性集 A 中有 k 个属性，那么称信息向量 $INF_A(Q)$ 为对象集合 Q 在属性集合 A 上的 k 元信息向量集，记为 $INF_A^k(Q)$。

定义 4[8] 对象集合 $Q\subseteq U, A\subseteq C\cup D$，信息向量 $t\in INF_A(U)$。t 所决定的 Q 上的等价类 $\|t\|_Q=\{x\in Q|inf_A(x)=t\}$（$\|t\|_U$ 简记为 $\|t\|$，$card(\|t\|)$ 表示 $\|t\|$ 的基数）。

定义 5 对象集合 $Q\subseteq U, A\subseteq C\cup D$，信息向量 $t\in INF_A(U)$，t 所决定的 Q 上的等价类的覆盖度：

$$cov(t)=\frac{card(\|t\|_Q)}{card(U)}$$

2.2 不一致信息系统中决策规则的评价

定义 6 决策规则具有如下形式：$cond\rightarrow deci$。

称 $cond$ 和 $deci$ 分别为规则的条件部分和决策部分，其中 $cond\in INF_A(U), A\subseteq C$；决策属性子集 $B\subseteq D$，决策向量 $deci\in INF_B(\|cond\|)$。

若条件向量 $cond$ 对应一个决策向量，当且仅当 $\|cond\|\subseteq\|deci\|$，称该规则为确定性规则，否则，称之为不确定规则。

定义 7 设 $s_1\rightarrow t, s_2\rightarrow t$ 是两条结论都为 t 的规则，若 s_1 是 s_2 的子句，那么 $s_2\rightarrow t$ 是一条冗余规则。

定义 8[8] 条件向量 $cond$ 对决策向量 $deci$ 的决策强度：

$$strength(deci|cond)=\frac{card(\|deci\|_{\|cond\|})}{card(\|cond\|)}$$

作为规则 $cond\rightarrow deci$ 的置信度，一定程度上表示了在用该规则进行推理时正确的概率，反映了规则的确定性。$strength(deci|cond)=1$ 时，能产生确定性规则，否则，成为不确定性规则。

考虑到规则的随机性，引入了规则支持度的概念[1]。

定义 9 条件向量 $cond$ 对决策向量 $deci$ 的支持度：

$$support(deci|cond)=\frac{card(\|deci\|_{\|cond\|})}{card(U)}$$

即 $cond\rightarrow deci$ 规则的支持度，表示该规则的支持数在论域 U 中的比重。在一定程度上反映了规则的可靠性。

3 基于粗糙信息向量的决策规则获取算法

针对文献[7]中 Skowron 提出的从不一致性的数据中提取命题缺省规则的方法存在的问题，提出了基于粗糙信息向量的决策规则挖掘算法。在对规则可靠性的分析基础上，引入规则支持度概念，有效地过滤噪声，避免因噪声影响而产生的随机规则；其次基于粗糙信息向量，利用条件向量对决策向量的决策支持能力，挖掘出符合条件的决策规则，且不损失条件属性值对的决策支持能力；在规则挖掘过程中，改变通过移去核属性来获取缺省规则的方法，采用逐步扩充属性集的规则挖掘策略，以便得到不同简化层次上的决策规则集合，并且避免了冗余规则的产生。

3.1 算法原理

提出的决策规则获取算法的原理是：给定目标规则的置信度和支持度阈值，从样本数据出发，先生成第一层所有的属性节点，计算所有条件属性的一元条件向量集 T_1，挖掘出第一层的所有满足规则支持度和置信度要求的一组一元规则 $Rule_1$，接着，依次为每个节点按一定方式加入一个条件属性，得到该节点的后继节点，衍生成第二层节点，递推计算每一个节点上二元条件向量集 T_2，挖掘出另一组满足要求的二元规则 $Rule_2$，……，直至所有进行挖掘的条件向量包含全部的属性或者某一层所有条件向量的所产生的规则都不满足预先定义的支持度阈值（$T_k=\phi$）时，规则挖掘结束。

（1）节点的衍生

这里规定：每个节点的属性集中的条件属性的排列顺序按属性的字典序排列。

以空条件属性集作为初始节点，在衍生下一层节点时，采取为当前节点属性集添加条件属性的方式，为了保证不重复生成条件向量，添加时按照字典序加入一个属性，依次生成每个节点的后继节点。

在对每一层节点进行规则挖掘时，考虑几个比较耗时的因素：挖掘的深度，条件属性等价类的生成等，减少不必要的运算，提高规则挖掘的效率。

（2）挖掘的深度

定理 1 对象集合 $Q\subseteq U$，设条件属性子集 $A\subseteq C$，条件向量 $cond\in INF_A(U)$，若 $cov(cond)<\alpha$，其中 α 为某一阈值，则由条件向量 $cond$ 所产生的规则 $cond\rightarrow deci$ 的支持度必然小于 α，其中 $\forall deci\in INF_B(\|cond\|)$。

证明：若 $cov(cond)<\alpha$，$cov(cond)=card(\|cond\|_Q)/card(U)$，而规则 $cond\rightarrow deci$ 的支持度 $support(deci|cond)=card(\|deci\|_{\|cond\|})/card(U)$，而 $card(\|deci\|_{\|cond\|})\leqslant card(\|cond\|)$，则 $support(deci|cond)<\alpha$，得证。

由定理 1 可以在算法设计中对集合中的条件向量进行筛选操作，认为由不满足阈值要求的条件向量不能产生任何满足支持度阈值要求的规则，进行删除，通过删除操作可以减少大量不必要的计算。

定理 2 设条件属性子集 $A\subseteq C$，k 元条件向量 $cond^k\in INF_A(U)$，决策属性子集 $B\subseteq D$，某一决策向量 $deci\in INF_B(\|cond\|)$ 若条件向量 $cond^k$ 的任一子向量 $cond^l$（$l\leqslant k$）所产生的规则 $cond^l\rightarrow deci$ 的支持度不大于阈值 α，那么规则 $cond^k\rightarrow deci$ 的支持度必然也不大于支持度阈值 α，反之，若规则 $cond^k\rightarrow deci$ 的支持度大于 α，则 $cond^l\rightarrow deci$ 的支持度必然大于 α。

证明：$cond^l\subseteq cond^k$，则 $\|cond^l\|\supseteq\|cond^k\|$，支持规则 $cond^l\rightarrow deci$ 的元组数必然小于支持规则 $cond^l\rightarrow deci$ 的元组数，即若 $support(cond^l\rightarrow deci)\leqslant\alpha$ 则 $support(cond^k\rightarrow deci)\leqslant\alpha$，反之亦然。

因此，当某一 k 元条件向量产生的所有规则的支持度都小于事先规定的支持度阈值 α 时，则以它为子向量的所有 k+1 条件向量所产生的规则支持度也将小于 α。而当第 k 层所有的 k 元条件向量产生的规则的支持度都小于 α 时，第 k+1 层产生的规则的支持度也必然都小于 α，此时，就可以停止对其所有上层节点的规则挖掘，从而提前结束对规则的挖掘。

根据定理 2 算法加入了对第 k 层每个节点生成的 k 元条件向量的所有 k–1 元子条件向量进行检查的操作，如果发现有某个 k–1 元条件向量不在 T_{k-1} 中，则将该条件向量从 T_k 中删除。当 T_k 为空的时候挖掘结束。

（3）条件属性等价类的生成

在规则挖掘过程中，每一个节点属性等价类的生成都可以利用上一层节点的等价类的计算结果，这样就避免了在每个节点上重复计算条件属性类，提高算法的效率。

设对象集合 $Q\subseteq U$ 在条件属性子集 $A\subset C$ 上的信息向量集 $INF_A(Q)=\{t_1,t_2,\cdots,t_s\}$，信息向量 $t_i\in INF_A(Q)$，$i=1\cdots s$ 所决定的 Q 上的等价类集为 $\{\|t_i\|, i=1\cdots s\}$，当添加属性 $a, a\in C-A$，其对应的属性集为 $A\cup\{a\}$，则对象集合 $Q\subseteq U$ 在属性集 $A\cup\{a\}$ 上的条件向量集：

$INF_{A\cup\{a\}}(Q)=\{INF_a(\|t_1\|),INF_a(\|t_2\|),\cdots,INF_a(\|t_s\|)\}$

由上式可以根据上层节点的等价类递推出下层节点的等价类。对 T_k 集合中的 k 元条件向量进行递推运算生成 k+1 元向量集合 T_{k+1},为了保证 T_{k+1} 中包含所有使规则成立的所有 k+1 元向量,又避免生成的 T_{k+1} 集合中包含使冗余规则成立的条件向量,因此在每次生成决策规则时,要同时从 T_k 删去该向量,这样就避免了冗余规则的生成。

依据上述的原理,设计出下面的决策规则挖掘算法。

3.2 算法实现

该算法挖掘出各个简化层次上满足置信度和支持度阈值要求的确定性规则和缺省规则的无冗余规则集。

算法描述:

设条件属性集合 $C=\{a_1,a_2,\cdots,a_n\}$,则$|C|=n$;第 i 层上的节点数为 C_n^i;第 i 层上的规则集为 R_i。

记 N_{ij} 表示第 i 层的第 j 个节点,$C_{N_{ij}}$ 表示节点 N_{ij} 对应的属性集。

输入:决策系统 $S=<U,C\cup D,V,f>$,决策规则的最小支持度阈值 α 和置信度阈值 β。

输出:决策系统上的每一层的确定性规则和缺省规则集合。

(1)生成决策属性向量集 $INF_D(U)=\{inf_D(x),x\in U\}$;

(2)$T_1=\{INF_{\{a_i\}}(U),i=1\cdots s\}$,其中 $INF_{\{a_i\}}(U)=\{\inf_{\{a_i\}}(x)|x\in U\}$;//生成一元候选集 T_1

(3)i 从 1 到 s 循环执行步骤(4)~(21);//生成第 i 层上的规则集 R_i

(4)计算 T_i 中每个条件向量的覆盖度,若小于阈值 α,则从 T_i 删除 $cond$;

(5)j 从 1 到 C_n^i 循环执行步骤(6)~(17);//生成节点 N_{ij} 上的规则

(6)若 $INF_{C_{N_{ij}}}(U)\neq\Phi$,对每一个 $cond\in INF_{C_{N_{ij}}}(U)$,反复执行步骤(7)~(17),否则转(5);

(7)flag⇐true;//设置标志位来判断条件向量对决策向量的支持度是否满足阈值

(8)对于每一个 $deci\in INF_D(\|cond\|)$,反复执行(9)~(15);

(9)计算条件向量 $cond$ 对决策向量 $deci$ 的支持度 $support(deci|cond)$;

(10)若 $support(deci|cond)>\alpha$,执行步骤(11)~(15),否则转步骤(8);

(11)flag⇐flase;

(12)计算条件向量 $cond$ 对决策向量 $deci$ 的决策强度 $strength(deci|cond)$;

(13)若 $strength(deci|cond)>\beta$,执行步骤(14)~(15),否则转步骤(8);

(14)将决策规则 $cond\xrightarrow{strength(deci|cond),support(deci|cond)}deci$ 存入 $Rule_i$;

(15)将条件向量 $cond$ 从 T_i 中删除,并转步骤(6);

(16)若 flag 为真,将条件向量 $cond$ 从 T_i 中删除;

(17)转步骤(6);

(18)若 $T_i\neq\phi$,执行步骤(19)~(20),否则转步骤(22);

(19)对 T_i 中的 i 元条件向量确定的等价类进行递推运算生成第 i+1 每个节点上的条件向量集 T_{i+1};

(20)$\forall cond^{i+1}\subset T_{i+1}$,若存在 $cond^i\in cond^{i+1},cond^i\notin T_i$,则进行删除//根据定理 2 删除不满足支持度要求的向量;

(21)转步骤(3);

(22)算法结束。

4 算法分析

采用文献[7]中的例子作为示例,与 Skowron 算法[7]获取的规则进行比较,保持了经典粗糙集挖掘出的规则集的精简的特点:规则集所含规则的数目较少,每条规则所相关的属性也较少。具体如表 1 所示。

表 1 Skowron 算法与本文算法结果对照表

	规则数	规则所含条件属性数
Skowron 算法	9	含 1 个属性 5 条,含 2 个属性 4 条
本文算法	4	含 1 个属性 3 条,含 2 个属性 1 条

由上述算例分析可知,算法充分利用了条件属性值对的决策支持能力,并且引入了规则支持度,一定程度上抑制了随机规则的产生;另一方面在算法设计中避免了冗余规则,而且规则的描述长度短,使产生的规则集大为精炼,挖掘出既简便又易于理解和实用的规则,提高了规则产生和实际分类的效率,因此具有一定的实用价值。在算法实现效率上,一方面由于算法没有进行差别矩阵的计算,因此节省了计算时间;另一方面在算法设计过程中省去了大量不必要的运算,在前面作了较为详细的论述,提高了算法效率。

5 结论

基于粗糙信息向量的方法,利用条件向量对决策属性向量的支持能力,直接从决策表中挖掘出满足置信度和支持度阈值要求的确定性规则和缺省规则的无冗余规则集合,且不损失条件属性值对的决策支持能力。对于不一致信息系统中的决策规则挖掘有一定的实用性,理论分析和实例表明该算法在不一致信息系统中的决策规则获取上是可行的。

利用该算法可以挖掘出决策系统在各个简化层次上的规则集合。在实际问题中,应用得到的规则进行推理或决策,根据已有的信息在模型上逐层匹配,因此在待识样本信息不完备的状况下都能尽可能给出问题的决策,是很有实际意义的。

该算法计算简单,其效率主要与属性的个数相关,当属性个数不大、取值不多时,是一个高效算法。但在实际系统应用中如何合理确定置信度和支持度阈值,以便保证在高效挖掘的同时,能挖掘出既简便,又实用的规则还要做进一步研究。

参考文献:

[1] 常犁云,王国胤,吴渝.一种 Rough Set 理论的属性约简及规则提取方法[J].软件学报,1999,10(11):1206-1211.

[2] Lan S,Mo Z W,Hu D.Methods of learning rules based on rough set:LBR and LEM3[C]//IFSA World Congress and 20th NAFIPS International Conference,2001,2:753-756.

[3] Xia Y J,Li S Y,Xi Y G.A method of inducing decision rules based on rough set theory[J].Control and Decision,2000,16(5):577-580.

[4] 尹旭日,陈世福.一种基于 Rough 集的缺省规则挖掘算法[J].计算机研究与发展,37(12):1441-1445.

(下转 224 页)