文章编号:1000 - 582X(2004)04 - 0044 - 05

一种递增式归纳学习算法*

程玉胜[1],张春梅[1],胡学钢[2]
(1. 安庆师范学院 数学计算机系,安徽 安庆 246011; 2. 合肥工业大学 计算机学院,安徽 合肥 230009)

摘 要:笔者提出的递增式决策函数生成算法是在改造后的分辨矩阵下完成的,更有利于编程实现;避免了传统粗集方法对数据作一次性处理生成庞大的决策矩阵,从而有效解决了在处理大规模数据库时内存不足问题;由于决策函数是递增式生成的,因此该算法适应了目前数据变化的特点,实现了新例的动态学习,因此这种对信息的重用,减少了数据挖掘的时间;同时,递增式决策函数生成算法从根本上解决了多类决策的递增式学习问题。

关键词:机器学习;粗集;分辨矩阵;决策函数
中图分类号:TP18 **文献标识码**:A

学习是使系统做一些适应性的变化,使得系统下一次完成同样或类似的任务时比上一次更有效(西蒙)[4]。机器学习是一种利用计算机模仿人类学习的过程,其中归纳学习就是其中一种,它是一种从原始数据出发,通过归纳产生一般性规则的学习方法。但是现实生活中,数据是在不断变化的,如何高效利用已有的知识和规则来指导我们的行为或者是利用已有的规则对我们现有的情况进行预测判断是目前研究的一个热点。文献[2]在信息系统基础之上生成决策分辨矩阵,提出了利用该矩阵生成决策函数,最后生成规则的方法,但是由于它直接对原始数据进行处理,没有进行约简(去掉冗余信息),以致于对生成的决策函数的化简非常烦琐,其次该文利用属性 - 属性值生成分辨矩阵,因此对每类问题都需要单独生成矩阵,比较复杂。文献[3]增量算法是在原有分辨矩阵上进行,因此没能有效利用约简的中间结果,计算量比较大。这两种方法都是先生成分辨矩阵,因此需占用大量内存,不适用于对大规模数据库数据的处理,而且也仅讨论了两类决策的递增式学习问题。

笔者利用扩展分辨矩阵定义,并结合 MATLAB 语言编程,将对象间的可区分性质转化为用'1'和'0'来表示的区分矩阵,避免了决策函数的复杂逻辑化简,使得约简算法更有利于计算机实现,本文第 2 节介绍了这种方法。第 3 节提出递增式决策函数的生成算法,解决了大规模数据的处理问题,避免了传统粗集方法对大量数据作一次性处理而造成的内存不足问题,同时又具备了处理动态新增数据的能力;并给出新例的动态学习过程算法,由于重用了以前的信息,大大减少了数据挖掘时间,通过参数传递,从根本上解决了多类决策的学习问题。最后举例进行了说明。

1 属性约简的相关定义与算法

设 $S=\{U,A\}$ 是一个信息系统,其中 U 是一个非空集合,$U=\{s_1,s_2,\cdots\cdots s_n\}$ 是个体域,$A=C\cup\{d\}$ 是属性(Attribute)集合,子集 $C=\{a_1,a_2\cdots\cdots a_m\}$ 和 $\{d\}$ 分别称为条件属性集和决策属性集。

1.1 经典决策矩阵定义

定义 1 经典决策矩阵[9]

$$C_D(i,j)=\begin{cases}\{a_k: a_k\in C\wedge a_k(s_i)\neq a_k(s_j)\} & \mathrm{d}(s_i)\neq\mathrm{d}(s_j)\\ \varphi & \mathrm{d}(s_i)=\mathrm{d}(s_j)\end{cases}$$

在该定义中,决策矩阵的每个值反映了不同类别对象间的可区分属性构成的集合。

* 收稿日期:2003 - 11 - 28
基金项目:国家自然科学基金课题资助项目(69985004);安徽省教育厅自然科学研究项目(2004kj264)
作者简介:程玉胜(1969 -),男,安徽省桐城市人,讲师,硕士,主要从事 Rough 集和神经网络方向的研究。

1.2 扩展决策矩阵定义

定义 2 扩展的决策矩阵

修改定义 1 为扩展的决策矩阵，将对象间可区分性质转换为用‘1’来表达，从而对象间可区分性质便可转换为在矩阵 A 中，判断两个不同向量对应位是否相等：

$$C_k(i,j) = \begin{cases} 1 & if \quad a_k(s_i) \neq a_i(s_j) \wedge d(s_i) \neq d(s_j) \\ 0 & \text{if} \quad d(s_i) = d(s_j) \end{cases}$$

因此，修改后的扩展矩阵 $C_k(i,j)$ 与 $C_D(i,j)$ 之间联系：

$$C_D(i,j) = \{a_k : C_k(i,j) = 1 \qquad k \in 1 \cdots N\}$$

它反映了经典决策矩阵与修改后决策矩阵定义即集合变量到整型变量之间的转换关系，从而为约简在计算机上高效实现成为可能。以表 1 决策表系统为例，假设 $C_D(1,2) = \{a_1, a_3, a_4\}$，则 $C_k(i,j) = 1011$，显然二进制数长度等于条件 C 的基数。

1.3 约简条件和方法

为高效的求出约简，在 B 中增加一列，用于统计两对象间可区分属性的个数，记为 NumberOfOne

$$\text{NumberOfOne}(i,j) = \sum C_k(i,j)$$

所以区分矩阵 B 中元素为：$B_{ij} = \begin{cases} C_k(i,j) \\ \text{Number Of One}(i,j) \end{cases}$

设 $B1$ 和 $B2$ 是矩阵 B 中任意的两个元素，如果存在最少某些属性能将任意两个对象区分，则称该属性可被提取的。于是所有可被提取的属性便构成了系统的可能约简。因此约简条件是：

$$RED = \min(\{a_k : \forall(i,j)\ \exists k \in C_k(i,j) = 1\})$$

其中 min 表示条件 a_i 是最少的。由于约简是求最少的属性，因此与 B 中每行‘1’的个数有关，所以提取约简的过程：

1）对 B 按 NumberOfOne 递增排序；

2）将 $B1$ 和 $B2$ 进行“与”操作（不妨假设前者‘1’的个数少）；

3）保留最少的 $B1$。

1.4 数据结构定义

a_i 是信息系统条件属性；Object_pairs 为在 B 中第一列前增加一列所得，表示属于不同类的两个对象；b 表示为对象 (i,j) 在条件 a_i 下的取值，由 $C_k(i,j)$ 求得。如图 1 所示。

Object_Pairs	a_1	a_2	a_3	……	a_n	NumberOfOne
(i, j)	b	b	b	……	b	

图 1　扩展矩阵数据结构定义

例 1　表 1 为购房者特征模式[4]所示的信息系统，其中，$a_1 = \{0,1\}$ 表示性别 = {男，女}；$a_2 = \{0,1,2\}$ 表示收入{700 元以下，700 - 1800 元，1800 元以上}根据本地情况得出；$a_3 = \{0,1\}$ 表示年龄 = {中年，青年}；$a_4 = \{0,1\}$ 表示婚否 = {已婚，未婚}；$d = \{0,1\}$ 表示购房者可承受单价 = {2000 元以下，2000 元以上}。

表 1　决策信息系统

调查对象	条件属性				决策属性
	a_1	a_2	a_3	a_4	d
obj1	0	1	0	0	0
obj2	1	1	1	1	1
obj3	1	0	0	0	0
obj4	0	2	0	1	1
obj5	1	2	0	0	1
obj6	1	1	0	0	0
obj7	1	2	1	1	1
obj8	0	1	1	1	1

按照上述定义和文献[5,8]算法，可求得表 1 系统所有可能约简集的约简表，如表 2 所示，记为 New$B$。最后结果：核为 $\{a2\}$，约简为 $\{a2,a3\}$，$\{a2,a4\}$。

表 2　约简表

Object_Pairs	属性集 A				NumberOfOne
	a_1	a_2	a_3	a_4	
3,5	0	1	0	0	1
1,8	0	0	1	1	2

2　递增式学习相关算法

为了表达方便，记 $b_{ij} = \overset{M}{\underset{k=1}{C}}_k(i,j)$，即矩阵 B 中的每一行，M 为信息系统条件属性的个数。假设决策信息系统在决策属性 d 下不可分辨类 $\text{IND}(d) = \{C1, C2, \cdots Ci \cdots Cn\}$。

2.1 递增式决策函数生成算法

input：$IS = < U, A, i, [i]_{IND(d)} >$　　//$[i]_{IND(d)}$ 表示包含 i 对象的等价类

1）置 fCi 为空；//fCi 为 Ci 类的决策函数；

2）令 $U^* = U - [i]_{IND(d)}$，从 U^* 取出一对象 j；

3）求出 b_{ij}；$U^* = U^* - \{j\}$；

4）当 U^* 不为空时做：从 U^* 中取出一个对象 p；$U^* = U^* - \{p\}$，求出 b_{ip}，计算 $b_{ij} \& b_{ip} \rightarrow b_{ij}$；

5) 将 b_{ij} 中所有出现的属性,用 i 对象对应的属性值对 $(a,a(i))$ 代替→FCi;

6) 返回 FCi 的值。

output: FCi

递增式决策函数算法表明,通过修改后扩展矩阵的定义对取自不同决策类的对象操作,不断调整决策函数的值,避免了传统粗集方法对数据作一次性处理生成一个庞大的决策矩阵,从而有效解决了在处理大规模数据库时内存不足问题,同时为递增式学习提供了基础。

例 2 继例 1。

$U=\{obj1,obj2,obj3,obj4,obj7\}$, $A=\{a_2,a_3\}$ IND $(d)=\{C_1,C_2\}$,其中 $C_1=\{obj1,obj3\}$, $C_2=\{obj2,obj4,obj7\}$

根据上述算法很容易求出 $\{d=1\}$ 类的决策函数 $FC2$:$\{(a_3,1),(a_2,2)\}$

$\{d=0\}$ 类的决策函数 $FC1$:$\{(a_3,0)\wedge(a_2,1),(a_2,0)\}$

2.2 动态学习算法

通常把知识系统中的对象集称为训练样本(train example),上述规则的生成实际上是在训练样本中产生的;当有新例(test/learning example)到来时,系统应有能力来区分或者识别该对象所属的类别。新例的到来,可能破坏原有的约简,进而破坏已生成的规则库。笔者的方法是考虑新例对约简的影响入手,最终实现动态式学习。

动态学习算法如下:

step1:判断新例 new 所属类别,如果不属于 IND(d)中,则新例构成新类别,记为 C_{n+1},加入 IND(d)中,调用递增式决策函数学习算法,否则 goto step2;

step2:假设属于第 Ci 类,对任意的 $j\in U-Ci$,重复:在 new$B$ 约简表中加入一行,存放 $b_{new,j}$,值;

step3:判断新例是否影响约简集[4],若是记为 flag = 1否则 flag = 0;

step4:如果 $flag=0$,增加 Ci 类规则:

将 $IS=<U,A,new,Ci>$ 作为入口参数;

调用递增式决策函数生成算法→$t1$

$FC_i=FC_i\vee t1$

更新其它类规则,重复直到 IND(d)为空:

从 IND(d)中取出 Cj 类;

IND(d) $-Cj$→IND(d)

将 $IS=<Cj,A,new,\varphi>$ 作为入口参数;

// * φ 为空

调用递增式决策函数生成算法→t_1

$FC_j=FC_j\vee t_1$

否则:

计算出新的约简表,重新化简(如合并相同记录等)原有信息系统,得 $IS=<U',A'>$

更新 C_i 类规则:

将 $IS=<U',A',new,C_i>$ 作为入口参数;

调用递增式决策函数生成算法→t_1

$FC_i=FC_i\vee t_1$ // * FC_i 为规则库中的规则,不能因为一条新例对约简的影响而改变所有规则

更新其它类规则,重复直到 IND(d)为空:

从 IND(d)中取出 C_j 类;

IND(d) $-C_j$→IND(d)

将 $IS=<Cj,A,new,\varphi>$ 作为入口参数;

// * φ 为空

调用递增式决策函数生成算法→t_1

$FC_j=FC_j\vee t_1$

例 3 继例 2。

第一种情况,如表 3 中 obj9。(新例不影响约简):

表 3 新例

	a_1	a_2	a_3	a_4	d
Obj9	1	0	1	1	1
Obj10	1	0	0	1	1

因为 $d=1$ 属于正例,由递增式学习算法,在 new$\boldsymbol{B}$ 表中增加 3 行:

$b_{91}=1111$, $b_{93}=0011$, $b_{96}=0111$

计算得 new$\boldsymbol{B}$ = {0100,0011},与原有约简集 new$\boldsymbol{B}$ 相同

将 $IS=<U,A,obj_9,C_2>$ 作为入口参数,调用递增式决策函数算法得 $t_1=\{(a_3,1)\}$

由例 2 结果 FC_2:$\{(a_3,1),(a_2,2)\}$, $t1$ 包含在 FC_2 中,故不影响 C_2 类规则集。

更新 C_1 规则:

取出 C_1,此时 IND(d)为空,将 $IS=<C1,A,obj9,\varphi>$ 作为入口参数;调用递增式决策函数算法得

$t_1=\{(a_2,1)\wedge(a_3,0),(a_2,0)\wedge(a_3,0)\}$

结合 FC_1,故对应的 C_1 类规则为:$\{(a_2,1)\wedge(a_3,0),(a_2,0)\wedge(a_3,0)\}$

由上计算可知:新例可被正确识别。

第二种情况,如表 3 中 obj10。(新例影响约简):

因为在原有约简和生成的规则下,由 FC_1:{(a_3,

0) $\wedge$ (a_2,1),(a_2,0)}知新例应为 $d=0$ 类,因此不能被正确识别。分析原因:上述规则是在{{a_2,a_3}}产生的。由本算法计算得表 4 如下:

因为 $d=1$ 属于 C_2 例,按照同样办法,计算得

$b_{10,1}$ = 1101 $b_{10,3}$ = 0001 $b_{10,6}$ = 0101 new$\boldsymbol{B}$ = {0100,0001},从而约简为{{a_2,a_4}},故 flag = 1

得 $IS=<U',A'>$,其中 $U'=\{1,2,3,4,5,10\}$, $A'=\{a_2,a_4\}$

表 4　更新值

Object_Pairs	a_2	a_4	NumberOfOne
1,10	1	1	2
3,10	0	1	1

更新 C_2 类规则:

将 IS = $<U',A',10,C_2>$作为入口参数;

调用递增式决策函数学习算法得 $t_1=\{(a_4,1)\}$

$C_2=FC_2\vee t_1=\{(a_3,1),(a_2,2)\}\vee\{(a_4,1)\}=\{(a_3,1),(a_2,2),(a_4,1)\}$

对应的规则可描述为:

if $a_3=1$ then $d=1\vee$ if $a_2=2$ then $d=1\vee$ if $a_4=1$ then $d=1$,可见新例属于 $d=1$ 类,被正确识别。更新 C_1 类规则不再赘述。

2.3　算法比较与分析

从时间复杂度来看:在两类决策系统下,假设系统有 n 个对象(其中 $n1$ 个正例类),m 个属性,用经典决策矩阵定义求解区分矩阵比较次数[2-3]为 $m\times n^2$,而文献[8]至少也要 $m\times\frac{n(n-1)}{2}$次,笔者仅比较 $m\times\frac{n1(n-n1)}{2}$次;在求约简时,文献[2-3]分别与比较次数相同,而笔者则为$\sqrt{m}\times\frac{n1(n-n1)}{2}$,因为按照 NumberOfOne 定义,它的最大值为 m,从而在矩阵中有近一半的元素$\frac{m}{2}\times\frac{n1(n-n1)}{2}$将被化简,理论上可以达到$\sqrt{m}\times\frac{n1(n-n1)}{2}$,从而高效的求出约简。

从存储空间看:对象 x,y 在属性 a 下能否区分占一个 bit 位,每行占用空间字节(byte);共有行,所占总空间为 byte;而传统的基于区分矩阵的约简则为:假设每一个属性在存储时仅占用一个字节空间,则空间为 $(m+1)\times$byte。

从递增式学习算法来看,文献[2-3]都是在决策矩阵基础上通过增加行或列的方法实现递增式学习,因此约简求解过程中产生的中间结果没能有效利用;笔者则将约简的中间结果保存在 new$\boldsymbol{B}$ 约简表中,因此新例对约简的影响和约简的过程都直接通过在 new$\boldsymbol{B}$ 中增加相应的行而实现,这种对信息的重用避免了区分矩阵重新生成和约简的重新计算。而且通过对新例的类别判断和集合的差(-)操作实现了多类决策约简的递增式求解,从而有效实现了多类决策系统的递增式学习。

3　结论

本递增式学习算法是在扩展分辨矩阵的基础之上,将对象间的区分性质转换为 0 和 1 来表达,避免了繁杂的逻辑化简,使得约简算法在计算机上执行更加有效;而且递增式决策函数算法解决了在大规模数据挖掘过程中,传统粗集方法对大量数据一次性处理造成的内存不足问题,能够重用以前数据挖掘的结果,并处理新增数据,大大减少了数据挖掘的时间;并且通过调用递增式决策函数算法,有效的解决了文献[2-3]提出的多决策类的学习问题。

参考文献:

[1] MARZENA KRYSZKIEWICZ. Finding Reduces in Composed Information Systems[A]. Rough Sets, Fuzzy Sets and Knowledge Discovery[C]. 1993,261-273.

[2] NING SHAN, WOJCIECH ZIARKO. An Incremental Learning Algorithm for Constructing Decision Rules[A]. Rough Sets, Fuzzy Sets and Knowledge Discovery[C]. 1993,261-273.

[3] 王亚英,邵惠鹤.一种两类决策系统的递增式粗集归纳学习方法[J].信息与控制,2000,29(6):521-525.

[4] 陈云化,叶东毅.基于粗糙集理论的一个增量算法[A].第二届中国 Rough 集与软计算学术研讨会[C].北京:计算机科学出版社,2002.53-55.

[5] 程玉胜.粗集理论约简及其应用的研究[D].合肥:合肥工业大学,2003.

[6] 苏健,高济.基于元信息的粗糙集规则增式生成算法[J].模式识别与人工智能,2001,14(4):428-433.

[7] 吴泉源,刘江宁.人工智能与专家系统[M].北京:国防科技大学出版社,1999.310-326.

[8] 王国胤. Rough 理论与知识获取[M].西安:西安交通大学出版社,2001.

Incremental Learning Algorithm of Induction

CHENG Yu-sheng[1], *ZHANG Chun-mei*[1], *HU Xue-gang*[2]

(1. Mathematics and Computer Department of Anqing Normal College, Anqing 246011, China;
2. College of Computer Science, Heifei Technology University, Heifei 230009, China)

Abstract The algorithm about incremental generating decision function is built by modifying the definition of discernibility matrixes. So it is programmed effectively in the computer. The discernibility matrixes is not generated because the great quantity of data is fetched one after another. The lack of memory is solved by handing data in large database. The algorithm is adapted to the dynamic data and is able to deal with dynamic learning. The time of Data Ming decreases because of information reuse. At the same time, the question of learning about multi-classes is solved in essence by using the incremental generating decision function.

Key words: machine learning; rough sets; discernibility matrixes; decision function

(编辑 吕赛英)

(上接第 26 页)

Optimization of Worm Gearing's Geometric Parameters Under the Condition of the Errors and the Load

XU Wu-jiao, QIN Da-tong, SHI Wan-kai

(State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400030, China)

Abstract: Double enveloping hourglass worm gearing has been widely used in many fields because of its heavy - duty capacity, good lubrication, high transmission efficiency and excellent manufacturing accuracy. Nevertheless, this kind of worm gearing is sensitive to the influence of the error and the load. The effects of important parameters such as inclinable angle β of generating surface, diametral coefficient of worm reference circle k_1, diametral coefficient of base circle k_b and transmission ratio i on the mating performance and loading capacity have been analyzed in a system way. Results show that selection of relatively larger β and k_1 is beneficial to transmission quality. The analysis of worm gearing's mating performance is based on 3 - D solid modeling and tooth contact analysis method(TCA) has been utilized.

Key words: double enveloping hourglass worm gearing; geometric parameters; error; deformation; 3 - D contact FEM

(编辑 成孝义)