一种改进的区分矩阵属性约简算法①

An Improved Algorithm for Attribute Reduction of Discernibility Matrix

李智玲　胡 彧　（山西财经大学 信息管理学院 山西 太原 030031）
（太原理工大学 计算机与软件学院 山西 太原 030024）

摘　要: 属性约简是粗糙集理论研究的关键问题之一。文章以属性在区分矩阵中出现的频率作为启发,对HORAFA 算法做了一些改进。提出了 HORAFA – AFVDM(HORAFA base on Attribute frequency value of discernibility matrix)算法。它是以核为基础,加入属性重要性最大的属性,直到不能再加。为了能找到信息系统的最优约简,在此基础上加了一个反向消除过程,直到不能再删为止。最后在 MATLAB 环境下进行了实验,通过比较改进前后两种算法,表明 HORAFA – AFVDM 算法在属性约简情况和算法运行时间上都比 HORAFA 算法有明显的改进。

关键词: 粗糙集 属性约简 区分矩阵 最优约简 MATLAB

1　引言

粗糙集理论是一种新的数据挖掘方法,波兰科学家 Pawlak 在 1982 年首先提出。属性约简一直是粗糙集理论研究的主要内容之一,数据库中的属性并不是同等重要的。甚至其中某些知识是冗余的,特别是当数据采集没有明确的目的性时,其冗余性更为普遍,通过属性约简,可以去除数据库中的冗余、无用的成分,揭示数据中隐含的规律。从粗糙集理论的角度来理解,在一个决策表中,有些属性对于分类来说多余的,去掉这些属性后,决策表的分类能力不会改变,所以属性约简后仍然反映了一个决策表的本质信息。本文在深入研究 HORAFA 算法的基础上,对其进行了改进,使得该算法一定能找到决策表的最优约简,并给出了算法的实验结果。

2　基本概念

定义 1: 一个信息系统 S 可以表式为:$S = <U, A, V, F>$ 其中,U 为对象的集合,即论域;A 是属性集合,$V = \bigcup_{a \in A} Va$, Va 表示属性 a 的值域;$F: U \times A \rightarrow V$ 是一信息函数,它指的 U 中的每一个对象 x 的属性值,即 $x \in U$, $a_k \in A$,有 $F(x, a_k) \in V_a$ 对于任意 $B \in A$,记 $ind(B) = \{(x, y): F(x, a_k) = F(y, a_k), \forall a_k \in B\}$ 则 $ind(B)$ 是 U 上的等价关系,称为由 B 决定的不可区分关系,它产生的 U 上的划分别为:$U/ind(B) = : \{[x]_B : x \in U\}$,其中:$[x]_B = \{y: (x, y) \in ind(B)\}$,是 x 关于 B 的等价类[1-3]。

定义 2:设信息系统 $S = <U, A, V, F>$, $R \in A$, R 是一族等价关系,$r \in R$,如果:$ind(R) = ind(R - \{r\})$,则称 r 是 R 中不必要的,否则称 r 为 R 中必要的;如果每一个 $r \in R$ 都为 R 中必要的,则称 R 为独立的,否则称 R 为信赖的。设 $Q \in R$,如果 Q 是独立的,且 $ind(Q) = ind(P)$,则称 Q 为论域 U 在属性集 P 上的约简[1]。

定义 3:设信息系统,$S = <U, A, V, F>$, $A = C \cup D$, $C \cap D = \varnothing$, C 中所有的 D 不可省略的元素构成的集合称为 C 的 D 核,记作 $core_D(C)$[1-2]。

定义 4:区分矩阵[1]:给定一个信息系统 $S = <U, A, V, F>$, $A = C \cup D$ 是属性集合,C, D 分别是条件属性和决策属性。区分矩阵 定义为:$M = (m_{ij})$ 定义为:

① 基金项目:太原理工大学 211 青年资金资助项目(100231 – 10991);山西省高科技资助项目(101049)

$$m_{ij}=\begin{cases}\{a\in c:a(xi)\neq a(xj)\} & \text{当 } D(xi)\neq D(xj)\\ \varnothing & \text{当 } D(xi)=D(xj)\end{cases}$$

其中 a(x)是元组 x 在属性 C 上的取值,D(x)是 x 在决策属性 D 上的取值[1-2]。

3 HORAFA 算法

清华大学的胡可云博士提出了 HORAFA 算法。这个算法利用属性在区分矩阵中出现的频率作为启发,从而对信息系统进行约简。它完全摆脱了对原来的信息系统的依赖,仅仅利用区分矩阵中属性出现的频率来计算。因为不用计算粗糙集的基本概念,而仅仅使用简单的计算,所以提高了效率。

3.1 算法原理

一个约简和区分矩阵的每一项的交都不能为空。如果它和区分矩阵的某项 cij 的交为空的话,对象 i 和对象 j 对于该约简就是不可区分的了。这和约简能够区分所有对象的最小属性集合相矛盾(这里假定数据集是一致的)。

利用这个事实算法可以这样设计。首先设候选约简集合 R=φ。检查区分矩阵的每一项 cij 和候选约简集合 R 的交。如果交为空,随机从 cij 中选择一个属性插入到 R 中;否则就跳过这一项。重复这一过程,直到区分矩阵的每一项都检查过了。这时候,我们在 R 中得到的就是一个“约简”。

此算法很简单而且意义明确。但是在大多数情况下我们得到将是约简的超集而不是约简本身。例如,假定在区分矩阵中有这样三项:{a1,a3},{a2,a3},{a3}。根据此算法,我们可能会得到这样一个候选约简集合 {a1,a2,a3} 或者 {a1,a3}。尽管很明显 {a3} 是唯一的约简。为什么会这样呢?答案是这个事实是约简的必要而非充分条件。对于约简来说,它还必须是满足这些条件中最小的一个。而上面的算法没有考虑到这一点。为了找到约简,尤其是最小约简,我们需要更强有力的的启发知识。

一个简单而有效的方法是根据|cij|来对区分矩阵排序。这样就能正确的解出上面的例子了。我们知道,如果 cij 中只有一个属性,该属性一定是约简的成员。我们因此可以想象,区分矩阵某项的长度越短,该项就对分类所起的作用就越大。而且该项出现的越频繁,该项就越重要。因此我们对区分矩阵排序时,除了按长度外,在长度相同的情况下,出现频率高的项就排在前面。

但是,如果在区分矩阵的某项和当前的候选集的交为空,并且该项有多于一个的属性组成的时候,如何知道优先选择哪个属性呢?当生成区分矩阵的时候,每个属性出现的频率同时被记录,供以后使用。用这些频率来评估属性的重要性,并用于属性优先级的选择。这一点是基于对一个属性出现的越频繁,它的潜在区分能力就越大来考虑的。在计算属性出现的频率时 并不是简单的计数,而是要加权的。加权大小的根据依然是属性所出现在区分矩阵项中的长度。

每当计算出一个新的区分矩阵项 c,相应的属性频率函数 f(a)就更新为:

f(a)=f(a)+|A|/|c|,对于每个 a∈c;

其中|A| 是信息系统总的条件属性个数。例如,设 f(a1)=3,f(a3)=4,系统总共有 10 个属性,新的区分函数项是 {a1,a3}。则处理这项后,属性的频率被更新为:f(a1)=3+10/2=8;f(a3)=4+10/2=9。这充分体现了两个重要的启发式思想:

(1)属性在区分矩阵中出现的次数越多,该属性的重要性越大。

(2)属性所出现的区分矩阵的项越短,属性的重要性越大。

3.2 算法描述

输入:信息系统 (U,A∪{d}),其中 A=∪ai,i=1,...,n.

输出:约简 red

(1) Red=φ,count(ai)=0,对于 i=1,...n.;

(2) 计算区分矩阵 M 并同时计算属性的加权频率 count(ai);

(3) 合并相同的项并且按照每项的长度和频率对区分矩阵排序;

(4) For M 中的每项 m do

(5) If (m∩Red = =φ)

(6) 选择 m 中的具有最大的 count(a)的属性 a

(7) Red=Red∪{a}

(8) Endif

(9) EndFor

(10) Return Red

在第二行中,每计算 M 的一个新项 c,count(ai)就更新为 count(ai):= count(ai) + n/|c|,其中 ai(|c|。

3.3 HORAFA 算法的时间复杂度分析

区分矩阵 M 中有|U|(|U|-1)/2 项,因此计算区分矩阵的代价是 O(|A| $|U|^2$)。排序算法需要的时间复杂度为 O(2$|U|^2$log(|U|))。循环计算中,区分矩阵最多有|U|(|U|-1)/2 项,每项最多包含|A|个属性,时间复杂度为 O(|A| $|U|^2$)。所以整个算法的时间复杂度为 O((|A| + log|U|) $|U|^2$)。

4 改进的 H0RAFA 算法:HORAFA - AFVDM

实验证明[4] HORAFA 算法并不是一定能够找到信息系统的最优约简或者是较优约简,甚至是约简。为了便于说明问题,我们先看下面一个例子。

下表 4-1 所示为一信息系统:

表 4-1 信息系统

	a	b	c	d	e	f	k	y
1	1	1	1	1	1	1	1	1
2	1	0	0	0	1	1	1	2
3	0	0	1	1	1	1	1	2
4	1	1	1	1	1	1	0	2
5	0	0	1	1	1	1	0	2
6	0	1	1	1	0	1	0	2
7	0	1	1	1	1	0	0	2

其中 a,b,c,d,e,f,k 是条件属性,Y 是决策属性。

HORAFA 算法的运算结果是(k,a,b)。然而,通过简单的计算,我们可以发现,(k,a,b)根本就不是这个信息系统的约简。而这个信息系统的约简应该是(k,b),它也是最优约简。HORAFA 算法的运算过程如下所示。

根据 HORAFA 算法有:

count (a) =l0. 5,count (b) =8.17,count (c) =2. 33,count (d) =2. 33,count (e) =2. 33, count (f) =2. 33,count (k) =14 于是,可得到排序后的区分矩阵是 {k,ab,bcd,abk,aek, afk};然后对区分矩阵进行循环,依次加入属性 k,a,b,于是得到的约简就是:(k,a, b)。

下面分析 HORAFA 算法没有找到约简的原因。

一个属性之所以对信息系统重要,是因为这个属性对我们区分某些对象提供了帮助。但是属性对系统的重要性是有重复的,属性之间的重复性越大,对系统贡献的重复性也就越大。比如说,信息系统中某一项为 abc,则属性 a 属性 b 和属性 c,对系统的重要性就是重复的。这三个属性中任何一个都可以区分这一项所对应的两个对象。如果信息系统已经需要属性 a,则对于这一项来说,我们已经可以区分,那么,b 和 c 便对这一项不起作用。然而,属性频率函数却对这一项的 b 和 c 也进行了计算。在当前的这种情况下,就属性频率函数对整个信息系统的重要性而言,属性频率函数大的未必就比属性频率函数小的属性重要性大。

基于以上因素,同时为了得到信息系统的最优约简,文中对 HORAFA 算法提出了以下两点的改进,提出了 HORAFA - AFVDM (HORAFA base on Attribute frequency value of discernibility matrix)算法:

(1)属性频率函数等于区分矩阵中删除当前属性所在的元素之后,属性出现的频率,属性频率函数表示为 f(a) = f(a) + |A|/|c'|,对于每个 a(c,其中|A| 是信息系统总的条件属性个数,|c'|为区分函数项中删除加入到核中的属性之后还剩的属性个数;

(2)以核为基础,加入属性重要性最大的属性,直到不能再加。在此基础上,加上一个反向消除过程,在已得到的 core 中删除可以删除的属性,直到不能再删为止,保证了算法的完备性。

4.1 算法描述

输入:信息系统 S = (U,A,V,f),其中 U 为论域,A 为属性集,A = C ∪ D,C ∩ D = Φ,C 为条件属性集合,D 为决策属性集合

输出:约简 red

(1) red = Φ,core = Φ,count(ai) = 0,(i =1,2, …n),n = 0;

(2) 计算区分矩阵 M,合并矩阵中相同的项,同时

将只有一个元素的属性加入 core，即：core = core ∪ {ai}，n + +

（3）red = core，删除 M 中包含核的所有项，计算 count(ai)，同时计算 |M|

（4）if |M| ≠ ∅

（5）将 count(ci) 最大的 ai 加 red，n + +，删除 M 中包含 ai 的所有项，重新计算 count(ai)，并计算 |M|

（6）endif

（7）if $POS_{red-ai}(D) \neq POS_C(D)$

（8）保留 ai，否则删除 ai，n - -

（9）endif

（10）输出：red

4.2 算法的时间复杂度分析

区分矩阵 M 有 |U|(|U| -1)/2 项，因此计算区分矩阵的代价是 O(|A| |U|²)。循环计算中，区分矩阵最多有 |U|(|U| -1)/2 项，每项最多包含 |A| 个属性，时间复杂度为 O(|A| |U|²)，在第七步中首先计算分类的时间复杂是 O(|U|²) 共有 n 属性，那么时间复杂度是 O(2n|U|²) 所以整个算法的时间复杂度为 O((|A| +n) |U|²)。

改进后的算法与改进前的相比：在计算区分矩阵的同时，计算信息系统的核，并且计算约简是在核的基础上进行的，同时不用对区分矩阵进行排序，将某个属性加入约简后，便删除包含该属性的元素，使区分矩阵大大减小，所以这提高了算法的效率；每加入一个属性，都要重新计算属性的重要性，这样就使得每次加入的属性都是当前情况下属性最重要的属性，从而更能够能够找到约简。

5 实验结果

算法测试的硬件环境是运行 windowsXP 的个人计算机。该计算机使用 Celeron 1.00GHz 处理器并装有 256M 内存。软件环境是 MATLAB7.0，SQL SERVER2000。

文中使用 UCI 数据集测试算法，这些数据集是 SGI 经过整理后用于 MLC 的数据集。对于具有连续属性的数据集，首先使用 MLC 工具（Kohavi，Sommerfield and Dougherty，1996）中的离散化工具用熵方法进行离散化。其约简结果如表 5 -1，其中 Red1 表示 HORAFA 算法约简的属性个数；Red2 表示 HORAFA - AFVDM 算法约简的属性个数；Time1 表示 HORAFA 算法的运行时间；Time2 表示 HORAFA - AFVDM 算法的运行时间；h 表示小时；m 表示分钟；s 表示秒。

表 5 -1　属性约简结果

数据集	实例数	条件属性	Red1	Red2	Time1	Time2
auto_mpg	392	7	3	2	1h24m 57.047s	26m 3.516s
ctr	21	9	7	4	0.11s	0.016s
Echocardi - ogram	61	8	1	1	1.234s	0.328s
flag	194	27	14	6	16m 26.36s	13m 30.375s
glass	214	9	2	2	14m 25.078s	5m 19.843s
heart	62	16	6	4	10.89s	3.344s
heart_ statlog	270	13	3	3	17m 58.046s	8m 54.625s
house_ votes	226	16	16	8	9m 29.437s	5m 28.156s
iris	150	4	3	3	2m 26.766s	14.375s
ionosphere	351	34	3	3	1h1m 47.391s	58m 27.047s
lymph	148	18	8	6	1m 58.609s	1m 10.359s
mushromm	352	22	5	4	9m 7.922s	6m 39.797s
machine	209	7	4	3	6m 51.329s	1m 51.75s
promoters	106	57	5	4	50.704s	57.625s
solar	333	10	9	7	9m 0.078s	6m 1.745s
tic - tac - toe	201	9	1	1	4m 26.469s	1m 26.703s
vehicle	150	18	2	2	4m 7.453s	2m 33.906s
zoo	101	16	7	5	49.75s	21.703s

从表 5 -1 中可以看出改进后的算法对属性的约简能力均比改进前的算法强，即使针对个别数据集，改进前后两种算法对其属性约简个数是相同的，但整个算法的运行时间得到了大大的改进，就比如 glass、iris 等数据集。除 promoters 数据集外，针对其它数据集，

改进后算法的运行时间都不同程度的得到了改善。对于 promoters 数据集之所以改进后的算法运行时间比改进前的多了 7 秒多，是因为这个数据集本的的实例数不多，而属性个比较多。通过比较这两个算法的时间复杂度得出这样的结果是不为过的。

5.1 属性约简比较

从表 5 -1 中看出改进后的算法约简属性的个数均小于等于改进前的算法约简的个数，从而使数据集得到了最小的属性约简，即最优约简。

现在取 auto_mpg、ctr、solar、heart、lymph、mushroom、flat 七个数据集，针对改进前后的两个算法运行结果，对它们约简属性进行比较，结果如图 5 -2 所示。从图中可以看出 HORAFA - AFVDM 算法的属性约简效率均高于 HORAFA 算法。对于 auto_mpg、ctr、solar、heart、lymph、mushroom、flat 这七个数据集，经计算约简个数分别别减少了 1、3、2、2、2、1、8，约简效率分别提高了 14% 、33% 、20% 、12.5% 、11% 、4.5% 、30% 。从计算结果看出对 ctr 数据集的约简率提高最大，HORAFA 算法对其的约简率是 22% ，HORAFA - AFVDM 算法对其的约简率为 55% 。那么这样改进后算法的约简率就提高了 33% 。

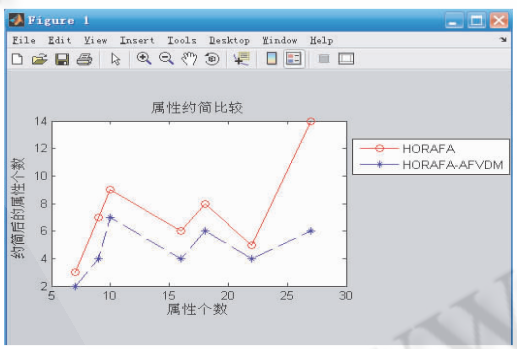

图 5 -2 属性约简比较

5.2 算法运行时间比较

取 vehicle、iris、solar、machine、lymph、mushroom、flat 七个数据集为算法运行时间的比较对象，在运行时间比较图中横坐标用 1、2、3、4、5、6、7 分别代表 vehile、iris、solar、machine、lymph、mushroom、flat 这七个数据集，比较结果如图 5 -3 所示。

从图中可以看出 HORAFA - AFVDM 算法的运行时间均少于 HORAFA 算法。对于 vehile、iris、solar、machine、lymph、mushroom、flat 这七个数据集，经计算运行时间分别减少了 1m33.547s、2m12.391s、2m58.33s、4m59.579s、48.25s、2m28.125s、2m55.985s。运行时间率分别提高了 37.8% 、90% 、33% 、72.8% 、40.7% 、27% 、17.8% 。从计算结果看出对 machine 数据集的运行时间减少最大，减少了 4m59.579s。对运行时间率提高最大的是 iris 数据集，提高了 90% 。

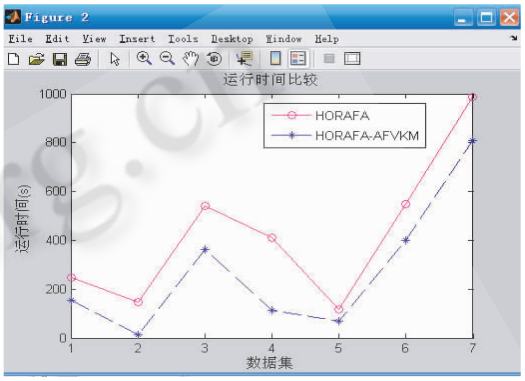

图 5 -3 运行时间比较

6 结论

本文我们提出一种有效的启发式最优约简算法。此算法利用区分矩阵中项的长度和每个属性的频率信息，并根据此对属性频率进行加权，属性频率函数等于区分矩阵中删除当前约简之后的属性出现的频率，用来进行属性的选择，之后加了一个反向消除的过程式。此算法不仅能找到信息系统的约简，而且能找到其最优约简，实验表明该算法的结果优于相关算法。

参考文献

1 张文修，吴伟志，梁吉业等. 粗糙集理论与方法. 北京：科学出版社，2005.

2 刘清. Rough 集及 Rough 推理. 北京：科学出社，2001.

3 Paw lak Z. Rough sets. International Journal of Computer and Information Science，1982，11（5）：341 - 356.

4 胡可云. 基于概念格和粗糙集的数据挖掘方法研究[博士学位论文]. 北京：清华大学，2001.

5 Wang Jue，Wang Ju. Reduction algorithms based on discernibility matrix：the ordered attributes method. Journal of Computer Sc ience and Technology，2001，16（6）：489 - 504.