

基于属性层次的规则挖掘方法研究

杨伟 苗夺谦 卫志华

(同济大学电子与信息工程学院计算机科学与技术系 上海 201804)

(同济大学嵌入式系统与服务计算教育部重点实验室 上海 201804)

摘要 针对决策表中有隐含属性层次的决策表,讨论了属性层次树,定义出属性层次决策表,提出一个算法来对属性层次决策表进行规则挖掘,并通过实例分析了算法的有效性。实验结果表明,高属性层次决策表的对象数明显少于原决策表对象数。在对象数明显减少的高属性层次决策表上的约简运算量比直接在原决策表上的要减小很多。

关键词 属性层次树,属性层次决策表,粗糙集,属性约简

中图分类号 TP391 文献标识码 A

Research on Method of Rules Mining Based on Attributes Hierarchy

YANG Wei MIAO Duo-qian WEI Zhi-hua

(Department of Computer Science and Technology, School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

(Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China)

Abstract Based on the attributes hierarchy decision table, the attributes hierarchy trees were introduced and the concept of attributes hierarchy decision table was defined. An algorithm was proposed for the rules mining in the attributes hierarchy decision table and numerical examples were employed to substantiate this algorithm. And the experiment indicates that the number of objects in attributes hierarchy decision table is less than original decision table, and the time of implementing attributes reduction is less than that.

Keywords Attributes hierarchy tree, Attribute hierarchy decision table, Rough sets, Attributes reduction

1 引言

粒计算是一种世界观与方法论,在数据处理、机器学习与规则挖掘等领域具有重要的作用。Zadeh^[1-4]文中提出了信息粒与模糊信息粒。信息粒的概念已经应用到许多领域中,例如:粗糙集、模糊集等。接下来,一些与粒计算相关的工作^[5-16]逐渐提出。T. Y. Lin^[5]于1997年首次提出粒计算这一术语。Yao^[6]对粒计算的基本问题与可能的问题求解方法进行了讨论。Zadeh^[7]在2007年讨论了粒计算与粗糙集中的粒化问题。Pawlak^[8]1982年提出了粗糙集。粗糙集主要是用来处理不精确、不一致、不完备等信息的。张铃、张钺^[9]认为人类的智能表现在人可以从不同粒度(Granularity)上观察和分析同一问题,提出了商空间理论,并把商空间理论推广到了模糊粒度世界,给出了两个重要的结论^[10]。张铃、张钺^[11]在文中深入浅出地介绍了粒计算未来的发展方向。张铃、张钺^[12]在文中为解决动态环境下的问题求解,引入了时间变量把原来的商空间模型扩展为动态商空间模型。李等^[13]对粒度计算方法进行了综述。Yao^[14,15]介绍了值集与粒度上的度量。

在实际处理问题中,我们会从不同的角度去观察问题,比

如:对于一个人的年龄,可以直接是数字上的表示(1~200),也可以表示为幼儿、少年、青年、中年、老年。这样,不自觉地加入了先验知识,从不同角度、不同层次上分析问题,可以很好地反映数据的特征,根据需求合理地分析。粒计算的思想就是要从多角度、多层次上分析问题。粒计算的一个重要模型——粗糙集,其处理信息表或决策表时,所用到的数据有时是具体的数据,比如:一个人的工资,是具体的数字。经典粗糙集只是从这一层次去处理问题,没有使用数据中隐含的关系,没有在具体数据上进一步抽象,这样处理问题时所需要的时间就比较多。因此,关于这方面的工作就开始了。对经典粗糙集中的论域进行粒度化^[17],通过调整属性的个数,形成一个嵌套的属性队列,可以对论域进行不同层次的分类,得出不同大小的粒。Hu^[18,19]在文中研究了最大一般化规则。Shan^[20]在文中提到了信息泛化、约简以及规则生成。Wei^[21]在文中提出了基于粗糙熵的约简算法,并分析了属性的层次。Zhang^[22]在文中构造决策树与设计分类器时分析了属性值上隐含的层次性。Feng^[23]在文中基于属性值的层次性提出了层次的决策表,并对不同层次进行了数据挖掘。

基于粒计算理论,我们发现处理实际问题中有一类决策表,它的属性间隐含着层次性,即加入先验知识的情况下,可

收到日期:2012-12-27 返修日期:2013-03-11 本文受国家自然科学基金项目(60970061,61075056,61103067),中央高校基本科研业务费专项资金资助。

杨伟 博士生,主要研究方向为模式识别与智能系统、粗糙集理论、粒计算,E-mail: yangw312@yahoo.com.cn; 苗夺谦 教授,博士生导师,主要研究方向为人工智能、模式识别、知识发现、粗糙集理论等。

以生成属性层次树,基于属性层次树我们可以研究高属性层次的决策表。本文主要定义出属性层次决策表,提出一个算法对属性层次决策表进行规则挖掘。在给定一个属性层次决策表后,我们通过算法可以挖掘出任意给定的属性层次的规则。通过实例分析了算法的可行性,得出高属性层次决策表可以明显地减少对象的个数,比直接在原决策表上求约简更容易。

本文第2节介绍信息系统、决策表,并给出属性层次决策表的定义;第3节介绍一种算法,对怎样生成高属性层次决策表以及在高属性层次决策表上挖掘规则的方法进行分析;第4节通过一个实验对算法1进行实际情况分析,证明高属性层次决策表可以有效地减少对象的个数,在后面的约简以及规则生成中有很好的优势。

2 基于属性层次的粗糙集理论和方法

为了讨论方便,这里给出信息系统与决策表的概念。

信息系统 $\zeta = (U, AT, V, f)$, 其中 U 是一个有限非空的对象集, AT 是一个有限非空的属性集, V_a 是属性 a 的值域, $V = \bigcup_{a \in AT} V_a$ 是属性值的集合, f 是一个信息函数, $f: U \times AT \rightarrow V$ 。

决策表 $\zeta = (U, AT \cup D, V, f)$, 其中 U 是一个有限非空的对象集, AT 是一个有限非空的属性集, D 是一个有限非空的决策属性集, V_a 是属性 a 的值域, $V = \bigcup_{a \in AT \cup D} V_a$ 是条件属性值与决策属性值的集合, f 是一个信息函数, $f: U \times AT \cup D \rightarrow V$ 。

决策表 $\zeta = (U, AT \cup D, V, f)$, 对任意 $x, y \in U$, 如果这两个对象所有条件属性的取值相等, 并且所有决策属性的取值也相等, 则称这个决策表是一致的; 如果这两个对象所有条件属性的取值相等, 并且只要有一个决策属性的取值不相等, 就称这个决策表是不一致的。

例1 表1是一个属性间隐含层次的决策表。表中数据来源于UCI数据集的部分数据, 数据集名称为汽车评估数据集, 作者是Marko Bohanec, 提供者是Marko Bohanec与Blaz Zupan。

表1 决策表

CAR	buying	maint	doors	persons	lug_boot	Safety	Acceptability
1	vhigh	vhigh	2	2	small	low	unacc
2	vhigh	vhigh	5more	more	big	high	unacc
3	vhigh	high	2	2	small	low	unacc
4	vhigh	high	2	2	small	med	unacc
5	high	vhigh	5more	more	small	low	unacc
6	high	vhigh	5more	2	small	high	unacc
7	vhigh	med	3	more	med	med	acc
8	vhigh	med	3	more	med	high	acc
9	low	low	4	more	small	high	good
10	vhigh	med	3	more	big	med	acc
11	vhigh	med	3	more	big	high	acc
12	med	med	4	4	med	high	vgood
13	med	med	4	4	big	low	unacc
14	med	med	4	4	big	med	acc
15	med	med	4	4	big	high	vgood
16	low	low	2	4	big	med	good
17	low	low	2	4	big	high	vgood
18	med	low	2	more	med	med	acc
19	low	low	5more	more	big	med	good
20	low	low	5more	more	big	high	vgood

在一个决策表中,通过预先给定的先验知识,属性中可能存在某些属性对应一个层次树。我们这里在预先给定的先验知识下,定义关于属性的层次树。

例2 给定表1,通过先验知识给出属性层次树,并对所有非叶结点进行标记,如图1所示。

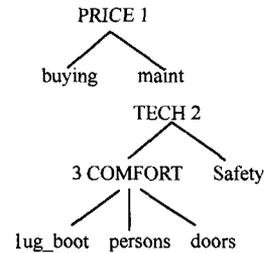


图1 属性层次树

如例2所示,在给定先验知识的情况下,一个决策表中的属性有可能形成属性层次树。

定义1 $\zeta = (U, AT \cup D, V, f)$ 是一个决策表, 给定属性层次树, 设 B 是所有属性层次树中非叶结点的集合, 则:

(1) 如果 B 为空集, 则决策表中所有属性无法形成属性层次树, 不能抽象到更高的层次。

(2) 如果 B 为非空集, 则有 $C \subseteq B$, 我们可以使用集合 C 中的元素去替换它们各自的叶子结点, 以形成新的决策表, 属性层次决策表如下:

$$\zeta_C = (U_C, AT_C \cup D_C, V_C, f_C)$$

式中, U_C 为对象集, AT_C 是条件属性集, D_C 是决策属性集, V_C 是所有条件属性与决策属性的值域, $f_C: U_C \times AT_C \cup D_C \rightarrow V_C$ 。

对于一个属性层次决策表 $\zeta_C = (U_C, AT_C \cup D_C, V_C, f_C)$, 有如下讨论:

(1) 如果 $C = \emptyset$, 则 ζ_C 称为初始决策表;

(2) 如果 $C = B$, 则 ζ_C 称为最高决策表。

定义2 $\zeta = (U, AT \cup D, V, f)$ 是一个决策表, 给定属性层次树, 设 B 是所有属性层次树中非叶结点的集合, $x \in U$, $C \subseteq B, D \subseteq C, \zeta_C = (U_C, AT_C \cup D_C, V_C, f_C), \zeta_D = (U_D, AT_D \cup D_D, V_D, f_D)$, 则有: 若对象 $x \in U_C$, 一定存在 $x \in U_D$, 记为 $U_C \leq U_D$ 。

定义3 $\zeta = (U, AT \cup D, V, f)$ 是一个决策表, 给定属性层次树, 设 B 是所有属性层次树中非叶结点的集合, $x \in U, C \subseteq B, D \subseteq B, \zeta_C = (U_C, AT_C \cup D_C, V_C, f_C), \zeta_D = (U_D, AT_D \cup D_D, V_D, f_D)$, 当 D 中所有元素都是 C 中元素的子结点或相同时, 则有: 若对象 $x \in U_C$, 一定存在 $x \in U_D$, 记为 $U_C \leq U_D$ 。

定义4 $\zeta = (U, AT \cup D, V, f)$ 是一个决策表, $A \subseteq AT, A$ 是一个约简当且仅当:

$$POS_A(D) = POS_{AT}(D)$$

$$\forall a \in A, POS_{A-(a)}(D) \neq POS_{AT}(D)$$

3 规则挖掘

这一小节,我们将基于属性层次树,提出相应的算法对决策表进行规则挖掘。

算法1

输入: 一个给定的决策表 $\zeta = (U, AT \cup D, V, f)$, 给定属性层次树, B

是所有属性层次树中非叶结点的集合, $C \subseteq B$ 。

输出: 给定决策表 ζ_c 的规则。

1. 使用集合 C 中的属性去替换相应叶子结点的属性, 新属性的属性值由其叶子结点的属性的属性值合成, 得到新的决策表 ζ_1 。
2. 对决策表 ζ_1 因提升属性层次而可能出现的重复对象进行删除, 生成决策表 ζ_c , 具体操作如下:

- (1) 生成决策表 ζ_1 的所有属性的划分。
- (2) 给定决策表 ζ_1 , 条件属性的集合 A , 决策属性集合 D , 令 $B=A$ 。
- (3) 任意 $b \in B$, 令 $B=B-\{b\}$ 。
- (4) 令属性 b 的划分中的每个元素, 即对象集, 与 B 中任意属性 c 的划分中的每个元素求交集。当所得所有交集的基数都小于等于 1 时, 则算法结束; 当所得交集中有大于 1 的时, 所有大于 1 的交集记为 $U_1 U_2 \dots U_n$ 。
- (5) 令 $B=B-\{c\}$, 若 B 为空集, 则表示 $U_1 U_2 \dots U_n$ 中的对象为重复对象, 对决策表删除每个交集中的决策属性取值相等的对象, 直到只剩下一个对象或决策属性值不相等为止, 并在编号中记录, 则算法结束; 若 B 不为空集, $U_1 U_2 \dots U_n$ 与 B 中任意属性 c 的划分中的每个元素求交集:
 - 当所得交集中有大于 1 的时, 所有大于 1 的交集记为 $U_1 U_2 \dots U_n$, 则返回第(5)步;
 - 当所得所有交集的基数都小于等于 1 时, 则算法结束。

3. 求得属性约简。

4. 生成决策规则。当出现不一致时, 可生成确定性规则与不确定规则, 并为不确定规则给定支持率, 支持率为任意不确定规则支持集的基数除以冲突规则支持集的基数之和。

下面给出一个示意性的例子, 使用表 1 中的决策表, 其中有 20 个对象, 6 个条件属性, 1 个决策属性。

例 3 给定表 1, 基于此表给出属性层次树, 如图 1 所示。根据属性层次树 $B=\{1, 2, 3\}$, 给定 $C=\{1, 3\}$ 。

(1) 使用集合 C 中的属性去替换相应叶子结点的属性, 这里需要给定合成标准。这里是根据经验所得, 不是唯一标准, 也可以是其它合理标准。下面合成标准类同。

buying 与 maint 合成为 PRICE, 合成标准:

vhigh 与 vhigh, high 合成为 vhigh;

vhigh 与 med 合成为 high;

vhigh 与 low 合成为 med;

high 与 high, med 合成为 high;

high 与 low 合成为 med;

med 与 med, low 合成为 med;

low 与 low 合成为 low;

doors, persons 与 lug_boot 合成为 COMFORT, 合成标准, 统一 3 个属性的取值如下:

doors: 2-small, 3, 4-med, big;

persons: 2-small, 4-med, big;

lug_boot: small, med, big;

big, big, big 合成为 big;

big, big, med 合成为 big;

big, big, small 合成为 med;

big, med, med 合成为 med;

big, med, small 合成为 med;

big, small, small 合成为 med;

• 200 •

med, med, med 合成为 med;

med, med, small 合成为 med;

med, small, small 合成为 small;

small, small, small 合成为 small。

对表 1 进行算法中第 1 步提升属性层次操作, 可以得出高层属性决策表 ζ_1 ; 对 ζ_1 进行算法中第 2 步删除重复对象操作, 可以得出高层属性决策表 ζ_c , 如表 2 所列, 其中对象 3, 15, 18 被删除。加粗对象 9 与 17 不一致。

表 2 高属性决策表

CAR	PRICE	COMFORT	Safety	acceptability
1,3	vhigh	small	low	unacc
2	vhigh	big	high	unacc
4	vhigh	small	med	unacc
5	vhigh	med	low	unacc
6	vhigh	med	high	unacc
7	high	med	med	acc
8	high	med	high	acc
9	low	med	high	good
10	high	big	med	acc
11	high	big	high	acc
12,15	med	med	high	vgood
13	med	med	low	unacc
14,18	med	med	med	acc
16	low	med	med	good
17	low	med	high	vgood
19	low	big	med	good
20	low	big	high	vgood

对表 2 求属性约简可得: PRICE, COMFORT, Safety。

生成的确定性规则如表 3 所列, 生成的不确定性规则如表 4 所列。

表 3 确定性规则集

决策规则	支持集
1 如果 $P(vhigh)$, 则 unacc	{1,2,3,4,5,6}
2 如果 $S(low)$, 则 unacc	{5,13}
3 如果 $P(high), C \geq (med), S \geq (med)$, 则 acc	{7,8,10,11}
4 如果 $P(med), C(med), S(med)$, 则 acc	{14,18}
5 如果 $P(low), C \geq (med), S \geq (med)$, 则 good	{16,19}
6 如果 $P \leq (med), C \geq (med), S(high)$, 则 vgood	{12,15,20}

表 4 不确定性规则集

决策规则	支持集	支持率
1 如果 $P(low), C(med), S(high)$, 则 good	{9}	0.5
2 如果 $P(low), C(med), S(high)$, 则 vgood	{17}	0.5

4 实验结果与分析

在这一节中, 我们对汽车评估数据集提升属性层次, 并生成决策表。汽车评估数据集是在 UCI 数据集中找到的。其中对象数为 1728; 条件属性数为 6; 决策属性数为 1; 没有缺失数据, 即决策表为完备决策表。

对汽车评估数据集进行属性层次提升, $B=\{1, 2, 3\}$, 给定 $C_1=\{1\}, C_2=\{3\}, C_3=\{2\}, C_4=\{1, 3\}, C_5=\{1, 2\}$, 因为结点 3 是结点 2 的子结点, 所以当有结点 2 时, 已经隐含结点 3 上的属性层次提升。对上面 C_1 到 C_5 依次提升属性层次, 生成新的决策表, 不同的决策表对象数与不一致对象数如表 5 所列, 属性间的关系如图 2 所示。计算中合成为属性 TECH 时, 需要统一属性 COMFORT 与属性 Safety 的取值以及设定属性值合成标准, 如下:

COMFORT; small-1, med-2, big-3;

Safety; low-1, med-2, high-3;

1,1,合成为 1; 1,2,合成为 1;

1,3,合成为 2; 2,2,合成为 2;

2,3,合成为 2; 3,3,合成为 3。

表 5 高层属性决策表对象数与不一致对象数

	原对象数	剩余对象数	不一致对象数	W
C ₁	1728	363	76	0.209
C ₂	1728	196	96	0.490
C ₃	1728	79	54	0.684
C ₄	1728	55	32	0.582
C ₅	1728	24	19	0.792

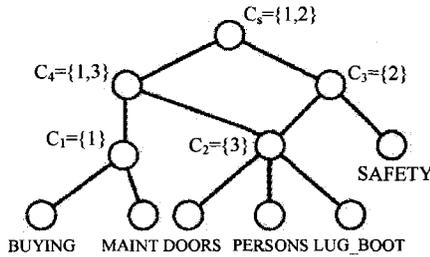


图 2 属性间的关系

使用 C₁ 与 C₂ 分别提升层次时,两者之间没有相互重叠或嵌套部分;使用 C₃ 提升层次时,因为结点 3 是结点 2 的子结点,所以提升过程中已经使用 C₂ 提升层次,因此,只提升 C₃ 层次后的对象数一定小于只提升 C₂ 层次后的对象数,即有 79 小于 196。同样,只提升 C₄ 层次时,由于 C₄ 中包含了 C₁ 与 C₃,因此,从表 6 中可以看出,只提升 C₄ 层次后的对象数一定小于只提升 C₁ 或 C₃ 层次后的对象数,即有 55 小于 363,55 小于 79。只提升 C₅ 层次时,由于提升时包含了所有的属性层次,因此只提升 C₅ 层次后的对象数一定是最小的。表 5 中显示,高属性层次决策表中的对象数明显少于原决策表的对象数,在后面求约简以及生成规则对比于原决策表要更加容易。

提升 C₅ 层次后生成的决策表,我们规定决策属性取值为 acc, good, vgood 时,可以用 ACC 表示,删除重复对象 7 个。其中一致对象如表 6 所列,求约简可得:PRICE, TECH。

表 6 高属性一致决策表

CAR	PRICE	TECH	acceptability
1	vhigh	low	unacc
2	vhigh	med	unacc
3	vhigh	high	unacc
4	high	low	unacc
5	high	high	ACC
6	med	high	ACC
7	low	high	ACC

生成的规则如表 7 所列。

表 7 确定性规则集

	决策规则	支持集
1	如果 P(vhigh), 则 unacc	{1,2,3}
2	如果 P(high), TECH(low), 则 unacc	{4}
3	如果 P(high), TECH(high), 则 ACC	{5}
4	如果 P(med), TECH(high), 则 ACC	{6}
5	如果 P(low), TECH(high), 则 ACC	{7}

不一致对象如表 8 所列。

表 8 高属性不一致决策表

CAR	PRICE	TECH	acceptability
1	high	med	unacc
2	high	med	ACC
3	med	low	unacc
4	med	low	ACC
5	med	med	unacc
6,7,8	med	med	ACC
9	low	low	unacc
10	low	low	ACC
11	low	med	unacc
12,13,14	low	med	ACC

生成的规则如表 9 所列。

表 9 不确定性规则集

	决策规则	支持集	支持率
1	如果 P(high), TECH(med), 则 unacc	{1}	0.5
2	如果 P(high), TECH(med), 则 ACC	{2}	0.5
3	如果 P(med), TECH(low), 则 unacc	{3}	0.5
4	如果 P(med), TECH(low), 则 ACC	{4}	0.5
5	如果 P(med), TECH(med), 则 unacc	{5}	0.25
6	如果 P(med), TECH(med), 则 ACC	{6,7,8}	0.75
7	如果 P(low), TECH(low), 则 unacc	{9}	0.5
8	如果 P(low), TECH(low), 则 ACC	{10}	0.5
9	如果 P(low), TECH(med), 则 unacc	{11}	0.25
10	如果 P(low), TECH(med), 则 ACC	{12,13,14}	0.75

结束语 本文主要工作是对存在有属性层次的决策表进行研究,定义出属性层次决策表,并对其上的规则挖掘提出算法,可以挖掘出任意给定属性层次的规则,通过实验分析了算法的有效性。但在提升属性层次后,可能会出现不一致对象。在进一步工作中,我们需要对因为提升属性层次而出现的不一致对象进行分析处理,看是否可以既提升层次,又可以不出现在不一致对象。本文工作虽然只是在决策表上讨论,但一样适用于信息表,可以扩展到信息表中。

参考文献

- [1] Zadeh L A. Fuzzy sets[J]. Advances in Fuzzy Information and Control, 1965, 8: 338-353
- [2] Zadeh L A. Fuzzy sets and information granularity[M]// Gupta M, Ragade R, Yager R, eds. Advances in Fuzzy Set Theory and applications. Amstrdam: North-Holland publishing, 1979: 3-18
- [3] Zadeh L A. Fuzzy logic-computing with words [J]. IEEE Transaction on Fuzzy System, 1996, 4(2): 103-111
- [4] Zadeh L A. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic[J]. Fuzzy Sets and Systems, 1997, 90: 111-121
- [5] Lin T Y. Granular computing[R]. Announcement of the BISC special interest group on granular computing, 1997
- [6] Yao Y Y. Granularity computing: basic issues and possible solutions[C]// Proceedings of the 5th Joint Conference on Information Sciences, 2000: 186-189
- [7] Zadeh L A. Granular computing and rough set theory[C]// LNCS. 2007, 4585: 1-4
- [8] Pawlak Z. Rough sets[J]. International Journal of Information and Computer Sciences, 1982, 11: 341-356
- [9] 张钹, 张铃. 问题求解理论及应用[M]. 北京: 清华大学出版社, 1990
- [10] 张铃, 张钹. 模糊商空间理论(模糊粒度计算方法)[J]. 软件学报, 2003, 14(4): 770-776
- [11] 张铃, 张钹. 粒计算未来发展方向探讨[J]. 重庆邮电大学学报:

自然科学版,2010,5:538-540

[12] 张铃,张钺. 动态商空间模型及其基本性质[J]. 模式识别与人工智能,2012,25(2):181-185

[13] 李道国,苗夺谦,等. 粒度计算研究综述[J]. 计算机科学,2005,32(9):1-12

[14] Yao Y Y, Zhang N, Miao D Q, et al. Set-theoretic approaches to granular computing[J]. Fundamenta Informaticae, 2012, 115(2/3):247-264

[15] Yao Y Y, Zhao L Q. A measurement theory view on the granularity of partitions[J]. Information Sciences, 2012, 213:1-13

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

[17] Yao Y Y. Stratified rough sets and granular computing[C]// Dave R N, Sudkamp T, eds. Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society. New York, USA, IEEE Press, 1999:800-804

[18] Hu Xiao-hua, Cercone N. Discovery maximal generalized decision rules through horizontal and vertical data reduction[J]. Computational Intelligence, 2001, 17(4):685-702

[19] Hu Xiao-hua, Cercone N. Learning maximal generalized decision rules via discretization, generalization and rough set feature selection[C]// Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence. Newport Beach, CA, USA. IEEE Computer Society, 1997:548-556

[20] Shan Ning, Hamilton H J, Cercone N. GRG: Knowledge Discovery Using Information Generalization, Information Reduction, and Rule Generation[C]// Proceedings of the seventh international conference on tools with artificial intelligence. Washington, DC, USA, IEEE Computer Society, 1995:372-379

[21] Dong Wei, Wang Jian-hui, Xu Lin, et al. Algorithm of Hierarchical Reduction Based on Rough Entropy[C]// Proceedings of the sixth world congress on intelligent control and automation. 2006:4374-4377

[22] Zhang J, Kang D K, et al. Learning accurate and concise naive Bayes classifiers from attribute value taxonomies and data[J]. Knowledge and information systems, 2006, 9(2):157-159

[23] Feng Q R, Miao D Q, Cheng Y. Hierarchical decision rules mining[J]. Expert Systems with Applications, 2010, 37:2081-2091

(上接第 165 页)

持续期参数都没有发生变化,但它被分配的资源却在变化。这是由于,在各测试阶段,为保证在可靠性指标的基础上实现资源最少,就要对复杂度较大的模块和在系统结构中影响更大的模块分配更多的测试资源,所以导致了一些模块即使在测试过程中系统结果和模块自身复杂度没有改变,但其被分配的资源却发生了变化。

将表 2 数据代入式(19),可得到用动态资源优化分配方法所需总测试时间资源为 5617h。若用静态资源优化分配法对此软件系统资源优化分配,即将 a_{11} 、 b_{11} 代入式(17)解得的 T_{11} 值作为优化资源分配的结果,所得到的总测试时间为 6268h。对于此模拟系统,在总的测试时间资源上,动态方法可比静态方法少 651h,较静态方法节约 10.4% 的测试资源。总的测试资源比较图如图 9 所示。



图 9 总测试资源对比图

结束语 基于 Markov 结构的软件可靠性测试资源动态分配方法用模块的运行比例系数 π_i 体现软件系统体系结构特点,用模块可靠性模型参数体现模块自身复杂度特点。通过阶段测试过程中动态调整可靠性模型参数,不仅保证了软件可靠性模型的准确性,也使得可靠性测试资源在测试过程中根据各软件模块复杂度的变化得到最合理而有效的分配和测试资源动态分配方法可根据软件 Markov 使用结构和自身复杂度合理分配可靠性测试资源,并有效减少总的软件可靠性测试时间资源。

参考文献

Lyu Micheal R. Software Reliability Engineering: A Roadmap

[A]// Future of Software Engineering, 2007[C]. Minneapolis, 2007:153-170

[2] 覃志东. 高可信软件可靠性和防危性测试与评价理论研究[D]. 成都:电子科技大学,2005

[3] Ohtera H, Yamada S. Optimal Allocation and Control Problems for Software-Testing Resources[J]. IEEE Trans. on Reliab., 1990, 39(2):171-176

[4] Lyu Micheal R, Rangarajan S, van Moorsel Aad P A. Optimal Allocation of Test Resources for Software Reliability Growth Modeling in Software Development[J]. IEEE Transactions on Reliability, 2002, 51(2):183-192

[5] Dai Y S, Xie M, Poh K L. Optimal Testing-Resource Allocation with Genetic Algorithm for Modular Software Systems[J]. The Journal of Systems and Software, 2003, 66(1):47-55

[6] 樊林波,吴智,赵明. 基于构件的软件可靠性分析[J]. 计算机科学, 2007, 34(5):266-269

[7] Rajgopal J, Mazumdar M. Modular Operational Test Plan for Inferences on Software Reliability Based on a Markov Model[J]. IEEE Transactions on Software Engineering, 2002, 28(4):356-363

[8] Pietrantuo R, Russo S, Trivedi K S. Software Reliability and Testing Time Allocation: An Architecture-Based Approach[J]. IEEE Transactions on Software Engineering, 2010, 36(3):323-337

[9] Lenug Y W. Dynamic Resource-Allocation for Software-Module Tesing[J]. Journal of Systems and Software, 1997, 37(2):129-139

[10] Cheung R C. A User-Oriented Software Reliability Model[J]. IEEE Transactions on Software Engineering, 1980, 6(2):118-125

[11] Yang B, Xie M. A Study of Operational and Testing Reliability in Software Reliability Analysis[J]. Reliability Engineering & System Safety, 2000, 70(3):323-329

[12] Goel A L, Okumoto K. Time-Dependent Error-Detection Rate Model for Software Reliability and Control