图表示下的知识约简

苗夺谦[1],陈玉明[1,2],王睿智[1],张红云[1]

(1.同济大学计算机科学与技术系,上海 201804; 2.厦门理工学院计算机科学与技术系,福建厦门 361024)

摘　要： 知识约简主要有代数表示下的知识约简和信息表示下的知识约简.本文提出图表示下的知识约简,给出图表示下求最小约简的完备递归算法.借鉴人工智能理论中的图搜索技术,提出旋转剪枝和回溯剪枝两个搜索算子求最小约简,并证明了在这种表示下求最小约简的完备性,理论分析和实验结果表明,在图表示下求最小约简是有效可行的.

关键词： 粗糙集；约简；幂图；图表示

中图分类号： TP18　　**文献标识码：** A　　**文章编号：** 0372-2112 (2010) 08-1952-06

Knowledge Reduction Algorithm under Graph View

MIAO Duo-qian[1], CHEN Yu-ming[1,2], WANG rui-zhi[1], ZHANG Hong-yun[1]

(1. *Department of Computer Science and Technology*, *Tongji University*, *Shanghai* 201804, *China*;
2. *Department of Computer Science and Technology*, *Xiamen University of Technology*, *Xiamen*, *Fujian* 361024, *China*)

Abstract: Knowledge reduction is widely studied under algebra view and information view. In this paper, knowledge reduction under graph view is presented. A complete recursive algorithm for minimal reduction under graph view is designed. In virtue of graph searching methods of artificial intelligence, rotation pruning operator and backtracking pruning operator for answering the minimal reduction question are proposed. These methods' completeness for the minimal reduction is proved. In order to test the efficiency of the algorithm, some experiments are made on simulative data. Theoretical analysis and experimental results show that the reduction algorithm under graph view is efficient and feasible.

Key words: rough sets; reduction; power graph; graph view

1　引言

Pawlak Z 提出的粗糙集理论[1]中所有的概念和运算都是通过代数学的等价关系和集合运算来定义的,被称为粗糙集理论的代数表示.Skowron A 在这种表示下提出基于差别矩阵的知识约简[2].Kryszkiewicz M 研究了代数表示下不一致决策系统中各种约简之间的关系[3],张文修等发展了 Kryszkiewicz M 的思想,进一步研究了代数表示下各种约简的关系,提出了最大分布约简的概念[4].

在代数表示下,粗糙集理论中的许多概念与运算的直观性较差,不容易使人理解其本质,并且在此表示下许多算法的效率也不高.苗夺谦等提出知识约简的信息表示[5,6],王国胤等研究代数表示下的约简和信息表示下的约简之间的关系[7].信息表示是以信息论为基础,通过信息熵来表示知识和度量知识,这种表示从更深层次上揭示了知识的本质,苗夺谦等在这种表示下提出基于信息熵的信息系统知识约简算法[5]和基于互信息的决策表知识约简算法[6],杨明提出基于条件信息熵的近似约简算法[8].

代数表示下的知识约简,难于理解,算法效率不高,信息表示下的知识约简解释了约简的信息含义,提高了算法的效率,但在代数表示下和信息表示下都没有考虑约简的空间拓扑结构,求最小约简算法的完备性也有待于进一步的研究.刘少辉等[9]提出的完备算法针对约简是完备的,但针对最小约简并不完备.知识约简包括信息系统的知识约简和决策表的知识约简.本文对信息系统的知识约简进行研究,结合信息表示下约简的判定,考虑到知识约简的空间拓扑结构,构建一种新的知识表示方式—幂图和幂树,用于知识约简当中,在这种新的表示方式基础上,借鉴人工智能理论中的图搜索技术,提出旋转剪枝法和回溯剪枝法两个搜索算子求最小约简,提出求最小约简的完备递归算法,分析了算法的时间和空间复杂度,证明了图表示下求最小约简的完备性.理论分析和实验结果表明,图表示下的知识约简是有效可行的.

收稿日期:2008-06-18;修回日期:2010-03-25
基金项目:国家自然科学基金(No.60475019,No.60775036,No.60970061)

2 基本概念

粗糙集理论把知识看作是对论域的划分,知识库是知识的集合.知识库可以形式地定义为序对 $K=(U,R')$,其中 U 为论域,R' 为 U 上的等价关系簇,称等价关系 $R\in R'$ 为知识,称 R 生成的等价类$[U_R]$为基本知识颗粒,称商集 $U/R=\{[U_R]\mid u\in U\}$为论域 U 的 R-粒划分.

定义 1[6]　设 P 是知识库 $K=(U,R')$中的知识,$U/P=\{X_1,X_2,\cdots,X_n\}$,定义知识 P 的熵为

$$H(P)=-\sum_{i=1}^{n}p(X_i)\log_2 p(X_i)$$

定义 2[6]　设 U 是论域,P、Q 为 U 上的两个等价关系,P、Q 在 U 上导出的划分为 A、B.其中 $A=\{X_1,X_2,\cdots,X_n\}$,$B=\{Y_1,Y_2,\cdots,Y_m\}$,知识 Q 相对于知识 P 的条件熵为

$$H(Q\mid P)=-\sum_{i=1}^{n}p(X_i)\sum_{j=1}^{m}p(Y_j\mid X_i)\log_2 p(Y_j\mid X_i)$$

定义 3[6]　设信息系统 $IS=(U,A,V,f)$,令 $R\subseteq A$,若满足:

(1)$H(R)=H(A)$;

(2)$\forall a\in R,H(a\mid R-a)>0$.

则称 R 是 IS 的约简.

定义 4　设信息系统 $IS=(U,A,V,f)$,令 $R\subseteq A$,若满足 $H(R)=H(A)$,则称 R 为 IS 的简化.

定理 1　是约简必为简化,是简化不一定为约简.

根据约简和简化的定义可以很容易得到,证明略.

定理 2　不是简化则其所有子集都不存在约简.

证明:反证法.已知 R 不是简化,设 R 的子集中存在某一约简 r,r 是约简必为简化,$r\subseteq R$,R 也必为简化,这与已知 R 不是简化相矛盾,定理得证.

定理 3　设信息系统 $IS=(U,A,V,f)$,$Sub(A)$为属性集合 A 全部子集的集合,$b\in Sub(A)$是约简,若 $\forall a\in Sub(A)$且$|a|>|b|$,则 a 不是最小约简.

证明:　设最小约简的元素个数为 k,由 $b\in Sub(A)$且是约简可知,$|b|\geqslant k$,已知$|a|>|b|$,则$|a|>k$,所以 a 不是最小约简.

定义 5[10]　设 $Power(A)$为属性集合 A 的幂集,给定有向图 G,G 的顶点为 $Power(A)$的元素,G 的边满足条件:$\forall P,Q,R\in Power(A)$,若 $Card(P)-1=Card(Q)=Card(R)+1$ 且$(P\cap R)\subset Q\subset(P\cup R)$,则存在 R 到 Q,Q 到 P 的有向边,称此有向图 G 为 A 的幂图.

例 1　属性集 $A=\{a,b,c,d\}$,其幂图为图 1 所示.

从幂图的定义可知幂图结点之间有相互的交叉,这不利于剪枝,因而可用一棵没有交叉的树来表示求解空间,这将提高剪枝的效率.称由幂图转化的树为幂树,下面给出幂树的递归定义.

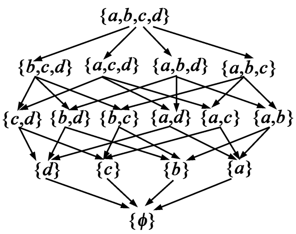

图1　幂图

定义 6　设属性集 A 有 n 个属性,属性集 A 的幂树是包含 2^n 个结点的有限集 PT,PT 为空时为空幂树,否则满足以下条件:

(1)有且仅有一个特定的称为根的结点,此结点有 n 个属性,n 个子结点;

(2)其余的结点可分为 n 个互不相交的子集 PT_1,PT_2,$\cdots$,PT_n,其中 PT_1 有 $n-1$ 个属性,$n-1$ 个子结点,PT_2 有 $n-1$ 个属性,$n-2$ 个子结点,$\cdots$,PT_n 有 $n-1$ 个属性,0 个子结点,并且每个子集本身又是一棵 $n-1$ 个属性的幂树,并称为根的子幂树.

定义 7　设属性集 A 有 n 个属性,PT 为 A 的幂树,幂树中基数为 n 的结点构成幂树的第 0 层,基数为 $n-1$ 的结点构成幂树的第 1 层,以此类推,基数为 0 的结点构成幂树的第 n 层.

为控制幂树的结构,对幂树进行标记,幂树结点属性个数标记为 *attnum*,子结点个数标记为 *child*,子结点是通过父结点顺次删除一个属性继承下来的,被删除的属性标记为 *delete*.

同时对同层每个结点的属性顺序进行属性内调整.首先,计算同层每个结点的属性个数减去子结点个数为 r,则 r 为每个结点应当调整顺序的属性个数.若某结点 r 为 0,则该结点不用调整;若 r 为 n,则该结点前面 n 个兄弟被删除的属性依次后置,其它属性与父结点的顺序相同并前置.

定义 8　经过标记和属性内调整的幂树,称为调整的幂树,为方便计,简称为幂树.

例 2　属性集 $A=\{a,b,c,d\}$,其幂树为图 2 所示.

图 2 表示调整后的幂树.单个字母表示被删除的属性,数字表示扩展的子结点数,也表示最多扩展的层数.

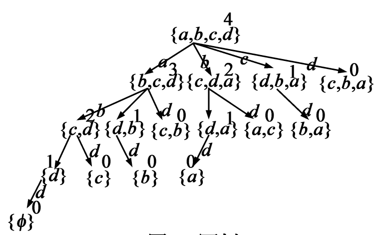

图2　幂树

3 旋转剪枝法和回溯剪枝法

幂树的特点是左边的树枝大,右边的树枝小.因此,尽量剪去左边的树枝,效率就高.根据定理 2,不是简化则其所有子集都不存在约简,在逐步扩展过程中,

可以根据是否是简化来剪枝，每次扩展过程中把不是简化的树枝旋转到左边来，作为大树枝剪掉，同时调整幂树，保持幂树的结构，我们称这种剪枝法为旋转剪枝法．把旋转剪枝法运用于知识约简中，称为旋转搜索算子，记为 R．

例 3 属性集 $A=\{a,b,c,d\}$，依次删除一个属性扩展，若第一个子结点 $\{b,c,d\}$ 不是简化，则剪去，如图 3所示；若最后一个子结点 $\{a,b,c\}$ 不是简化，则旋转到最左边成为第一个子结点，作为大树枝剪去并属性内调整，如图 4 所示．

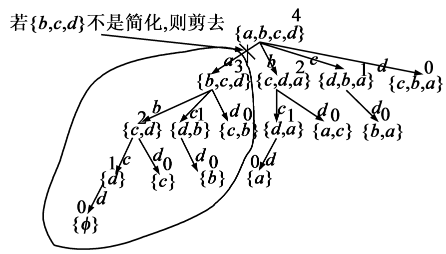

图3 剪枝

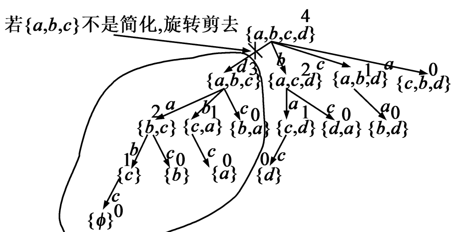

图4 旋转剪枝

旋转剪枝法总是剪去左边的大树枝，但右边的树枝没有很好的剪枝效率．根据定理 3，找到一个约简后可以剪去不存在最小约简的部分．当深度搜索找到一个约简后，暂时作为最小约简，并计算此最小约简的属性个数为 r．回溯到父结点的右边兄弟搜索下去，当前要扩展结点的属性个数为 t，要扩展的子结点个数(层数)为 s，若 $t-s\geqslant r$ 则不用扩展，剪去当前结点．根据找到的最小约简回溯找下一个最小约简，这种回溯剪枝是按宽度搜索剪枝的，相当于尽量剪去右边部分的树枝．这种剪枝方法我们称为回溯剪枝法．把回溯剪枝法运用于知识约简中，称为回溯搜索算子，记为 B．

例 4 属性集 $A=\{a,b,c,d\}$，如图 5 所示，设经过旋转剪枝搜索到 $\{b,a\}$ 是约简，把 $\{b,a\}$ 暂时作为最

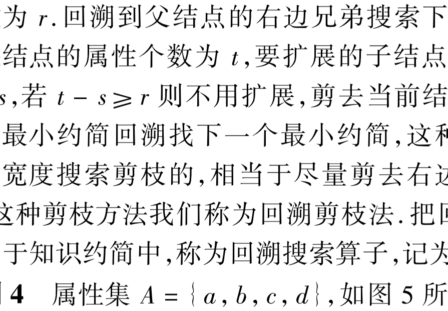

图5 回溯剪枝

小约简，回溯到父结点的兄弟 $\{a,c,d\}$ 搜索下去，父结点的其它两个兄弟 $\{a,b,d\}$ 和 $\{c,b,d\}$ 剪去，因为 $\{a,b,d\}$ 标记子结点数为 1，表示最多扩展 1 层，扩展 1 层后，属性个数是 2 个，与暂时找到的约简 $\{b,a\}$ 属性个数相同，不必扩展，同样 $\{c,b,d\}$ 标记子结点数为 0，也不必扩展．

旋转剪枝法是尽量剪去左边部分树枝，回溯剪枝法是尽量剪去右边部分树枝，两种方法同时采用，这对求最小约简有着非常高的效率，后面的实验也验证了这点．

4 图表示下的知识约简算法及完备性证明

4.1 图表示下的知识约简算法

信息表示下的约简是采用信息熵来判定的，我们结合信息表示下约简的判定，转化成图表示下约简的判定，下面给出图表示下约简的判定．

定义 9 设信息系统 $IS=(U,A,V,f)$，定义该信息系统在图表示下的知识约简系统 KRS 为一 6 元组 (U,A,G,PT,R,B)，U 为对象集，A 为属性集，G 为 IS 的幂图，PT 为 IS 的幂树，R 为旋转算子，B 为回溯算子．

定理 4 设信息系统 $IS=(U,A,V,f)$，该信息系统图表示下的知识约简系统 $KRS=(U,A,G,PT,R,B)$，设 t 为幂树 PT 中某一叶子结点，s 为 t 的父结点，在同时采用搜索算子 R 和 B 的条件下，产生搜索树 ST，$s,t\in ST$，

(1)若 $H(t)=H(A)$，则 t 为约简；

(2)若 $H(t)\neq H(A)$，则 s 为约简．

证明： (1)t 为叶子结点，根据幂树的定义易知 t 的子集必由 t 的兄弟扩展，假设 t 的某个兄弟 b 是简化，则约简是 b 或者存在于 b 的子集中，则根据回溯算子 B，t 被剪去，即 t 不是搜索树的结点，这和已知 t 是搜索树中的结点相矛盾，因而 t 的所有兄弟都不是简化，因而 t 的子集都不是简化，因此 $\forall a\in t$，$H(a|t-a)>0$．若 $H(t)=H(A)$，由 $\forall a\in t$，$H(a|t-a)>0$，则 t 为约简．

(2)假设 s 不是简化，根据旋转算子 R，s 被剪去，t 是 s 的子结点，则 t 不是搜索树的结点，这和 t 是搜索树中的结点相矛盾，因而 s 为简化，即 $H(s)=H(A)$．已知 $H(t)\neq H(A)$，则 t 不是简化；前面已证明 t 的所有兄弟都不是简化；所以 s 的所有子结点都不是简化，因此，$\forall a\in s$，$H(a|s-a)>0$．由 $H(s)=H(A)$，$\forall a\in s$，$H(a|s-a)>0$，故 s 为约简．

知识约简解空间是幂级空间，考虑到知识约简的空间拓扑结构，我们构造了幂图来表示知识约简解空间，进而把幂图转化为幂树，引入旋转剪枝法和回溯剪

枝法两个搜索算子,并给出了在图表示下约简的判定,下面给出图表示下基于幂树的知识约简算法.

算法KRAPT (Knowledge Reduction Algorithm based on Power Tree)

输入:信息系统 $IS=(U,A,V,f)$

输出:最小约简 *MinReduct*

a $MinReduct=\{A\}$, $attrib=\{A\}$, $attnum=|A|$, $child=|A|$.

b Expand(*attrib*, *attnum*, *child*).

c 输出最小约简 *MinReduct*.

递归函数名及参数: void Expand(*attrib*, *attnum*, *child*)

输入参数: *attrib*(属性集), *attnum*(属性个数), *child*(扩展的子结点数)

Step1 当前结点 $N=attrib$.

Step2 判当前结点 N 是否为叶结点,若是,转 Step3,否则,转 Step5.

Step3 判当前结点 N 是否为简化,若是,则 $reduct=N$;否则 $reduct=$ 父结点 F.

Step4 判 *reduct* 的属性个数是否小于 *MinReduct* 的属性个数,若是, $MinReduct=reduct$,返回;否则 *MinReduct* 不更新,返回.(注: *MinReduct* 为全局变量, *reduct* 为局部变量).

Step5 依次删除一个属性扩展当前结点 N,扩展后的结点为 $N_1,N_2,\cdots,N_{child}$.

Step6 判扩展的结点是否是简化,若是简化则旋转到幂树右边,不是简化则旋转到幂树左边剪去;旋转剪枝后的结点为 $M_1,M_2,\cdots,M_m,M_{m+1},\cdots,M_{child}$,其中 $M_1,M_2,\cdots,M_m$ 为被剪枝的部分, $M_{m+1},\cdots,M_{child}$ 为未被剪枝的部分.(即:旋转剪枝)

Step7 对未被剪枝的部分 $M_{m+1},\cdots,M_{child}$ 进行属性内调整.

Step8 对未被剪枝的部分 $M_{m+1},\cdots,M_{child}$ 进行递归调用,从 $i=m+1$ 到 *child* 重复以下操作:

若(M_i 属性个数 $-$ M_i 要扩展的子结点数) $<$ *MinReduct* 的属性个数,则递归调用 Expand(M_i, $attnum-1$, $child-i$);否则,这次循环跳过,进行下次循环(即:回溯剪枝)

4.2 算法的完备性证明

文献[9]中的完备算法针对约简是完备的,即能保证找到约简,但是算法没有搜索整个解空间,不能保证找到最小约简,所以针对最小约简是不完备的.图表示下基于幂树的知识约简算法针对最小约简是完备的,下面给出完备性证明.

定义 10 设 A 为属性集,定义 $Sub(A)$ 为 A 的全部子集的集合, $Sub(A)$ 即是 A 的幂树全部结点的集合,定义 $Sub\_level(A,i)$ 为 A 的全部子集中基数是 $|A|-i$ 的属性集集合, $Sub\_level(A,i)$ 即是 A 的幂树的第 i 层全部结点的集合.例:若 $A=\{a_1,a_2\}$,则 $Sub\_level(A,0)=\{\{a_1,a_2\}\}$, $Sub\_level(A,1)=\{\{a_1\},\{a_2\}\}$.

性质 1 $Sub(A)=\sum_{i=0}^{n}Sub\_level(A,i)$

定理 5 算法 KRAPT 求最小约简是完备的.

证明: 全部遍历算法:检查 $Sub(A)$ 中的每个元素是否是约简,找到所有约简,在所有约简中找出一个属性个数最小的约简,即为最小约简.全部遍历算法是完备的.

根据定理 2,在遍历过程中可以剪去不是简化的属性集;根据定理 3,在找到一个约简的基础上可以剪去不存在最小约简的部分;因而可以对全部遍历算法进行改进.

改进算法:

(1) 检查 $Sub\_level(A,i)$ 中的所有元素是否为简化($i=0,\cdots,n$):

(a) 若 $B\in Sub\_level(A,i)$ 且不是简化,则令 $Sub(A)=Sub(A)-\sum_{j=0}^{n-i}Sub\_level(B,j)$,

(b) 若 $B\in Sub\_level(A,i)$ 且是约简,则令 $Sub(A)=Sub(A)-\sum_{j=0}^{i}Sub\_level(A,i-j)$,

(c) 若 B 是约简且 $|B|<|Minreduct|$,则 $Minreduct=B$.

(2) 全部遍历完后, *Minreduct* 即为最小约简.

改进算法在全部遍历算法的基础上,根据定理 2 剪去不是简化的属性集,根据定理 3 剪去不存在最小约简的属性集,因而与全部遍历算法是等价的,所以是完备的.

算法 KRAPT 是改进算法的具体实现过程.改进算法(1)对应于算法 KRAPT 递归函数的步骤 Step1, Step2, Step3;改进算法(a)对应于算法 KRAPT 递归函数的步骤 Step5, Step6, Step7;改进算法(b)对应于算法 KRAPT 递归函数的步骤 Step8;改进算法(c)对应于算法 KRAPT 递归函数的步骤 Step4.已经证明改进算法是完备的,所以算法 KRAPT 是完备的.

5 实验分析

为了验证本文方法的有效性及剪枝效率,我们进行了多组实验.首先,我们预先产生所有约简来测试本文算法在约简不同分布情况下的剪枝效率;然后,通过大量随机离散数据测试本文算法的性能.本实验的硬件测试环境是: CPU 为 Inter Pentium4 2.4GHz,内存为

1G,操作系统为 WindowsXP,开发工具为 VC + +6.0.

5.1 模拟数据集测试

为了考察约简不同分布情况下的剪枝效率,我们采用实验数据如下,例如 5 个属性{a,b,c,d,e},解空间共 32 个结点,随机生成 8 个约简,假设预先随机产生约简{a,b}、{a,c}、{a,d}、{a,e}、{b,c}、{b,d}、{b,e}和{c,d,e}为一组,共 20 组进行测试,计算平均访问结点.在实验 1,属性个数逐个增长,解空间指数增长,随机产生的约简个数保持占整个解空间的 25%.在实验 2,属性个数逐个增长,解空间指数增长,随机产生的约简个数不变,保持 20 个.在实验 3,属性个数为 10,随机产生的约简个数指数增长.实验用平均访问结点数来表示剪枝的性能.

从表 1、表 2 和表 3 可知,实验 1、实验 2 和实验 3 均找到最小约简,平均访问结点数缓慢增长,表明旋转剪枝和回溯剪枝有着很好的剪枝效率.图 6 给出了实验 1 和实验 2 的对比结果,从中可以看出约简大量分布(占解空间的四分之一)比约简稀疏分布有着更好的剪枝性能.图 7 给出了实验 1 与实验 2 对比于解空间的剪枝性能,从中可以看出,解空间是随属性增加而指数增长的,而实验 1 与实验 2 中的平均访问结点数缓慢增长,结果表明本文方法有着较高的剪枝性能.

表 1 实验 1

属性个数	解空大间大小	随机产生约简个数	平均访问结点数	是否找到最小约简
5	32	8	14.3	是
6	64	16	24.1	是
7	128	32	29.6	是
8	256	64	40.9	是
9	512	128	54.8	是
10	1024	256	81.3	是
11	2048	512	86.1	是

表 2 实验 2

属性个数	解空大间大小	随机产生约简个数	平均访问结点数	是否找到最小约简
5	32	20	14.1	是
6	64	20	20.5	是
7	128	20	33.8	是
8	256	20	46.8	是
9	512	20	97.5	是
10	1024	20	113.3	是
11	2048	20	159.1	是

表 3 实验 3

属性个数	解空大间大小	随机产生约简个数	平均访问结点数	是否找到最小约简
10	1024	4	56.3	是
10	1024	8	131.7	是
10	1024	16	151.3	是
10	1024	32	170.7	是
10	1024	64	157.5	是
10	1024	128	106.3	是
10	1024	256	81.3	是

图 8 给出了实验 3 的结果,从中可以看出,平均访问结点数开始随约简的增加而增加,达到峰值后,随约简的增加而减少.实验表明约简稀疏分布和约简大量分布都有较好的剪枝性能.约简稀疏分布时,则幂树中大量的结点不是简化,这有利于旋转剪枝,但不利于回溯剪枝.约简大量分布时,则幂树中大量的结点为简化,这不利于旋转剪枝,但有利于回溯剪枝.

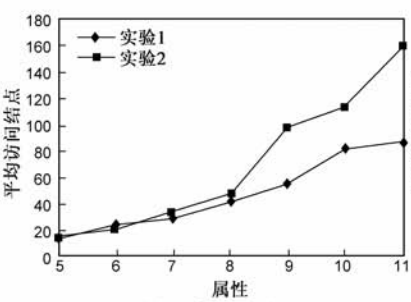

图6 实验对比

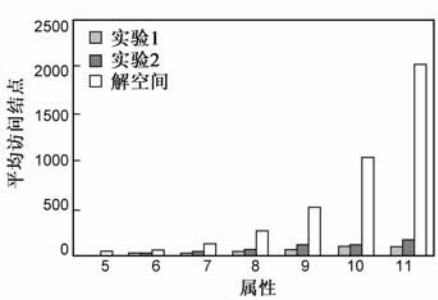

图7 实验对比

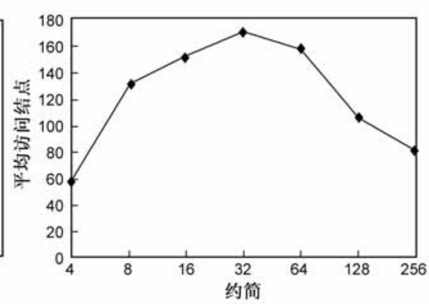

图8 实验3结果

5.2 随机生成数据集测试

Srarzyk J 采用扩展法则简化分明矩阵求出所有约简[11],从而可以得到最小约简,但只能处理 40 个属性的数据.下面实验采用 Srarzyk J 的扩展法则约简算法[11]与本文的 KRAPT 算法进行对比测试.实验数据采用文献[11]中的方法,设计数据产生器随机生成大量数据进行测试,数据值为 0~8 之间的随机整数数据.图 9 为扩展法则算法的测试结果,数据对象数从 25,5 个对象一递增,递增到 40,属性数从 25,5 个属性一递增,递增到 40.图 10 为 KRAPT 算法的测试结果,数据对象数从 60,20 个对象一递增,递增到 120,属性数从 60,20 个属性一递增,递增到 120.平均运行时间以秒为计算单位,并取对数.

从图 9 与图 10 的对比可知,扩展法则算法能处理 40 个属性的数据集,而 KRAPT 算法可以处理 120 个属性的数据集.求最小约简已被证明是 NP 问题,从实验

可知,扩展法则算法和 KRAPT 算法仍然是指数增长的,扩展法则算法运行时间大致按 5 个属性指数增长,而 KRAPT 算法大致按 20 个属性指数增长.

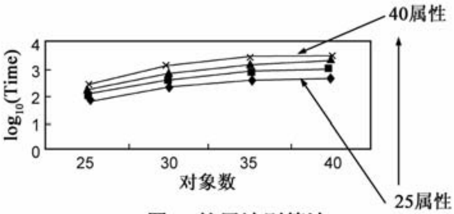

图9 扩展法则算法

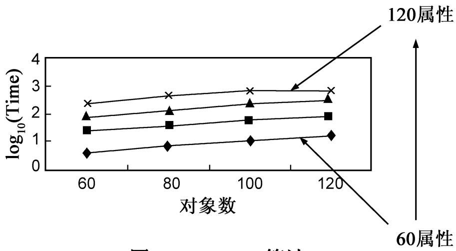

图10 KRAPT算法

6 结论

知识约简主要有代数表示下的知识约简和信息表示下的知识约简.代数表示下的知识约简和信息表示下的知识约简都未考虑到解的空间拓扑结构.本文考虑到解的空间拓扑结构,提出图表示下的知识约简,采用幂图和幂树表示解空间,并给出基于幂树求最小约简的完备递归算法,采用旋转剪枝法和回溯剪枝法进行有效剪枝,理论分析和实验结果表明,在图表示下求最小约简是有效可行的,这为最小约简的求解提供了一条新的途径.

参考文献:

[1] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Science, 1982, 11(5): 341 – 356.

[2] Skowron A, Rauszer C. The discernibility matrices and functions in information systems[A]. Intelligent Decision Support, Handbook of Applications and Advances of the Rough Set Theory[C]. Dordrecht: Kluwer Academic Publishers, 1992. 331 – 362.

[3] Kryszkiewicz M. Comparative studies of alternative type of knowledge reduction in inconsistent systems[J]. International Journal of Intelligent Systems, 2001, 16(1): 105 – 120.

[4] 张文修,米据生,吴伟志.不协调目标信息系统的知识约简[J].计算机学报,2003,26(1):12 – 18.
Zhang Wen-xiu, Mi Ju-sheng, Wu Wei-zhi. Knowledge reductions in inconsistent information systems[J]. Chinese Journal of Computers, 2003, 26(1): 12 – 18. (in Chinese)

[5] 苗夺谦,王珏.粗糙集理论中概念与运算的信息表示[J].软件学报,1999,10(2):113 – 116.
Miao Duo-qian, Wang Jue. An information representation of the concepts and operations in rough set theory[J]. Journal of Software, 1999, 10(2): 113 – 116. (in Chinese)

[6] 苗夺谦,王国胤,刘清,等.粒计算:过去、现在与展望[M].北京:科学出版社,2007.

[7] Wang Guo-yin. Rough reduction in algebra view and information view[J]. International Journal of Intelligent Systems, 2003, 18(6): 679 – 688.

[8] 杨明.决策表中基于条件信息熵的近似约简[J].电子学报,2007,35(11):2156 – 2160.
Yang Ming. Approximate reduction based on conditional information entropy in decision tables[J], Acta Electronica Sinica, 2007, 35(11): 2156 – 2160. (in Chinese)

[9] 刘少辉,盛球戬,吴斌,等. Rough 集理论论高效算法的研究[J].计算机学报,2003,26(5):524 – 529.
Liu Shao-hui, Sheng Qiu-jian, Wu Bin, et al. Research on efficient algorithms for rough set methods[J]. Chinese Journal of Computers, 2003, 26(5): 524 – 529. (in Chinese)

[10] 陈玉明,苗夺谦.基于幂图的属性约简搜索式算法[J].计算机学报,2009,32(8):1486 – 1492.
Chen Yu-ming, Miao Duo-qian. Searching algorithm for attribute reduction based on power graph[J]. Chinese Journal of Computers, 2009, 32(8): 1486 – 1492. (in Chinese)

[11] Srarzyk J, Nelson D E, Sturtz K. Reduct generation in information systems[J]. Bulletin of International Rough Set Society, 1999, 3(1 – 2): 19 – 22.

作者简介:

苗夺谦 男,1964 年生于山西晋中,教授,博士生导师,研究方向为粗糙集理论、粒计算、数据挖掘与 Web 智能等.
E-mail: miaoquoqian@163.com

陈玉明 男,1977 年生于江西吉安,博士生,研究方向为粗糙集理论与数据挖掘等.
E-mail: cym0620@163.com