Computer Science and Application 计算机科学与应用, 2017, 7(10), 1015-1025
Published Online October 2017 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2017.710115

Hans汉斯

The Study and Implementation of Induction Value Reduction

Chengxia Liu[1,2], Meishu Zhang[2]

[1]Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing Information and Technology University, Beijing
[2]Computer School, Beijing Information and Technology University, Beijing
Email: cecilia7812@163.com

Received: Oct. 8th, 2017; accepted: Oct. 22nd, 2017; published: Oct. 26th, 2017

Abstract

Based on the research of rough set theory, this paper studies the process of induction value reduction. The induction value reduction algorithm uses the minimum decision algorithm to solve the decision table of the knowledge representation system to obtain the reduction. It can be realized by solving the minimum decision algorithm of each decision rule class. For each rule in each decision rule class, the algorithm computes its core attributes and then determines whether the core attributes can determine the rule. If can, then it will output the rule and remove its equivalent rules. Otherwise, it will gradually add the non-core attributes until they are able to determine the rule, then output the rule and remove its equivalent rules. At last the test system is implemented.

Keywords

Induction Value Reduction, Minimum Decision Algorithm, Rough Set

基于归纳的值约简算法的研究与实现

刘城霞[1,2]，张梅舒[2]

[1]北京信息科技大学网络文化与数字传播北京市重点实验室，北京
[2]北京信息科技大学计算机学院，北京
Email: cecilia7812@163.com

收稿日期：2017年10月8日；录用日期：2017年10月22日；发布日期：2017年10月26日

摘　要

在粗糙集理论的基础上，本文研究了归纳值约简过程。归纳值约简算法采用求解知识表达系统决策表的

文章引用: 刘城霞, 张梅舒. 基于归纳的值约简算法的研究与实现[J]. 计算机科学与应用, 2017, 7(10): 1015-1025.
DOI: 10.12677/csa.2017.710115

最小决策算法来求其约简，它可以通过分别求解各个决策规则类的最小决策算法来实现。对于每个决策规则类中的规则，首先计算其核值属性，然后判断核值属性是否能够决定该规则，如果能够决定，则输出规则并删除其等价规则；否则，逐渐加入非核值属性，直到能够决定该规则，然后输出规则并删除其等价规则。最终实现了其测试系统。

关键词

归纳值约简，最小决策算法，粗糙集

1. 引言

属性约简和值约简是粗糙集理论研究中的两个重要内容，属性约简是在保持与原有的数据库决策能力相同的情况下，选择问题最小属性子集，剔除数据中的没有利用价值成分的过程。在现实世界的问题中，由于噪音、误导和不相关属性的存在，使得属性约仅是在一定程度上去除了决策表中的冗余属性，但并没有完全去掉决策表中的不必要的信息。为此，还需要对决策表进行更深层次的处理，即对决策表进行值约简。值约简是去掉多余的属性值，用最少的条件属性值来区分每一个决策类，在不改变决策能力的基础上得到更加简化的规则集。值约简的研究方法有很多，比如一般的值约简算法、启发式值约简算法、基于决策矩阵的值约简算法、归纳值约简算法和 Skowron 算法等。本文主要研究基于归纳的值约简算法，并对算法的执行效果进行了实验验证，以及与启发式值约简算法进行了比较。

2. 粗糙集基本概念

粗糙集理论是一种对不确定性数据进行分析的理论，它的主要思想就是在保持信息系统分类能力不变的条件下，通过知识约简剔除数据中冗余的信息，从而得到问题的正确决策或数据分类。

2.1. 信息表和决策表

$S=(U,V,A,f)$ 为一个信息表[1]，其中 U 为论域，是一非空有限对象集，即 $U=\{x_1,x_2,\cdots,x_n\}$；$A=\{a_1,a_2,\cdots,a_n\}$ 是非空有限的属性集合；Va 是属性 a 的值域，即 $V=\bigcup Va$，$f:U\times A\xrightarrow{a\in U}V$ 成为信息函数，使得对每一 $a\in A$，$x\in U$，有 $f(x,a)\in Va$。在粗糙集理论中，信息表可简化 $S=(U,A)$ 或 $S=(U,A,V)$。

在信息表 S 中，如果属性集 A 由条件属性集 C 和决策属性集 D 组成，并且满足 $C\cup D=A$，$C\cap D=\varnothing$，则称 S 为决策表，记为 $S=(U,C\cup D)$。在决策表 S 中，若存在两行信息，其全部条件属性值相同，而决策属性值不相同，则称 S 为不相容决策表，否则为相容决策表。这里仅考虑相容决策表。

2.2. 知识和不可分辨关系

定义 1：(知识和知识库)给定论域 U 和其对应的一个等价关系 R，在等价关系 R 下对论域 U 的划分，称为知识，记为 U/R。U 上的一簇划分称为关于 U 的一个知识库。

设 R 是 U 上的一个等价关系，U/R 表示 R 的所有等价类(或者 U 上的分类)构成的集合，$[x]_R$ 表示包含元素 $x\in U$ 的 R 等价类。一个知识库就是一个关系系统 $K=(U,R)$，其中 R 是论域 U 上的一簇等价关系。若 $P\subseteq R$，且 $P\neq\varnothing$，则$\cap P$(P 中所有等价关系的交集)也是一个等价关系，称为 P 上的不可分辨关系，

记为 $ind(P)$，且有 $[x]_{ind(P)}=\bigcap[x]_R(R\in P)$。不可分辨关系 $ind(P)$是 U 上的等价关系，它是粗糙集理论中最基本的概念，若 $\langle x,y\rangle\in ind(P)$，则称对象 x 与 y 是 P 不可分辨的，即 x，y 存在于不可分辨关系 $ind(P)$ 的同一个等价类中，依据等价关系簇 P 形成的分类知识，x 与 y 无法分辨。

2.3. 约简和核

知识约简是粗糙集理论中的核心内容之一。所谓知识约简，就是在保证知识库分类能力不变的条件下，删除不相关或不重要的知识，它涉及的两个基础概念就是约简和核。

令 A 为一属性集，$a\in A$，如果 $ind(A)=ind(A-\{a\})$，则称 a 为 A 中不必要的；否则 a 为 A 中必要的。

如果 $a\in A$ 都为 A 中必要的，则称 A 是独立的；否则称 A 是依赖的。

定理 1：如果 A 是独立的，$P\subseteq A$，则 P 也是独立的。

设 $Q\subseteq P$，如果 Q 是独立的，且 $ind(Q)=ind(P)$，则称 Q 为 P 的一个约简。显然，P 可以由多个约简。P 中所有的必要属性组成的集合称为 P 的核，记作 $core(P)$。

定理 2： $core(P)=\bigcap red(P)$。其中，$red(P)$表示 P 的所有约简的集合。

由上述定理可以看出，核这个概念的用处包含两个方面：一方面，核能够作为计算所有约简的基础，这是因为所有约简都包含它的核；另一方面，核可解释为在属性约简中不能去除的知识特征部分的集合。

定义 2：相容决策信息系统 $IS=(U,C\cup D,V,f)$，对决策规则 d_x 有 $[x]_C\subseteq[x]_D$。如果对于 $a\in C$，有 $[x]_{C-\{a\}}\not\subset[x]_D$，则属性 a 为决策规则 d_x 的核值属性，a 为 d_x 中不可省略的；如果 $[x]_{C-\{a\}}\subseteq[x]_D$，则属性 a 为决策规则 d_x 的非核值属性，a 为 d_x 中可以省略的。

如图 1 所示，对于第一条决策规则 $a_1b_0d_1\to e_1$，$[1]_{e_1}=\{1,2\}$，去掉属性 a，得 $[1]_{b_0d_1}=\{1\}\subseteq\{1,2\}$，所以属性 a 为该规则的非核值属性；去掉属性 b，得 $[1]_{a_1d_1}=\{1,4\}\not\subset\{1,2\}$，所以属性 b 为该规则的核值属性。即对于这条决策规则，属性 a 可以省略，属性 b 不可以省略。

U	a	b	d	e
1	1	0	1	1
2	1	0	0	1
3	0	0	0	0
4	1	1	1	0
5	1	1	2	2
6	2	1	2	2
7	2	2	2	2

Figure 1. An instance of core attributes based decision rule
图 1. 一个关于决策规则核值属性的例子

2.4. 值约简相关概念

对于一个决策表而言，它的约简主要有两方面：属性约简和值约简。属性约简是删除决策表中的不必要的条件属性，而值约简的目的在于删除论域中各条记录的多余属性值，也就是删除与决策规则不相关的条件属性的值，进一步简化决策表。

定义 3： 令 $U/D=\{y_1,y_2,\cdots,y_n\}$ 表示论域 U 上有决策属性划分的决策类集，对每一个决策等价类，定义决策规则类 DRC 为

$$DRC(y)=\left\{d_x:des([x]_C)\Rightarrow des([x]_D)\middle|x\in U\text{ 且 }[x]_C\subseteq y\right\},\ \forall y\in U/D$$

其中 $des(Xi)$表示对等价类 Xi 的描述，即等价类 Xi 对于各条件属性值的特定取值。

用 $core(y)$，$\forall y\in U/D$ 表示决策类 y 的核值属性集，$core(d_x)$表示决策规则 d_x 的核值属性集，则有 $core(y)\subseteq C$，$core(dx)\subseteq C$，且 $core(y)=\bigcup_{d_x\in DRC(y)} core(d_x)$。

集合的幂集就是集合所有子集组成的集合。

定义 4： 令 T(OA)为集合 OA 的幂子集，$T_1(OA)$为集合 OA 的一阶幂集，给 $T_1(OA)$中元素赋以权值，有 $\forall A'\in T_1(OA)$，$w(A')=w(a_i')$，$a_i'\in A$。按 $w(A')$大小对 $T_1(OA)$中的元素进行排序，得到一阶有序幂子集 $OT_1(OA)$。

同理，$T_i(OA)$为集合 OA 的 i 阶幂集$(1\le i\le m)$，给 $T_i(OA)$中元素赋以权值，有 $\forall A'\in T_i(OA)$，$w(A')=\sum w(a_j')\,(j=1,2,\cdots,i)$，$a_j'\in A'$。按 $w(A')$大小对 $T_i(OA)$中的元素进行排序，得到一阶有序幂子集 $OT_i(OA)$。

3. 归纳值约简算法的实现

值约简算法很多学者都在研究，比如文献[2]-[10]，这里主要实现归纳值约简算法。归纳值约简算法采用求解知识表达系统决策表的最小决策算法来求其约简，它可以通过分别求解各个决策规则类的最小决策算法来实现。对于每个决策规则类中的规则，首先计算其核值属性，然后判断核值属性是否能够决定该规则，如果能够决定，则输出规则并删除其等价规则；否则逐渐加入非核值属性，直到能够决定该规则，然后输出规则并删除其等价规则。具体实现方法如下：

步骤 1： 任意 $d_X\in DRC(y)$；

步骤 2： 如果$[X]core(d_X)\subseteq y$，则输出决策规则

$$d_X:des[X]_{core(d_X)}\Rightarrow des([X]_D),\ DRC(y)=DRC(y)/[X]_{core(d_X)}$$，转步骤 9；

其中，$DRC(y)=DRC(y)/[X]_{core(d_X)}$，表示从 $DRC(y)$ 中删除规则 $d_{x'}$：$des([X']_C)\Rightarrow des([X']_D)$，这里 $[X']\in[X]_{core(d_X)}$。

步骤 3： 令 $A1=core(y)-core(d_X)$，$A2=C-core(y)$，在测度函数 $w(a)=|POSC-\{a\}(D)|/|U|$ 下对 $A1$，$A2$ 中的元素排序，得有序集 $OA1$，$OA2$，则有序集 $OA=OA1\cup OA2$，且 $|OA|=m$，OA 的 m 个有序幂子集分别为 $T_1(OA),\cdots,T_m(OA)$，相应的元素个数为 $n_1,n_2,\cdots,n_m$；

步骤 4： $j=1$；

步骤 5： $i=1$；

步骤 6： 令 $B=core[d_X]\cup T_j^i(OA)$，如果 $[x]_B\subseteq y$，输出 $d_X:des[X]_B\Rightarrow des([X]_D)$

$DRC(y)=DRC(y)/[X]_B$，转步骤 9；

步骤 7： $i=i+1$，如果 $i\le n_j$，转(6)；

步骤 8： $j=j+1$，如果 $j\le m$，转(5)；

步骤 9：如果 $DRC(y)\neq 0$，转(1)；

步骤 10：结束。

通过以上步骤，可以求得各个决策规则类的最小决策算法，进而得到整个决策表的最小决策算法，达到对决策表进一步约简的目的。

系统流程图如图 2 所示。

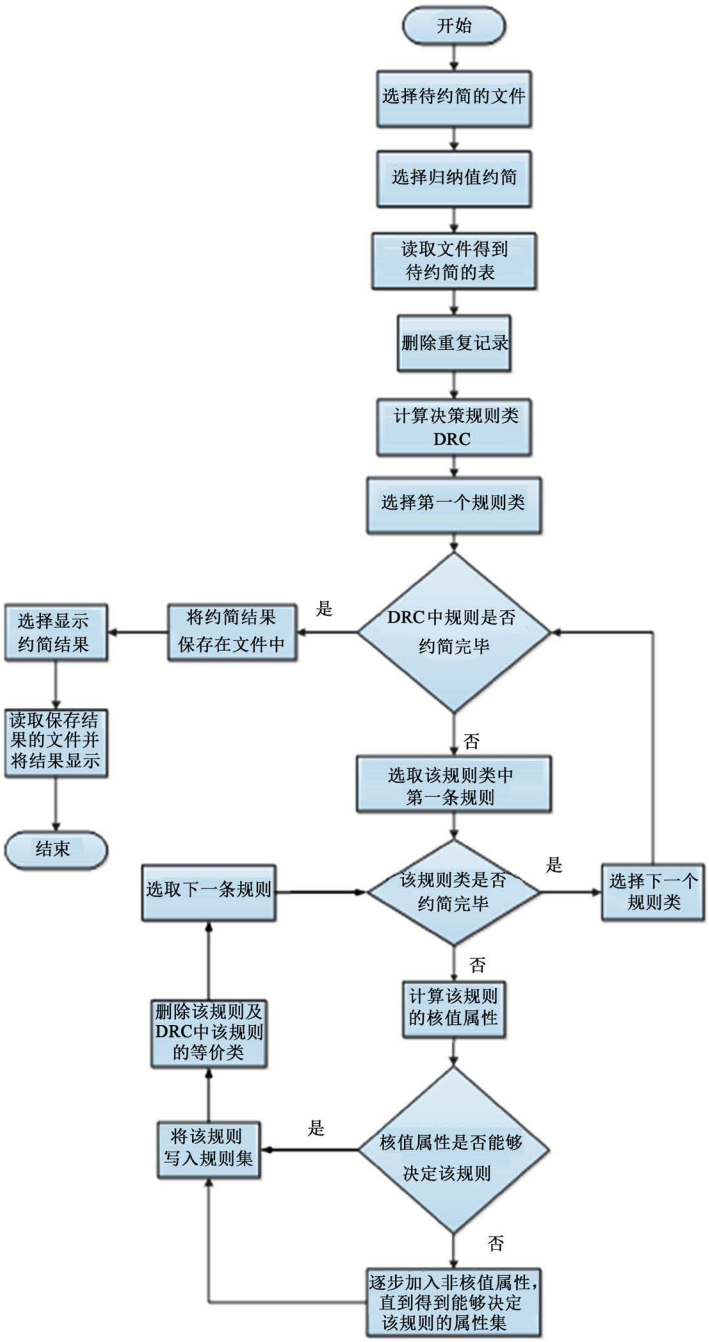

Figure 2. Flow chart of induction value reduction algorithm
图 2. 归纳值约简算法

归纳值约简算法的核心内容是最小决策算法。求解知识表达系统的最小决策算法，可以通过分别求解各个决策规则类的最小决策算法来实现。对于每个决策规则类中的规则，首先计算其核值属性，然后判断核值属性是否能够决定该规则，如果能够决定，则输出规则并删除其等价规则；否则，逐渐加入非核值属性，直到能够决定该规则，然后输出规则并删除其等价规则。流程图如图 3 所示。

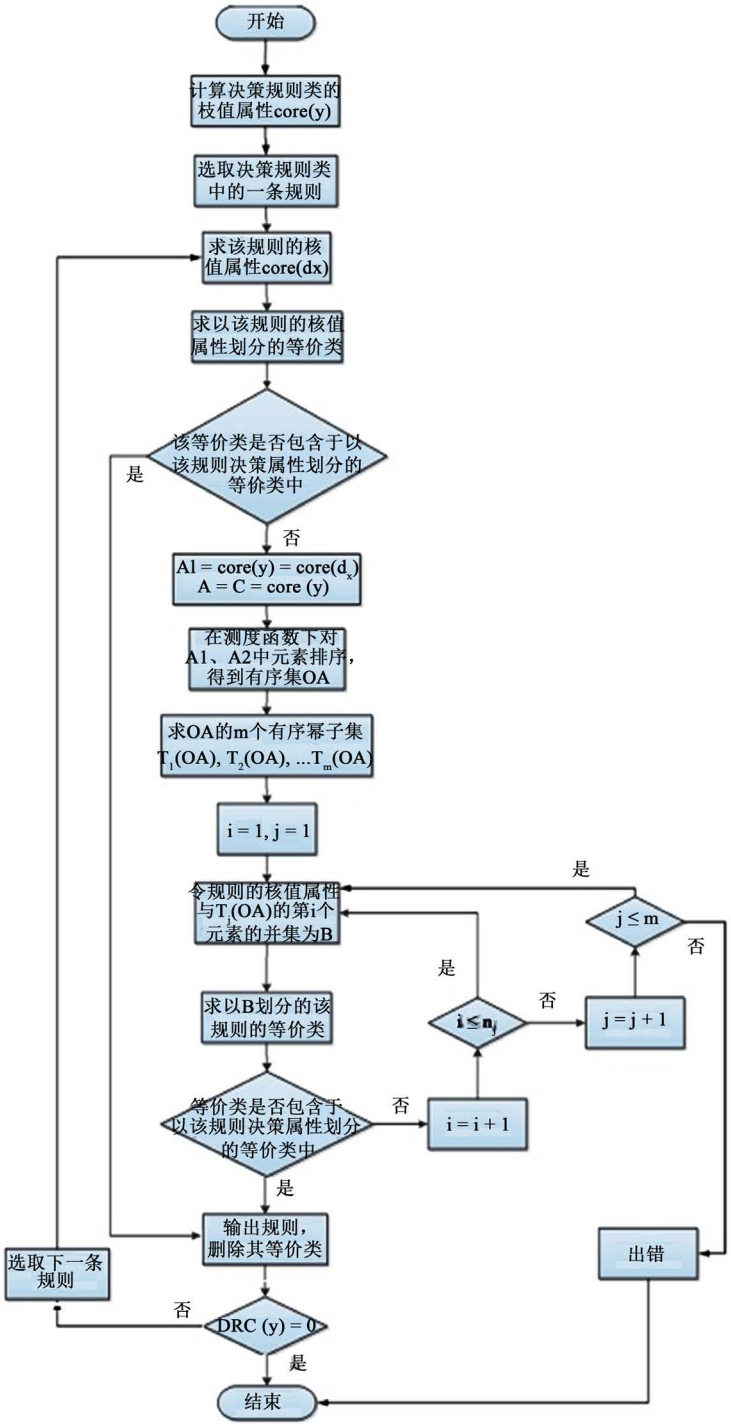

Figure 3. Flow chart of minimum decision algorithm
图 3. 最小决策算法流程图

由于本算法只考虑相容决策表，即不存在两条规则，条件属性值都相等而决策属性值不同，所以决策规则类 $DRC(y)$可由 U/D 得到。

4. 结果分析

4.1. 简单例子

测试一个较为简单的例子，数据如图 2 所示。算出规则等价类 $DRC(y)$。

$$U/D = U/\{e\} = \{\{1,2\},\{3,4\},\{5,6,7\}\}$$

所以 7 条规则划分为 3 个规则等价类，分别为：

$$DRC(y1): a_1b_0d_1 \to e_1\ a_1b_0d_0 \to e_1$$

$$DRC(y2): a_0b_0d_0 \to e_0\ a_1b_1d_1 \to e_0$$

$$DRC(y3): a_1b_1d_2 \to e_2\ a_2b_1d_2 \to e_2\ a_2b_2d_2 \to e_2$$

1) 对于第一个规则等价类

选取第一条规则 $a_1b_0d_1 \to e_1$，经计算其核值属性为 b，由于 $[1]_{b_0} = \{1,2,3\} \not\subset y1 = \{1,2\}$，即其核值属性不能决定该规则，需要进行进一步约简。

由于 $a_1b_0d_0 \to e_1$ 的核值属性为 a，所以等价类 $y1$ 的核值属性 $core(y1) = \{a,b\}$，则

$A1 = \{a\}$， $A2 = \{d\}$， $U/\{b,d\} = \{\{1\},\{2,3\},\{4\},\{5,6\},\{7\}\}$， $POS_{C-\{a\}}(D) = \{1,4,5,6,7\}$，

$w(a) = 5/7$， $U/\{a,b\} = \{\{1,2\},\{3\},\{4,5\},\{6\},\{7\}\}$， $POS_{C-\{d\}}(D) = \{1,2,3,6,7\}$， $w(d) = 5/7$

则 $OA = \{a,d\}$。

令 $B = \{a,b\}$， $[1]_{\{a_1,b_0\}} = \{1,2\} \subseteq y1 = \{1,2\}$，则输出规则： $a_1b_0 \to e_1$，将该规则从规则等价类中删除。又 $\{2\} \subseteq [1]_{\{a_1,b_0\}}$，将第 2 条规则从规则等价类中删除，该规则等价类约简完毕。

约简结果为： $a_1b_0 \to e_1$

2) 对于第二个规则等价类

选取第一条规则 $a_0b_0d_0 \to e_0$，经计算其核值属性为 a，由于 $[3]_{a_0} = \{3\} \subseteq y2 = \{3,4\}$，即核值属性可以决定该规则，则输出规则： $a_0 \to e_0$，将该规则从规则等价类中删除。

选取第二条规则 $a_1b_1d_1 \to e_0$，经计算其核值属性为 b 和 d，由于 $[4]_{\{b_1,d_1\}} = \{4\} \subseteq y2 = \{3,4\}$，即核值属性可以决定该规则，则输出规则： $b_1d_1 \to e_0$，将该规则从规则等价类中删除。

该规则等价类约简完毕，约简结果为： $a_0 \to e_0$， $b_1d_1 \to e_0$

3) 对于第三个规则等价类

选取第一条规则 $a_1b_1d_2 \to e_2$，经计算其核值属性为 d，由于 $[5]_{d_0} = \{5,6,7\} \subseteq y3 = \{5,6,7\}$，即核值属性可以决定该规则，则输出规则 $d_2 \to e_2$，将该规则从规则等价类中删除。又

$\{6,7\} \subseteq [5]_{d_0}$，将第 5，6 条规则从规则等价类中删除。

该规则等价类约简完毕，约简结果为： $d_2 \to e_2$

然后计算每条规则覆盖的记录数：规则 $a_1b_0 \to e_1$，覆盖记录 1、2，覆盖记录数为 2；规则 $a_0 \to e_0$，覆盖记录 3，覆盖记录数为 1；规则 $a_1b_1d_1 \to e_0$，覆盖记录 4，覆盖记录数为 1；规则 $d_2 \to e_2$，覆盖记录 4、5 和 6，覆盖记录数为 3。

约简结果如表 1 所示。

程序运行结果如图 4 所示，和预期结果一致。

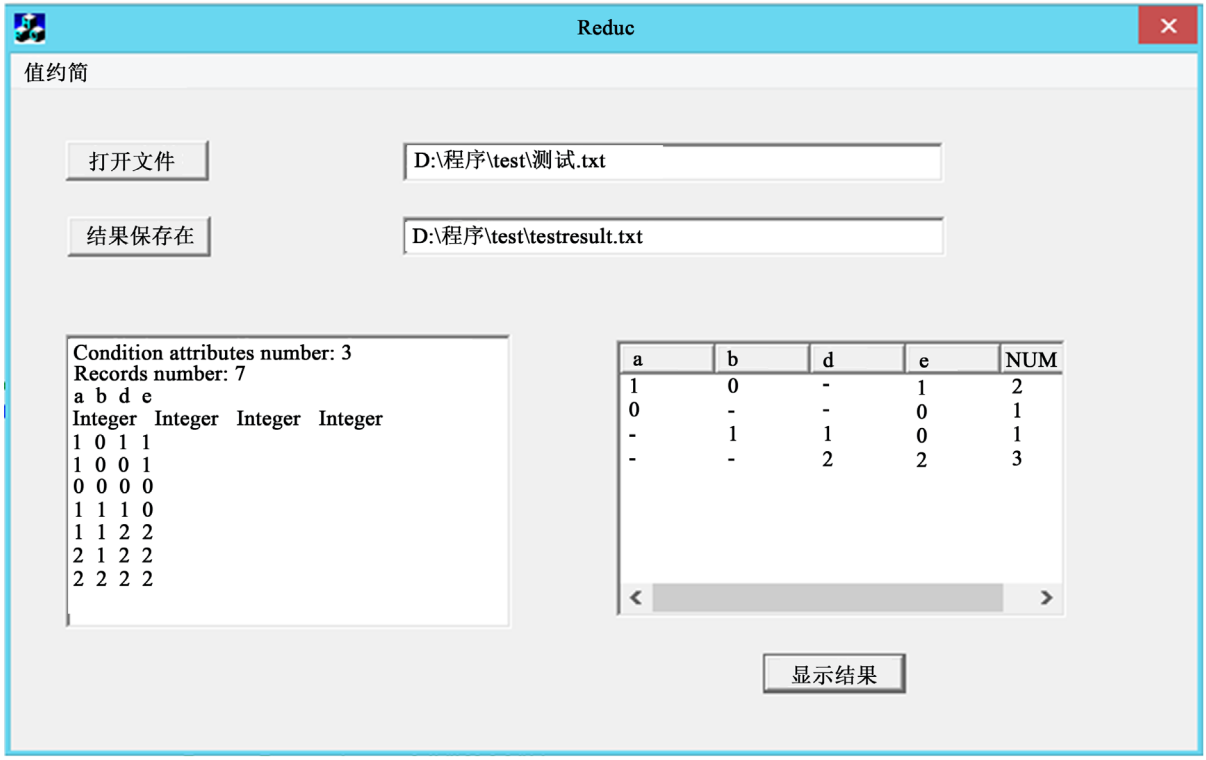

Figure 4. Reduction result
图 4. 约简结果图

Table 1. Reduction result
表 1. 约简结果

U	a	b	d	e
1, 2	1	0	×	1
4	×	1	1	0
5, 6, 7	×	×	2	2

其中：×表示可为任意值，不影响结果

4.2. 性能分析

表 2 记录了测试的情况，包括约简前的条件属性数和记录数，以及约简后的规则数和约简时间。

由测试记录可知，约简时间与条件属性数和记录数都有关系。约简时间随着记录数的增加而增加，为了清楚地展现其增加趋势，绘制了随着记录数的增加，约简时间的变化的折线图图 5。

由图 5 可知，随着记录数的增加，约简时间大致呈平方增加。分析其约简过程，发现删除重复记录的函数时间复杂度为 $O(n^2)$，程序主算法最小约简算法的时间复杂度也是 $O(n^2)$，所以，该程序的时间复杂度为 $O(n^2)$ (其中 n 指约简前的记录条数)。

4.3. 对比分析

现在用相同的数据测试归纳值约简和启发式值约简，对比约简结果，验证结果是否正确。

分别使用两种算法对几个不同的数据集进行约简，结果对比如表 3 所示。

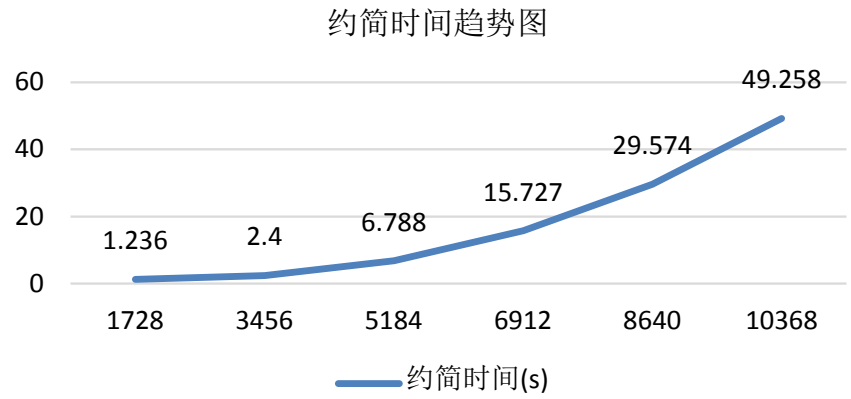

Figure 5. The tendency of reduction time
图 5. 约简时间趋势图

Table 2. Analysis of test time
表 2. 测试时间分析

测试编号	约简前		约简后	
	条件属性数	记录数	规则条数	约简时间
1	6	1728	246	1.236 s
2	6	3456	261	2.4 s
3	6	5184	423	6.788 s
4	6	6912	585	15.727 s
5	6	8640	747	29.574 s
6	6	10368	909	49.358 s
7	4	625	81	0.125 s
8	9	699	287	10.461 s

Table 3. The comparison of two reduction algorithms results
表 3. 两种算法约简结果对比

条件属性数	记录数	规则条数	
		归纳值约简	启发式值约简
3	75	8	8
4	625	81	81
9	699	287	435
6	1728	246	246

由上面的结果可知：归纳值约简和启发式值约简只是约简方法不同，但是约简结果是大致一致的。为了对比两种算法的时间效率，分别用两种算法计算同样的数据，记录数和约简时间如表 4 所示(其中这些数据的条件属性数为 6)。

为了更直观地反映其随着记录数增加，约简时间的变化趋势，绘制了图 6 所示折线图。

由图 6 可知：记录较少时两种算法所用时间差不多；随着记录数量的增加，归纳值约简的约简时间优于启发式值约简。

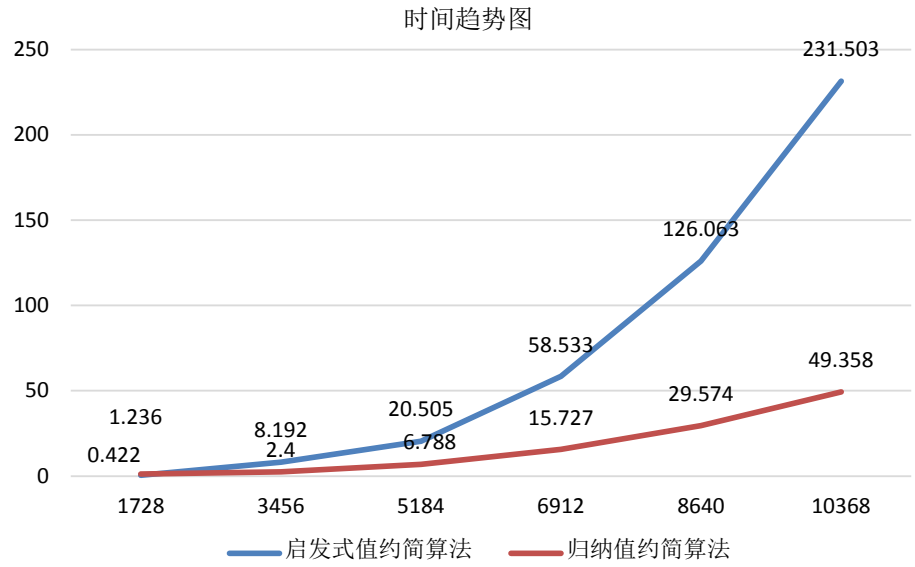

Figure 6. The time tendency comparison of two reduction algorithms
图 6. 两种约简算法的时间趋势对比图

Table 4. The comparison of two reduction algorithms time
表 4. 两种算法约简时间对比

测试记录条数	启发式值约简算法	归纳值约简算法
1728	0.422 s	1.236 s
3456	8.192 s	2.4 s
5184	20.505 s	6.788 s
6912	58.533 s	15.727 s
8640	126.063 s	29.574 s
10,368	231.503 s	49.358 s

5. 总结与展望

本文研究并实现了基础的基于归纳的值约简算法，它可以有效的去掉多余的属性值，在不改变决策能力的基础上得到更加简化的规则集，如此可以提高挖掘的效率，并帮助企业及用户更有效的挖掘需要的数据。

基金项目

本项目得到网络文化与数字传播北京市重点实验室开放课题资助；2017 实培计划(毕设)项目资助。

参考文献 (References)

[1] 张文修, 吴伟志, 梁吉业, 李德玉. 粗糙集理论与方法[M]. 北京: 北京科学出版社, 2001.
[2] 罗秋瑾, 陈世联. 基于值约简和决策树的最简规则提取算法[J]. 计算机应用, 2005, 25(8): 141-143.
[3] 林嘉宜, 彭宏, 郑启伦. 一种新的基于粗糙集的值约简算法[J]. 计算机工程, 2003, 29(4): 71-129.
[4] 杨振峰, 郭景峰, 常峰. 一种基于粗集的值约简方法[J]. 计算机工程, 2003, 29(9): 96-97.
[5] 刘艳丽, 王海涌, 郑丽英. 基于粗集理论的决策规则约简算法的研究与应用[J]. 兰州交通大学学报(自然科学版),

2004, 23(6): 78-111.
[6] 叶明凤. 基于核值的决策规则算法的研究[J]. 煤炭技术, 2014, 33(3): 257-259.
[7] 林嘉宜, 彭宏, 郑启伦. 一种新的基于粗糙集的值约简算法[J]. 计算机工程, 2003(4): 70-7l.
[8] 王珍, 余昭平. 一种基于粗糙集的最小约简算法[J]. 微计算机信息, 2006(22): 218-220.
[9] 王清毅, 范焱, 蔡庆生. 知识的约简研究[J]. 小型微型计算机系统, 2000, 21(6): 623-627.
[10] 顾军华, 周艳聪, 宋洁, 晏俊秋. 一种新的求解属性值约简算法[J]. 南开大学学报(自然科学版), 36(4): 38-42.