Hans Journal of Data Mining 数据挖掘, 2018, 8(1), 23-35
Published Online January 2018 in Hans. http://www.hanspub.org/journal/hjdm
https://doi.org/10.12677/hjdm.2018.81004

Hans汉斯

The Study and Implementation of Heuristic Value Reduction

Chengxia Liu[1,2], Limei 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: Jan. 12th, 2018; accepted: Jan. 24th, 2018; published: Jan. 31st, 2018

Abstract

Based on the research of rough set theory, this paper studies the process of heuristic value reduction. It usually constructs the decision table composed of the reduced attribute set and the decision attribute. Then, the heuristic information is used to perform the de-operation and delete the duplicate information. Finally, the new decision table works as the initial decision table and heuristic algorithm is used to judge whether the attribute values in the records are redundant or necessary. The redundant attribute values are deleted and the attribute values of the records are reduced to get the approximate minimum rule set. At last the test system is implemented.

Keywords

Heuristic Value Reduction, Approximate Minimum Rule Set, Rough Set

启发式值约简算法的研究与实现

刘城霞[1,2]，张李梅[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单位，北京
Email: cecilia7812@163.com

收稿日期：2018年1月12日；录用日期：2018年1月24日；发布日期：2018年1月31日

摘　要

在粗糙集理论的基础上，本文研究了启发式值约简的过程。本文研究的就是在属性约简完成后的启发式

文章引用: 刘城霞, 张李梅. 启发式值约简算法的研究与实现[J]. 数据挖掘, 2018, 8(1): 23-35.
DOI: 10.12677/hjdm.2018.81004

值约简算法，它一般先构造由约简属性集合和决策属性组成的决策表；然后利用一定的启发式信息对其进行去重操作，得到的新表将作为值约简的初始决策表；最后，判断记录中的各个属性值是否冗余，删除冗余属性值，对记录的属性值进行约简得到近似最小规则集。最终实现了其测试系统。

关键词

启发式值约简，最小规则集，粗糙集

1. 引言

如今社会中，无论是线下实际工作还是线上电子商务，时时刻刻都会产生数据，它的总量超过我们的想象。而在这大量的数据中，如何挖掘出隐含的、有利用价值的信息，是人类在智能信息处理方面遇到的空前未有的挑战，为此数据挖掘应运而生。粗糙集理论是由波兰数学家 Pawlak Z 在 1982 年提出的一种对不确定性数据分析的理论。该理论能够有效地对数据中隐含的、有利用价值的信息进行挖掘，它可以有效的指导在复杂系统在中进行的数据挖掘。

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。

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 $U=\{x1,x2,\cdots,xn\}$；$A=\{a1,a2,\cdots,an\}$ 是非空有限的属性集合；Va 是属性 a 的值域，即 $V=\cup Va$，$f:U\times A\to V$ 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。

如表 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。

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

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

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

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

$$core(y)\subseteq C,\ core(d_x)\subseteq C,\ \text{且}\ core(y)=\bigcup_{d_x\in DRC(y)}core(d_x)。$$

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. 启发式值约简算法

值约简算法很多学者都在研究，比如文献[2]-[10]，这里主要实现启发式值约简算法。

3.1. 算法步骤

对启发式值约简算法整体的基本思路分步骤概要说明如下：

算法的输入：信息系统 T，即假设信息系统 T：记录条数为 n，条件属性数为 m-1，决策属性数为 1。

算法的输出：T 的启发式值约简结果 T'。

第一步：对信息表中的各条件属性逐列进行考察。在删除其中的某一列后，如果有冲突记录出现，则保留原该属性值；否则，若出现了重复记录，则将该属性值标记为“*”；对于其他的记录，则将该属性值标记为“？”。

For(j = 1 To m-1)

For(i = 1 To n) {

If $\exists_k\left(k\ne i\wedge\forall_l\left(\left(l\ne j\wedge l\ne m\wedge T_{il}'\ne *\wedge T_{il}'\ne ?\right)\to T_{il}==T_{kl}\right)\wedge T_{im}\ne T_{km}\right)$

$T_{ij}'=T_{ij}$;

Else if $\exists_k\left(k\ne i\wedge\forall_l\left(l\ne j\wedge T_{il}'\ne *\wedge T_{il}'\ne ?\to T_{il}==T_{kl}\right)\right)$

$T_{ij}'=*$;

Else $T_{ij}'=?$;

}

For(i=1 To n) $T_{im}'=T_{im}$;

第二步：删除第一步完成后信息表中可能产生的重复记录，并对表中每条含有被标记“？”的条件属性的记录进行处理。如果只由未被标记的条件属性值就可以判断出决策属性值，则将标记“？”改为“*”；否则，将标记“？”改为原属性值；若存在某条记录的全部条件属性都被标记为“？”或“*”，则标记“？”改为原属性值。

For(j=1 To m-1)

For(i=1 To n) {

If $T'_{ij} == ?$ {

If $\forall_l \left(l \neq m \to \left(T'_{il} == ? \vee T'_{il} == * \right) \right)$

$T'_{ij} = T_{ij}$;

Else If $\forall_k \left(\forall_l \left(l \neq m \wedge T'_{il} \neq ? \wedge T'_{il} \neq * \to T_{il} == T_{kl} \right) \to T_{im} == T_{km} \right)$

$T'_{ij} = *$;

Else $T'_{ij} = T_{ij}$;

}

}

第三步：删除全部条件属性都被标记为“*”的记录和第二步完成后可能形成的重复记录(假定 $Card\left(T'\right) = n'$)。

第四步：在存在两条仅有一个条件属性值不一样，且其中一条记录的该条件属性被标记为“*”的记录的前提下，那么，对于该记录，如果仅由未被标记的条件属性值即可判断出决策属性值，则删除另外一条记录；否则，删除本记录。

For each tuple (i) in T' {

If $\exists_k \exists_l \left(l \neq m \wedge T'_{il} \neq T'_{kl} \wedge T'_{il} == * \wedge \forall_j \left(j \neq l \to T'_{ij} == T'_{kj} \right) \right)$ {

If $\forall_h \left(\forall_j \left(\left(j \neq m \wedge T'_{ij} \neq * \right) \to T_{hj} == T'_{ij} \right) \to T_{hm} == T'_{im} \right)$

除记录 k;

Else 删除记录 i;

}

Else If $\exists_k \exists_l \left(l \neq m \wedge T'_{il} \neq T'_{kl} \wedge T'_{kl} == * \wedge \forall_j \left(j \neq l \to T'_{ij} == T'_{kj} \right) \right)$ {

If $\forall_h \left(\forall_j \left(\left(j \neq m \wedge T'_{kj} \neq * \right) \to T_{hj} == T'_{kj} \right) \to T_{hm} == T'_{km} \right)$

删除记录 i;

Else 删除记录 k;

}

}

经过上述四个步骤的约简过程后得到的决策表，所有属性值都为该决策表的值核，即得到了启发式值约简后的规则表。

3.2. 算法实现

第一步：逐列删除表中各个条件属性，对产生的三种情况(冲突记录、重复记录、其他记录)进行处理，即分别进行标记或保留原属性值。其流程图如图 1 所示。

第二步：删除重复记录，并对存在被标记为“？”的条件属性的记录的三种情况分别进行处理。图 2 为其流程图。

第三步：删除重复记录及全部条件属性都为“*”的记录。流程图如图 3 所示。

第四步：对只有一个条件属性值不同，且其中一条为“*”的两条记录的情形，分情况进行处理。

在这个部分，判断哪条记录需要被删除，将要删除的规则即记录返回并存储，然后用与删除重复记录类似的方法从表中删除记录。流程图如图 4。

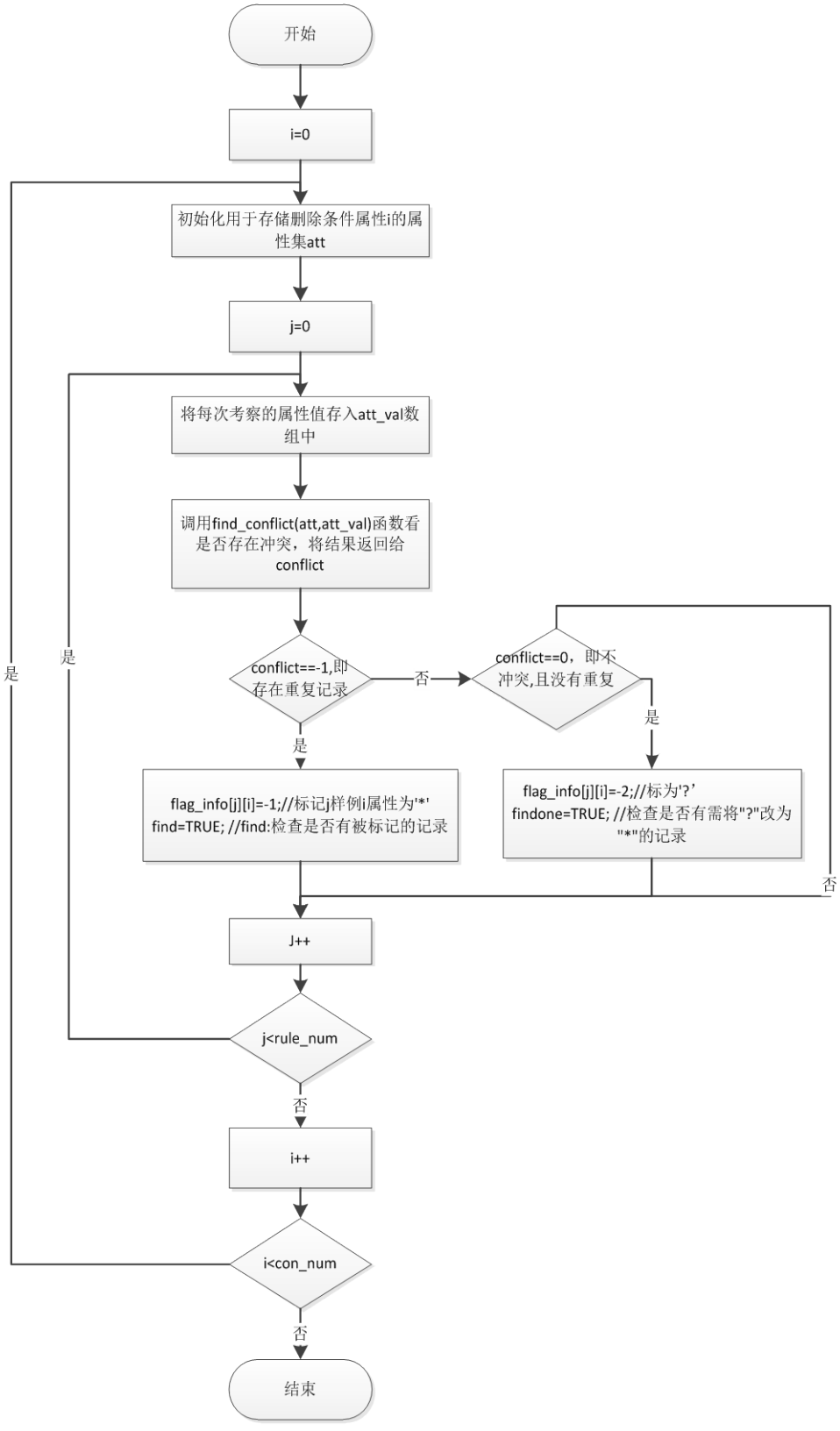

Figure 1. Diagram of deleting attribute
图 1. 删除属性流程图

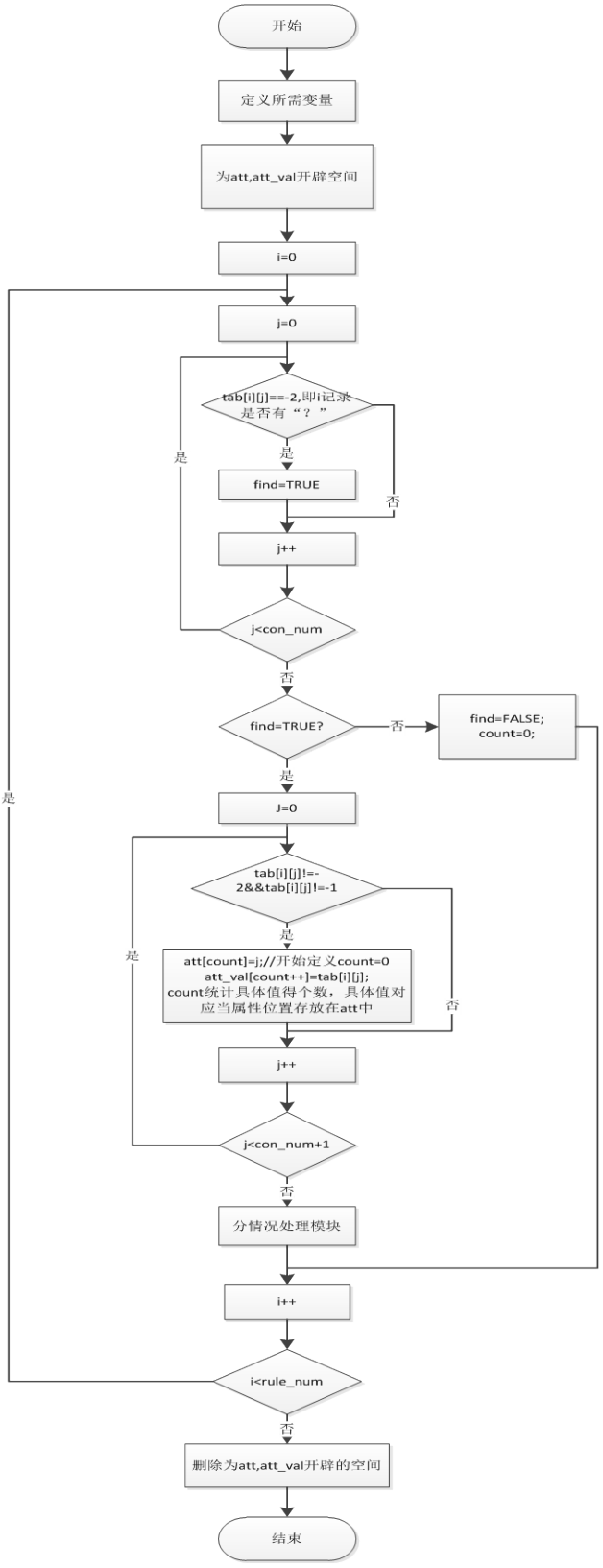

Figure 2. Diagram of deleting record “?”
图 2. 删除“？”记录流程图

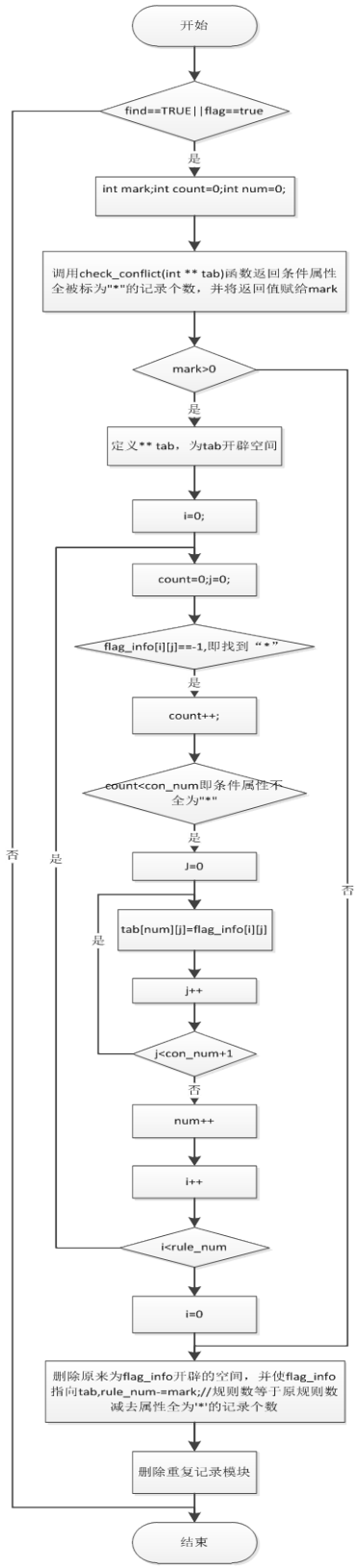

Figure 3. Diagram of deleting record “*”
图 3. 删除“*”记录流程图

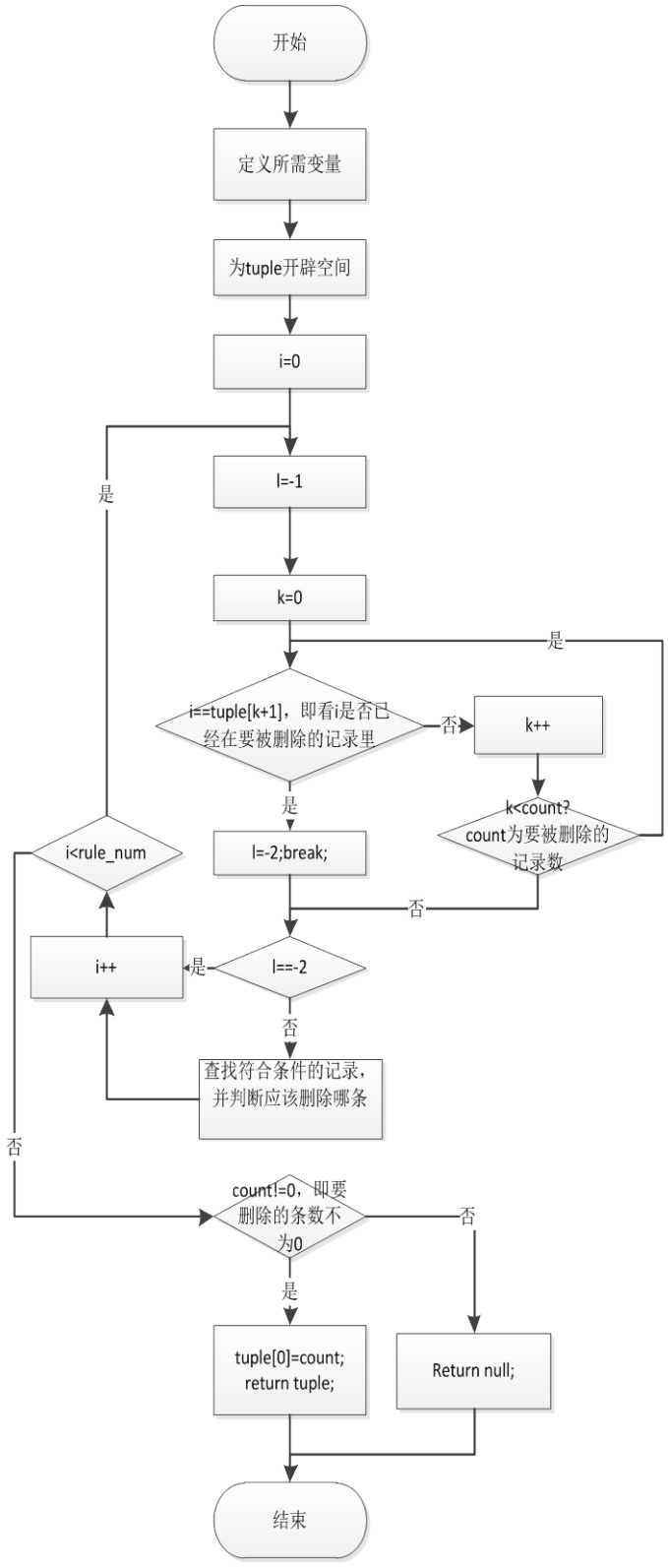

Figure 4. The diagram of deleting record which is * and has only one different attribute

图 4. 删除只有一个条件属性值不同的*记录流程图

4. 结果分析

4.1. 简单例子

为了测试系统的约简结果是否正确，可简单测试一个小例子，并手动计算约简结果与程序约简的结果进行对比。示例：此为车辆价值评估的一个决策表，其中 A、B、C、D、E、F 是车辆价值评估的评估标准，G 是车辆价值评估的结论，即条件属性：A、B、C、D、E、F，决策属性：G，表 2 为已经属性约简后的数据：

其中，

A: buying (购买价格): vhigh, high, med, low. (4, 3, 2, 1)

B: maint (维修价格): vhigh, high, med, low. (4, 3, 2, 1)

C: doors (车门数量): 2, 3, 4, 5more. (2, 3, 4, 5)

D: persons (车载人数): 2, 4, more. (2, 4, 5)

E: lug_boot (汽车后备箱型号): small, med, big. (1, 2, 3)

F: safety (安全性): low, med, high. (1, 2, 3)

G: class values (车辆价值): unacc, acc, good, vgood (1, 2, 3, 4）

根据启发式值约简算法的四个步骤，可得约简流程如下：

第一步：对信息表中的条件属性逐列进行删除，对产生的三种情况(冲突记录、重复记录、其他记录)进行处理，即分别进行标记或保留原属性值，得到表 3 结果。

1) 冲突记录----保存原该属性值；

2) 重复记录----将该属性值标记为“*”；

3) 其他剩余记录----将该属性值标记为“？”。

第二步：删除重复记录，并对含“？”标记的记录的三种情况分别进行处理，结果表如表 4。

1) 所有条件属性均被标记----标记“？”改为原属性值；

Table 2. Table after attribute reduction
表 2. 属性约简后的表

记录	A	B	C	D	E	F	G
1	3	3	5	5	3	2	2
2	3	3	5	5	3	3	2
3	3	2	2	2	1	1	1
4	3	2	2	2	1	2	1
5	3	2	2	2	1	3	1
6	3	2	2	2	2	1	1
7	2	1	5	5	2	3	4
8	2	1	5	5	3	1	1
9	2	1	5	5	3	2	3
10	2	1	5	5	3	3	4
11	1	4	2	2	1	1	1
12	1	4	2	2	1	2	1

2) 能只根据没有被标记的条件属性值就判断出决策属性值----标记“？”改为“*”；

3) 不能判断出决策----标记“？”改为原属性值。

第三步：删除重复记录及全部条件属性都为“*”的记录。

由上表所示可知，其中不存在重复记录，也不存在所有条件属性都被标记为“*”的记录，因此，第三步结果与第二步相同，第三步结果如表 4。

第四步：对只有一个条件属性的值不同，且其中一条为“*”的两条记录的情况，进行处理，约简结果如表 5。

1) 根据未被标记的属性值就能得到决策属性值----删除另一条记录；

2) 不能判断出决策属性值----删除本记录。

此表即为启发式值约简算法的最终约简结果。

Table 3. The reduction result of first step
表 3. 第一步约简结果

规则	A	B	C	D	E	F	G
1	?	?	?	?	?	*	2
2	?	?	?	?	?	*	2
3	?	?	?	?	*	*	1
4	?	?	?	?	?	*	1
5	?	?	?	?	?	*	1
6	?	?	?	?	*	?	1
7	?	?	?	?	*	?	4
8	?	?	?	?	?	1	1
9	?	?	?	?	?	2	3
10	?	?	?	?	*	3	4
11	?	?	?	?	?	*	1
12	?	?	?	?	?	*	1

Table 4. The reduction result of second and third step
表 4. 第二/三步约简结果

规则	A	B	C	D	E	F	G
1	3	3	5	5	3	*	2
2	3	2	2	2	*	*	1
3	3	2	2	2	1	*	1
4	3	2	2	2	*	1	1
5	2	1	5	5	*	3	4
6	*	*	*	*	*	1	1
7	2	1	5	5	3	2	3
8	*	*	*	*	*	3	4
9	1	4	2	2	1	*	1

Table 5. The reduction result of the forth step
表 5. 第四步约简结果

规则	A	B	C	D	E	F	G
1	3	3	5	5	3	*	2
2	3	2	2	2	*	*	1
3	2	1	5	5	*	3	4
4	*	*	*	*	*	1	1
5	2	1	5	5	3	2	3
6	1	4	2	2	1	*	1

Table 6. The comparison of different size data of heuristic value reduction
表 6. 不同规模数据的启发式值约简比较

记录条数	条件属性数	规则条数	约简率	约简时间
1728	6	247	14.3%	0.422s
3456	6	374	10.8%	8.192s
5184	6	561	10.8%	20.505s
6912	6	748	10.8%	58.533s
8640	6	935	10.8%	126.063s
10368	6	1122	10.8%	231.503s

Figure 5. The reduction time comparison of different data size
图 5. 约简时间对比图

4.2. 性能分析

为了更加直观的了解和感受启发式值约简算法的性能以及约简率，可从以下几个方面进行比较，得到表 6 比较数据：

为了更加直观的了解和感受启发式值约简算法，可根据上表 6 绘制折线图如图 5。

由上述数据可得出结论：随着约简前的数据量的逐步上升，启发式值约简算法的约简率是相对比较稳定的；随着约简前的数据量的逐步上升，启发式值约简算法所用的约简时间增加的幅度是相对较大的，曲线也是相对较陡的，曲线的斜率即约简时间增加率逐渐增大。

5. 总结与展望

本文研究并实现了基础的启发式值约简算法，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 2017 到网络文化与数字传播北京市重点实验室开放课题资助，课程建设“实培计划毕设(论文)项目”资助。

参考文献 (References)

[1] 张文修, 吴伟志, 梁吉业, 李德玉.粗糙集理论与方法第一版[M]. 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), 2003, 36(4): 38-42.