基于冲突域的高效属性约简算法

葛 浩[1)] 李龙澍[2)] 杨传健[3)]

[1)](滁州学院机械与电子工程学院 安徽 滁州 239012)

[2)](安徽大学计算机科学与技术学院 合肥 230039)

[3)](滁州学院计算机与信息工程学院 安徽 滁州 239012)

摘 要 引入冲突域的概念,研究冲突域的性质.以冲突域中冲突对象数目的变化为度量标准,给出核属性和属性重要性的计算方法,并设计了快速求解核属性和属性重要性的算法.在此基础上,给出高效属性约简算法,该算法以核属性为初始约简集,以属性重要性为启发式信息.在最坏情况下,算法的时间复杂度为 $O(|C|^2|U|)$,空间复杂度为 $O(|U|)$;实验结果表明,该算法是正确的、高效的.

关键词 决策表;粗糙集;属性约简;正区域;冲突域;核属性

中图法分类号 TP18 **DOI 号**: 10.3724/SP.J.1016.2012.00342

An Efficient Attribute Reduction Algorithm Based on Conflict Region

GE Hao[1)] LI Long-Shu[2)] YANG Chuan-Jian[3)]

[1)](*School of Mechanical and Electronic Engineering, Chuzhou University, Chuzhou, Anhui* 239012)

[2)](*School of Computer Science and Technology, Anhui University, Hefei* 230039)

[3)](*School of Computer and Information Engineering, Chuzhou University, Chuzhou, Anhui* 239012)

Abstract The notion of the conflict region is given, and natures of the conflict region are researched. The features of the core attributes and attribution importance, which are based on the change of the number of conflict objects, are studied. The algorithms for computing core attributes and attribute significance are designed. Based on these conditions, using core attributes as the initial reduction sets and the significance of attribute as heuristic information, an efficient attribute reduction algorithm is proposed. In the worst case, the time complexity and space complexity of the algorithm are $O(|C|^2|U|)$ and $O(|U|)$ respectively. The experimental results show that the algorithm is correct and efficient.

Keywords decision table; rough set; attribute reduction; positive region; conflict region; core attribute

1 引 言

粗糙集理论[1-2]是波兰数学家 Pawlak 教授于 1982 年提出的一种处理含糊和不精确性知识的数学工具,它能有效地分析和处理不精确、不一致、不完备的信息,从海量数据中发现隐含的知识.属性约简是粗糙集理论的核心内容之一.

属性约简是在保持知识库分类能力不变的条件下,删除不相关或不重要的冗余属性.现已经证明求

收稿日期:2009-02-15;最终修改稿收到日期:2011-10-12.本课题得到安徽省自然科学基金(050420204,090412054)、安徽高等学校省级自然科学研究项目基金(KJ2011Z276)、安徽省高等学校省级优秀青年人才基金(2011SQRL123)、滁州学院自然科学研究项目基金(2010kj014B,2011kj003Z)资助.**葛 浩**,男,1976 年生,硕士,副教授,中国计算机学会(CCF)会员,主要研究方向为数据挖掘和粗糙集.E-mail: togehao@126.com. **李龙澍**,男,1956 年生,教授,博士生导师,主要研究领域为不精确信息处理和智能软件. **杨传健**,女,1978 年生,硕士,副教授,主要研究方向为数据挖掘和粗糙集.

决策表所有约简和最优约简是一个 NP-Hard 问题[3]. 属性约简方法一般采用启发式算法，常用的启发式算法有三类：基于信息熵、基于可分辨矩阵和基于正区域. 苗夺谦等人[4]提出了基于互信息的MIBARK算法，其时间复杂度为 $O(|C||U|^2)+O(|U|^3)$，但不能保证算法的完备性；王国胤等人[5]给出基于信息熵的 CEBARKNC 算法，算法的时间复杂度为 $O(|C|^2|U|^2)$. Hu 等人[6]根据 Skowron 可分辨矩阵[7]提出一种属性约简算法，其时间复杂度和空间复杂度分别为 $O(|C|^2|U|^2)$ 和 $O(|C||U|^2)$，但因为没有考虑到决策表中存在不相容问题，Hu 的算法在处理不相容决策表时，不能保证获得正确的约简；刘文军等人[8]给出基于改进的可分辨矩阵的约简算法，可以很好地处理不相容问题，但该算法的时间和空间复杂度与 Hu 的算法是相同的；蔡卫东等人[9]和徐章艳等人[10]进一步改进了可分辨矩阵，给出了时间复杂度和空间复杂度为 $\max\{O(|C|^2(|U'_{pos}||U/C|)),O(|C||U|)\}$ 和 $\max\{O(|C|(|U'_{pos}||U/C|)),O(|C||U|)\}$ 的算法. 以上这两类启发式约简算法的时间复杂度均比较大，并且空间复杂度也不理想.

正区域的属性约简方法因不需要建立可分辨矩阵，其空间和时间开销相对较小，许多学者对这种方法做了大量研究. 叶东毅[11]给出对 Jelonek 约简算法的改进算法，其时间复杂度最坏情况下为 $O(|C|^2|U|^2)$. 由于基于正区域求解约简过程中，等价类划分是约简算法的关键步骤，因此刘少辉等人[12]采用了快速排序法对属性集排序后划分等价类，其时间复杂度为 $O(|C||U|\log|U|)$，由此设计的属性约简算法时间复杂度为 $O(|C|^2|U|\log|U|)$；徐章艳等人[13]对刘少辉等价类划分算法进行改进，采用链式基数排序算法划分等价类，其时间复杂度为 $O(|C||U|)$，使整个约简算法时间复杂度降低为 $\max\{O(|C||U|),O(|C|^2|U/C|)\}$，但在某些情况下该算法是不完备的.

针对目前约简算法中存在的不足：约简算法不完备、处理大数据集时的效率不够理想，本文提出一个基于冲突域的高效的完备属性约简方法. 由于等价类划分是本文约简算法中的重要操作，因而首先采用分布计数的基数排序思想，求解等价类，使求解 U/C 的时间复杂度降为 $O(|C||U|)$；然后给出冲突域的概念和相关性质，并提出基于冲突域的求核属性和属性重要性的方法；接着以核属性为初始约简集，以属性重要性为启发式信息，给出一个高效的属性约简算法，算法最坏情况下时间复杂度为 $O(|C|^2|U|)$，空间复杂度为 $O(|U|)$.

本文第 2 节介绍相关粗糙集理论；第 3 节提出一种分布计数的基数排序方法，并设计快速求解等价类算法，该算法提供给后面的操作使用；第 4 节给出了冲突域的性质和冲突域一般求解算法以及改进算法；在第 5 节中首先给出基于冲突域的快速求核方法，解决了因不相容性造成的求核错误，然后给出属性重要性的定义和性质，最后给出高效的属性约简算法；第 6 节通过一个实例说明本文的方法，并采用 UCI 数据库中数据集进行实验测试，且对实验结果进行分析；最后一节为全文总结.

2　基本概念

定义 1.　一个决策表可以定义为

$$S=(U,A,V,f),$$

其中，U 为论域，是对象的集合，$U=\{x_1,x_2,\cdots,x_n\}$；A 为属性集，$A=\{a_1,a_2,\cdots,a_m\}$，A 由两个部分组成 $A=C\cup D$ 且 $C\cap D=\varnothing$，C 为条件属性集，D 为决策属性集，一般情况下 D 中只含有一个属性 $D=\{d\}$；V 为属性的值域，$V=\{V_{a_1},V_{a_2},\cdots,V_{a_m}\}$；$f$ 为信息函数 $f:U\times A\to V$，$\forall a\in A,x\in U$，有 $f(x,a)\in V_a$.

定义 2.　对于决策表 $S=(U,A,V,f)$，令 $P\subseteq A$，$ind(P)=\{(x_i,x_j)\mid f(x_i,b)=f(x_j,b),\forall b\in P\}$ 称为 S 的不可区分关系. 显然不可区分关系为一个等价类，含 x 的等价类记为 $[x]_P$. P 在 U 上导出的划分记为 U/P.

定义 3.　对于决策表 $S=(U,A,V,f)$，令 $P\subseteq A$，$P\_X=\{x\in U\mid[x]_P\subseteq X\}$ 称为 X 的 P 下近似集；$P^-X=\{x\in U\mid[x]_P\cap X\neq\varnothing\}$ 称为 X 的 P 上近似集；$POS_P(X)=P\_X$ 称为 X 的 P 的正区域.

定义 4.　决策表 $S=(U,C\cup D,V,f)$ 中，$P\subseteq C$，称 $POS_P(D)=\bigcup\limits_{X\in U/D}P\_X$ 为 P 关于 D 的正区域.

定义 5.　决策表 $S=(U,C\cup D,V,f)$，若存在 $x_i,x_j\in U$，当 $i\neq j$ 时，若有 $f(x_i,C)=f(x_j,C)$ 且 $f(x_i,D)\neq f(x_j,D)$，则称该系统为不相容决策表，x_i 与 x_j 称为不相容对象. 否则称为相容决策表.

定理 1[14].　给定决策表 $S=(U,C\cup D,V,f)$，$\forall a\in C$ 为核的充分必要条件是 $POS_{C-\{a\}}(D)\neq POS_C(D)$.

3　等价类性质

在求解正区域和核属性的过程中，等价类划分是一个关键步骤. 求等价类的一般方法是对样本集 U 中未分类的对象进行两两比较，比较它们对条件

属性集 C 每个属性取值是否相同，如果相同，则属于同一个等价类. 上述方法等价类的划分的时间复杂度为 $O(|C||U|^2)$.

性质 1. 一个决策表 $S=(U,A=C\cup D,V,f)$，两个样本 $x_i,x_j\in U$ 相对于属性集 C 同属于一个等价类当且仅当 $\forall a\in A$，有 $f(x_i,a)=f(x_j,a)$.

由定义 2，可以得证.

根据性质 1，可先对决策系统 S 按属性集 C 排序，然后分析排序后的决策系统 S，划分等价类. 刘少辉等人[12]和赵军等人[15]使用了快速排序，使等价类划分算法的时间复杂度降低为 $O(|C||U|\log|U|)$，徐章艳等人[10,13]利用链式基数排序算法，将时间复杂度降低到 $O(|C||U|)$. 本文提出一种分布计数的基数排序方法，按属性集 C 对决策表 S 排序，该算法的时间复杂度也为 $O(|C||U|)$，空间复杂度为 $O(|U|)$. 该算法相对于徐章艳的方法更加易于处理决策信息系统.

对决策表采用分布计数的基数排序思想：设 $S=(U,C\cup D)$，其中 $C=\{a_i|i=1\cdots m\}$，$D=\{d\}$，决策表一行为一个数据对象，则 S 是数据对象的集合：$S=\{S_i|i=1\cdots n\}$，其中 S_i 为一个 $m+2$ 的元组：$S_i=(x_i,a_1,a_2,\cdots,a_m,d)$，其中，$S_i.x_i$ 为对象的编号，$S_i.a_j$ 表示对象 i 的 a_j 属性值，$S_i.d$ 表示对象 i 的决策属性值.

按照属性集 C 对 S 排序，即依次以每个属性 a_i 对 S 排序. 首先，把需要离散化的属性 a_i 离散化，将其分布在整型区间 $[1\cdots e]$（其中，$0<e\leq|U|$）；然后，构造一个计数表 $countPos[0\cdots e]$，$countPos$ 中元素个数为 $U/\{a_i\}$ 中等价类的个数，每个元素用于存放 $U/\{a_i\}$ 中每个等价类当前最后一个元素在有序决策表中的位置，根据 $countPos$ 表，可以直接将每个对象 S_i 放到有序决策表最终的位置. 在这个过程中，需要使用两个辅助空间：一个是 $countPos$；一个是存放有序决策表的 $sortedS$.

算法 1. 等价类划分算法.

输入：决策系统 $S=(U,C\cup D,V,f)$，$U=\{x_i|i=1\cdots n\}$，$C=\{a_i|i=1\cdots|U|\}$

输出：U/C

1. for $i=1$ to $|C|$ do //分布计数的基数排序
1.1. { 初始化 $countPos$ 表：$countPos[0\cdots e]=0$；
1.2. 对属性 a_i，统计 $U/\{a_i\}$ 中每个等价类中对象的个数，
1.3. 计算 $U/\{a_i\}$ 中每个等价类最后一个对象在有序决策表的位置，存放到 $countPos$ 中；
1.4. for $j=|U|$ to 1 do
1.4.1. { 根据 $S_j.a_i$ 的值，在 $countPos$ 表中找到 S_j 在有序决策表中的位置 pos；
1.4.2. 将 S_j 存入有序决策表 $sortedS$ 的第 pos 位置；
1.4.3. 修改等价类 $[S_j.a_i]_{\{a_i\}}$ 当前最后一个元素在有序表中的位置；}//end_for_j
}//end_for_i
2. $s=1$，$E_1=\{x_1\}$；
3. for $i=2$ to $|U|$ do
if($f(x_i,C)=f(x_{i-1},C)$) then $E_s=E_s\cup\{x_i\}$；
else $\{s=s+1,E_s=\{x_i\};\}$
4. 输出等价类集合 E（即 U/C）和等价类数目 s.

算法中，步 1 循环体的时间复杂度为 $O(|U|)$，循环次数为 $O(|C|)$，因而步 1 总的时间复杂度为 $O(|C||U|)$；步 3 的时间复杂度为 $O(|C||U|)$，故算法 1 的时间复杂度为 $O(|C||U|)+O(|C||U|)=O(|C||U|)$. 空间开销方面，在步 1 中辅助空间 $countPos$ 其容量最大为 $|U|+1$，步 3 中辅助空间 $sortedS$ 的容量可以控制为 $2|U|$. 因而，算法 1 的空间复杂度为 $O(|U|)$. 该算法思想将用于后面的算法中.

4 冲突域的性质及其算法

4.1 冲突域性质

定理 2. 在决策表 $S=(U,C\cup D,V,f)$ 中，设 $P\subseteq C$，则 $POS_P(D)=\bigcup\{Y|Y\in U/P\wedge Y\subseteq U/D\}$.

证明. 设 $U/P=\{Y_1,Y_2,\cdots,Y_m\}$，$U/D=\{X_1,X_2,\cdots,X_n\}$，若有 $Y_i\in POS_P(D)(1\leq i\leq m)$，设 $\exists x\in Y_i$，有 $[x]_P=Y_i\in POS_P(D)$，则 $[x]_D=X_j(1\leq j\leq n)\in U/D$. 根据正区域定义，$\forall x\in Y_i$ 有 $x\in P\_X_j$，即 $[x]_P\subseteq Y_j$. 因此，有 $POS_P(D)=\bigcup\{Y|Y\in U/P\wedge Y\subseteq U/D\}$. 证毕.

定义 6. 决策表 $S=(U,C\cup D,V,f)$ 中，$P\subseteq C$，P 关于 D 的冲突域记为 $ConSet(P)$，定义为

$$ConSet(P)=U/P-POS_P(D).$$

由定义 6 可知 $ConSet(P)$ 是对 S 按照条件属性集 P 划分后，冲突对象类的集合.

性质 2. 在决策表 $S=(U,C\cup D,V,f)$ 中，设 $P\subseteq C$，$ConSet(P)$ 为 P 关于 D 的冲突域，则对于 $\forall x_i\in ConSet(P)$，$\exists x_j\in ConSet(P)$，满足 $f(x_i,C)=f(x_j,C)\wedge f(x_i,D)\neq f(x_j,D)$.

由定义 6 和定理 2 可以得证.

4.2 冲突域一般求解算法

根据定义 6 和性质 2，可以得到 $ConSet(P)$ 的基本算法.

算法 2.　冲突域 $ConSet(R)$的基本算法.

输入：$S=(U,C\cup D,V,f)$中，设$\varnothing\neq P\subseteq C$

输出：$ConSet(P)$

1. $ConSet(P)=\varnothing$；
2. 依据算法 1，按属性 P 排序决策表 S，得 E 和 s；
   //E 是 U/P，s 是等价类的个数
3. for $i=1$ to s do
   if($\exists x_t x_k\in E_i$，有 $f(x_t,D)\neq f(x_k,D)$)
   then $ConSet(P)=ConSet(P)\cup E_i$；
4. 输出 $ConSet(P)$.

算法中，步 2 的时间复杂度为 $O(|P||U|)$，步 3 的时间复杂度 $O(|U|)$，则算法 2 总的时间复杂度为 $O(|P||U|)$，空间复杂度为 $O(|U|)$.

4.3　改进的冲突域求解算法

定理 3.　在决策表 $S=(U,C\cup D,V,f)$中，设 $P\subset R\subseteq C$，则 $POS_R(D)=POS_P(D)\bigcup\cup\{Z|Z\in Y/(R-P)\wedge Y\in U'/P\wedge Z\subseteq U'/D\}$，$ConSet(R)=ConSet(P)-\cup\{Z|Z\in ConSet(P)/(R-P)\wedge Z\subseteq U'/D\}$，其中 $U'=U-POS_P(D)$.

证明.　设 $U/P=\{Y_1,Y_2,\cdots,Y_t,Y_{t+1},\cdots,Y_m\}$，$POS_P(D)=\{Y_1,Y_2,\cdots,Y_t\}$，$ConSet(P)=\{Y_{t+1},\cdots,Y_m\}$，令 $Y_i\in POS_P(D)$，其中 $i=1,2,\cdots,t$；$Y_j\in U/P-POS_P(D)$，其中 $j=t+1,t+2,\cdots,m$；由于，U/R 是对 U/P 的加细，则

$$\begin{aligned}U/R&=U/(P\cup(R-P))\\&=\{Y_1/(R-P),Y_2/(R-P),\cdots,Y_t/(R-P),\\&\quad Y_{t+1}/(R-P),\cdots,Y_m/(R-P)\}\\&=\cup\{Z|Z\in Y/(R-P)\wedge Y\subseteq U/P\}.\end{aligned}$$

又有

$$\begin{aligned}&POS_R(D)=\cup\{Z|Z\in U/R\wedge Z\subseteq U/D\}\\&=\cup\{Z|Z\in Y/(R-P)\wedge Y\in U/P\wedge Z\subseteq U/D\}\\&=\cup\{Z|Z\in Y_i/(R-P)\wedge Y_i\in U/P\wedge Z\subseteq U/D\}\bigcup\\&\quad\cup\{Z|Z\in Y_j/(R-P)\wedge Y_j\in U/P\wedge Z\subseteq U/D\}\\&=POS_P(D)\bigcup\cup\{Z|Z\in Y_j/(R-P)\wedge\\&\quad Y_j\in U/P\wedge Z\subseteq U/D\}\\&=POS_P(D)\bigcup\cup\{Z|Z\in Y/(R-P)\wedge\\&\quad Y\in U'/P\wedge Z\subseteq U'/D\}.\end{aligned}$$

由于 $ConSet(R)=U/R-POS_R(D)$，则

$$\begin{aligned}&ConSet(R)=U/R-(POS_P(D)\bigcup\\&\quad\cup\{Z|Z\in Y/(R-P)\wedge Y\in U'/P\wedge Z\subseteq U'/D\})\\&=(U/R-POS_P(D))-\\&\quad\cup\{Z|Z\in Y/(R-P)\wedge Y\in U'/P\wedge Z\subseteq U'/D\}\\&=ConSet(P)-\cup\{Z|Z\in Y/(R-P)\wedge\\&\quad Y\in U'/P\wedge Z\subseteq U'/D\}\\&=ConSet(P)-\cup\{Z|Z\in ConSet(P)/\\&\quad(R-P)\wedge Z\subseteq U'/D\}.\end{aligned}$$

根据定理 3，若$\varnothing\neq P\subset R\subseteq C$ 则 $ConSet(R)$可以在 $ConSet(P)$的基础上运算求得，这样可以避免求解过程中的一些重复运算，从而提高算法效率.

进一步研究发现，在已知 $ConSet(C)$的前提下，$ConSet(R)$的计算可以在定理 3 的基础上进一步简化. 为了便于说明，下面先给出相关的定义.

定义 7.　在决策表 $S=(U,C\cup D,V,f)$中，设 $\varnothing\neq P\subset R\subseteq C$，$U'=U-POS_P(D)$，记 $U^*=U'-\cup\{Y'|Y'\in ConSet(P)\wedge\forall x\in Y'$有 $x\in ConSet(C)\}$，$ConSet'(P)=ConSet(P)-\cup\{Y'|Y'\in ConSet(P)\wedge\forall x\in Y'$有 $x\in ConSet(C)\}$.

定理 4.　在决策表 $S=(U,C\cup D,V,f)$中，设 $\varnothing\neq P\subset R\subseteq C$，已知 $ConSet(C)$，则 $ConSet(R)=ConSet(P)-\cup\{Z|Z\in ConSet'(P)/(R-P)\wedge Z\subseteq U^*/D\}$.

证明.　由定理 3，$ConSet(R)=ConSet(P)-\cup\{Z|Z\in ConSet(P)/(R-P)\wedge Z\subseteq U'/D\}$

$$\begin{aligned}&=ConSet(P)-\cup\{Z|Z\in(ConSet'(P)+\cup\{Y'|\\&\quad Y'\in ConSet(P)\wedge\forall x\in Y'\text{有 }x\in ConSet(C)\})/\\&\quad(R-P)\wedge Z\subseteq(U^*+\cup\{Y'|Y'\in ConSet(P)\wedge\\&\quad\forall x\in Y'\text{有 }x\in ConSet(C)\})/D\\&=ConSet(P)-\cup\{Z|Z\in(ConSet'(P)/(R-P)+\\&\quad\cup\{Y'|Y'\in ConSet(P)\wedge\forall x\in Y'\text{有 }x\in\\&\quad ConSet(C)\}/(R-P))\wedge Z\subseteq(U^*/D+\cup\{Y'|Y'\in\\&\quad ConSet(P)\wedge\forall x\in Y'\text{有 }x\in ConSet(C)\})/D\\&=ConSet(P)-\cup\{Z|Z\in ConSet'(P)/(R-P)\wedge\\&\quad Z\subseteq U^*/D\}.\end{aligned}$$

证毕.

定理 4 表明了，如果 $ConSet(P)$中某个等价类 Z'是 $ConSet(C)$中某个等价类或某几个等价类的并集，则 Z'中的元素也必然属于 $ConSet(R)$，在进行求解 $ConSet(R)$过程中则不需要对这样的等价类 Z'判断，直接放入 $ConSet(R)$中即可.

根据定理 4，下面给出改进的冲突域求解算法.

算法 3.　冲突域 $ConSet(R)$的改进算法.

输入：$S=(U,C\cup D,V,f)$中，设$\varnothing\neq P\subset R\subseteq C$，
$ConSet(P)$，$ConSet(C)$

输出：$ConSet(R)$

1. 根据 $ConSet(C)$对 $ConSet(P)$遍历，得 $ConSet'(P)$；
2. $ConSet(R)=ConSet(P)-ConSet'(P)$；
3. 对 $ConSet'(P)$的每个等价类 E_i按属性 $R-P$ 排序，得 E'和 s'；
4. for $j=1$ to s' do
   if($\exists x_t x_k\in E'_j$，有 $f(x_t,D)\neq f(x_k,D)$)
   then $ConSet(R)=ConSet(R)\cup E'_j$；
5. 输出 $ConSet(R)$.

算法 3 中，步 1 和步 2 的时间复杂度为 $O(|U-POS_P(D)|$，步 3 时间复杂度为 $O(|R-P||U-$

$POS_P(D)|)$，步 4 的时间复杂度为 $O(|U-POS_P(D)|$，因此算法 3 总的时间复杂度为 $O(|R-P||U-POS_P(D)|)$，空间复杂度为 $O(|U|)$.

5 约简算法设计和分析

Hu 等人[6]和赵军等人[15]的求核方法对于相容的决策表处理是正确的，但对不相容决策表却不能保证获得正确的核. 而采用启发式方法进行属性约简，通常是以核属性为初始约简集，下面先研究核属性的求解.

5.1 快速求核算法

针对决策表中存在的不相容性，下面给出一种求核方法，该方法建立在冲突域的基础上.

定义 8. 决策表 $S=(U,C\cup D,V,f)$ 中，$a\in C$，核属性集 $GCore(C)$ 表示为

$$GCore(C)=\begin{cases}\{a|a\in C\}, & |ConSet(C-\{a\})|>|ConSet(C)|\\ \varnothing, & \text{其它}\end{cases}.$$

$|ConSet(C)|$ 表示 $ConSet(C)$ 中冲突对象的数目.

定义 8 说明了，若删除某个属性 a 后，如果冲突对象的个数增加了，说明 a 为核属性；否则 a 不是核属性. 该方法避免了采用正区域方法中的一些繁琐操作，减少了计算量；也避免了采用可分辨矩阵方法中建立可分辨矩阵所需的大量时间和空间开销.

定理 5. 对于决策表 $S=(U,C\cup D,V,f)$，有 $Core(C)=GCore(C)$.

证明. 设 $U/C=\{Y_1,Y_2,\cdots,Y_m\}$，$U/D=\{X_1,X_2,\cdots,X_n\}$.

(1) 首先，证明 $GCore(C)\subseteq Core(C)$.

对 $\forall a\in GCore(C)$，由定义 8 可知，删除属性 a 后，$ConSet$ 中冲突对象个数增加了. 设 $x\notin ConSet(C)$，$x\in ConSet(C-\{a\})$，则存在 $y\notin[x]_C$ 且 $y\in[x]_{C-\{a\}}$，有 $f(x,D)\neq f(y,D)$. 设 $x\in X_q$，$y\in X_p$，$y\notin X_q$，由于 $y\in[x]_{C-\{a\}}$ 并且 $y\notin Y_q$，由下近似定义知 $x\notin(C-\{a\})\_X_q$，故 $x\notin POS_{C-\{a\}}(D)$. 又因 x 在原来系统 S 中无冲突对象且 $x\in Y_q$，由于 $x\in C\_Y_q$，故 $x\in POS_C(D)$，于是 $POS_C(D)\neq POS_{C-\{a\}}(D)$. 所以 $a\in Core(C)$. 因此，$GCore(C)\subseteq Core(C)$.

(2) 然后，证明 $Core(C)\subseteq GCore(C)$.

对 $\forall a\in Core(C)$，有 $POS_C(D)\neq POS_{C-\{a\}}(D)$，则 $\exists x$ 使得 $[x]_C\subseteq X_k$ 且 $[x]_{C-\{a\}}\not\subset X_k$. 于是 $\exists y,y\in[x]_{C-\{a\}}$ 且 $y\notin[x]_C$. 有 $f(x,C-\{a\})=f(y,C-\{a\})$ 且 $f(x,D)\neq f(y,D)$. 因此，删除属性 a 后，会产生冲突. 下面要证明 x 在原来系统 S 中为非冲突对象. 采用反证法证明. 假设 $\exists z,z,x\in[x]_C\subseteq X_k$ 且 $f(x,D)\neq f(z,D)$. 因此，x,z 属于 U/D 的不同等价类，此与 $z,x\in[x]_C\subseteq X_k$ 相矛盾. 所以，x 在原来系统 S 中没有与之相冲突的对象. 因此删除 a 后，新增加了冲突对象，也就是 $|ConSet(C-\{a\})|>|ConSet(C)|$，即 $a\in GCore(C)$. 故 $Core(C)\subseteq GCore(C)$.

由(1)(2)得 $Core(C)=GCore(C)$. 证毕.

由定理 5，给出一个快速求核算法.

算法 4. 快速求核算法.

输入：决策表 $S=(U,A,V,f)$，其中 $A=C\cup D$ 且 $C\cap D=\varnothing$，C 为条件属性集，D 为决策属性集

输出：决策表的核 $Core(C)$

1. $Core(C)=\varnothing$，$ConSet(C)=\varnothing$；
2. 根据算法 2，求得 $ConSet(C)$；
3. for $i=1$ to $|C|$ do
3.1. {根据算法 2，求得 $ConSet(C-\{a_i\})$；
3.2. if($|ConSet(C-\{a_i\})|>|ConSet(C)|$)
then $Core(C)=Core(C)\cup\{a_i\}$；
} //end_for_i
4. 输出 $Core(C)$.

算法 4 中，步 2 的时间复杂度为 $O(|C||U|)$. 步 3.1 的时间复杂度为 $O(|C||U|)$，步 3 循环次数为 $|C|$ 次，则步 3 的时间复杂度为 $O(|C|^2|U|)$. 因此，算法 4 的时间复杂度为 $O(|C||U|)+O(|C|^2|U|)=O(|C|^2|U|)$，空间复杂度为 $O(|U|)$.

5.2 属性重要性

定义 9. 在决策表 $S=(U,C\cup D,V,f)$ 中，C 为条件属性，D 为决策属性，$R\subset C$，$a\in C-R$，$U'=U-POS_R(D)$，则 $POS'_a(D)$ 定义为 $POS'_a(D)=\cup\{Z|Z\in U'/\{a\}\wedge Z\subseteq U'/D\}$.

定义 10[12]. 在决策表 $S=(U,C\cup D,V,f)$ 中，C 为条件属性，D 为决策属性，$R\subset C$，$a\in C-R$，则属性 a 的重要性 $SGF(a,R,D)$ 定义为 $SGF(a,R,D)=\gamma_{R\cup\{a\}}-\gamma_R$. 其中，$\gamma_R=|POS_R(D)|/|U|$.

定义 11. 在决策表 $S=(U,C\cup D,V,f)$ 中，C 为条件属性，D 为决策属性，$R\subset C$，$a\in C-R$，则属性 a 的重要性 $Sig(a,R,D)$ 定义为 $Sig(a,R,D)=|ConSet(R)|-|ConSet(R\cup\{a\})|$.

定理 6. 在决策表 $S=(U,C\cup D,V,f)$ 中，$\varnothing\neq R\subset C$，$a\in C-R$，$SGF(a,R,D)$ 与 $Sig(a,R,D)$ 是等价的.

证明. 设 $U'=U-POS_R(D)$，则

$$SGF(a,R,D)=\gamma_{R\cup\{a\}}-\gamma_R$$
$$=|POS_{R\cup\{a\}}(D)|/|U|-|POS_R(D)|/|U|$$
$$=(|POS_{R\cup\{a\}}(D)|-|POS_R(D)|)/|U|$$
$$=(|POS_R(D)\bigcup\bigcup\{Z|Z\in Y/\{a\}\wedge Y\in U/R\wedge Z\subseteq U/D\}|-|POS_R(D)|)/|U|$$
$$=|\bigcup\{Z|Z\in Y/\{a\}\wedge Y\in U'/R\wedge Z\subseteq U'/D\}|/|U|,$$
$$Sig(a,R,D)=|ConSet(R)|-|ConSet(R\bigcup\{a\})|$$
$$=|U-POS_R(D)|-|U-POS_{R\cup\{a\}}(D)|$$
$$=|POS_{R\cup\{a\}}(D)-POS_R(D)|$$
$$=|POS_R(D)\bigcup\bigcup\{Z|Z\in Y/\{a\}\wedge Y\in U/R\wedge Z\subseteq U/D\}-POS_R(D)|$$
$$=|\bigcup\{Z|Z\in Y/\{a\}\wedge Y\in U'/R\wedge Z\subseteq U'/D\}|,$$

可见，$SGF(a,R,D)$与$Sig(a,R,D)$是等价的. 证毕.

根据定理 6，给出属性重要性的算法.

算法 5. 属性重要性 $Sig(a,R,D)$算法.

输入：$S=(U,C\cup D,V,f)$中，设 $P\subset R\subseteq C$，$ConSet(P)$

输出：$Sig(a,R,D)$

算法 5 可以由算法 3 得到，在算法 3 中只需要令 $R-P=\{a\}$，即可获得. 算法略.

因为 $R-P=\{a\}$，则算法 5 的时间复杂度为 $O(|U-POS_P(D)|)$.

5.3 属性约简算法

定理 7. 决策 $S=(U,C\cup D,V,f)$中，设 $R\subseteq C$，如果 R 是决策表 S 的约简当且仅当$|ConSet(R)|=|ConSet(C)|$.

证明. 只要证明$|ConSet(R)|=|ConSet(C)|\Leftrightarrow POS_R(D)=POS_C(D)$，即可.

(1) 必要性，即 $POS_R(D)=POS_C(D)\Rightarrow|ConSet(R)|=|ConSet(C)|$.

$POS_C(D)=POS_R(D)$，必有$|POS_R(D)|=|POS_C(D)|$，则$|U-POS_R(D)|=|U-POS_C(D)|$，即$|ConSet(R)|=|ConSet(C)|$.

(2) 充分性，即$|ConSet(R)|=|ConSet(C)|\Rightarrow POS_R(D)=POS_C(D)$.

① 由于 $R\subseteq C$，则有 $POS_R(D)\subseteq POS_C(D)$.

② 假设，$POS_R(D)\supseteq POS_C(D)$ 不成立，则 $POS_R(D)\subset POS_C(D)$. 则至少 $\exists x_0\in U$，有 $x_0\in POS_C(D)$，但 $x_0\notin POS_R(D)$. 因此有 $x_0\in U-POS_R(D)$，但$x_0\notin U-POS_C(D)$，则$|U-POS_R(D)|\neq|U-POS_C(D)|$. 又因 $POS_R(D)\subset POS_C(D)$，从而$|POS_R(D)|<|POS_C(D)|$，则$|U-POS_R(D)|>|U-POS_C(D)|$，即$|ConSet(R)|>|ConSet(C)|$，此与$|ConSet(R)|=|ConSet(C)|$相矛盾. 故$POS_R(D)\subseteq POS_C(D)$.

由①②得 $POS_R(D)=POS_C(D)$.

由(1)(2)定理 7 得证. 证毕.

根据定理 7，同时综合前面的算法，下面给出高效的属性约简算法.

算法 6. 决策表属性约简算法.

输入：决策 $S=(U,C\cup D,V,f)$，$U=\{x_i|i=1\cdots n\}$，$C=\{a_i|i=1\cdots|C|\}$为条件属性集，D 为决策属性集

输出：约简属性集 R

1. 根据算法 4，求得 $ConSet(C)$和核 $Core(C)$，令 $R=Core(C)$；
2. if($R=\varnothing$) then $ConSet(R)=U$；
   else 根据算法 2，求得 $ConSet(R)$；
3. while($|ConSet(R)|\neq|ConSet(C)|$)
3.1. {for $i=1$ to $|C-R|$ do
   根据算法 3，计算 $ConSet(R\bigcup\{a_i\})$，获得 $Sig(a_i,R,D)$；
3.2. 选择最大 $Sig(a_i,R,D)$的 a_i；如果这样的 a_i有多个，在 $ConSet(R)/\{a_i\}$中选择具有最少等价类个数的 a_i，若这样的 a_i也有多个，则任选一个；
3.3. $R=R+\{a_i\}$；
3.4. 更新 $ConSet(R)$；
   }//end_while
4. for $i=1$ to $|R-Core|$ do
   {$R'=R-\{a_i\}$；
   在 $ConSet(C)$的基础上，采用算法 3 得 $ConSet(R')$；
   if($|ConSet(R')|=|ConSet(C)|$) then $R=R'$；
   }
5. 输出 R.

算法 6 中，步 1 的时间复杂度为 $O(|C|^2|U|)$；步 2 最坏情况下的时间复杂度为 $O(|C||U|)$；步 3.1 的时间复杂度为 $O(|C-R||U-POS_R(D)|)$，步 3.2 的时间复杂度为 $O(|U-POS_R(D)|)$，步 3.4 的时间复杂度为 $O(1)$，步 3 内部循环体循环的次数最大为 $|C-R|$ 次，则步 3 的时间复杂度为 $O(|C-R|^2|U-POS_P(D)|)$；步 4 最坏情况下时间复杂度为 $O(|C|^2|U|)$. 因此，算法 6 总的时间复杂度最坏情况下为 $O(|C|^2|U|)$，空间复杂度为 $O(|U|)$.

6 实例和实验分析

6.1 实例分析

表 1 为一个决策表，表中共有 11 个样本对象，其中$\{c1,c2,c3,c4,c5\}$为条件属性集 C，D 为决策属性.

表 1 一个决策表

U	$c1$	$c2$	$c3$	$c4$	$c5$	D
$x1$	0	0	1	1	1	1
$x2$	0	0	1	1	1	0
$x3$	0	1	1	1	1	0
$x4$	0	1	1	1	1	1
$x5$	1	1	0	1	1	1
$x6$	1	1	0	1	1	0
$x7$	0	1	0	0	0	0
$x8$	0	1	1	0	0	0
$x9$	1	0	0	1	1	1
$x10$	1	1	0	0	1	1
$x11$	1	0	1	0	1	0

分析决策表表 1,可知 $x1$ 和 $x2$、$x3$ 和 $x4$、$x5$ 和 $x6$ 分别是冲突对象,因此该决策表为不相容决策表.

由于$ConSet(C)=\{\{x1,x2\},\{x3,x4\},\{x5,x6\}\}$, $|ConSet(C)|=6$. 分别计算

$ConSet(C-\{c1\})=\{\{x1,x2\},\{x3,x4\},\{x5,x6\}\}$, $|ConSet(C-\{c1\})|=6$;

$ConSet(C-\{c2\})=\{\{x1,x2,x3,x4\},\{x5,x6,x9\}\}$, $|ConSet(C-\{c2\})|=7$;

$ConSet(C-\{c3\})=\{\{x1,x2\},\{x3,x4\},\{x5,x6\}\}$, $|ConSet(C-\{c3\})|=6$;

$ConSet(C-\{c4\})=\{\{x1,x2\},\{x3,x4\},\{x5,x6,x10\}\}$, $|ConSet(C-\{c4\})|=7$;

$ConSet(C-\{c5\})=\{\{x1,x2\},\{x3,x4\},\{x5,x6\}\}$, $|ConSet(C-\{c5\})|=6$.

只有 $|ConSet(C-\{c2\})|>|ConSet(C)|$ 和 $|ConSet(C-\{c4\})|>|ConSet(C)|$,因此 $Core(C)=\{c2,c4\}$,令 $R=\{c2,c4\}$. 则 $ConSet(R)=\{\{x1,x2,x9\},\{x3,x4,x5,x6\},\{x7,x8,x10\}\}$, $|ConSet(R)|=10$. 接着,以核属性集为初始约简集 R,以 $Sig(a_i,R,D)$ 为启发式信息选择属性进行约简($i=1,3,5$),分别对 $c1,c3,c5$ 采用算法 3,在 $ConSet(R)$ 基础上进一步计算,得

$ConSet(R\cup\{c1\})=\{\{x1,x2\},\{x3,x4\},\{x5,x6\}\}$, $|ConSet(R\cup\{c1\})|=6$;

$ConSet(R\cup\{c3\})=\{\{x1,x2\},\{x3,x4\},\{x5,x6\},\{x7,x10\}\}$, $|ConSet(R\cup\{c3\})|=8$;

$ConSet(R\cup\{c5\})=\{\{x1,x2,x9\},\{x3,x4,x5,x6\}\}$, $|ConSet(R\cup\{c5\})|=7$;

则

$Sig(c1,R,D)=|ConSet(R)|-|ConSet(R\cup\{c1\})|=4$,

$Sig(c3,R,D)=|ConSet(R)|-|ConSet(R\cup\{c3\})|=2$,

$Sig(c5,R,D)=|ConSet(R)|-|ConSet(R\cup\{c5\})|=1$.

可见 $Sig(c1,R,D)$ 的值最大,故选择属性 $c1$ 加入 R 中,有 $R=\{c1,c2,c4\}$, $ConSet(R)=\{\{x1,x2\},\{x3,x4\},\{x5,x6\}\}$;此时 $|ConSet(R)|=|ConSet(C)|$. 最后,对 R 中非核属性 $\{c1\}$ 反向消除检查有 $|ConSet(C)|\neq|ConSet(R-\{c1\})|$,因此属性约简为 $R=\{c1,c2,c4\}$. 如果采用文献[13]的方法得到约简属性集为 $\{c1,c2,c3,c4,c5\}$,可见该约简集中 $c2,c5$ 为冗余属性. 文献[13]的方法错误的原因是忽略了最重要的核属性,没有从核属性出发求解约简;并且,约简结束后没有对约简集中的非核属性进行反向消除冗余检查.

6.2 实验比较

采用本文决策表 1 和 UCI 数据库中 6 个决策表为实验数据,对 3 个约简算法进行测试(实验环境为 Petium4 2.8GHz, RAM512MB 微机, VS. NET 2005 平台的 VC++),结果如表 2(刘少辉的属性约简算法记为算法 A,文献[12]的属性约简算法记为算法 B,本文的属性约简算法记为算法 C;DT1 表示本文决策表 1;$|Core|$ 表示核属性的数目).

表 2 3 个约简算法的比较

决策表	$\|U\|$	$\|C\|$	$\|Core\|$	最小约简属性数目	算法 A			算法 B			算法 C		
					约简中属性数	是否含有冗余属性	执行时间/ms	约简中属性数	是否含有冗余属性	执行时间/ms	约简中属性数	是否含有冗余属性	执行时间/ms
DT1	11	5	2	3	3	No	1.75	5	Yes	0.09	3	No	0.08
Patient	90	8	8	8	8	No	10.41	8	No	1.58	8	No	0.73
Monks-1	432	6	3	3	3	No	16.56	4	Yes	2.97	3	No	1.32
Vote	435	16	7	9	9	No	70.91	10	Yes	24.05	9	No	18.12
Tic-Tac-Toe	958	9	0	8	8	No	101.91	8	No	17.90	8	No	23.57
Led17	2000	22	14	18	18	No	912.34	22	Yes	452.13	18	No	226.86
Poker	25010	10	5	7	7	No	3143.32	7	No	902.27	7	No	760.57

分析表 2 可以发现,算法 C 和算法 B 的执行时间要远低于算法 A,并且随着数据集的增大,这种优越性更明显.

对算法 C 和算法 B 的比较发现,算法 C 花费时

间要普遍低于算法 B，并且随着数据集规模的增大，优越性也越明显. 深入分析表 2 可以发现一个规律：当约简集与核属性集属性数目比值较大时，算法 B 的性能要优于算法 C；当约简集与核属性集属性数目比值较小(<2)时，算法 C 的性能要明显好. 出现这种结果的原因是：在算法 C 的最后阶段(即本文算法 6 中的步 4)对非核属性进行了反向消除检查，增加了一些计算量所造成的. 例如，数据集 Patient 和 Monks-1 的约简集和核属性集是相同的，按照本文的约简算法，当求出核属性集以后就等于约简算法完成了(也就是，算法 6 中步 2 完成后，整个约简算法就等于结束了)，因此算法 C 的性能明显优于算法 B. 而在处理数据集 Tic-Tac-Toe 时，算法 C 花费的时间却大于算法 B，这是因为：Tic-Tac-Toe 数据集中核属性数目为 0，约简集中 8 个属性均为非核属性，因此在算法 C 的最后阶段(即，算法 6 的步 4)需要对约简集中所有属性进行反向消除检查，这必然要花费一些时间. 如果算法 B 最后也加上消除冗余操作的话，在处理数据集 Tic-Tac-Toe 总的时间开销将增加到 31.86 ms，这个时间将大于算法 C 的 23.57 ms.

另外，分析表中属性约简结果可以看出，算法 C 和算法 A 是完备的，并且获得的属性约简是最小约简. 但算法 B 却不一定能保证约简的完备性，由表 2 可以看出，算法 B 在处理决策表 DT1、Monks-1、Vote 和 Led17 时，所求的约简中均存在冗余的属性. 这是因为算法 B 没有从核属性着手进行约简求解，并且在获得约简后没有对非核属性进行冗余性检查.

7　结　论

属性约简是粗糙集理论一个重要研究内容. 在属性约简中，等价类划分是一个关键步骤. 本文首先研究了等价类的性质，利用分布计数的基数排序方法，对决策表排序，设计了对属性集 C 进行快速等价类划分算法，其时间复杂度为 $O(|C||U|)$. 然后，给出冲突域的定义，讨论冲突域的性质，以冲突域中对象的个数的变化作为核属性和属性重要性判断的依据，并设计求核和属性重要性的算法. 在上述基础上，设计属性约简算法，算法以核属性为初始约简集，不断将重要性大的属性加入约简集中，并对非核属性进行反向消除检查，以保证约简的完备性. 该算法的时间复杂度为 $O(|C|^2|U|)$，空间复杂度为 $O(|U|)$. 实验结果表明该算法是正确的、高效的.

参　考　文　献

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

[2] Zhang Wen-Xiu, Wu Wei-Zhi, Liang Ji-Ye, Li De-Yu. Theory and Method of Rough Set. Beijing: Science Press, 2001 (in Chinese)
(张文修，吴伟志，梁吉业，李德玉. 粗糙集理论与方法. 北京：科学出版社，2001)

[3] Wong S K M, Ziarko W. Optimal decision rules in decision table. Bulletin of Polish Academy of Sciences, 1985, 33(11-12): 693-696

[4] Miao Duo-Qian, Hu Gui-Rong. A heuristic algorithm for knowledge reduction. Computer Research and Development, 1999, 36(6): 681-684(in Chinese)
(苗夺谦，桂荣. 识约简的一种启发式算法. 计算机研究与发展，1999，36(6)：681-684)

[5] Wang Guo-Yin, Yu Hong, Yang Da-Chun. Decision table reduction based on conditional information entropy. Chinese Journal of Computers, 2002, 25(7): 759-766(in Chinese)
(王国胤，于洪，杨大春. 基于条件信息熵的决策表约简. 计算机学报，2002，25(7)：759-766)

[6] Hu X H, Cercone N. Learning in relational databases: A Rough Set approach. Computational Intelligence, 1995, 11(2): 323-337

[7] Skowron A, Rauszer C. The discernibility matrices and functions in information systems//Slowinski R. Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic Publishers, 1991: 331-362

[8] Liu Wen-Jun, Gu Yun-Dong, Feng Yan-Bin, Wang Jia-Yi. An improved attribute reduction algorithm of decision table. Pattern Recognition and Artificial Intelligence, 2004, 17(1): 119-123(in Chinese)
(刘文军，谷云东，冯艳宾，王家银. 基于可辨别矩阵和逻辑运算的属性约简算法. 模式识别与人工智能，2004，17(1)：119-123)

[9] Cai Wei-Dong, Li Fan, Xu Zhang-Yan, Yang Bing-Ru. An efficient attribute reduction algorithm by modificatory discernibility matrix. Journal of Huazhong University of Science and Technology (Nature Science Edition), 2007, 35(9): 110-113(in Chinese)
(蔡卫东，李凡，徐章艳，杨炳儒. 基于修正差别矩阵的高效属性约简算法. 华中科技大学学报(自然科学版)，2007，35(9)：110-113)

[10] Xu Zhang-Yan, Yang Bing-Ru, Song Wei. Complete reduction algorithm based on simple discernibility matrix. Computer Engineering and Applications, 2006, (26): 167-169(in Chinese)
(徐章艳，杨炳儒，宋威. 基于简化差别矩阵的完备属性约简算法. 计算机工程与应用，2006，(26)：167-169)

[11] Ye Dong-Yi. An improvement to Jelonek's attribution reduction algorithm. Acta Electronica Sinica, 2000, 28(12): 81-82(in Chinese)
(叶东毅. Jelonek 属性约简算法的一个改进. 电子学报, 2000, 28(12): 81-81)

[12] Liu Shao-Hui, Sheng Qiu-Jian, Wu Bin, Shi Zhong-Zhi, Hu Fei. Research on efficient algorithms for Rough set methods. Chinese Journal of Computers, 2003, 26(5): 524-529(in Chinese)
(刘少辉, 盛秋戬, 吴斌, 史忠植, 胡斐. Rough 集高效算法的研究. 计算机学报, 2003, 26(5): 524-529)

[13] Xu Zhang-Yan, Liu Zuo-Peng, Yang Bing-Ru, Song Wei. A quick attribute reduction algorithm with complexity of $\max(O(|C||U|), O(|C|^2|U/C|))$. Chinese Journal of Computers, 2006, 29(3): 391-399(in Chinese)
(徐章艳, 刘作鹏, 杨炳儒, 宋威. 一个复杂度为 $\max(O(|C||U|), O(|C|^2|U/C|))$ 的快速属性约简算法. 计算机学报, 2006, 29(3): 391-399)

[14] Wang Guo-Yin, Zhao Jun, An Jiu-Jiang, Wu Yun. A comparative study of algebra viewpoint and information viewpoint in attribute reduction. Fundamenta Informaticae, 2005, 68(6): 289-301

[15] Zhao Jun, Wang Guo-Yin, Wu Zhong-Fu, Tang Hong, Li Hua, Liao Xiao-Feng. An efficient approach to computer feature core. Mini MicroSystems, 2003, 24(11): 1590-1593(in Chinese)
(赵军, 王国胤, 吴中福, 唐宏, 李华, 廖晓峰. 一种高效的属性核计算方法. 小型微型计算机系统, 2003, 24(11): 1590-1593)

GE Hao, born in 1976, M. S., associate professor. His research interests include data mining and rough set theory.

LI Long-Shu, born in 1956, professor, Ph. D. supervisor. His research interests include inaccurate information processing and intelligent software.

YANG Chuan-Jian, born in 1978, M. S., associate professor. Her research interests include data mining and rough set theory.

Background

As a mathematical tool, rough set has been successfully applied in a lot of fields, such as pattern recognition, data mining, knowledge discovery, and so on. The attribute reduction is an important issue of rough set. Many researchers have proposed a number of attribute reduction algorithms. But some algorithms for attribute reduction are not completeness and its time and space complexity are not ideal.

In the paper, the notion of the conflict region is given, and natures of the conflict region are researched. An efficient algorithm for attribute reduction is proposed based on conflict region. In the worst case, its time complexity is $O(|C|^2|U|)$ and space complexity is $O(|U|)$. Theoretical analysis and experimental results show that the algorithm is effective. Compared to the existing algorithm for attribute reduction, the algorithm overcome defects of some algorithms for attribute reductions.

This paper is partially supported by the Natural Science Foundation of Anhui Province of China under Grant Nos. 050420204, 090412054, the Natural Science Foundation of Education of Anhui Province of China under Grant No. KJ2011Z276, and the Foundation for Outstanding Young Talents in Higher Education Institutions of Anhui Province under Grant No. 2011SQRL123.