Hash 快速属性约简算法

刘 勇 熊 蓉 褚 健
(浙江大学工业控制国家重点实验室 杭州 310027)
(浙江大学智能系统与控制研究所 杭州 310027)

摘 要 从决策系统的不一致情况出发,给出了不一致度的概念及其性质,并证明了不一致记录与正区域的等价关系.在此基础上,提出了基于 Hash 的正区域计算方法,时间复杂度下降为 $O(|U|)$;利用不一致情况的性质设计了一个基于不一致记录数的属性重要性测量参数,用新的测量参数设计了一个基于二次 Hash 的约简算法,其复杂度下降为 $O(|C|^2|U/C|)$,并证明采用该测量参数所获得约简的完备性.最后通过实验证明该文正区域算法和约简算法的高效性.

关键词 粗糙集;正区域;约简;Hash;不一致度
中图法分类号 TP18 **DOI 号**: 10.3724/SP.J.1016.2009.01493

Quick Attribute Reduction Algorithm with Hash

LIU Yong XIONG Rong CHU Jian
(*State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou* 310027)
(*Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou* 310027)

Abstract This paper presents the concept and property of inconsistency from the inconsistent condition of decision system. It also presents the relationship between the positive region and inconsistent records. A hash based algorithm calculating positive region has been presented and its temporal complexity decreases to $O(|U|)$. Based on the characteristics of inconsistency, a new attribute measure has been introduced, then a corresponding reduction algorithm with twice-hash is presented, and its temporal complexity is $O(|C|^2|U/C|)$, this paper also proves this algorithm is complete. The efficiency of the algorithms is proved by the experiments.

Keywords rough set; positive region; reduction; Hash; inconsistency

1 引 言

求信息系统的约简和最小约简是粗糙集理论研究中的基本问题之一.与其它数据降维的方法[1-2]相比,使用粗糙集方法进行约简的优势在于能够保持其数据本身的语义特征(semantic features)[3].许多学者都对属性约简问题进行过研究[3-9],其中已经公认的结论是求解信息系统最短约简是一个 NP 难问题[8].

通常情况下,约简算法并不需要计算出信息系统的所有约简,仅需获得用户感兴趣或者可用的约简即可.依据粗糙集理论中获取约简方法的不同,常用的计算方法有 Greedy 法和差别矩阵法.

在 Greedy 法[5]中,待求约简集通常初始化为核(Core)[5,10]或空集(NULL)[3],然后依次扫描剩下

收稿日期:2007-08-28;最终修改稿收到日期:2008-10-31.本课题得到国家自然科学基金(60803053,60675049)、浙江省自然科学基金(Y106414)、国家"八六三"高技术研究发展计划项目基金(2008AA04Z209)、国家博士后科学基金(20081459)和武器装备预研基金(9140A06050609JW0402)资助.**刘 勇**,男,博士,讲师,研究方向为人工智能、信息处理.E-mail: yongliu@iipc.zju.edu.cn.**熊 蓉**,女,副教授,研究方向为智能机器人.**褚 健**,男,教授,研究领域为智能控制.

的属性,从中选取一个使得分类质量增益最大的属性,并把该属性加入到待求约简集中,直到当前候选集计算出的分类质量等于所有条件属性计算出的分类质量.

差别矩阵(也称为差异矩阵或分明矩阵、区分矩阵)法[11-13],首先计算数据集的差别矩阵和差别函数,然后通过求差别矩阵中所有项的最小析取范式来获得约简.该算法的优点在于直观、易于理解,而且能够很容易地计算出核与所有约简.但这个算法也存在着不足之处,即,由于在矩阵中会出现大量的重复元素(或元素之间存在着包含关系),这就大大降低了属性约简算法的效率.通常此类算法的复杂度为 $O(|C|^2|U|^2)$,其中 $|C|$ 为属性个数,$|U|$ 为数据纪录数.

上述约简方法中差别矩阵方法计算过程需耗费巨大的时间和空间,故不常采用.其它方法中都需要频繁计算决策表的正区域,因而计算正区域的时间复杂度直接决定了约简算法的时间复杂度.

通常采用的正区域计算方法时间复杂度为 $O(|C||U|^2)$,文献[14]中基于快速排序思想给出了一种快速求解正区域算法,其时间复杂度可降为 $O(|C||U|\log(|U|))$.在此基础上文献[15]中类似地提出了一种基数排序的计算正区域算法,其时间复杂度下降至 $O(|C||U|)$.在本文的属性约简算法中,提出了一种基于 Hash 表的正区域计算方法,能够将时间复杂度降为 $O(|U|)$.在对采用正区域作为启发式搜索条件的约简算法进行充分分析后,给出了一个适用于 Hash 方法计算的属性重要性度量参数,并基于此参数设计了一个完备二次 Hash 属性约简算法,将时间复杂度下降到 $O(|C|^2|U/C|)$,并通过实验和分析证明本文算法的高效性.

本文第 2 节介绍相关的基本概念及不一致的定义和性质,证明了不一致记录数与正区域间的等价性;第 3 节介绍基于 Hash 表的快速正区域计算方法,并且给出其时间复杂度分析;第 4 节提出一个建立在不一致性质上的二次 Hash 快速属性约简算法,并通过实例予以说明;第 5 节通过实验证明本文算法的高效性;最后给出结论和展望.

2 不一致的定义及其性质

属性约简[11]过程实际上是寻找一个属性子集,使该子集能够保有与原条件属性集合相同的记录辨识能力.因而其本质上是通过分类能力来衡量属性子集是否构成一个约简.正区域[11]的定义恰好描述一个属性子集能够辨别区分的记录数.

从条件属性集区分纪录的能力考虑,给出如下定义.

定义 1. 不一致.信息系统 $U(C,D)$ 中,C 为条件属性集,D 为决策属性集,$P\subseteq C$,$x_1,x_2\in U$,若 $\forall a\in P, a(x_1)=a(x_2)$ 且有 $\exists d\in D, d(x_1)\neq d(x_2)$,则称此时 x_1 与 x_2 之间在属性 P 上存在不一致情况,若 $P=C$,此时的 U 也称为不一致信息系统.

定义 2. 不一致记录数/不一致记录信息系统.信息系统 $U(C,D)$ 中,C 为条件属性集,D 为决策属性集,$U'\subseteq U$,若有 $P\subseteq C, E_i\in U'/P, i=1,2,3,\cdots,|U'/P|$,记属性集 P 的在 U' 上不一致记录数为 $IN^{U'}(P)$(通常 $U=U'$ 时,可简写为 $IN(P)$),计算如下:

$IN^{U'}(P)=\sum|E_k|$, E_k 是在属性 P 上存在不一致情况的等价类,若 $P=\varnothing$, $IN^{U'}(P)=|U'|$.

我们将 U 中所有在 P 上的不一致记录构成的信息系统称为相对属性 P 的不一致记录信息系统,记为 I_P.即有

$$I_P=\{\bigcup X_i\,|\,X_i\in U/P,\ |X_i/D|\neq 1\},$$
$$\text{若 } P=\varnothing,\ I_P=U.$$

这里显然有

$$|I_P|=IN(P)=IN^{I_P}(P).$$

在给出了不一致情况的相关定义后,可得如下几个性质.

性质 1. 信息系统 $U(C,D)$ 中,C 为条件属性集,D 为决策属性集,$\forall Q\subseteq C$,信息系统 $U(Q,D)$ 是一致的(consistent)当且仅当 $IN(Q)=0$.

证明. 若信息系统 $U(Q,D)$ 是一致的(consistent)则显然有 $IN(Q)=0$,反过来若 $IN(Q)=0$,根据不一致记录数的定义,表明信息系统中无不一致情况,即信息系统是一致的. 证毕.

性质 2. 信息系统 $U(C,D)$ 中,C 为条件属性集,D 为决策属性集,$\forall P\subset C$,$|POS_P(D)|=|POS_C(D)|$ 当且仅当 $IN(P)=IN(C)$.

证明. 根据定义有 $POS_P(D)=\bigcup\limits_{X\in U/D}\underline{P}X$,也就是在信息系统 $U(C,D)$ 中有

$$\underline{P}X=\{E_i\,|\,E_i\in U/P\ \&\ E_i\subseteq X_j, X_j\in U/D\},$$

这里 $i=1,2,3,\cdots,|U/P|$, $j=1,2,\cdots,|U/D|$.

由 $E_i\subseteq X_j$ 可得对任意的 $x_m,x_n\in E_i$ 有 $\forall a\in P$, $a(x_m)=a(x_n)$,且 $\forall d\in D, d(x_m)=d(x_n)$,对于 $E\notin\underline{P}X$ 的等价类则表明 E 中存在不一致情况,故 U/P

中的等价类可以分为两种,一种是属于$\underline{P}X$,另外一种是不属$\underline{P}X$ 存在不一致情况的等价类.故有

$$card(\bigcup_{X\in U/D}\underline{P}X)=|U|-IN(P),$$

即

$$|POS_P(D)|=|U|-IN(P).$$

同样可得

$$|POS_C(D)|=|U|-IN(C),$$

故,若$|POS_Q(D)|=|POS_C(D)|$,显然有 $IN(Q)=IN(C)$.　　证毕.

通过上面给出的性质,可得如下定理.

定理 1.　信息系统 $U(C,D)$中,C 为条件属性集,D 为决策属性集,R 是 C 的一个约简,当且仅当 $IN(R)=IN(C)$,且有 $\forall Q\subset R, IN(Q)\neq IN(C)$.

证明.　由性质 2 可得

$$|POS_R(D)|=|POS_C(D)|,$$

即 $\gamma_R(D)=\gamma_c(D)$,且有 $\forall Q\subset R$,$|POS_Q(D)|\neq|POS_C(D)|$,故 R 是 C 的一个约简.　　证毕.

3　正区域快速计算方法

由上节中的不一致定义和性质可知,正区域可通过计算不一致纪录数间接获得,即$|POS_Q(D)|=|U|-IN(Q)$.式中$|U|$为常数,故只需计算不一致纪录数 $IN(P)$.通常正区域计算方法需逐个比较 U 中纪录的每一属性是否都相同,若相同则归入同一等价类,再比较其决策是否相同,以获得决策一致的等价类,从而计算出正区域,其时间复杂度达到 $O(|C||U|^2)$.文献[15]中采用基数排序思想设计的算法能够将复杂度下降为 $O(|C||U|)$,但是此算法需预先统计每一个属性的取值范围,并多次遍历决策表.本文中根据不一致纪录数与正区域的计算关系,设计了一个通过 *hash* 表来实现的正区域计算方法,其时间复杂度可控制在 $O(|U|)$.

算法 1.　计算正区域 $POS_R(D)$.

输入:信息系统 $U(C,D)$,属性集合 $R,R\subseteq C$

输出:$|POS_R(D)|$ 和 $POS_R(D)$

1. $Count=0$,初始化 *hash* 表 $H(h_i.count=0,\ h_i.cons=$ true)
2. 对每一 $x_j\in U$ 有如下操作
   2.1. $h_i=h_i\bigcup hash(R(x_j))$ //将纪录放入对应 *hash* //到的表分项中,$h_i\in H$
   2.2. $h_i.count$++
   2.3. 若 $D(h_i)\neq D(x_j)$, $h_i.cons=$ false //比较决策是否相等,即是否一致
3. 对每一 $h_i\in H$　//遍历 *hash* 表,计算正区域记录数
   3.1. 若 $h_i.cons==$ true, $Count=Count+h_i.count$
4. 返回计数 *Count*, *hash* 表 H,算法结束.

其中 $hash(R(x_j))$为 *hash* 编码计算函数,计算纪录 x_j在属性集 R 上的编码值,实现过程中可直接取纪录 x_j在属性 R 上所有取值的并,即 $hash(R(x_j))=\bigcup_{a\in R}a(x_j)$.$H$ 表示 *hash* 表,h_i是 *hash* 表中当前纪录命中项,步 2.1 中对 x_j进行 *hash* 后将纪录加入到 *hash* 表的对应分项 h_i中,同时递增分项计数(步 2.2)以及检测是否存在不一致情况(步 2.3).算法步 1 至步 2 完成对 U 遍历一次,将每一条纪录 *hash* 操作至对应的分项(即等价类),其时间复杂度为 $O(|U|)$.步 1 中的 *hash* 表的初始化可放在 *hash* 表中的项第一次命中时,其总时间耗费为 $O(|U/P|)$.步 3 对 *hash* 表遍历完成正区域的计算,其时间复杂度为 $O(|U/P|)$,故算法 1 的总时间复杂度为 $O(|U|)$.

对表 1 所示的决策表中 6 个对象,用算法 1 计算 $POS_{\{a,b,c\}}(D)$,过程如下:

对 U 中纪录,例如 $X1$,$hash(X1)=111$,做 *hash* 操作后有 $h_1=\{X1\}$,$h_1.count=1$, $h_1.cons=$ true,依次遍历完记录后,可得如图 1 的 *hash* 表.

表 1　信息系统

	a	*b*	*c*	*D*
$X1$	1	1	1	0
$X2$	1	2	3	2
$X3$	2	3	2	0
$X4$	3	1	2	1
$X5$	1	1	1	1
$X6$	1	2	3	2

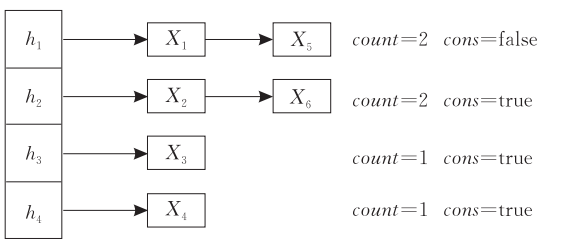

图 1　遍历完成后的 *hash* 表 H

再对 H 遍历即可得 $POS_{\{a,b,c\}}(D)$为$\{X2,X6\}$,$\{X3\}$,$\{X4\}$.

4　二次 Hash 属性约简算法

4.1　属性重要性参数及其计算

定理 1 说明属性约简过程可建立在对信息系统中不一致情况的判定基础上.若能够找到一个属性子集,使用该集合获得的不一致纪录数等于采用全部属性计算得到的不一致记录数,则此属性子集是信息系统的一个约简或其子集中包含有约简.据此

可以设计出本文约简算法的搜索判定参数.此外,对于本文的属性重要性参数也可采用文献[14-15]中的搜索空间递减方法,以期提高算法效率.

基于不一致的定义和性质,本文给出一个新的属性重要性参数如下.

定义 3. 基于不一致记录数的属性重要性参数信息系统 $U(C,D)$ 中,$P\subseteq C$, $\forall\alpha\in(C-P)$ 的重要性定义为 $SGF(P,\alpha)=IN^{I_P}(P\cup\{\alpha\})$,其中 I_P 为相对属性 P 的不一致记录信息系统.

定理 2. 信息系统 $U(C,D)$ 中,$P\subset C$, $\forall\alpha\in(C-P)$,若 $SGF(R,\alpha)=IN(C)$,则 $P\cup\{\alpha\}$ 是信息系统的一个约简或其子集中包含有信息系统的约简.

证明. 若 $P=\varnothing$,此时 $I_P=U$,即 $SGF(R,\alpha)=IN(\alpha)=IN(C)$,由性质 2 和约简的定义可得 $P\cup\{\alpha\}$ 是信息系统的一个约简.若 $P\neq\varnothing$,此时 $SGF(P,\alpha)=IN^{I_P}(P\cup\{\alpha\})=|I_P|-|POS^{I_P}_{P\cup\{\alpha\}}(D)|$,此处,$POS^I_R(D)$ 表示是属性集 R 相对于决策 D 在信息系统 I 上的正区域.

由不一致记录信息系统的定义可得

$$I_P=U-POS^U_P(D),$$

即 $|POS^U_P(D)|=|U|-|I_P|$.可得

$$SGF(P,\alpha)=|U|-|POS^U_P(D)|-|POS^{I_P}_{P\cup\{\alpha\}}(D)|.$$

由已知条件

$SGF(R,\alpha)=IN(C)$ 且 $|POS_C(D)|=|U|-IN(C)$,可得

$$\begin{aligned}SGF(P,\alpha)&=|U|-|POS^U_P(D)|-|POS^{I_P}_{P\cup\{\alpha\}}(D)|\\&=|U|-POS^U_C(D),\end{aligned}$$

即 $|POS^U_C(D)|=|POS^U_P(D)|+|POS^{I_P}_{P\cup\{\alpha\}}(D)|$.

由正区域的定义及性质 $U/(P\cup\{a\})=\bigcup_{X\in U/P}(X/\{a\})(a\in C-P)$ 有如下式子

$$|POS^U_{P\cup\{\alpha\}}(D)|=|POS^{U-I_P}_{P\cup\{\alpha\}}(D)|+|POS^{I_P}_{P\cup\{\alpha\}}(D)|.$$

由 I_P 的定义这里显然有

$$\begin{aligned}|POS^{U-I_P}_{P\cup\{\alpha\}}(D)|&=|POS^{U-I_P}_P(D)|=|U|-|I_P|\\&=|POS^U_P(D)|.\end{aligned}$$

由上可得

$$\begin{aligned}|POS^U_C(D)|&=|POS^U_P(D)|+|POS^{I_P}_{P\cup\{\alpha\}}(D)|\\&=|POS^U_{P\cup\{\alpha\}}(D)|,\end{aligned}$$

故可得 $P\cup\{\alpha\}$ 为信息系统约简,或其子集中包含信息系统约简,证明完毕. 证毕.

上述定义中的属性重要性,可直观上理解为属性未能够区分的记录数.根据约简定义,未能够区分的记录数越少越好,故本文中的属性重要性值越小代表属性的重要程度越高.此外,在属性重要性计算过程中,没有采用 U 作为每次的搜索空间,而是采用了相对属性集 P 的不一致记录信息系统 I_P 作为搜索空间.这是因为,对于属性 P 已经可以区分的数据记录来说,增加一个新的属性 a 后其仍然是可被区分的,此部分不会对未区分记录数构成影响.基于此点,仅需考虑属性 P 未能区分的信息系统 I_P,可大大减少搜索范围,提高算法效率.下面给出可递归的属性重要性参数计算方法.

算法 2. 计算属性重要参数 $SGF(R,\alpha)$.

输入:对信息系统 $U(C,D)$ 中属性 C,调用算法 1 后的 $hash$ 表 $H_C(H_1,H_2,\cdots,H_n)$, $n=|U/C|$

输出:$SGF(R,\alpha)=IN^{I_R}(R\cup\{\alpha\})$

1. $INS=0$,初始化新的 $hash$ 表 H_R, $H_{R\cup\{\alpha\}}$ ($count=0$, $cons=$true)
2. 对输入 $hash$ 表 H_C 中的每一项 $H(H\in H_C)$ 进行如下操作:
   2.1. $h_{R_i}=h_{R_i}\cup hash(R(H))$
   //将 H 作为纪录放入 $hash$ 到的表分项中,$h_{R_i}\in H_R$
   2.2. $h_{R_i}.count+=H.count$
   //加上 H 中所包含记录数
   2.3. 若 $D(h_{R_i})\neq D(H)$, $h_{R_i}.cons=$false
   2.4. 若 $H.cons==$false, $h_{R_i}.cons=$false
   //如 H 自身是存在不一致的等价类
3. 遍历 H_R 中的每一个 $hash$ 项 h_{R_i} ($h_{R_i}\in H_R$)
4. 若 $h_{R_i}.cons==$false,对 h_{R_i} 中每一项 $H(H\in H_C)$ 进行如下操作:
   4.1. $h^i_{R\cup\{\alpha\}}=h^i_{R\cup\{\alpha\}}\cup hash([R\cup\{\alpha\}](H))$
   4.2. $h^i_{R\cup\{\alpha\}}.count+=H.count$
   //加上 H 中所包含记录数
   4.3. 若 $D(h^i_{R\cup\{\alpha\}})\neq D(H)$, $h^i_{R\cup\{\alpha\}}.cons=$false
   //新等价类中有不一致
   4.4. 若 $H.cons==$false, $h^i_{R\cup\{\alpha\}}.cons=$false
   //H 自身是存在不一致的等价类
5. 遍历 $H_{R\cup\{\alpha\}}$ 中每一个 $hash$ 项 $h^i_{R\cup\{\alpha\}}\in H_{R\cup\{\alpha\}}$
   5.1. 若 $h^i_{R\cup\{\alpha\}}.cons==$false,
   $INS=INS+h^i_{R\cup\{\alpha\}}.count$
6. 返回 INS 值.

算法 2 中求 SGF 的过程是将全部条件属性 C 进行 $hash$ 后的 $hash$ 表项作为输入重新进行 $hash$ 计算,易得步 2 的复杂度为 $O(|H_C|)$,步 3 和 4 的复杂度为 $O(|H_C|-|Q_R|)$,$Q_R=\{x_i\,|\,x_i\in H_C/R\,\&\,|x_i/D|=1\,\&\,\forall H\in x_i, H.cons==\text{ture}\}$,步 5 的复杂度为 $O(|(H_C-Q_R)/R\cup\{\alpha\}|)$,故算法复杂度为 $O(|H_C|)$,即 $O(|U/C|)$. 进一步分析可知,算法 2 中步 2 的计算复杂度最高,其目的是计算 U/R 和

$POS_R(D)$,而在此两项已知的情况下计算复杂度可下降到 $O(|H_C|-|Q_R|)$,因此可以考虑将约简算法设计为逐步递增计算,即在保留 U/R 的基础上计算 $SGF(R,\alpha)$.

4.2　约简算法

约简算法给出如下.

算法 3.　二次 Hash 快速属性约简算法.

输入:信息系统 $U(C,D)$

输出:C 相对 D 的某个约简 R

1. 置 R 为空集合.
2. 用算法 1 计算 U/C, H_C.
3. 对每一个属性 α, $\alpha\in C-R$ 有如下操作.
4. 计算属性重要性参数 $SGF(R,\alpha)=IN^{I_R}(R\cup\{\alpha\})$.
5. 在 $C-R$ 中找出使得 $SGF(R,a)$ 取值最小的属性值 α(若存在多个这样的属性则任选一个),并记 $\chi=SGF(R,\alpha)$.
6. 将 α 加入 R 中,即 $R=R\cup\{\alpha\}$.
7. 若 $\chi\neq IN(C)$,转步 3,否则继续.
8. 从 R 的尾部开始从后往前对每个属性 b 进行判断是否可省,若 $|POS_C(D)|=|POS_{R-\{b\}}(D)|$,则说明 b 是可省的,从 R 中把 b 删除.
9. 否则算法结束,返回 R.

上述约简算法首先对原决策表 *hash*(步 2),然后在此结果上再次通过 *hash* 来计算约简,因而称之为二次 Hash 约简算法.下面给出算法的正确性分析,算法的终止条件为 $\chi\neq IN(C)$, 即 R 所不能区分的记录数等于 C 所不能区分的记录数,由定理 2 可知,此条件是充分的,故此时的 R 是信息系统的一个约简或者包括了约简,可通过步 8 来消除冗余属性获得最终约简.

算法复杂度分析如下,步 2 中计算所需时间为 $O(|U|)$,由算法 2 中分析可得步 4 中计算一次 $SGF(R,\alpha)$需要的时间复杂度为 $|U/C|$,从而步 3 至步 7 计算的最差时间复杂度为 $O(|C|^2|U/C|)$,步 8 最坏情况下时间复杂度为 $O(|C||U/C|)$,因而算法总的时间复杂度为 $O(|C|^2|U/C|+|C||U/C|+|U|)$,即 $O(|C|^2|U/C|)$.

下面以表 1 中的数据为例说明算法 3 的执行过程,决策表经第一次 *hash* 后的结构如图 1 所示,对表 1 中数据执行算法 3 的步 2 后可得如下表 2 的以 *hash* 项为单位的信息系统.此后在表 2 上继续 *hash*,由算法 3 的第 3、4 步可得如下的 *hash* 表.

$$
\begin{aligned}
H_{\{a\}}=&\{\{h1,\ h2,\ cons=\text{false},\ count=4\},\\
&\{h3,\ cons=\text{true},\ count=1\},\\
&\{h4,\ cons=\text{true},\ count=1\}\},\\
&SGF(a)=4,\\
H_{\{b\}}=&\{\{h1,\ h4,\ cons=\text{false},\ count=3\},\\
&\{h2,\ cons=\text{true},\ count=2\},\\
&\{h3,\ cons=\text{true},\ count=1\}\},\\
&SGF(b)=3,\\
H_{\{c\}}=&\{\{h1,\ cons=\text{false},\ count=2\},\\
&\{h2,\ cons=\text{true},\ count=2\},\\
&\{h3,\ h4,\ cons=\text{false},\ count=2\}\},\\
&SGF(a)=4.
\end{aligned}
$$

表 2　第一次 *hash* 后的决策表

	a	*b*	*c*	*D*	*Cons*	*Count*
*h*1	1	1	1	0	False	2
*h*2	1	2	3	2	True	2
*h*3	2	3	2	0	True	1
*h*4	3	1	2	1	True	1

由算法 3 的步 5 取获得最小 *SGF* 的属性 b, $R=\{b\}$, $I_R=\{h1,h4\}$. 转算法步 3,*hash* 后有

$$
\begin{aligned}
H_{R\cup\{a\}}=&\{\{h1,\ cons=\text{false},\ count=2\},\\
&\{h4,\ cons=\text{true},\ count=1\}\},\\
&SGF(a,b)=2,\\
H_{R\cup\{c\}}=&\{\{h1,\ cons=\text{false},\ count=2\},\\
&\{h4,\ cons=\text{true},\ count=1\}\},\\
&SGF(a,b)=2.
\end{aligned}
$$

信息系统中 $IN(C)=2$,故此时 $\{a,b\}$ 和 $\{b,c\}$ 都是信息系统的约简.

5　实验及结果分析

本节中采用 UCI 机器学习数据库中数据,如表 3 所示,在 PC(Dell GX520,1GB 内存,Windows XP, Intel P4 2.8GHz)上进行实验.实验 1 采用本文中算法和文献[15]以及文献[16]中的 Semi-minimal Reduct 算法(即 Johnson Reduct Algorithm)进行属性约简,上述 3 种算法皆采用 JAVA 语言实现,运行在同一系统硬件/软件平台下,且每种算法运行 10 次,分别去除最初一次和最后一次的运算结果,对中间 8 次的运行结果取平均值.实验结果如表 3 所示,其中 $POS_C(D)$表示计算正区域的时间.

表 3　约简算法执行时间对比

数据集	实例数	属性数	本文方法/ms		Semi-minimal Reduct /ms		文献[15]算法/ms	
			约简	$POS_C(D)$	约简	$POS_C(D)$	约简	$POS_C(D)$
Mushroom	8124	22	2359	104	7759	157	6703	153
Vote	435	16	182	1	450	3	406	3
Tic-tac-toe	958	9	125	3	202	7	188	6

为进一步对比本文提出的正域计算方法与其它计算方法的性能差异,设计了第 2 个实验.实验 2 在实验 1 相同的硬件软件环境下实现了 Jensen 的 Quickreduct 算法[3],算法中正域的计算方法分别采用本文的 Hash 方法和文献[16]中的快速排序方法.实验结果如表 4.

表 4 相同约简策略下不同正区域算法性能对比

数据集	实例数	属性数	哈希/ms		快速排序/ms	
			约简	$POS_C(D)$	约简	$POS_C(D)$
lung-cancer	32	57	47	1	58	1
German-credit	1000	21	78	3	139	10
pendigits	10992	17	4627	111	6281	139
letter	20000	17	21355	202	26297	232

从表 3 和表 4 可以看出,本文的 Hash 约简方法无论在约简还是在单个正区域计算上性能都优于文献[15]和 Semi-minimal reduct 方法.本文方法和文献[15]中算法在时间复杂度上同为 $O(C^2|U/C|)$,但是本文中采用的 *hash* 表数据结构能够更有效地减少计算量,具体原因分析如下:

(1)在 *hash* 过程天然可划分好等价类,不需要文献[15]中算法 1 的步 4 来搜索等价类(约 $O(|C||U|)$ 的复杂度耗费);

(2)在 *hash* 时通过一次比较操作,直接可以得知当前等价类是否可归于正区域内,不需要额外的计算,也不需要额外的开销来维护正区域和负区域记录集;

(3)*hash* 表中的 *hash* 项 $h_i(h_i \in H_C)$ 可以直接作为基本处理单元(类似文献[15]中简化决策表的项)重新哈希,效率得到提高.

(4)文献[15]中的算法 2 需要判断等价类是否被包含在负区域或者正区域集中,带来了大量额外运算.

综上所述,*hash* 表的存储结构和操作效率要明显好于文献[15]中采用的离散双向链表数据管理方式.此外本文算法是一个完备的属性约简算法,能够严格保证计算结果为约简.基于上面的实验结果和分析表明本文算法是可靠高效的,且尤其适用于海量数据约简.

6 总结与展望

约简是粗糙集理论中的一个重要研究部分,但是由于约简计算的复杂性,限制了其在机器学习、数据挖掘等领域的应用和推广.因而研究快速高效的约简算法对粗糙集理论的应用和推广有着极其重大的意义.现有约简算法中,或需数据集保持一致,或计算效率过低,很大程度上限制了约简算法的应用.实际问题中的数据集很多都难以保证其一致性,并且由于误差或干扰等因素造成的数据不一致情况也非常普遍,因而需要约简算法在能够很好地支持不一致数据的同时具有良好的计算效率.

本文从数据不一致性出发,证明了决策系统中不一致情况与正区域的等价关系,并在此基础上给出一个基于不一致记录数的二次 Hash 快速约简算法,该算法通过扫描数据集中的不一致情况从而计算约简.算法具有如下优点:

(1)能够非常好地支持不一致数据,可处理因噪音干扰等因素所形成的不一致数据集.

(2)能够有效去除冗余无关属性,保证输出约简的完备性.

(3)扫描获得不一致数和计算正区域的时间复杂度为 $O(|U|)$,整个约简算法复杂度控制在 $O(|C|^2|U/C|)$之内,能够大大降低算法时间耗费.

在未来的工作中将进一步考虑约简算法在数据特征选择中的应用,利用约简后的结果来构建机器学习中的分类器,达到提高分类器泛化能力的目的.

参 考 文 献

[1] Kira K, Rendell L A. The feature selection problem: Traditional methods and a new algorithm//Proceedings of the 9th International Workshop on Machine Learning (ML 1992). Aberdeen, Scotland, UK, 1992: 129-134

[2] Blum A, Langley P. Selection of relevant features and examples in machine learning. Artificial Intelligence, 1997, 97(1-2): 245-271

[3] Jensen R, Shen Q. Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(12): 1457-1471

[4] Chouchoulas A, Shen Q. Rough set-aided keyword reduction for text categorization. Applied Artificial Intelligence, 2001, 15(9): 843-873

[5] Hu X H. Knowledge discovery in database: An attribute-oriented rough set approach [Ph. D. dissertation]. Regina, Canada: University of Regina, 1995

[6] Jelonek J, Krawiec K, Slowinski R. Rough set reduction of attributes and their domains for neural networks. Computational Intelligence, 1995, 11(2): 339-347

[7] Lin T Y, Yin P. Heuristically fast finding of the shortest reducts//Proceedings of the Rough Sets and Current Trends in Computing (RSCTC2004). Uppsala, Sweden, 2004: 465-470

[8] Skowron A, Rauszer C. The discernibility matrices and functions in information systems//Slowinski R ed. Intelligent Decision Support: Handbook of Applications and Advances to Rough Sets Theory. Dordrecht: Kluwer Academic, 1992: 331-362

[9] Swiniarski R W, Skowron A. Rough set methods in feature selection and recognition. Pattern Recognition Letters, 2003, 24(6): 833-849

[10] Zhong N, Dong J, Ohsuga S. Using rough sets with heuristics for feature selection. Journal of Intelligent Information System, 2001, 16(3): 199-214

[11] Pawlak Z, Rough Sets: Theoretical Aspects and Reasoning About Data. Dordrecht: Kluwer Academic Publishers, 1991

[12] Zhang Wen-Xiu, Mi Ju-Sheng, Wu Wei-Zhi. Approaches to knowledge reductions in inconsistent systems. International Journal of Intelligent System, 2003, 18(9): 989-1000

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

[14] 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)

[15] 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)

[16] Nguyen S H. Some efficient algorithms for rough set methods//Proceedings of the Conference on Information Processing and Managements of Uncertainty in Knowledge Based Systems (IPMU1996). Granada, Spain, 1996: 1451-1456

LIU Yong, Ph. D., lecturer. His research interests include intelligent artificial intelligent and information processing etc.

XIONG Rong, associate professor. Her research interests include intelligent robots, environment modeling.

CHU Jian, professor. His research interests include system optimization and robots, process control theory and applications.

Background

The attribute reduction is one of the most important concepts in rough set theory. With the attribute reduction, the irrelevant attributes can be removed from the original attribute set and the remained attributes can keep the discriminability as same as the original attribute set and also keep the internal semantic correlations among attributes. The attribute reduction has been widely used in data mining, which normally needs to deal with very large data set, so a fast and easy attribute reduction algorithm is especially important when implementing the reduction concept in data mining applications with huge data sets.

The mainly reduction algorithms fall in two categories: the discernibility matrix based approaches and positive region based approaches. The previous approaches are almost not able to implement in large data set condition due to the huge spatial requirement; and the latter approaches need to calculate the positive region iteratively. The procedure of positive region is a calculation intensive and tactical task. With the increasing of the attribute dimension and the number of instances, the processing time for positive region has grown tremendously. So it is significant to present a fast positive region method in attribute reduction algorithm. The approach in this paper belongs to the latter.

This paper addresses the problem to design fast and easy attribute reduction algorithm which is suit for the large data set. Firstly, we propose the concept of inconsistency and also present the properties of inconsistency, based on the inconsistency, we present a hash based quick positive region calculation method with linear temporal complexity. Based on the characteristics of inconsistency, a new attribute measure for attribute reduction has been introduced, then a corresponding reduction algorithm with twice-hash is presented, and its temporal complexity is $O(|C|2|U/C|)$. Finally, comparable experiments between our algorithm and other two fast reduction algorithms are carried out under UCI data set. The experimental results show the efficiency of our novel, fast hash based attribute reduction algorithm.

This paper is supported by National Nature Science Foundation of China (grant Nos. 60803053, 60675049), Nature Science Foundation of Zhejiang Province (grant No. Y106414), National High Technology Research and Develop Program (863 Program) of China (grant No. 2008AA04Z209), Excellent Postdoctoral Science Foundation of China (grant No. 20081459), and Defense Advanced Research Foundation of the General Armaments Department of the PLA (grant No. 9140A06050609JW0402).