

文章编号: 1001-0920(2006)12-1374-05

基于数据库系统的Rough集模型的扩展

刘启和, 陈雷霆, 闵帆, 蔡洪斌

(电子科技大学 计算机科学与工程学院, 成都 610054)

摘要: 针对基于数据库系统的Rough集模型中的知识约简算法对一致决策表适用, 而对不一致决策表不适用的局限性, 给出了将不一致决策表转换为一致决策表的算法, 证明该算法能保持核和约简集合不变, 并分析了该算法的时间复杂度. 在此基础上, 利用数据库系统的集合操作和SQL语言描述转换算法, 将基于数据库系统的Rough集模型中的知识约简算法扩展到不一致决策表. 理论分析和实验结果表明, 扩展后的算法仍是高效的.

关键词: Rough集理论; 数据挖掘; 数据库系统; 决策表

中图分类号: TP301 **文献标识码:** A

Extension of Rough Set Model Based on Database Systems

L I U Q i-he, C H E N L e i-t i n g, M I N F a n, C A I H o n g-b i n

(College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China. Correspondent: L I U Q i-he, E-mail: qiheliu@uestc.edu.cn)

Abstract: To the problem that the knowledge reduction algorithms in the rough set model based on database systems are only suitable for consistent decision tables not for inconsistent ones, a converting algorithm is proposed to convert decision tables into consistent decision tables and keep the core and the set of reducts. The correctness of converting algorithm is proved and its time complexity is analyzed. Based on that, the converting algorithm is designed by utilizing the database set operations and SQL statements, so the knowledge reduction algorithms in the model are extended to being suitable for inconsistent decision tables. Theoretical analysis and experimental results show that the extension of the knowledge reduction algorithms is still efficient.

Key words: Rough set theory; Data mining; Database systems; Decision tables

1 引言

Rough集理论提供两个基本概念来处理知识约简: 核和约简^[1]. 已经证明计算所有约简或最小约简是一个NP-难问题^[2]. 一些研究者设计了计算核和约简的有效算法^[2-4], 但进一步研究发现^[6-9], 这些算法只适用于数据存储在扁平文件的情况; 对于大数据集的数据库系统, 因其没有与数据库系统整合, 所以它们不是高效的, 从而限制了Rough集理论在数据挖掘中的应用^[6].

Han和Hu等人^[6-9]针对该问题, 提出了基于数据库系统的Rough集模型, 该模型是经典Rough集理论^[1]的数据库描述, 其核心思想是利用数据库系

统中的集合操作(如计数、投影等)来设计计算核和约简的算法. 与其他算法相比, 在大数据集的数据库系统中它们是高效的, 但该算法只对一致决策表适用, 对不一致决策表不适用.

Han等人^[6]认为导致决策表不一致的原因是噪音和知识不足, 并提出相应的解决方法: 删除不一致数据后计算核和约简. 经分析发现, 删除不一致数据后计算的核和约简可能并不是原表的核和约简. 在实际应用中, 导致决策表不一致的原因多种多样, 并且这种方法也违背Rough集理论处理数据集的原则: 它只依赖数据本身, 而对数据不作任何假设.

本文在保持核和约简集合不变的前提下, 将决

收稿日期: 2005-09-26; 修回日期: 2006-01-11.

基金项目: 国家自然科学基金项目(60471055).

作者简介: 刘启和(1973—), 男, 重庆人, 讲师, 博士, 从事机器学习、Rough集理论的研究; 陈雷霆(1966—), 男, 四川大竹人, 教授, 硕士, 从事三维图形、数字图像处理等研究.

策表转换为一致决策表, 给出转换算法并证明其正确性 利用转换算法, 使基于数据库系统的Rough 集模型中计算核和约简的算法对不一致决策表也适用

2 基本概念

三元组 $S = U, C, D$ 是一个决策表, 其中 U 是对象集合, C 是条件属性集合, D 是决策属性集合, 且 $C \cap D = \emptyset$. 对于任意 $x \in U$ 和 $a \in C \cup D, a(x)$ 表示 x 在属性 a 下的属性值 决策属性集 D 导出的 U 上划分记为 $U/D = \{Y_1, Y_2, \dots, Y_m\}$. 对于任意 $\emptyset \neq P \subseteq C, X \subseteq U$ 及 $x \in U, P_X$ 或 $P(X)$ 表示集合 X 的 P 下近似, $POS_P(D)$ 表示 D 的 P -正区域, $[x]_P$ 表示由 P 所导出的包含 x 的等价类

定义 1^[3] 若 $POS_C(D) = U$, 则称决策表 S 为一致决策表, 否则称 S 为不一致决策表

定义 2^[5] $B \subseteq C$, 如果 $POS_B(D) = POS_C(D)$, 且对任意的 $a \in B$, 有 $POS_{B - \{a\}}(D) \neq POS_C(D)$, 则称 B 为决策表 S 的约简 S 的所有约简记为 $RED_D(C)$.

定义 3^[5] 称 $CORE_D(C) = \bigcap_{B \in RED_D(C)} B$ 为决策表 S 的核

在决策表 $S = U, C, D$ 中, 正区域 $POS_C(D) = \bigcup_{i=1}^m CY_i$, 令 $Y_0 = U - POS_C(D)$, 易知 $CY_0 = Y_0$ 称 CY_0 为决策属性集 D 相对于 C 的边界域, 其中的元素为不一致元素或不一致数据 由定义 1, 如果 S 是一致决策表, 则 $CY_0 = \emptyset$.

3 决策表的转换

本节首先分析 Han 方法的不足, 给出转换算法并证明其正确性; 然后分析转换算法的时间复杂度

3.1 Han 方法的不足

Han 等人^[6] 通过删除不一致数据来获得一致决策表, 并将此决策表的核和约简看作是原决策表的核和约简 为了说明该方法的不足, 本文引用文献 [6] 中的例子来说明

表 1 一个决策表系统 S

Tuple-id	Door	Size	Cylinder	M ileage
t_1	2	compact	4	high
t_2	4	sub	6	low
t_3	4	compact	4	high
t_4	2	compact	6	low
t_5	4	compact	4	low
t_6	4	compact	4	high
t_7	4	sub	6	low
t_8	2	sub	6	low

例 1 如表 1 所示, 设决策表 $S = U, C, D$, 其中 $U = \{t_1, t_2, \dots, t_8\}, C = \{\text{Door}, \text{Size}, \text{Cylinder}\}, D = \{\text{M ileage}\}$.

在 S 中, $POS_C(D) = \{t_1, t_2, t_4, t_7, t_8\}, Y_0 = \bigcup_{i=1}^m CY_i = \{t_3, t_5, t_6\} \neq \emptyset$, 所以 S 是不一致决策表 当然, 给表 1 增加新的条件属性, 可使表 1 变为一致决策表 但在实际应用中, 更多的情况是没有新的条件属性可增加 在此情况下, 由 Han 方法在表 1 中删除 t_3, t_5, t_6 后, 获得一致决策表 S_1 (表 2).

表 2 删除 t_3, t_5, t_6 后的一致决策表 S_1

Tuple-id	Door	Size	Cylinder	M ileage
t_1	2	compact	4	high
t_2	4	sub	6	low
t_4	2	compact	6	low
t_7	4	sub	6	low
t_8	2	sub	6	low

表 3 给出了决策表 S 和一致决策表 S_1 的核和约简集合

表 3 决策表 S 和 S_1 的核和约简集合

	S	S_1
$CORE_D(C)$	{Door, Cylinder}	{Cylinder}
$RED_D(C)$	{{Door, Cylinder}}	{{Cylinder}}

从表 3 可以看出, 决策表 S 与 S_1 的核和约简集合并不相同, 因此 Han 方法并不适用

3.2 转换算法

应在保持核和约简集合不变的前提下, 把不一致决策表转换为一致决策表 为此, 给出如下转换算法:

算法 1 输入: 决策表 $S = U, C, D = \{d\}$; 输出: 一致决策表 $T_S = U_1, C_1, D_1 = \{d_1\}$.

Step 1: 计算 $POS_C(D)$ 和 $CY_0 = U - POS_C(D)$;

Step 2: 如果 $|CY_0| = 0$, 则 $d_1 = d$, 并转 Step 5;

Step 3: 取定值 val, 对于任意 $x \in POS_C(D)$, 有 val = $d(x)$;

Step 4: 任意 $x \in POS_C(D), d_1(x) = d(x)$; 任意 $x \in CY_0, d_1(x) = \text{val}$;

Step 5: $U_1 = U, C_1 = C$.

根据算法 1, 如果 S 是一致决策表, 即 $|CY_0| = 0$, 则 T_S 就是 S ; 如果 S 是不一致决策表, 则 T_S 是将 CY_0 中所有元素的决策属性值修改为一个相同的值 val, 且这个值应与 $POS_C(D)$ 中所有元素的决策属性值不同 在例 1 中, 取 val = other, 则决策表 S 对应的 T_S 如表 4 所示

表 4 例 1 中 S 的一致化决策表 T_S

Tuple-id	Door	Size	Cylinder	M ileage
t_1	2	compact	4	high
t_2	4	sub	6	low
t_3	4	compact	4	other
t_4	2	compact	6	low
t_5	4	compact	4	other
t_6	4	compact	4	other
t_7	4	sub	6	low
t_8	2	sub	6	low

比较表 1 和表 4, 除了 t_3, t_5, t_6 在决策属性 M ileage 上的值不同以外, 表 1 和表 4 没有其他不同 对于 $T_S = U_1, C_1, D_1$, 有 $U_1 = \{t_1, t_2, \dots, t_8\}, C_1 = \{\text{Door}, \text{Size}, \text{Cylinder}\}, D_1 = \{\text{M ileage}\}$.

表 5 给出了表 1 与表 4 的比较结果

表 5 S(表 1) 与 T_S (表 4) 的比较

	S	T_S
正区域	$\{t_1, t_2, t_4, t_7, t_8\}$	$\{t_1, t_2, \dots, t_8\}$
边界域	$\{t_3, t_5, t_6\}$	\emptyset
是否一致	否	是
核	$\{\text{Door}, \text{Cylinder}\}$	$\{\text{Door}, \text{Cylinder}\}$
约简集合	$\{\{\text{Door}, \text{Cylinder}\}\}$	$\{\{\text{Door}, \text{Cylinder}\}\}$

从表 5 可以看出, T_S 是一致决策表, 且与 S 有相同的核和约简集合 因此, 在基于数据库系统的 Rough 集模型中, 可通过计算一致决策表 T_S 的核和约简来获得 S 的核和约简 应当注意, T_S 仅被用来获得 S 的核和约简, 决策规则的生成仍在 S 中进行 如在例 1 中, 通过表 4 可获得表 1 的约简 $\{\text{Door}, \text{Cylinder}\}$, 此约简在表 1 中可获得如下确定性决策规则^[10]:

- 1) Door = 2 Cylinder = 4 M ileage = high;
- 2) Door = 4 Cylinder = 6 M ileage = low;
- 3) Door = 2 Cylinder = 6 M ileage = low.

3.3 算法正确性证明

本节证明 S 的一致化决策表 T_S 满足如下性质:

- 1) T_S 是一致决策表;
- 2) T_S 与 S 有相同的核和约简集合

根据算法 1, 显然 T_S 是一致决策表

引理 1 $B \subseteq C$ 是决策表 S 的约简的充要条件是:

$$\forall i \in \{1, 2, \dots, m\}, B Y_i = C Y_i; \quad (1)$$

$$\forall a \in B, \exists i \in \{1, 2, \dots, m\},$$

$$B - \{a\} Y_i \neq C Y_i; \quad (2)$$

由 $POS_B(D) = POS_C(D) \Rightarrow B Y_0 = B(U - POS_C(D)) = B(U - POS_B(D)) = U - POS_B(D) = Y_0 = C Y_0$, 即 $B Y_0 = C Y_0$ 因此由引理 1 可得:

推论 1 $B \subseteq C$ 是决策表 S 的约简的充要条件是:

$$\forall i \in \{0, 1, \dots, m\}, B Y_i = C Y_i; \quad (3)$$

$$\forall a \in B, \exists i \in \{0, 1, \dots, m\},$$

$$B - \{a\} Y_i \neq C Y_i; \quad (4)$$

根据引理 1 可得:

推论 2 $B \subseteq C$ 是一致化决策表 T_S 的约简的充要条件是:

$$\forall i \in \{0, 1, \dots, m\}, B(C Y_i) = C Y_i; \quad (5)$$

$$\forall a \in B, \exists i \in \{0, 1, \dots, m\},$$

$$B - \{a\}(C Y_i) \neq C Y_i; \quad (6)$$

引理 2 $B \subseteq C$, 则 $\forall i \in \{0, 1, \dots, m\}$, 有

$$B(C Y_i) = C Y_i \Leftrightarrow B Y_i = C Y_i; \quad (7)$$

由推论 2 和引理 2 可得:

推论 3 $B \subseteq C$ 是 T_S 的约简的充要条件是:

$$\forall i \in \{0, 1, \dots, m\}, B Y_i = C Y_i; \quad (8)$$

$$\forall a \in B, \exists i \in \{0, 1, \dots, m\},$$

$$B - \{a\} Y_i \neq C Y_i; \quad (9)$$

比较引理 1 和推论 3 发现, $B \subseteq C$ 是 S 的约简, 当且仅当 B 是 T_S 的约简 因此, 根据引理 1 和推论 3 有:

性质 1 $RED_D(C) = RED_{D_1}(C_1)$.

根据性质 1 和定义 3, 有:

性质 2 $CORE_D(C) = CORE_{D_1}(C_1)$.

性质 1 和性质 2 说明, S 与 T_S 有相同的核和约简集合, 因此可通过计算 T_S 的核和约简来获得 S 的核和约简 这样便证明了转换算法的正确性

3.4 转换算法复杂度分析

根据文献[4]中计算 $POS_C(D)$ 的方法, 则转换算法的时间复杂度为 $O(|C - D| |U| \log |U|)$.

文献[4]给出了计算核和约简的高效算法, 其中计算核和约简的时间复杂度分别为 $O(|C - D| |U| \log |U|)$ 和 $O(|C - D|^2 |U| \log |U|)$.

显然, 转换算法的时间复杂度小于或等于计算核和约简的时间复杂度 当通过计算 T_S 的核和约简来获得 S 的核和约简时, 其时间复杂度与直接计算 S 约简的时间复杂度相同

4 基于数据库系统的 Rough 集模型的扩展

本节利用数据库系统的集合操作和 SQL 来描述转换算法(算法 2), 将基于数据库系统的 Rough 集模型中计算核和约简算法扩展到对不一致决策表也适用(算法 3), 并分析这些算法的时间复杂度

利用转换算法可以克服基于数据库系统的 Rough 模型中知识约简算法的局限性。首先利用转换算法获得一致化决策表 T_S ; 然后利用基于数据库系统的 Rough 集模型中的高效算法来计算 T_S 的核和约简。根据性质 1 和性质 2, T_S 的核和约简就是 S 的核和约简。为了实现整个过程的高效性, 需用数据库系统的集合操作来描述转换算法。

设 X 和 Y 是数据库系统中的表, 则有:

Count: $\text{Card}(X)$ 表示 X 中不同元组的个数;

Select: $\sigma_{\text{condition}}(X)$ 表示从 X 中选择满足 condition 的所有元组;

Projection: $\Pi_A(X)$ 表示从 X 中选择出若干属性列组成新的关系表, 其中 A 是 X 中的属性列;

Rename: $\rho_Z(X)$ 表示与 X 相同的表, 表名为 Z ;

θ -Join: $X \theta_{\text{condition}} Y$ 表示 X 与 Y 的笛卡儿积中选取满足 condition 中的元组;

Add: $\text{add}_{\{b\}}(X)$ 表示给 X 增加新的属性列 b , 其中 X 中的元组在 b 上取空值;

Update: $\text{update}_{b(x)=a(x)}(\text{condition})$ 表示将满足 condition 条件的元组 x 在属性列 b 上的值更新为 $a(x)$ 。

假设数据库系统为关系数据库系统, 决策表 S 被存储在数据库的一张表中, 表中有条件属性集 C 和决策属性集 D , 还有一个主码属性, 记为 ID 。例如表 1 中的 Tuple-id 就是主码属性。

由算法 1 可知, 如果 S 是一致决策表, 则 $S = T_S$; 如果 S 是不一致决策表, 则必须构造一个新的决策属性 d_1 , 可将 d_1 存储在 S 所对应的数据库表中, 即只需在这个数据库表中增加一个新的属性, 并按算法 1 进行赋值。因此, 必须给出判别一致决策表的方法以及计算边界域的方法。为此给出如下定理:

定理 1^[6] S 是一致决策表的充要条件是

$$\text{Card}(\Pi_C(S)) = \text{Card}(\Pi_{C \cup D}(S)). \quad (10)$$

定理 2^[6] 在决策表 S 中, 有

$$\mathcal{L}Y_0 = \Pi_{\{ID\}}(S \theta_{c=v.c \cup s.d \cup v.d}(\rho_V(S))). \quad (11)$$

根据定理 1 和定理 2, 可将转换算法改写为:

算法 2 输入: 决策表 $S = U, C, D = \{d\}$; 输出: 一致决策表 $T_S = U, C, D_1 = \{d_1\}$ 。

Step 1: 如果 $\text{Card}(\Pi_C(S)) = \text{Card}(\Pi_{C \cup D}(S))$, 则 S 是一致决策表, 返回;

Step 2: $\text{add}_{\{d_1\}}(S)$;

Step 3: 取定一个值 val, 使得对于任意

$$x \in \Pi_{\{ID\}}(S \theta_{c=v.c \cup u.d \cup v.d}(\rho_V(S))),$$

有 $\text{val} = d(x)$;

Step 4: $\text{update}_{d_1(x)=d(x)}(S)$;

Step 5: $\text{update}_{d_1(x)=\text{val}}(x)$

$$\Pi_{\{ID\}}(S \theta_{c=v.c \cup u.d \cup v.d}(\rho_V(S))).$$

在算法 2 中, Step 1 用来判别 S 是否为一一致决策表, 若是, 则 $T_S = S$ 。如果 S 不是一致决策表, 则 Step 2 在 S 所对应的数据库表中增加一个新的属性 d_1 。Step 3 用来取定一个值 val, 使得对于任意 $x \in \text{POS}_C(D)$, 有 $\text{val} = d(x)$ 。例如, 如果 d 的值是可比较的, 设 $\text{POS}_C(D)$ 中的元组在 d 上的最大值为 max , 则可取 $\text{val} = 2\text{max}$ 。Step 4 和 Step 5 的目标是对 S 中的元组在 d_1 上赋值: 任意 $x \in \mathcal{L}Y_0, d_1(x) = \text{val}$, 任意 $x \in \text{POS}_C(D), d_1(x) = d(x)$ 。

算法 2 中相应语句可用 SQL 语言来实现, 如 Step 1 的 SQL 语句如下:

$\text{Card}(\Pi_B(S))$:

$\text{SELECT COUNT}(DISTINCT B) FROM S$ 。

根据文献[6], 如果数据库系统使用 Hash 表技术来实现 θ -Join 操作, 则算法 2 的时间复杂度为 $O(|U|)$ 。

利用算法 2 和文献[7, 8] 中计算核和约简的算法, 可以设计如下算法:

算法 3 输入: 决策表 $S = U, C, D = \{d\}$; 输出: S 的核和约简

Step 1: 用算法 2 获得 T_S ;

Step 2: 用文献[7, 8] 中计算核和约简的算法获得 T_S 的核和约简, 则 T_S 的核和约简就是 S 的核和约简。

算法 3 对不一致决策表也成立, 并且 Step 1 和 Step 2 都主要利用数据库系统中的集合操作来完成。因此算法 3 将基于数据库系统的 Rough 模型中的知识约简算法扩展到不一致决策表。

文献[7, 8] 中计算核和约简的算法时间复杂度均为 $O(|C \cup D| |U|)$ 。使用算法 3 计算核和约简的时间复杂度均为 $O(|C \cup D| |U|)$ 。由此可见, 用算法 3 计算核和约简的时间复杂度与文献[7, 8] 中计算核和约简的算法时间复杂度相同。因此, 对于大数据集的数据库系统, 算法 3 是高效的。

5 仿真实验

实验中使用例 1 和 UCI 机器学习数据库中的部分决策表。UCI 数据库中的数据是一致数据, 因此删除数据集中的一些条件属性便可获得不一致决策表。对于这些数据集, 使用文献[6, 7] 中的算法计算核和约简, 对于不一致决策表, 可用删除不一致数据的方法来计算核和约简^[6] (简称算法 4)。同时使用本文的算法 3 来计算其核和约简, 并比较两种算法的执行结果。实验所使用的数据库系统为 Oracle9i, 在 PC 机上进行实验, 其运行结果如表 6 所示。

表6 实验结果

决策表	实例数	算法 3				算法 4			
		核属性数	执行时间/s	约简属性数	执行时间/s	核属性数	执行时间/s	约简属性数	执行时间/s
例1	8	2	0.65	2	0.70	1	0.26	1	0.30
operative	90	8	0.86	8	0.86	8	0.13	8	0.13
diabetes	768	0	0.29	3	0.41	0	0.14	3	0.24
tic	958	0	0.24	8	0.36	0	0.16	8	0.32
hepatitis	155	10	1.27	10	1.23	9	0.18	9	0.24
horse	300	5	1.36	10	1.42	3	0.35	8	0.57

从表6可以看出,算法4不能正确计算这些决策表的核和约简,而算法3则能正确计算这些决策表的核和约简,但其执行时间要略高于算法4的执行时间。其原因在于转换算法增加了一些集合操作,并且算法4通过删除不一致数据能够减少对象数。尽管算法3与算法4时间复杂度相同,但算法3的实际执行时间要略高于算法4。

总之,算法3能正确、高效计算决策表的核和约简,而且算法3与数据库系统紧密结合,一些关键步骤可由数据库系统完成。与其他算法相比^[2-4],算法3特别适用于海量数据存储在数据库系统的情形。

6 结 语

针对基于数据库系统的Rough集模型存在知识约简算法对不一致决策表不适用的局限性,本文首先分析Han等人在处理这种局限性时存在的不足,即通过删除不一致数据后获得的核和约简并不是原决策表的核和约简。然后针对这种不足,给出了转换算法(算法1),该算法能在保持核和约简集合不变的前提下,将决策表转换为一致决策表。在证明转换算法的正确性后,分析了该算法的时间复杂度,结果表明,转换算法并没有增加计算核和约简算法的时间复杂度。最后将转换算法用数据库系统中的集合操作和SQL来描述(算法2),从而将基于数据库系统的Rough集模型中知识约简算法扩展到不一致决策表(算法3)。理论分析和实验结果表明,对于大数据集的数据库系统,扩展后的算法仍然高效。

参考文献(References)

- [1] Pawlak Z, Grzymala-Busse J, Slowinski R, et al. Rough Sets[J]. *Communication of the ACM*, 1995, 38(11): 89-95.
- [2] Wang J, Wang J. Reduction Algorithms Based on Discernibility Matrix: The Ordered Attributes Method[J]. *J of Computer Science and Technology*, 2001, 16(6):

489-504.

- [3] 王国胤,于洪,杨大春.基于条件信息熵的决策表约简[J].*计算机学报*, 2002, 25(7): 759-766.
(Wang G Y, Yu H, Yang D C. Decision Table Reduction Based on Conditional Information Entropy [J]. *Chinese J of Computer*, 2002, 25(7): 759-766.)
- [4] 刘少辉,盛秋骥,吴斌,等. Rough集高效算法研究[J].*计算机学报*, 2003, 26(5): 524-529.
(Liu S H, Sheng Q J, Wu B, et al. Research on Efficient Algorithms for Rough Set Methods [J]. *Chinese J of Computer*, 2003, 26(5): 524-529.)
- [5] 张文修,等. *粗糙集理论与方法*[M].北京:科学出版社, 2001.
(Zhang W X, et al. *Rough Set Theory and Methods* [M]. Beijing: Science Press, 2001.)
- [6] Han J C, Hu X H, Lin T Y. An Efficient Algorithm for Computing Core Attributes in Database Systems [A]. *14th Int Symposium on Methodologies for Intelligent Systems* [C]. Springer-Verlag, 2003: 663-667.
- [7] Han J C, Hu X H, Lin T Y. A New Computation Model for Rough Set Theory Based on Database Systems [A]. *5th Int Conf on Data Warehousing and Knowledge Discovery* [C]. Springer-Verlag, 2003: 381-390.
- [8] Hu X H, Lin T Y, Han J C. A New Rough Sets Model Based on Database Systems [A]. *Proc of 9th Int Conf on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing* [C]. Springer-Verlag, 2003: 114-121.
- [9] Hu X H. Using Rough Sets Theory and Database Operations to Construct a Good Ensemble of Classifiers for Data Mining Applications [A]. *Proc of the 2001 IEEE Int Conf on Data Mining* [C]. Washington, 2001: 233-240.
- [10] Akhil Kumar. New Techniques for Data Reduction in a Database System for Knowledge Discovery Applications [J]. *J of Intelligent Information Systems*, 1998, 10(1): 31-48.