

Rough 关系数据库模型及其关系操作^{*})

Rough Relational Database Model and its Relational Operations

安秋生^{1,2} 徐久成^{1,3} 沈钧毅¹ 王国胤⁴

(西安交通大学软件所 西安710049)¹ (山西师范大学数计系 临汾041004)²

(河南师范大学计算机系 新乡453002)³ (重庆邮电学院计算机科学与技术研究所 重庆400065)⁴

Abstract In this paper, Rough Relational Database Model (RRDM) and several ways for extending relational database model (RDM) are introduced at first. Based on literature [2], some different definitions of RRDM and its relational operations are developed then. Some basic properties are discussed also. Finally, the difference and relationship between RRDM and RDM are studied, some interesting results of RRDM and some problems that are studied further in the future are presented.

Keywords Rough set, Union compatible, Decompose

1 引言

E. F. Codd 于七十年代提出了著名的关系数据库模型, 经过几十年发展, 以它为基础的关系数据库系统得到了广泛的流行和认可, 它有着层次模型和网状模型无可比拟的优点, 如灵活性、逻辑和物理独立性、数据完整性等。然而, 它也有自身的不足, 就是对于不确定性的数据处理能力较差。随着应用面的扩大和数据量的急剧膨胀, 出现了各种不确定信息、不完全信息、模糊信息, 而关系数据库的处理能力主要是对确定信息的处理, 因此, 十分有必要对关系数据库模型进行扩充。

到目前为止, 可以说对于关系数据库的扩充有六种方法。第一种是扩展关系代数使之能处理“null”值, 且提供“null”值的语义, 这种方法假设数据值是确定的或是未知的, 这对于大多数真实的应用太严格了, Lipski 讨论了使用“null”值表示数据不完备性的语义的问题。尽管他的模式比平常的“null”值解释更具有实用性, 但仍不能解释随机的自然数据问题^[1]。

第二种方法是 Wong、Mendelson、Saharia、Dey 等提出的。他们认为数据库项不能完全描述是由于较高的关联成本。他们从不确定性出发对信息检索和制定决策进行建模, 但忽略了与数据项自身相关联的内在的不确定性。这种方法适合于数据存储层的处理, 但不适合数据不确定性的表达^[1]。

第三种方法使用模糊集合论方法对数据不确定性建模。这种方法比前两种方法在处理不确定数据问题上有了很大的进步, 但它同样有一定的局限性, 它不太适合于商业处理^[1]。

第四种处理方法由多人的工作组成。Cavallo、Pittarelli 在处理不确定性数据时使用了众所周知的概率微积分学; Raju、Majumdar 使用模糊关系方法概括了关系的基本概念; Barbara 等提出了使用概率论来扩充关系模型; 基于 Barbara 的模型, Tseng 等开发了一个扩展的关系操作集合来表示由于数据异构所带来的不确定性。这些方法的主要问题是对于关系代数的完备性及关系运算的灵活性等^[1]。

第五种方法是 Debabrata Dey 和 Sumit Sarkar 于1996年

提出的“概率关系模型”, 它很好地处理了确定性和不确定性的关系, 并且其所有关系均符合第一范式, 其关系是封闭的。这种方法的局限性也很明显, 就是在这种模型的每个关系的元组中, 都要给出一个关联概率度量, 这对于大型数据库来说是不现实的。

最后一种方法是利用 Rough 集方法来处理不确定性数据问题。近年来, 由于 Rough 集理论 (Rough Sets Theory, 简称 RST) 的兴起, 由 RST 和关系数据库理论结合形成的 Rough 关系数据库模型便脱颖而出, 使得人们在处理不确定性问题上又向前迈了一步。Theresa Beaubouef、Frederick E. Petry 于1993年提出了 Rough 关系数据库模型 (the rough relational database model, 简称 RRDM), 并在随后的工作中对它进行了深入的研究, 由于 RST 本身具有坚实的数学理论基础, 由此而形成的 Rough 关系数据库模型比前面几种方法具有更多的优点。

2 RRDM

2.1 Rough 集的一些基本概念

设 U 是非空有限论域, R 是 U 上的二元等价关系, R 称为不可分辨关系, 把 $A = (U, R)$ 称为近似空间。 $\forall (x, y) \in U \times U$, 若 $(x, y) \in R$, 则称对象 x 与 y 在近似空间 A 中是不可分辨的。 U/R 是 U 上由 R 生成的等价类全体, 它构成了 U 上的一个划分。对于 U 上的任意子集 X , X 不一定能用知识库中的知识来精确描述, 即 X 可能为不可定义集, 这时就用 X 关于 A 的上、下近似算子来“近似”定义^[5,6]:

$$\overline{\text{apr}} = U[x] \mid [x] \cap X \neq \emptyset = \{x \in U \mid [x] \cap X \neq \emptyset\}$$

$$\underline{\text{apr}} = U[x] \mid [x] \subseteq X = \{x \in U \mid [x] \subseteq X\}$$

2.2 RRDM

RRDM 是结合了 RST 的一些基本概念如等价类、上近似、下近似和关系数据库的一些基本理论而形成的, 它是对关系数据库的扩展。在文[2]中, 对 RRDM 的一些基本概念进行了定义, 本节在此基础上进行了补充定义, 并得出一些相应的

^{*}) 本文得到国家自然科学基金 (No. 69803014)、攀登——特别支持费、教育部骨干教师资助计划、重庆市应用基础研究基金、信息产业部计算机网络与通信技术重点实验室研究基金资助。安秋生 博士研究生, 讲师, 主要研究方向: 数据库理论和 Rough 集。徐久成 博士研究生, 副教授, 主要研究方向: 数据库理论和 Rough 集。沈钧毅 教授, 博导, 主要研究方向: 数据库与软件集成, 数据仓库, 数据挖掘。王国胤 博士, 教授, 主要研究方向: 神经网络, Rough 集, 知识获取, 智能信息系统等。

结论。

RRDM 同普通的关系模型一样均是由包含元组的关系构成,元组 t_i 采用 $(d_{i1}, d_{i2}, \dots, d_{im})$ 的形式,其中 d_{ij} 是元组的一个属性值,隶属属性域 D_j 。在普通的关系数据库模型中, $d_{ij} \in D_j$, 在 RRDM 中, $d_{ij} \subseteq D_j, d_{ij} \neq \phi$ (注:文[2]中定义 $d_{ij} \neq \phi$, 我们将在随后的研究中对它进行扩充),我们用 $P(D_i)$ 表示 $D_i - \phi^{[2]}, P(D_i)$ 为某一属性域。这里,RRDM 与普通的关系数据库的一个不同点是它的属性值可以由多个原子值构成,而不象关系数据库是单属性值。

定义1^[2] Rough 关系 R 是 Rough 元组的有穷集合,它是集合叉积 $P(D_1) \times P(D_2) \times \dots \times P(D_m)$ 的一个子集。

在关系数据库中, R 的值用 r 或 $r(R)$ 表示,它是 n 个元组的集合, $r = \{t_1, t_2, \dots, t_n\}$, 每个元组 t_i 可以表示为 $(v_{i1}, v_{i2}, \dots, v_{im}), v_{ij} \in D_j, 1 \leq i \leq n$, 即 $t_i \in D_1 \times D_2 \times \dots \times D_m$, 而 $r \subseteq D_1 \times D_2 \times \dots \times D_m$ 。

Rough 关系的定义是由关系数据库中的关系定义引申而来,在 RRDM 中,它的一个元组可以称之为 Rough 元组,任意一个 Rough 元组 t_i 是 Rough 关系 R 的一个成员,这就意味着它也是 $P(D_1) \times P(D_2) \times \dots \times P(D_m)$ 的一个成员,即 $t_i \in P(D_1) \times P(D_2) \times \dots \times P(D_m)$ 。

上面是在文[2]中对 Rough 关系数据库给出的基本定义,下面我们结合 RST 的一些基本知识给出 Rough 关系数据库相关的命题。

命题1 给定 Rough 关系 R , 其值用 r 或 $r(R)$ 表示,它具有属性 $(a_1, a_2, \dots, a_m)$, R 是 n 个元组的集合, $R = \{r_1, r_2, \dots, r_n\}$, 元组 r_i 可以表示为 $(v_{i1}, v_{i2}, \dots, v_{im}), 1 \leq i \leq n$, 给出一个操作结果集合 $X = \{x_1, x_2, \dots, x_k\}$, 元组 x_j 可以表示为 $(x_{j1}, x_{j2}, \dots, x_{jm}), 1 \leq j \leq k$, 则集合 X 可用上、下近似表示为:

$$\underline{RX} = \{r_i | \exists i (r_i \in R \wedge r_i \cap X \neq \phi)\}$$

$$\overline{RX} = \{r_i | r_i \in R \wedge r_i \subseteq X\}$$

本命题可以由上、下近似的定义直接推得。

命题2 设有 Rough 关系 R , 它的任一属性为 a_i , 则它对应于一等价类或等价关系 $[a_i]$, 对于某一 Rough 关系操作结果集 X 来说,可用下列上、下近似来表示为:

$$\underline{RX} = \{[a_i] | [a_i] \in R \wedge [a_i] \cap X \neq \phi\}$$

$$\overline{RX} = \{[a_i] | [a_i] \in R \wedge [a_i] \subseteq X\}$$

本命题同样也可以由上、下近似的定义推得。

定义2^[2] Rough 元组 $t_i = (d_{i1}, d_{i2}, \dots, d_{im})$ 的一个解释 $\alpha = (a_1, a_2, \dots, a_m)$: 对 $\forall j (1 \leq j \leq m)$ 来说,是进行值的分配,即使 $a_j \in d_{ij}$ 。

传统的关系数据库由于其属性值是原子的,因此其解释即元组本身,而 Rough 关系数据库,其属性值可能是由多个原子值组合而成的属性值,因此便有了对它的多个指派,即对其每个 d_{ij} 抽取其一原子值而形成该元组的一个指派即解释。

命题3 若 Rough 元组 $t_i = (d_{i1}, d_{i2}, \dots, d_{im})$ 的所有解释为 $\alpha_1, \alpha_2, \dots, \alpha_n$, 则 $t_i = \alpha_1 \cup \alpha_2 \cup \dots \cup \alpha_n$, 这个结论是显然的。

命题4 若 Rough 元组 $t_i = (d_{i1}, d_{i2}, \dots, d_{im})$ 的一个解释为 $\alpha = (a_1, a_2, \dots, a_m)$, 则 $t_i \in \overline{R\alpha}$, 这是明显的,因为对它的任一元素 $a_i \in d_{ij}$, 因此有 $t_i \cap \alpha \neq \phi$, 故结论成立。

在特殊情况下,即关系数据库(单属性值)中,解释 $\alpha = (a_1, a_2, \dots, a_m)$ 即元组本身,这时 $t_i \in \overline{R\alpha}$, 并且 $t_i \in \underline{R\alpha}$ 。

命题5 对于元组 $t_i = (d_{i1}, d_{i2}, \dots, d_{im})$ 和 $t_k = (d_{k1}, d_{k2}, \dots, d_{km})$ 来说,对于所有的 $j = 1, 2, \dots, m$, 若满足 $(d_{ij}, d_{kj}) \in \text{ind}_R(A_j)$, 则 t_i, t_k 是冗余的。

这个结论可由不可分辨原理得出。

定义3 (Rough 关系数据库模式) 设 R 为一 Rough 关系, 它的属性为 $A_1, A_2, \dots, A_m$, 其对应的域为 $P(D_1), P(D_2), \dots, P(D_m)$, 则 Rough 关系可以表示为:

$$R = (A_1/P(D_1), A_2/P(D_2), \dots, A_m/P(D_m))$$

$$\text{或 } R = (A_1, A_2, \dots, A_m)$$

其中 A_i 也称为 Rough 关系 R 的等价类,或等价关系,它的全体构成了一个等价类簇,上式是对 Rough 关系的类型的描述,称为 Rough 关系 R 的模式^[7]。

定义4 (Rough 关系数据库的键) 对于关系数据库 $R = (A_1, A_2, \dots, A_m)$ 来说,若其中某一属性或属性组的值(原子的或非原子的)能够唯一地决定一个元组,而其它任何真子集无此性质,则该属性或属性组称为该 Rough 关系数据库的键(记做 $\text{RRD-KEY}(R)$)。

在普通的信息系统(单值)中,它的键为最小的恒等依赖集,核为所有重要属性的集合,而它的核为所有关键字(键)的交集^[3]。有了 Rough 关系数据库模式及 Rough 关系数据库的键的定义,我们便可以研究 Rough 关系数据库中的键与核、属性约简、值约简、属性依赖之间的关系了。

3 Rough 关系操作

3.1 Rough 关系代数

一般地,我们把由 $RS = (2^U, \cup, \cap, -, \text{apr}, \overline{\text{apr}})$ 所构成的代数系统称为 Rough 集合代数,它是标准集合代数 $(2^U, \cup, \cap, -)$ 的扩展, Rough 关系操作便是以它为基础的。

3.2 Rough 关系操作

Rough 关系操作是在传统的关系操作的基础上通过对关系操作进行修正后形成的,形成的 Rough 操作符算子集为 $(-, \cup, \cap, \sigma, \Pi, \infty)$, 在文[2]中对于这六种操作分别进行了定义,本节在这里对其定义分别予以介绍,并对其表达不规则之处进行了修正,然后定义了一种新的算子—Rough 关系分解算子 Γ , 并得出与之相关的一些结论。

定义5^[2] 并兼容两个 Rough 关系 $X(A_1, A_2, \dots, A_m), Y(B_1, B_2, \dots, B_m)$, 如果在它们的关系模式中有相同的属性数,并且对所有的 $i = 1, 2, \dots, m, A_i$ 与 B_i 的域相同,则称它们为并兼容。

3.2.1 Rough 差算子

定义6^[2] 对于两个并兼容的 Rough 关系 X, Y 来说, $X - Y$ 的结果是一个 Rough 关系 T , 它满足: $\underline{RT} = \{t | t \in \underline{RX} \wedge t \notin \underline{RY}\}$ 并且 $\overline{RT} = \{t | t \in \overline{RX} \wedge t \notin \overline{RY}\}$

3.2.2 Rough 并算子

定义7^[2] 对于两个并兼容的 Rough 关系 X, Y 来说, $X \cup Y$ 的结果是一个 Rough 关系 T , 它满足: $\underline{RT} = \{t | t \in \underline{RX} \cup \underline{RY}\}$ 并且 $\overline{RT} = \{t | t \in \overline{RX} \cup \overline{RY}\}$

3.2.3 Rough 交算子

定义8^[2] 对于两个并兼容的 Rough 关系 X, Y 来说, $X \cap Y$ 的结果是一个 Rough 关系 T , 它满足: $\underline{RT} = \{t | t \in \underline{RX} \cap \underline{RY}\}$ 并且 $\overline{RT} = \{t | t \in \overline{RX} \cap \overline{RY}\}$

3.2.4 Rough 选择算子

定义9^[2] Rough 选择 $\sigma_{A=a}(X)$ 是从 X 中选择元组,其结果是一个 Rough 关系 Y , 它具有与 X 相同的模式, 设 $a = \{a_i\}, a_i, b_i \in \text{dom}(A_i)$, 则:

$$\underline{RY} = \{t | t \in X \wedge U_i[a_i] \subseteq U_i[b_i], a_i \in a, b_i \in t(A_i)\}$$

$$\overline{RY} = \{t | t \in X \wedge U_i[a_i] = U_i[b_i], a_i \in a, b_i \in t(A_i)\}$$

3.2.5 Rough 投影算子

定义10^[2] X 在 B 上的 Rough 投影记为 $\prod_B(X)$, 是一个具有模式 $Y(B)$ 的关系 $Y: Y(B) = \{t(B) | t \in X\}$

3.2.6 Rough 连接算子

定义11 1) Rough 条件连接 设 $X(A_1, A_2, \dots, A_m), Y(B_1, B_2, \dots, B_n)$ 分别为具有 m 个和 n 个属性的 Rough 关系, 则:

$X \bowtie_{(\text{连接条件})} Y$ 的结果为 $T(C_1, C_2, \dots, C_{m+n})$, 这里 T 为:

$$\{t | \exists t_x \in X, t_y \in Y, t_x = t(A), t_y = t(B)\},$$

并且 $t_x(A \cap B) = t_y(A \cap B) \quad t \in \underline{RT}$

$t_x(A \cap B) \subseteq t_y(A \cap B) \text{ or } t_y(A \cap B) \subseteq t_x(A \cap B) \quad t \in \overline{RT}$.

其(连接条件)是一个或多个类似 $A=B$ 形式的条件。

2) 无条件 Rough 连接 Rough 连接也可以有无条件连接, 在这种情况下, 连接则变成了笛卡尔积。

3) Rough 自然连接 在上述条件连接(连接条件)中, 把这个条件表示为 $A_i \theta B_j$, 如果这个条件中的 θ 为 $=$, 则此种连接可以称为 Rough 对等连接。在对等连接中, 如果对应属性的值是相同的, 则出现列的冗余, 没有必要重复列出, 只要在两个对应属性中取一个就可以了, 这种消除冗余属性的对等连接称为 Rough 自然连接。

3.2.7 Rough 关系分解算子 Γ 我们知道, Rough 关系数据库的属性值具有非原子性, 在某些时候, 为了操作的需要, 我们需要对其属性值只取其中的一部分, 这样所形成的粗糙元组的集合便构成了 Rough 关系数据库的一个分解。

定义12 给定 Rough 关系 R 与 r , R 具有属性 $(a_1, a_2, \dots, a_m)$ 它是 n 个元组的集合, 其属性域为: $(D_1, D_2, \dots, D_m)$, 属性值用 $R(a_{ij})$ 表示, Rough 关系 r 与 R 具有相同的属性与属性域, 它的属性值用 $r(a_{ij})$ 表示, 称 Rough 关系 r 为 Rough 关系 R 的一个分解, 记为 $r = \Gamma(R)$, 其中 $r(a_{ij}) \subseteq R(a_{ij})$ 。

在特殊情况下, r 的每个元组均为 R 对应元组的一个解释, 则此时 Rough 关系数据库 r 变为标准的关系数据库。

命题6 设 Rough 关系 R 的元组集合为 $(R_1, R_2, \dots, R_n)$, $a_1, a_2, \dots, a_n$ 为 R 的 n 个解释, 则由 $(a_1, a_2, \dots, a_n)$ 形成的关系 r 必定为 R 的一个分解。

定义13 命题4中的关系 r 称为 Rough 关系 R 的最小分解, R 的最大分解为 R 本身。

定理1 若 X, Y 为 Rough 关系 R 的任两个分解, 则必有 $\overline{RX} = \overline{RY}$

证明: 设 X, Y 为 Rough 关系 R 的任两个分解, R 由元组集合 $(r_1, r_2, \dots, r_n)$ 组成, X 由元组集合 $(a_1, a_2, \dots, a_n)$ 构成, k_{ij} 为 X 的任一元组 $a_i (1 \leq i \leq n)$ 的任一属性值, v_{ij} 为 R 的任一元组 r_i 的任一属性值, 因为 X 为 R 的一个分解, 故必有 $k_{ij} \subseteq v_{ij}$, 因而 $r_i \cap a_i \neq \emptyset, r_i \subseteq \overline{RX}, \overline{RX} = \{r_1, r_2, \dots, r_n\} = R$, 同理得 $\overline{RY} = \{r_1, r_2, \dots, r_n\} = R$, 故 $\overline{RX} = \overline{RY}$ 。

推论1 设 X 为 Rough 关系 R 的任一分解, 则必有 $\overline{RX} = R$ 。

定理2 给定 X 为 Rough 关系 R 的任一分解, X 由元组集合 $(a_1, a_2, \dots, a_n)$ 构成, k_{ij} 为 X 的任一元组 $a_i (1 \leq i \leq n)$ 的任一属性值, v_{ij} 为 R 的任一元组 r_i 的任一属性值, $\forall a_i$, 若 $\exists k_{ij} \subset v_{ij}$, 则 $\overline{RX} = \emptyset, \overline{RX} = R$ 。

证明: 设 X 为 Rough 关系 R 的任一分解, 它由元组集合 $(a_1, a_2, \dots, a_n)$ 构成, k_{ij} 为 X 的任一元组 $a_i (1 \leq i \leq n)$ 的任一属性值, v_{ij} 为 R 的任一元组 r_i 的任一属性值, $\forall a_i$, 若 $\exists k_{ij} \subset v_{ij}$,

则按照定义, $r_i \in X$ 不成立, 没有一个元组是 X 的元素, 故 $\overline{RX} = \emptyset, \overline{RX} = R$ 的证明同定理1。

推论2 Rough 关系 r 为 Rough 关系 R 的分解的充要条件是 $k_{ij} \subseteq v_{ij}$ (其中 k_{ij} 为 X 的任一元组 $a_i (1 \leq i \leq n)$ 的任一属性值, v_{ij} 为 R 的任一元组 r_i 的任一属性值)。

4 RDM 与 RRDM 的比较分析

RRDM 是以关系模型和 Rough 集理论为基础, 是对关系模型的扩充, 但是它与关系模型又有很大的不同, 下面结合关系模型简要分析一下 RRDM。

两个模型的共同特点是它们都是以关系为基本单位, 而这种关系本质上是一个集合, 可以说, 它们都是笛卡尔积的子集; 另外, 同关系代数一样, Rough 关系同样具有封闭性, 即一个或多个 Rough 关系通过各种 Rough 操作后的结果仍然是 Rough 关系。两个模型的不同之处可以归结为以下几点:

1) 它们依赖的数学理论不同。传统的关系模型是以关系代数 $(2^U, \cup, \cap, -)$ 为基础, 而 RRDM 则是以 Pawlak 粗集代数系统 $(2^U, \cup, \cap, -, \text{apr}, \overline{\text{apr}})$ 为基础。

2) 传统的关系模型有自己成熟的范式理论, 它要求关系模式下的每个关系最起码要符合第一范式(1NF), 即其属性值是原子属性, 不可再分解, 而 RRDM 则不符合第一范式理论($\rightarrow$ 1NF), 它的属性值可以是多个原子值的组合, 它的理论有待完善。

3) 属性的原子性差异。传统的关系数据库的每个属性值是原子的, 它是它所对应的属性域的一个元素, 而 RRDM 的一个属性值可以是一个由多个原子值组合而成的非原子值, 它可以是一个集合, 它是所属属性域的一个子集, 这里, RRDM 对于传统的关系定义进行了拓展, 它的每个属性相当于一个等价关系或等价类。

4) 两个模型的操作及算子的差异。传统的关系模型的操作包括基本的集合的运算、专门的关系运算(选择、投影、连接、除), 即其算子集为 $(-, \cup, \cap, \sigma, \pi, \infty, \div)$, 而 RRDM 的操作中则去掉了其中的除运算, 增加了新的算子, 其操作符算子集为 $(-, \cup, \cap, \sigma, \prod, \infty, \Gamma)$, 并且相应的运算含义也发生了变化, 其操作的特性也有所不同。

5) 两个模型研究处理的范围不同。关系模型以处理确定性数据为主, 它对不确定知识的处理能力则较差, 而 RRDM 则是以处理不确定知识为主, 它是对关系数据库处理能力的补充。从一定意义上讲, 在处理不确定性问题时, RRDM 能够更好地表达属性间的语义联系, 这里, 我们可以把普通关系看作是 Rough 关系的特例, 当 Rough 关系中的属性值都为原子值时, 则 Rough 关系转化为传统的关系。可以说, 两种模型处理的能力是互补的。

6) 由于 RRDM 是由关系数据库与 Rough 集理论形成的, 因此为 RRDM 的研究增加了新的方法和工具。比如对于其中函数依赖 EFD: $X \rightarrow Y$ 的研究, 可以采用关系数据库中的函数依赖的方法, 也可以转化为 RS 中知识依赖的研究: $KD: Rcol(X) \Rightarrow Rcol(Y)$; 对于 RRDM 中候选键的研究可转化为 RS 中属性约简的研究等^[4]。

Rough 关系数据库模型从根本上说是基于非第一范式的, 它是对关系数据库模型的扩展, 是吸收了 Rough 集理论的不可分辨原理、等价类、等价关系和上、下近似概念而形成的。由于它的属性字段是丧失了原子性特性所致, 这在很大程度上比较符合真实世界的情况, 但同样是由于失去了原子性, 使得传统关系数据库的许多特性不再适合于它。

(下转第89页)

读入内存,导致大量使用虚拟内存,从而使性能迅速下降。而 GDBR 算法不会出现这样的情况。

4.2 基于抽样的概念层次挖掘算法分析

设 η 为抽样率, n 为初始相关数据集中元组的个数,则提取抽样数据时间为 $\eta \cdot n$ 。设对一概念进行概念层次调整时间为常数 c_1 ,即第一次动态调整概念层次的时间为 $c_1 \cdot \eta \cdot n$ 。设转化一个概念时间为一常数 c_2 ,则对整个初始相关数据集的扫描时间为 $c_2 \cdot n$ 。最后设第二次概念层次的调整和对 prime-relation 的调整时间为常数 c_3 ,所以总的时间复杂度为 $O(\eta \cdot n + c_1 \cdot \eta \cdot n + c_2 \cdot n + c_3) = O(n)$ 。

设元组的属性个数为 m ,各属性的不同概念数为 $n_i (1 \leq i \leq m)$, t_i 为第 i 属性的泛化阈值, $n(\max)$ 为属性的最大概念数,则 Buf 占空间为 $n(\max)$ 。初始概念层次 $p = \sum c_i \cdot n_i$,其中 c_i 为概念转化为概念层次的常数。最后的泛化规则 prime-relation 占空间为 $s = \prod t_i$,所以总的空间复杂度为 $O(n(\max) + p + s) = O(c)$, c 为常数。

五、基于抽样的概念层次挖掘算法模拟测试

我们在以下环境中对算法进行了测试。开发环境:Microsoft Visual C++ 6.0 Windows NT 4.0 Server;运行环境:IBM 兼容机,Intel Pentium II 233,128M RAM。对每一输入元组个数,模拟运算10次,取其平均值作为时间。具体数据如图6所示。

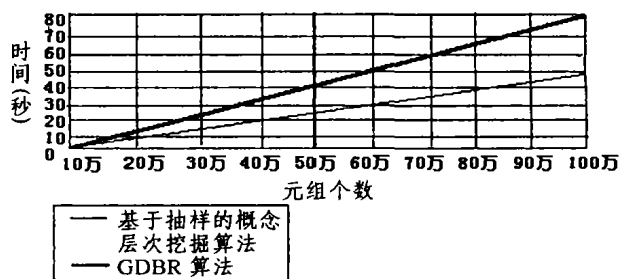


图6 测试结果

从图6我们可以看出基于抽样概念层次挖掘算法的执行时间与输入元组个数基本上是线性的,而且其处理大数据的能力是令人满意的,即具有较好的可扩展性。此外,我们还发

现基于抽样的概念层次挖掘算法比 GDBR 算法执行时间短,这主要是因为基于抽样的概念层次挖掘算法在概念转化表(即原概念层次中的概念映射到调整后的概念层次中的概念的映射表)建立后,读入概念后,可直接得到其对应的泛化概念,而不用象 GDBR 算法那样读入一个概念后就要在全局概念表中查找,从而加快了扫描速度。

总结 本文分析了几种传统属性归纳算法,针对它们的不足,提出了基于抽样的概念层次挖掘算法,它不仅可以处理不平衡的概念层次,而且得到的泛化规则可以反映实际的数据分布。此外,这种算法具有最优的时间和空间复杂性。实验证明本文算法是有效的、可行的。

由于数据库的数据经常改变,并且重新计算一次泛化规则的代价非常昂贵,因此如何将基于抽样的概念层次挖掘算法与增量泛化和动态维护相结合,是值得进一步研究的内容。

参考文献

- 1 Han Jiawei, Fu Yongjian. Exploration of the Power of Attribute-Oriented Induction in Data Mining
- 2 Han Jiawei, Fu Yongjian. Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Database. <http://www.cs.uregina.ca>.
- 3 Agrawal R, Imielinski T, Swami A. Mining Association Rules Between Sets of Items in Large Databases. In Proc. 1993 ACM-SIGMOD Int. conf. Management of Data, Washington, D. C. : ACM Press, 1993. 207~216
- 4 Piatetsky-Shapiro G, Frawley W J. Knowledge Discover in Database. AAAI/MIT Press, 1991
- 5 Han J, Cai Y, Cercone N. Data-Driven Discovery of Quantitative Rules in Relational Databases. IEEE Trans. Knowledge and Data Eng., Feb. 1993. 29~40
- 6 Cai Y N. Attribute-oriented Induction in Relational Databases. In: G. Piatetsky-Shapiro, W. J. Frawley, eds. Knowledge Discovery in Databases, AAAI/MIT Press, 1991. 213~228
- 7 Carter C L. Efficient Attribute-Oriented Algorithm for Knowledge Discover from Large Databases. IEEE trans on Knowledge and data Engineering, 1998, 10(2): 193~208

(上接第74页)

结束语 本文在对文[2]提出的 RRDM 深入研究的基础上,对 RRDM 的基本定义进行了扩充,并提出了一种新的 RRDM 操作算子—分解算子,针对 RRDM 的基本概念和 Rough 关系操作,得出了一些相应的结论。

Rough 关系数据库模型是对传统关系模型的扩充,它是为研究不确定性数据问题而发展起来的一个新学科,目前国内外对这个学科的研究还处于初级阶段,现在研究较多的是把 Rough 集理论和别的数学工具结合起来对信息系统处理,而真正把 Rough 集和关系数据库本身的理论结合起来进行研究的比较少。在 Rough 关系数据库理论研究方面还有很多不足,需要研究的问题还有很多,比如 Rough 关系演算、Rough 域演算,基于 Rough 关系数据库的数据约束、知识抽取、函数依赖理论、不确定知识的度量等、Rough 关系及其操作的性质等,从而建立它自己的完善的理论体系,使之更好地为实际应用服务。

参考文献

- 1 Dey D, Sarkar S. A probabilistic relational model and algebra. ACM Transactions on Database Systems, 1996, 21(3): 339~369
- 2 Theresa B, Petry F E, Buckles B P. Extension of the relational database and its algebra with rough set techniques. Computational Intelligence, 1995, 11: 233~245
- 3 Guan J W, Bell D A. Rough Computational methods for information systems. Artificial Intelligence, 1998, 105: 77~103
- 4 Lin T Y. An Overview of Rough Set Theory from the Point of View of Relational Databases. Bulletin of Internal Rough Set Society. Volume 1, Number 1
- 5 张文修, 吴伟志. Rough 集理论介绍和研究综述. 模糊系统与数学, 2000, 14(4)
- 6 王国胤编著. Rough 集理论与知识获取. 西安交通大学出版社, 2001, 5
- 7 王能斌编著. 数据库系统原理. 电子工业出版社, 2000, 1