第 34 卷 第 4 期 计 算 机 工 程 2008 年 2 月
Vol.34 No.4 Computer Engineering February 2008
·软件技术与数据库· 文章编号：1000—3428(2008)04—0113—03 文献标识码：A 中图分类号：TP311.13

# 粗关系数据库中的数据更新

魏玲玲，邱桃荣，刘 萍
(南昌大学计算机系，南昌 330031)

**摘 要**：根据粗关系数据库中数据的特性，借助邻接表、十字链表存储不确定性数据，其中邻接表用于等价类的存储，十字链表用于数据库中基本表的存储。与传统的关系数据库更新不同，在粗关系数据库中更新基本表时，相应地等价类也要随之更新，该存储结构加快了对数据库中的数据更新速度。将算法与实例相结合，根据用户条件详细地讨论对等价类和 RRDB 中基本表的数据更新。
**关键词**：粗关系数据库；数据更新；等价类；邻接表；十字链表

# Data Updating of Rough Relational Database

WEI Ling-ling, QIU Tao-rong, LIU Ping
(Department of Computer, Nanchang University, Nanchang 330031)

**【Abstract】** According to the data characteristic of the rough relational database, this paper solves the problem of uncertainty data storage with adjacency list and orthogonal list, and the adjacency list is for the equivalence classes storage and the orthogonal list for the basic table storage. The data updating of RRDB is different from the relational database, in which the equivalence class updates according to the basic table. This kind of storage structure can update the data quickly. In order to further discuss this problem, an algorithm for updating the data in RRDB is proposed and illustrated by using soil analysis example.
**【Key words】** rough relational database; data updating; equivalence class; adjacency list; orthogonal list

B.Theresa等人将Rough集理论与传统关系数据库模型相结合，于 1993 年提出了粗关系数据库模型(Rough Relational Database Model, RRDM)，主要处理不确定性数据的存储和查询，它的属性值可以划分成若干个等价类，属性值是这些等价类的并集[1]。然而在构建数据库时，并不能一次性预测属性值的所有取值情况，而是在实践中完善和补充，这样原来的等价类需要不断地更新、动态地改变等价类。在更新基本表时，也可能会更新等价类。因此在数据存储和更新时，要考虑到等价类的存储和更新。但是从目前研究者所研究的内容来看，大部分集中在RRDB的理论、查询等方面，如文献[2]分析了RRDM的空间结构，定义了属性集之间的部分、传递依赖关系和粗糙关系数据库的规范，定义粗糙关系数据库的连接算子等；如文献[3]讨论了RRDB的查询理论，并提出了粗糙数据查询。关于RRDB的数据存储结构，目前还没有得到研究，下面基于RRDB中数据的特性，提出RRDB中数据的存储结构。

## 1 RRDM 的定义和表示

RRDM是传统关系数据库模型的一种扩展[4]，打破了关系数据库中第一范式理论限制，属性值由多个原子值组合而成的非原子值，它的属性值可以划分成若干个等价类，属性值是这些等价类的并集，同时它也继承了Rough集中的基本特性，如上、下近似，等价类等。

关系的描述称为关系模式，将粗关系数据库模型形式化表示为三元组[5]：$R(U,A,D_j)$。其中，$U$表示所有元组的集合，$U=\{u_1,u_2,\cdots,u_i,\cdots\}$是对象的全体，为非空有限集；$A$表示数据库的属性集，$A=\{A_1,A_2,\cdots,A_i,\cdots\}$是属性的全体，也是非空有限集；$D_j(1\leq j\leq|A|)$为属性$j$的值域，它是由若干个等价类构成的等价关系。它的元组$t_i$采用传统关系数据库的形式$(d_{i1},d_{i2},\cdots,d_{in})$，所不同的是在传统关系数据中$d_{ij}\in D_j$，而在RRDB中$d_{ij}\subseteq D_j(d_{ij}\neq\Phi)$，这是因为粗关系数据库是由多值组合而成的属性值，见表 1。

**表 1 土壤信息**

| ID | COLOR | P-SIZE |
|---|---|---|
| P21 | Brown | Medium |
| P22 | {Black, tan} | Large |
| P23 | Gray | {Medium, Small} |
| T01 | Black | Tiny |
| T04 | {Gray, Brown} | Large |

## 2 等价类存储结构和 RRDB 中基本表存储结构

### 2.1 等价类存储方式

等价类是一组相近、相包含的元素组成的类。在 RRDB 中它是按属性来划分等价类，一个属性有一个或若干个等价类。在 RRDB 中，对基本表数据操作涉及到等价类，因此在存储数据时，也要考虑更新相应的等价类。从而本文考虑用一种存储结构来存储等价类——邻接表(图 1)。

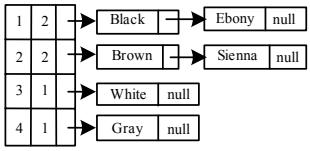


**图 1 Color 邻接表**

邻接表是图的一种链式存储结构[6]，在这将其作为RRDB

**基金项目**：江西省科技基金资助重点项目(20061B01002)
**作者简介**：魏玲玲(1981－)，女，硕士研究生，主研方向：数据库技术，人工智能及其应用；邱桃荣，副教授；刘 萍，讲师、硕士
**收稿日期**：2007-03-31 **E-mail**：rainweiling@163.com

中等价类的存储方式。其结构由头结点和表结点组成。头结点由 2 个域组成，其中信息域(Info)存储同一等价类元素个数，链域(Firstarc)存储等价类元素的地址；表结点也由 2 个域组成，其中邻接点域(Adjvex)存储等价类中的元素，链域(Nextarc)指向同一等价类中的下一个元素。

### 2.2 RRDB 基本表存储方式

传统关系数据库中数据存储是属性由定长或变长的字节序列表示，称为“字段”；然后字段被组装成定长或变长的集合，称为“记录”。数据就以这种方式存储在二级存储器中[7]。由于传统关系数据库处理的是确定性数据，而RRDB处理的是不确定性数据，属性值由多个原子值组合而成的非原子值，如果按传统的方式存储一方面会浪费存储空间；另一方面会对数据的更新带来不便，从而可能失去数据的一致性。在本文中，结合RRDB特性与存储器的特点，采用十字链表的方式存储RRDB中的数据。其结构是：每个结点有一个数据域(Data)，2 个指针域；数据域存放数据；一个指针域(Nexttuple)指向同一属性的下一个元组值；另一指针域(Nextattr)指向下一个属性的属性值。如一个属性值是多个值，用数组存储。如图 2 所示。

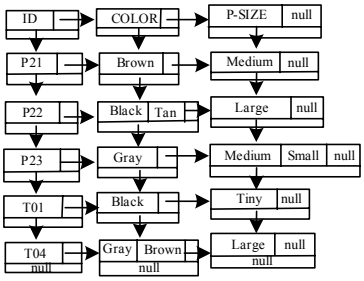


**图 2　表 1 的十字链表**

## 3　RRDB 数据的更新

一条记录更新，如果更新前的记录与更新后的记录在存储结构和存储空间上没有改变，对存储系统没有影响，因为两者占用的空间容量和结构是一样。但当一条记录与更新前的记录大小不一时，就要对链表施加插入和删除操作。如果更新前的记录比更新后的记录占用的空间小，一边修改原有记录内容，一边插入新结点到链表中；如果更新前的记录比更新后的记录占用的空间大，一边修改原有记录内容，一边删除结点，释放内存空间。

### 3.1 数据更新算法

当插入一个数据时，首先判断该集合中的元素在等价类中是否存在，如果存在，则将其插入到基本表中；如果不存在，则更新等价类，并将其元组插入到基本表中。当删除一个数据时，先查找要删除的数据是否存在，如果存在，再判断是删除元组，还是属性，如果删除元组，则直接删除该元组；如果删除属性，在删除属性值时，删除该属性的等价类。下面是算法具体描述。

#### 3.1.1　算法描述

算法：对 RRDB 数据更新(以表 1 为例)

输入 COLOR 和 P-SIZE 属性邻接表；表 1 的十字链表；用户输入数据

输出 更新后表 1 的结果

方法：

**Step1** If 插入，then 新建邻接表中表结点

If 用户输入的数据在等价类中不存在，then

If 用户输入的数据在邻接表中有相同的类，then

直接插入到邻接表同一等价类元素之后，并给 Info 值加 1；接着做 Step2；

Else 新建头结点,并给 Info 域赋值为 1，插入表结点；接着做 Step2；

Else 新建邻接表，接着做 step2；

Else if 查询满足用户输入条件的数据 then

If 元组删除 then step3；

Else 属性删除 then {等价类删除；Step3；}；

**Step2** 新建十字链表的结点，并给相应域赋值，将数据插入到原有基本表中；

**Step3** 删除基本表中的数据，释放空间；

**Step4** 显示更新后表 1 的结果。

#### 3.1.2　算法的时间性能分析

从上面的算法看，全部是判断操作，所以时间复杂度是 $O(1)$，但是要考虑插入和删除时的查询时间。采用传统的查询方法，不同查询方法，时间复杂度不一样。

### 3.2 RRDB 数据更新示例

#### 3.2.1　等价类数据更新

在做预测时，并不能一次预测所有等价类，而是在实践的过程中完善与补充，这样需要更新等价类的存储内容，从而涉及等价类的插入和删除。在本文中，等价类是采用图中的邻接表存储结构存储，下面用土壤的例子描述等价类的插入和删除过程。

土壤分析家通常从土壤的颜色、土壤划分的大小及土壤的质量等来对土壤进行测量、分析。假设土壤分析家在测量土壤时，将土壤颜色划分等价关系为：COLOR= {[Black, Ebony], [Brown, Sienna], [White], [gray]}，用邻接表的形式表示(见图 1)，但是在后来的测量中又发现一些等价类，如{Tan, Orange}的插入；{White}的删除，详细操作过程如下：

(1)等价类数据插入

1)Tan 的插入：①新建表结点，数据域为 Tan，指针域为 null；②对 Tan 进行归类，经测量发现 Tan 与 Sienna 是同一等价类，将 Tan 插入到第二类 Sienna 结点之后，并将头结点的 Info 域的值增 1。

2)Orange 的插入：①新建表结点，数据域为 Orange，指针域为 null；②对其进行归类，发现没有同一等价类，新增头结点并给相应的域赋值，插入 Orange 结点。如图 3 所示，为了方便，在邻接表图中增加一个序号。

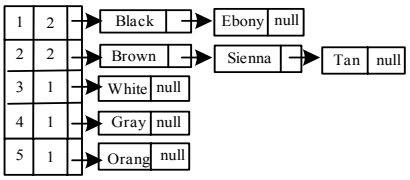


**图 3　Color 等价类插入结果**

(2)等价类数据删除

等价类的删除同样有头结点和表结点的删除，与数据结构中链表的删除一样，这里就不详述了。

#### 3.2.2　基本表数据更新

基本表数据的更新与关系数据库中数据更新是一样的，分属性和元组的更新。在本文中基本表采用的是十字链表的方式存储，将表 1[4]用十字链表的方式存储(见图 2)，然后对其进行插入和删除操作。

(1)基本表数据插入

给表 1 插入一个属性(Quality)及属性对应元组的取值{[Normal],[Normal,Fertile],[Poor],[Good,Productive],[Productive,Normal]}，操作后结果见图 4。

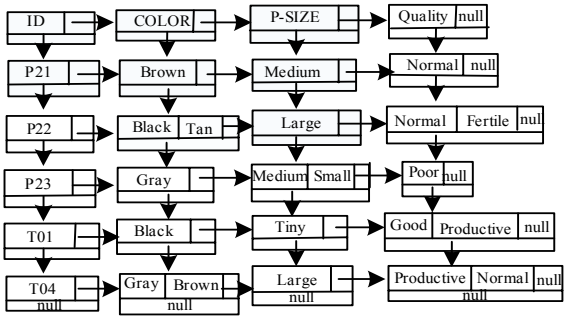


**图 4　插入属性(Quality)结果**

(2)基本表数据删除

假如要从表 1 中删除 P23 元组，操作后结果见图 5。

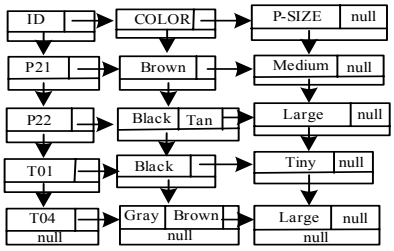


**图 5　删除 P23 元组结果**

插入和删除基本表，注意要相应地更新等价类，如上面基本表插入时，要新增一个 Quality 等价类的邻接表，删除属性时，也要删除相应的等价类邻接表。

## 4　结束语

在本文中，根据 RRDM 的特性，主要讨论了 RRDB 中数据的存储结构及数据更新，它借助数据结构中图的存储结构——邻接表和十字链表。这样，一方面解决 RRDB 中数据存储问题；另一方面给数据更新带来方便。从算法的时间分析性能看，在没有考虑算法的查找时间的前提下，时间复杂度为 $O(1)$。但是如果查找时间长、效率低，就给该算法带来极大的影响，因此，如何减少查找时间，从而提高算法的有效性是进一步研究的重点。

## 参考文献


[1] Frederick E, Buckles B P. Extension of the Relational Database and Its Algebra with Rough Set Techniques[J]. Computational Intelligence, 1995, 11(2): 233-245.

[2] 王　丹, 吴孟达, 刘银山. 粗糙关系数据库空间结构及其粗糙集模型[J]. 计算机工程与应用, 2005, 41(34): 163-167.

[3] 安秋生. 基于粗糙关系数据库的粗糙数据查询[J]. 西安交通大学学报, 2002, 36(8): 859-862.

[4] Beaubouef T. Information-theoretic Measures of Uncertainty for Rough Sets and Rough Relational Databases[J]. Information Sciences, 1998, 109(1-4): 185-195.

[5] 郭景峰, 李　莉, 宫继宾. 粗关系数据库中的粗函数依赖研究[J]. 计算机科学, 2004, 31(9): 90-95.

[6] 严蔚敏, 吴伟民. 数据结构(C 语言版)[M]. 北京: 清华大学出版社, 2002.

[7] HectorGarcia-Molina, Ullman J D, Widom J. 数据库系统实现[M]. 北京: 机械工业出版社, 2000.


(上接第 112 页)

**表 3　3 种算法的离散化效果**

| 判别标准 | 离散算法 | 测试数据集 | | | | 平均值 |
|---|---|---|---|---|---|---|
| | | Iris | Sat | Pid | Hea | |
| CAIR 平均值 | CADD | 0.393 | 0.191 | 0.027 | 0.032 | 0.161 |
| | CAIM | 0.473 | 0.209 | 0.024 | 0.046 | 0.188 |
| | CVM | 0.482 | 0.212 | 0.034 | 0.048 | 0.194 |
| 区间总数 | CADD | 16 | 253 | 112 | 61 | 110.5 |
| | CAIM | 12 | 216 | 16 | 12 | 64.0 |
| | CVM | 12 | 216 | 19 | 13 | 65.0 |
| 时间/s | CADD | 0.13 | 80.73 | 8.34 | 1.43 | 22.66 |
| | CAIM | 0.07 | 51.42 | 1.13 | 0.15 | 13.19 |
| | CVM | 0.07 | 56.31 | 1.26 | 0.15 | 14.45 |

**表 4　C4.5 分类的精度和规则数**

| 判别标准 | 离散算法 | 测试数据集 | | | | 平均值 |
|---|---|---|---|---|---|---|
| | | Iris | Sat | Pid | Hea | |
| 预测错误率和偏差 | CADD | 7.82±0.3 | 13.82±10.1 | 28.40±3.1 | 27.67±2.8 | 19.43±4.1 |
| | CAIM | 6.27±0.0 | 13.97±5.4 | 24.57±5.1 | 23.00±2.1 | 17.20±3.2 |
| | CVM | 6.13±0.1 | 14.09±3.5 | 22.90±2.9 | 22.85±2.9 | 16.49±2.4 |
| 规则数 | CADD | 4.21 | 764.6 | 202.60 | 68.34 | 259.89 |
| | CAIM | 3.51 | 752.7 | 7.57 | 21.51 | 196.32 |
| | CVM | 3.52 | 738.3 | 12.43 | 22.16 | 194.10 |

## 6　结束语

本文将统计学中的 Cramer's V 应用于连续属性的离散化中提出了基于 Cramer's V 的连续属性离散算法 CVM，为属性的离散化提供了一种新思路。实验结果表明，基于 Cramer's V 的连续属性离散算法 CVM 在保证类-属性相关度的前提下，减少了离散后的区间数，提高了分类器的预测精度。

## 参考文献


[1] Fayyad U M, Irani K B. Multi-Interval Discretization of Continuous-valued Attributes for Classification Learning[C]//Proc. of the 13th International Joint Conference on Artificial Intelligence. [S. l.]: Morgan Kaufmann, 1993: 1022-1027.

[2] Liu Xiaoyan, Wang Huaiqing. A Discretization Algorithm Based on a Heterogeneity Criterion[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 9(17): 1166-1173.

[3] Liu Huan, Setiono R. Chi2: Feature Selection and Discretization of Numeric Attributes[C]//Proc. of the 7th International Conference on Tools with Artificial Intelligence. Washington D. C.: [s. n.], 1995: 388.

[4] Chac-Ton S, Jyh-Hwa H. An Extended Chi2 Algorithm for Discretization of Real Value Attributes[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(3): 437-441.

[5] 陈秉正, 韩春鹏. 归纳式学习中连续型数据的区间划分问题[J]. 系统工程理论与实践, 2001, 21(4): 2-8.

[6] Ching J Y, Wong A K C, Chan K C C. Class-Dependent Discretization for Inductive Learning from Continuous and Mixed Mode Data[J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 1995, 17(7): 641-651.

[7] Kurgan L A, Cios K J. Discretization Algorithm that Uses Class-Attribute Interdependence Maximization[C]//Proc. of Int'l Conf. on Artificial Intelligence. Las Vegas: [s. n.], 2001: 980-987.

[8] 梅长林, 周家良. 实用统计方法[M]. 北京: 科学出版社, 2003.