第 28 卷 第 11 期
2005 年 11 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 28 No. 11
Nov. 2005

粗集理论集成 ORDBMS 的原型系统

孙惠琴[1),2)] 熊 璋[2)]
[1)](中国铁道科学研究院 北京 100081)
[2)](北京航空航天大学计算机学院 北京 100083)

摘 要 对等价矩阵代数进行了扩充,使它能够实现决策表的上下近似集、规则提取、相对约简、相对核等操作. 给出了一个原型系统 RSORDMS. 在原型系统中,采用了数据库 SQL 语言与粗集结合的技巧,使得数据清洗、求核等操作非常迅速. 整个原型系统运行效率很高.

关键词 粗集;ORDBMS;核;数据挖掘;等价矩阵代数
中图法分类号 TP18

The Prototype of Integrating ORDBMS and Rough Set Theory

SUN Hui-Qin[1),2)] XIONG Zhang[2)]
[1)](*China Academy of Railway Sciences*, *Beijing* 100081)
[2)](*School of Computer Science*, *Beihang University*, *Beijing* 100083)

Abstract The equivalence matrices algebra is extend to describe the low approximation, the upper approximation, rule extraction, relative core and relative reduction. A prototype system has been designed and implemented in authors' research, called RSORDMS, adding data mining capabilities to ORDBMS while preserving its traditional power. In the prototype system, a technique is adopted which combines rough set with SQL to make data cleaning and core computation fast. The whole prototype system has a good performance.

Keywords rough set; ORDBMS; core; data mining; equivalence matrices algebra

1 引 言

随着数据挖掘的进一步发展,一方面仍然应该继续研究如何改进各种方法的性能,另一方面则应该注重于研究数据挖掘的体系结构,好的体系结构是数据挖掘实用的基础. 目前的数据挖掘工作仍然集中于性能上的改进,但从长远来看,其体系结构很重要. 如果能将它集成在现有的成熟的数据库中,实现与数据库的紧耦合[1],将具有很多优点:(1)数据库发展了很多年,比较成熟. 通过将数据挖掘集成在现有的成熟的数据库中,数据挖掘可以平滑地集成到各种应用系统上,而不像当前那样每一个数据挖掘应用系统都得从头开发设计;(2)数据挖掘的研究工作更有继承性,通过数据库通用的查询语言来定义和查询数据挖掘的知识,好的查询优化(涉及数据挖掘性能的改进)可以形成标准的库函数而嵌入数据库系统中.

1997 年,ORACLE 公司推出了 ORACLE 第 8 版本,支持对象-关系特性,从此,新的以对象-关系模型为基础的 ORDBMS(Object Relation Database Management System)已经成为主流. ORDBMS 中支持抽象数据类型 ADT(Abstract Data Type)和用户定义函数 UDF(User Defined Function),达到扩

收稿日期:2003-09-09;修改稿收到日期:2005-06-14. 孙惠琴,1973 年生,主要研究方向为粗集、数据挖掘、粗集理论和数据挖掘方法在工程诊断方面的应用、分布式信息处理等. E-mail: sunhq11@sina.com. 熊 璋,教授,博士生导师,主要研究领域为多媒体技术、数据挖掘、分布式信息处理等. E-mail: xiongz@buaa.edu.cn.

充系统功能[2].

粗集具有较强的分类能力和规则提取能力,应用的领域也越来越广泛.如果能将粗集的挖掘方法融入数据库体系中,在数据库的关系代数基础之上,扩展适合粗集理论的代数,必然会将数据库的优点同粗集的优点结合,更快地推动两者的共同发展.

第 2 节介绍粗集和 ORDBMS 集成的难点和策略;第 3 节详细介绍基于粗集等价矩阵的扩充代数和提出的算法,并对算法的性能进行了分析和实验;第 4 节介绍本文所实现的原型系统 RSORDMS(Rough Set Object Relation Data Mining System)并对系统的能力和性能进行了分析;第 5 节进行总结.

2 粗集和 ORDBMS 集成的难点和策略

实现粗集和 ORDBMS 的紧耦合,有几个难点.

(1)传统的粗集代数是基于集合的操作,比如求一个集合是否是另一个集合的子集,ORDBMS 不提供这样的操作.本文采用了将传统的粗集代数转化为基于等价矩阵代数.但是,现有的等价矩阵代数并不完整,本文对其进行扩充.引入决策数组的概念,并证明其上的"$\leq$"和"$\cap$"操作与集合的"$\subseteq$"和"$\cap$"操作的代数同构性.针对原有等价矩阵代数仅仅定义了知识的绝对约简和绝对核,本文定义了知识的相对约简和相对核,在实际中更常用,也更容易获得规则.具体的扩充代数在第 3 节中详细介绍.

(2)如何实现这些代数.ORACLE8.0 以上的数据库提供了 ADT 和 UDF 的支持,可采用 ORACLE 的 PL/SQL 实现这些代数和算法.

(3)如何提高系统的性能.采用组合 SQL 和粗集方法对数据进行清洗和核计算,效率很高.核计算的方法、实验和性能分析在第 3 节介绍,本节介绍一种快速易行的方法进行清洗工作,在数据挖掘前清洗不一致和重复数据:

①将决策表中的重复对象去掉.决策表的每行元素为一个对象,如果对象的属性完全相同,那么对象必然为重复对象,可删除重复对象.在 ORACLE 数据库中,假设表 a 由条件属性 b 和 c 和决策属性 d 组成,实现的 SQL 语句如下:

```
delete from a where a.rowid !=(select max(rowid)
from a b where a.b=b.b and a.c=b.c and a.d=
b.d).
```

ORACLE 的 rowid 用来唯一标识表中的一条记录,是这条记录在数据库中存放的物理地址.ORACLE 中可以利用 rowid 查找和删除表中的重复记录.

②将决策表中的不一致对象减去.检查决策表中不一致元素存在的算法:两个对象 u_i, u_j 不一致意味着条件属性相同,决策属性不相同.对于①的例子,实现的 SQL 语句如下:

```
delete from a where a.rowid !=(select max(rowid)
from a b where a.b=b.b and a.c=b.c).
```

清洗工作可以由 SQL 语言快速地实现,比文献[9]所提供的数据清洗速度快得多.可看出数据库的 SQL 语言的强大功能,从而也说明了数据挖掘同 ORDBMS 结合的巨大潜力和很好的前景.

3 基于粗集理论的代数

本文对粗集代数进行扩充,使之可以很好地集成在 ORDBMS 中.

3.1 求粗集的上近似集、下近似集、正域的算法

在粗集中,知识系统可用一个 4 元组来描述:$S=(U,A,V,f)$,其中 U 表示数据集中的所有对象;A 表示数据集中的全部属性,$A=C\cup D$,C 为条件属性集合,D 为决策属性集合;V 是属性值组成的集合;f 是属性 a 和对象 e 的函数,$f(e,a)$ 的数值为对象 e 关于属性 a 的取值.

定义 1[3]. 假设信息系统 $S=\langle U,A,V,f\rangle$ 是一个知识系统,$X_i, X_j\in U$,二进制等价矩阵 $\boldsymbol{Mc}=[a_{ij}]$ 定义如下

$$a_{ij}=\begin{cases}1, & x_iE_cx_j\\ 0, & \text{否则}\end{cases},\quad i,j=1,2,\cdots,n;$$

其中 E_c 是等价关系,定义为

$$xE_cy\Leftrightarrow(\forall c\in C), f(x,c)=f(y,c).$$

粗集中最重要的概念是等价关系,文献[3]中说明了粗集中的等价关系同上述等价矩阵是同构的.

定义 2[3]. 假设 $\boldsymbol{M}_1=[r_{ij}]$,$\boldsymbol{M}_2=[r'_{ij}]$ 是两个 $n\times n$ 的等价矩阵.$\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 的相交操作定义如下:

$$\boldsymbol{M}_1\cap\boldsymbol{M}_2=[s_{ij}], s_{ij}=r_{ij}\cap r'_{ij}=\min(r_{ij}, r'_{ij}).$$

定理 1[9]. 假设信息系统 $S=\langle U,A,V,f\rangle$ 是一个知识系统,对于 $X,Y\subseteq A$,假设 $\boldsymbol{M}_X$,$\boldsymbol{M}_Y$ 是两个 $n\times n$ 的等价矩阵,有如下关系成立:$\boldsymbol{M}_X\cap\boldsymbol{M}_Y=\boldsymbol{M}_{X\cup Y}$.

仅仅使用等价矩阵,无法求粗集的上下近似集,因此本文设计了一个数组,即决策数组.决策数组的定义以及数组的"$\leq$"和"$\cap$"操作定义如下.

定义 3.

信息系统 $S=\langle U,A,V,f\rangle$,集合 X 定义为一个

数组 $x[length]$，$length$ 为信息系统 U 中的元素的个数. 如果 U 的第 i 个元素在集合 X 中，那么 $x[i]=1$；否则 $x[i]=0$. 两个数组 Sx 和 Sy，如果 Sx 的每个元素都"$\leq$"Sy 中对应的元素，那么 $Sx\leq Sy$. 显然有如下结论：X，Y 是两个集合，对应的两个数组 Sx 和 Sy，如果 $X\subseteq Y$，那么 $Sx\leq Sy$；如果 $Sx\leq Sy$，那么 $X\subseteq Y$.

求"$\cap$"的运算为 $Sx\cap Sy=Sz$，$Sx[i]=1$ 并且 $Sy[i]=1$，那么 $Sz[i]=1$. 其它情况下，$Sz[i]=0$.

定理 2.　数组上"$\leq$"的运算和集合上的"$\subseteq$"运算代数同构；数组"$\cap$"和集合"$\cap$"运算代数同构.

证明.　假设集合 X 和 Y 对应决策数组 Sx 和 Sy，$\forall Sx,Sy,Sx\leq Sy$ 当且仅当 $X\subseteq Y$；反过来也成立. 并且，$\forall Sx$ 仅仅对应集合 X. 因此，数组上"$\leq$"的运算和集合上的"$\subseteq$"运算代数同构.

一方面，$\forall Sx,Sy,Sx\cap Sy=Sz,Sz[i]=1$ 当且仅当 $Sx[i]=1$ 和 $Sy[i]=1$；另一方面，$X\cap Y=Z$，$ui\in Z$ 当且仅当 $ui\in X$ 和 $ui\in Y$，等价于 $Sz[i]=1$ 当且仅当 $Sx[i]=1$ 和 $Sy[i]=1$.　　证毕.

定义等价矩阵和决策数组后，求上下近似的操作非常简单. 设等价关系形成的等价矩阵为 $\boldsymbol{Mc}$，集合 X 对应的数组为 Sx；下近似数组为 Su，初始的元素为 0；上近似数组为 Sv，初始的元素为 0. 算法如下.

算法 1.　求粗集的上下近似集.

1. 形成等价矩阵 $\boldsymbol{Mc}$ 以及集合 X 的数组 Sx；

2. for $i=1$ to n

3. 计算 $\boldsymbol{Mc}$ 第 i 行是否是 Sx 的子集合，即 $\boldsymbol{Mc}$ 第 i 行(数组)和 Sx 之间是否存在"$\leq$"关系. 如果是，如果 $\boldsymbol{Mc}$ 第 i 行第 j 列为 1，将 Su 的第 j 个元素设置为 1；另外，计算它们的交集操作，如果结果中第 k 列为 1，将 Sv 的第 k 个元素设置为 1；

4. for $j=1$ to n 如果 a_{ij} ($\boldsymbol{Mc}$ 的第 i 行第 j 列元素)不为 1，next j；否则 $\boldsymbol{Mc}$ 第 j 行设置为 0，next j

5. next i

求得的 Su 为 X 下近似数组；Sv 为 X 上近似数组. Su 和 Sv 对应的元素是 X 的下近似集和上近似集.

求出上下近似集后，可进行规则的提取. 例如，对于决策属性 $D=1$，其下近似数组对应的元素集如图 1 所示.

a	b
0	1
1	0
1	1

图　1

因此，由下近似集所得到的规则为

if $a=0\wedge b=1$ then $D=1$，

if $a=1\wedge b=0$ then $D=1$，

if $a=1\wedge b=1$ then $D=1$.

单个集合 X 的下近似，即为集合的正域. 为了求集合的负域和边界域，定义 4 给出数组的差运算.

定义 4.　如果两个数组 Sx 和 Sy，$Sx\leq Sy$(见定义 3)，可以定义 $Sy-Sx$ 运算如下：

$Sy[i]-Sx[i]$和一般数学上的差运算一样，如果 $Sy[i]=1$，$Sx[i]=1$，那么结果为 0；如果 $Sy[i]=0$，$Sx[i]=0$，那么结果为 0；如果 $Sy[i]=1$，$Sx[i]=0$，那么结果为 1.

算法 2.　求某集合的负域和边界域.

在定义 5 的基础上，我们很容易求出某集合的负域和边界域. 假设信息系统的下近似集为 Su，上近似集为 Sv，信息系统的论域为 Sy(Sy 的每个元素都为 1)，那么此集合的负域为 $Sy-Su$；边界域为 $Sv-Su$.

在定义 3 所定义的"$\leq$"和"$\cap$"操作和算法 1 的基础上，很容易求出信息系统 $S=\langle U,A,V,f\rangle$ 的正域. 具体算法如下.

算法 3.　求信息系统的正域.

1. 形成等价矩阵以及决策信息的多个数组 S_{d1}，S_{d2}，…，S_{dk}；

2. $Sx:=S_{d1}$；

3. 重复算法 1 的步骤；

4. 将下一个决策信息数组赋值给 Sx，重复步 3. 直到步 1 中数组遍历完，求出的 Su 为信息系统正域.

算法 3 和算法 1 的原理基本相同，只是决策集合为多个集合，对应的数组也为多个数组；因此，算法中的比较需要和多个数组进行比较，而不仅仅是一个数组.

3.2　粗集的绝对约简、相对约简、核的定义和算法

定义 5[3]. 假设 X 是属性 A 的非空子集：$\varnothing\subset X\subseteq A$. 给定属性 $x\in X$，如果 $\boldsymbol{M}_X<\boldsymbol{M}_{X-\{x\}}$，我们称 x 对于 X 是重要属性；如果 $\boldsymbol{M}_X=\boldsymbol{M}_{X-\{x\}}$，我们称 x 对于 X 是不重要属性. 所有重要属性集合为 X 的核.

定义 6[3]. 假设 X 是属性 A 的非空子集：$\varnothing\subset X\subseteq A$. 给定属性 $x\in X$，我们定义 x 对于 X 的重要性为 $\boldsymbol{sig}_{X-\{x\}}=\boldsymbol{M}_{X-\{x\}}-\boldsymbol{M}_X$.

算法 4[3].　求核的算法.

1. 初始化设置 $i=1$；$C_X=\varnothing$；$\boldsymbol{M}_0$ 中所有元素都为 0.

2. 如果 $\boldsymbol{sig}_{X-\{xi\}}(xi)>\boldsymbol{M}_0$(即 $\boldsymbol{M}_X<\boldsymbol{M}_{X-\{xi\}}$)，那么从步 3 执行；如果 $\boldsymbol{sig}_{X-\{xi\}}(xi)=\boldsymbol{M}_0$(即 $\boldsymbol{M}_X=\boldsymbol{M}_{X-\{xi\}}$)，那么从步 4 执行.

3. $xi\in C_X$，将 xi 加入到 C_X 的集合中. 然后执行步 4.

4. 判断 i 是否等于 $|X|$. 如果成立，那么算法结束，C_X 是

X 的核. 如果 $i<|X|$,执行步 5.

5. 将 i 加 1,执行步 2.

如果 $X=C$,那么 C_X 是条件属性 C 的核.

但是这种算法在某些情况下,将非必要属性也加入到核中,求出的核出现异常情况,见表 1 所示的例子.

表 1 关于气象例子的决策表系统

U	条件属性				决策属性(d)
	Outlook($a1$)	Temperature($a2$)	Humidity($a3$)	Windy($a4$)	
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Sunny	Mild	Normal	True	P
12	Overcast	Mild	High	True	P
13	Overcast	Hot	Normal	False	P
14	Rain	Mild	High	True	N

上述决策表中,属性集为 $C=\{a1,a2,a3,a4\}$. 由文献[4]可以知道,$a2$ 不属于核,其核为$\{a1,a4\}$. 但是如果按照定义 1,矩阵 $\boldsymbol{M}_C$ 中只有对角线上的元素为 1,其它的元素都为 0;矩阵 $\boldsymbol{M}_{C-a2}$ 中,由于对象 1 和对象 8 除了属性 $a2$ 以外,其它属性的数值都相同,因此 $\boldsymbol{M}_{C-a2}$ 中除了对角线上的元素为 1 以外,第 1 行第 8 列、第 8 行第 1 列为 1. 因此必然有 $\boldsymbol{M}_X<\boldsymbol{M}_{X-a2}$,由定义 6 知 $a2$ 为核. 这与文献[4]中 $a2$ 不属于核相矛盾.

为什么会出现这种情况?经过认真的研究和分析后发现:文献[3]所提出的核是绝对核,约简是绝对约简,而对于相对核和相对约简没有定义和阐述. 在实际中,相对核和约简比绝对核和约简更常用. 本文给出了等价矩阵基础上的相对约简和相对核的定义和算法.

定义 7[3]. 信息系统 $S=\langle U,A,V,f\rangle$,X 是 A 的非空子集合,$X\subseteq A$,$X_0\subseteq X$,当下面条件满足时:

(1)$\boldsymbol{M}_X=\boldsymbol{M}_{X_0}$.

(2)X_0 是独立的.

X_0 是 X 的一个关键字. 如果 $X=C$,X_0 是条件属性 C 的一个绝对约简.

下面给出基于等价矩阵的相对约简和相对核的定义.

定义 8. 信息系统 $S=\langle U,A,V,f\rangle$,条件属性为 C,对应的矩阵为 $\boldsymbol{M}_C$;决策属性为 D,对应的多个数组 $S_{d1},S_{d2},\cdots,S_{dk}$,按照算法 3 求出 D 的 C 的正域记为 $POS_C(D)$.

定义 9. 信息系统 $S=\langle U,A,V,f\rangle$,条件属性为 C,对应的矩阵为 $\boldsymbol{M}_C$;决策属性为 D,对应的多个数组 $S_{d1},S_{d2},\cdots,S_{dk}$. 对于任一 $x\in C$,可按照算法 3 求出 D 的 C 的正域为 $POS_C(D)$和 D 的 $C-\{x\}$的正域为 $POS_{(C-\{x\})}(D)$. 如 $POS_C(D)=POS_{(C-\{x\})}(D)$,则称 x 为 C 中相对于 D 可省略(不必要的);否则,称 x 为 C 中相对于 D 不可省略.

定义 10. 信息系统 $S=\langle U,A,V,f\rangle$,条件属性为 C,对应的矩阵为 $\boldsymbol{M}_C$;决策属性为 D,对应的多个数组 $S_{d1},S_{d2},\cdots,S_{dk}$. C 相对于 D 的所有不可省略的属性的集合称为 C 关于 D 的相对核,记为 C_X.

定义 11. 信息系统 $S=\langle U,A,V,f\rangle$,条件属性为 C,对应的矩阵为 $\boldsymbol{M}_C$;决策属性为 D,对应的多个数组 $S_{d1},S_{d2},\cdots,S_{dk}$. $C_0\subseteq C$,如果(1)$POS_C(D)=POS_{C0}(D)$;(2)C_0 独立,即任一 $P\subset C_0$,$POS_{C0}(D)<>POS_P(D)$都满足,那么 C_0 是 D 的相对约简.

核是粗集非常重要的概念,是所有约简的共同属性,也是计算约简的基础. 本文提供了几种求核算法,对其性能进行分析,并进行了实验验证.

算法 5. 等价矩阵中求相对核方法 1.

从相对核的定义(定义 10)出发,首先求出条件属性 C 相对于决策属性 D 的正域 $POS_C(D)$,然后对于 $x\in C$,求出相对于决策属性 D 的正域 $POS_{C-\{x\}}(D)$,如果两者相同,说明 x 是不必要的,可以去掉;否则,保留 x. 所有的保留的 x 的集合是决策表相对核.

算法 6. 等价矩阵中求相对核方法 2.

假设 C 为条件属性,D 为决策属性. 求出决策属性的矩阵 $\boldsymbol{M}_D$,任一属性 $x\in C$,求出所有的 $\boldsymbol{M}_x$,$\boldsymbol{M}_{C-\{x\}}$. 假设 $\boldsymbol{M}_D$ 中第 i 行第 j 列为 0,如果 $\boldsymbol{M}_x$ 第 i 行第 j 列也为 0,$\boldsymbol{M}_{C-\{x\}}$ 第 i 行第 j 列为 1,那么 x 属于条件属性 C 相对于决策属性 D 的

核. 正确性证明见附录.

上述两种求核方法是从定义等出发求出的，效率相对比较低. ORACLE 中 SQL 语言的执行速度即使在大型数据量的情况下也很快. 下面就提供一种简单易行的方法，可以直接利用 SQL 语言实现求粗集的相对核.

算法 7.　一致的信息系统中利用 SQL 求相对核方法.

本文的 SQL 语言的设计是基于文献[5]的一些思想，并对这些思想进行了翻译，使其适合 SQL 语言的表述如下：在一致完备的知识系统，如果存在两个对象 u_i 和 u_j 的决策属性不一样，条件属性除了某一属性外，其余都相同，那么此属性是决策表的核；如果对于某一条件属性，这样的对象不存在，那么此属性不是决策表的核. 由此可以设计出简单易行的 PL/SQL 程序. 对于表 1 的例子，判断属性 $a1$，$a2$，$a3$，$a4$ 是否必要的 PL/SQL 程序如下：

```
DECLARE
   TYPE t_cursor IS REF CURSOR;
   v_a integer; v_cursor t_cursor;
BEGIN
--deciding if a1 is significant
OPEN v_cursor FOR
select count(*) from a,a b
where a.a2=b.a2 and a.a3=b.a3 and a.a4=b.a4
   and a.a1<>b.a1 and a.d<>b.d;
FETCH v_cursor into v_a;
IF v_a=0 THEN --If the number of chosen record is zero
  DBMS_OUTPUT.PUT_LINE('a1 is nonsignificant');
ELSE DBMS_OUTPUT.PUT_LINE('a1 is significant'); END IF;
CLOSE v_cursor;
--deciding if a2 is significant
OPEN v_cursor FOR
select count(*) from a,a b
where a.a1=b.a1 and a.a3=b.a3 and a.a4=b.a4
and a.a2<>b.a2 and a.d<>b.d;
FETCH v_cursor into v_a;
IF v_a=0 THEN--If the number of chosen record is zero
  DBMS_OUTPUT.PUT_LINE('a2 is nonsignificant');
ELSE DBMS_OUTPUT_LINE('a2 is sigificant');
END IF;
CLOSE v_cursor;
END;
```

判断 $a3$，$a4$ 属性是否必要的 SQL 语言和判断 $a1$，$a2$ 属性的语句相似，不再重复.

由于判断 $a1$，$a4$ 属性都选择出了记录，而判断 $a2$，$a3$ 语句没有选择出来任何记录，因此 $a1$，$a4$ 是此决策表的核；而 $a2$，$a3$ 不是此决策表的核.

我们选用 UCI 机器学习数据库中的决策表进行实验. 实验结果如表 2 所示.

表 2　实验结果

数据库	m	N	算法 6 所用时间(s)	算法 7 所用时间(s)
表 1 所示数据库	14	4	0.087	0.001
	70	7	0.120	0.006
vote	435	16	2.300	0.045
Chess	3196	36	116.800	0.580
Mushroom	8124	22	230.600	1.230

注：m 表示对象数目；N 表示条件属性数目.

算法 5 的执行效率和算法 6 相似，不再重复. 由实验结果可知，算法 7 比算法 5 和 6 快很多. 说明了算法 7 的实用性、高效性以及在大量的数据下的可行性.

在数据存在噪音(不一致数据)的情况下，本文也设计了一套 SQL 语言进行求核操作. 主要的思想参见文献[10,11].

算法 8.　不一致的信息系统中利用 SQL 求相对核方法.

首先创建一个临时表(创建临时表的规则参见文献[11]的定义 7)，然后对临时表按照算法 7 的思想创建 SQL 语句，判断属性是否是必要的.

4　系统的能力

本文在上述代数的基础上，在 ORDBMS 中进行了实现. 建立了一个原型系统 RSORDMS(Rough Set Object Relation Data Mining System)，可以使用粗集理论进行数据挖掘. 开发语言主要采用了 ORACLE 的 PL/SQL，数据类型和函数使用 ORACLE8.0 以上版本引入的 ADT 和 UDF. 原型系统 RSORDMS 具有如下特性：

(1)数据的离散化. 对象的属性可能是定量或定性的. 当处理定量属性时，必须进行离散化. 原型系统采用粗集理论中常用的几种离散化算法. 比如等距离离散化算法、等频率离散化算法、自然算法、半自然算法、布尔推理算法等.

(2)数据的清洗. 可以将重复数据和不一致数据从决策表中清除.

(3)数据的编码能力. 本原型系统对数据具有编码和解码能力，节省了存储空间，提高了效率.

(4)规则提取. 粗集求出上下近似之后，可以进

行规则提取等.

(5)求核操作. 可以求出决策表的核. 选用 UCI 机器学习数据库中的决策表,分别采用可辨识矩阵的求核算法[5]、本原型系统的求核算法,实验结果如表 3 所示.

表 3 求核实验结果

数据库	m	N	可辨识矩阵求核时间(s)	本系统求核时间(s)
表 1 所示数据库	14	4	0.019	0.004
	70	7	0.055	0.042
vote	435	16	0.812	0.720
Chess	3196	36	31.150	20.880
Mushroom	8124	22	61.430	27.060

注:m 表示对象数目;N 表示条件属性数目.

由实验结果可以看出:本原型系统具有较高的求核效率;数据量越大,本系统的优越性更明显,从而可以更好地适应数据挖掘中大量数据的要求. 此实验未采用并行化,系统的优越性体现得并非太明显,下面表 4 中的实验采用了并行化,更好地体现了本系统的优越性.

(6)属性约简. 约简算法发现数据的依赖和冗余,在进行数据挖掘之前将这些属性去掉. 在原型系统的实现中发现:从核出发求信息表的最小约简是非常有效的方法.

(7)支持并行化. 由于 ORACLE 支持并行化处理,对论域中多个个体并行处理,加快处理速度.

表 3 的实验未采用并行化,如果采用并行化操作,求核操作的实验结果如表 4 所示.

表 4 采用并行化操作的求核实验结果

数据库	m	N	可辨识矩阵求核时间(s)	本系统求核时间(s)
表 1 所示数据库	14	4	0.019	0.001
	70	7	0.055	0.006
vote	435	16	0.812	0.045
Chess	3196	36	31.150	0.580
Mushroom	8124	22	61.430	1.230

注:m 表示对象数目;N 表示条件属性数目.

由实验结果可以看出:采用了并行化之后,系统求核的速度大大提高了.

上述实验均在主频 PIII 733,内存为 512MB 的计算机上进行.

由于结合了 ORDBMS 中 SQL 语言的优点,因此,整个系统的运行效率较高,实用性很强.

5 总 结

将 Rough 集理论集成到新一代的 ORDBMS 中,实现数据挖掘与数据库的紧耦合,具有很好的应用前景[1]. 本文在这方面进行了研究和实现,并对其关键技术进行了分析和实验.

本文对粗集的数据挖掘方法集成到新一代 ORDBMS 中的代数进行了扩充,证明了一个代数同构定理,解决了相关的技术难点,设计和实现了一个原型系统 RSORDMS(Rough Set Object Relation Data Mining System). 本系统具有传统的 ORDBMS 的功能,又增加了求相对核、相对约简、规则提取等数据挖掘功能. 在原型系统中,也使用 SQL 语言来实现粗集的求核、数据清洗等功能,充分发挥了 ORDBMS 的优点,实验结果表明,原型系统运行良好,性能比较高. 但是还有几个难题尚未解决:(1)数据库中的数据非常庞大,如何在等价矩阵和 SQL 语言的基础上,设计更高效的约简算法?(2)粗集查询语言的设计. 好的查询语言可使 RSORDMS 更实用. 我们正对这些问题进一步研究.

参 考 文 献

1 Han Jia-Wei. Data Mining. Beijing: Higher Education Press, 2001(in Chinese)
(韩家炜等. 数据挖掘. 北京:高教出版社, 2001)

2 ONeil P.. Database: Principles, Programming and Performance. Beijing: China Machine Press, 2002(in Chinese)
(奥尼尔. 数据库原理、编程与性能. 北京:机械工业出版社, 2002)

3 Guan J. W., Bell D. A., Guan Z.. Matrix computation for information systems. Information Sciences, 2001, 131(1～4): 129～156

4 Wang Guo-Yin. Rough Set Theory and Knowledge Acquisition. Xi'an: Press of Xi'an Jiaotong University, 2001(in Chinese)
(王国胤. Rough 集理论与知识获取. 西安:西安交通大学出版社, 2001)

5 Hu Xiao-Hua, Cercone N.. Learning in relational databases: A rough set approach. Computational Intelligence, 1995, 11(2): 338～347

6 Ziarko W.. Rough sets: Trends, challenges, and prospect. In: Ziarko W., Yao Y. Y. eds. Rough Sets and Current Trends in Computing(RSCTC 2000). Berlin: Springer Verlag, 2001, 1～7

7 Liu Shao-Hui *et al*.. Research on efficient algorithms for rough set methods. Chinese Journal of Computers, 2003, 26(5): 524～529(in Chinese)
(刘少辉等. Rough 集高效算法的研究. 计算机学报, 2003, 26(5): 524～529)

8 Urman S.. ORACLE8i Advanced PL/SQL Programming. Bei-

jing: China Machine Press, 2001(in Chinese)
(Urman S. 著. 刘　谦, 苏建平等译. ORACLE8i PL/SQL 高级程序设计. 北京: 机械工业出版社, 2001)

9 Tan Tian-Le, Song Zhi-Huan. Matrix computation for data cleaning and rule extraction in knowledge system. In: Proceedings of the 1st International Conference on Machine Learning and Cybernetics, Beijing, 2002, 116～120

10 Ye Dong-Yi, Chen Zhao-Jiong. A new discernibility matrix and the computation of a core. Acta Electronic Sinica, 2002, 30(7): 1086～1088(in Chinese)
(叶东毅, 陈昭炯. 一个新的差别矩阵及其求核方法. 电子学报, 2002)

11 Zheng Zheng, Wang Guo-Yin, Wu Yu. Objects' combination based simple computation of attribute core. In: Proceedings of the 2002 IEEE International Symposium on Intelligent Control, Canada, 2002, 513～519

附　录. 算法 6 的证明.

在文献[5]中利用差别矩阵很容易地确定核, 差别矩阵 $\boldsymbol{M}=\{m_{ij}\}$ 定义为:

$$m_{ij}=\begin{cases}\{x\in C: F(u_i,x)\neq F(u_j,x)\}, \\ \qquad F(u_i,D)\neq F(u_j,D), \quad i,j=1,2,\cdots,n; \\ \phi, \quad \text{其它情况}\end{cases}$$

其中 C 为条件属性, x 为某一条件属性, D 为决策属性, u_i 和 u_j 为决策表中的两个对象, $F(u_i,D)\neq F(u_j,D)$ 表示对象 u_i 和 u_j 的决策属性 D 不相同, $F(u_i,x)\neq F(u_j,x)$ 表示对象 u_i 和 u_j 的条件属性 x 值不相同.

文献[5]有如下结论: 在一致完备的知识系统, 当且仅当某个 m_{ij} 为单个属性时, m_{ij} 属于核 C_X. 我们只需要证明上述的算法所得的结果和以上的结论等价即可.

假设算法的条件成立, 即 $\boldsymbol{M}_D$ 中第 i 行第 j 列为 0, 如果 $\boldsymbol{M}_x$ 第 i 行第 j 列也为 0, $\boldsymbol{M}_{C-\{x\}}$ 第 i 行第 j 列为 1. 对这些条件进行分析: 如果 $\boldsymbol{M}_D$ 中第 i 行第 j 列为 0, 说明相应的对象 u_i 和 u_j 的决策属性必然不相同, 即 $F(u_i,D)\neq F(u_j,D)$; 任一属性 $x\in X$, 如果 $\boldsymbol{M}_x$ 的 i 行 j 列为 0, 并且 $\boldsymbol{M}_{X-\{x\}}$ 的 i 行 j 列为 1, 说明对象 u_i 和 u_j 中除了属性 x 所有属性数值都相同(属性 x 的数值不相同, 而属性 $C-\{x\}$ 的数值相同), 即 $F(u_i,x)\neq F(u_j,x)$. 由于 x 为单个属性, 由文献[5]可知, x 属于核. 上述算法是正确的. 在实际计算中, 由于矩阵是对称的, 所以只需搜索上对称或者下对称矩阵即可.

SUN Hui-Qin, born in 1973. Her current interests include rough set, data mining, and its applications to engineering diagnosis, and distributed information processing.

XIONG Zhang, professor, Ph. D. supervisor. His current interests include multimedia, data mining, and distributed information processing.

Background

In recent years, data mining is a hot topic. With the development of data mining, on the one hand improving the methods' performance should be still on research, on the other hand its architecture should be emphasized which is the basis of data mining applied in practice. The current data mining work still focuses on the improvement of performance, but its architecture is more important in the long term. This paper discusses the architecture of integrating rough set and ORDBMS and solves the key problems to the core computation, equivalence matrix algebra, etc. A prototype is designed and implemented.

The previous work on this area include Reducts of incomplete information system, distributed data mining based on rough set, engineering diagnosis based on rough set and data mining.