学术论文

KDD 中双库协同机制的研究（Ⅱ）

杨炳儒，王建新，孙海洪
（北京科技大学信息工程学院，北京 100083）

[摘要] 在“KDD 中双库协同机制的研究（Ⅰ）”的理论基础上，实现了对知识库的简约并实现了双库协同机制的两个协调器，从而建立了由 KDD 融入双库协同机制的新结构模型——KDD*，这是一个在结构和功能上相对于 KDD 而言的开放的、优化的扩体。基于双库协同机制与 KDD*，提出了挖掘关联规则和聚类规则的新算法，充分体现了内在机理研究对主流发展的影响，开辟了一条全新的研究方向与路径。

[关键词] 知识发现；双库协同机制；启发型协调器；中断型协调器；Maradbcm 算法；数值域划分算法
[中图分类号] TP311 [文献标识码] A [文章编号] 1009－1742（2002）05－0034－10

1 构建供挖掘的知识库

1.1 规则的简约表示

文献［1］建立了论域中按数据子类结构形式所构成发掘数据库的可达范畴，与基于属性间关系的发掘知识库的推理范畴之间的等价关系，这两个范畴的等价关系为定向发掘和定向搜索奠定了理论基础。但在文献［1］中建立的知识库规模庞大，因此要把知识库中的规则用简约的形式表示，从而建立简约知识库，以便提高挖掘效率。

1.1.1 简约规则

定义 1 对 $n=\theta_0 a_1\theta_1 a_2\cdots\theta_{m-1}a_m\theta_m$，定义 n 的合取度为 n 的表达式中出现的合取号“∧”的总数，记为 DegCon（n）；定义 n 的析取度为 n 的表达式中出现的析取号“∨”的总数，记为 DegDis（n）。n 的度定义为 n 的析取度与合取度之和，记为 Deg（n）。

定义 2 规则 $r=(n\to k)$ 中的知识结点 n 称为规则 r 的始知识结点，k 称为 r 的终知识结点，规则 r 的度定义为始知识结点的度与终知识结点的度之和，记为 Deg（r）。

由于对不同的表示形式，知识结点的度不唯一，因此对不同形式给出的同一条规则，规则的度也是不唯一的。

定义 3 若论域的一个知识结点是一个或多个两两互不相同且两两取自不同属性的属性程度词合取的形式，则称为简约始知识结点；若一个知识结点是一个或多个两两互不相同且两两取自不同属性的属性程度词析取的形式，则称为简约终知识结点。所有简约始知识结点构成的集合称为简约始知识结点集，记为 N^s；所有简约终知识结点构成的集称为简约终知识结点，记为 N^t。

定义 4 若规则 $r=(n\to k)$ 中，n 是简约始知识结点，k 为简约终知识结点，则 $(n\to k)$ 称为简约规则；由所有的简约规则构成的集合 $\{(n_i\to k_j)\mid n_i\in N^s,\ k_j\in N^t\}$ 称为简约规则集，记为 R^r。一条规则若能等价地分解为一条或多条简约规则，则称该规则有简约分解。

1.1.2 规则的简约分解定理

引理 1 规则 $r=(L\to R)$ 有且仅有两种情况：或者它的始知识结点 L 为简约始知识结点，

[收稿日期] 2001－12－13；修回日期 2002－03－18
[基金项目] 国家自然科学基金重点资助项目（69835001）；教育部科技重点资助项目（［2000］175）
[作者简介] 杨炳儒（1943－），男，天津市人，北京科技大学教授，博士生导师

或者 r 能等价地表示成两条规则：$L_1 \rightarrow R$ 和 $L_2 \rightarrow R$，并且 $\mathrm{DegCon}(L_i) \leqslant \mathrm{DegCon}(L)$，$\mathrm{DegDis}(L_i) \leqslant \mathrm{DegDis}(L) - 1$，$i = 1, 2$。（证明略）

引理 2　论域 X 的知识库中的任意规则（$L \rightarrow R$）均可以等价地分解为若干条规则（$Li \rightarrow R$），$i = 1, 2, \cdots, s$，并且 L_i 为简约始知识结点，$\mathrm{DegCon}(L_i) \leqslant \mathrm{DegCon}(L)$；类似地，（$L \rightarrow R$）也可以等价地分解为若干条规则（$L \rightarrow R_j$），$j = 1, 2, \cdots, t$，并且 R_j 为简约的终知识结点，$\mathrm{DegDis}(L_i) \leqslant \mathrm{DegDis}(L)$。（证明略）

由引理 2 很容易地就可以得出：

定理 1　论域 X 的任意规则均有简约分解，并且分解后的规则的简约始知识结点的度不大于分解前规则的始知识结点的合取度；分解后的规则的简约终知识结点的度不大于分解前的规则的终知识结点的析取度。（证明略）

推论　论域 X 的任意规则经过简约分解后，规则的简约始知识结点的属性程度词的个数不大于分解前规则的始知识结点的合取度加 1，从而它也不大于分解前规则的始知识结点中属性程度词的个数；分解后规则的简约终结点的属性程度词的个数不大于分解前规则的终知识结点的析取度加 1，从而它也不大于分解前规则的终知识结点中属性程度词的个数。

定理 2　假设 α，p，I_1 以及 B 的意义分别由文（Ⅰ）的定理 11 给出，v_1 是 I_1 的势。当 $\alpha < p(1-B)/v_1$ 时，若知识库中的一条规则 r 能等价地分解成若干条简约规则 $r_1, r_2, \cdots, r_k$，则随着对应的数据子类结构库中元组数目的增加，相应的可达关系 $F_H(r)$ 能等价地分解成可达关系 $F_H(r_1)$，$F_H(r_2), \cdots, F_H(r_k)$ 的概率趋于 1；反之亦然。（证明略）

定义 5　论域 X 的简约始知识结点集、简约终知识结点集以及简约规则集共同构成 X 的简约知识库。

1.1.3　恒真规则与恒假规则

定理 3　若某一条简约规则中的始知识结点和终知识结点中出现属于相同属性的相同属性程度词，则这条规则是恒真规则；若某一规则的始知识结点和终知识结点中出现属于相同属性的不同属性程度词，则这一条规则等价于另一条简约规则，该规则的始知识结点同于原来规则的始知识结点；而终知识结点是把原来的终知识结点中去掉相应属性的属性程度词。（证明略）

定理 4　假设有简约规则（$L \rightarrow R$），若属性程度词 a 与 L 中的任何属性程度词均取自不同的属性，那么有简约规则（$a \wedge L \rightarrow R$）；若属性程度词 b 与 R 中的任何属性程度词均不相同，则有简约规则（$L \rightarrow b \vee R$）。（证明略）

定理 3 和定理 4 可以发现一些无须通过数据发掘就可以得到的假设规则，通过评价后就可入库，这将在一定程度上减小数据发掘算法的复杂度。

1.2　简约知识库的结构

1.2.1　*逻辑结构*　在双库协同机制中，知识库的逻辑结构是根据推理范畴的结构决定的。由于范畴是由对象集及对象间的态射集决定的，故知识库如图 1 所示的结构。图 1 中的点表示知识结点，有向弧线表示知识结点之间存在推理关系。如果有推理关系，则有向弧线存在，如果没有推理关系，则有向弧线不存在。但是，由于这样的知识库的知识结点数目庞大，并且推理弧线的数目是知识结点数目的方幂的形式，所以直接用这种方式建立知识库在具体实现上较为复杂。

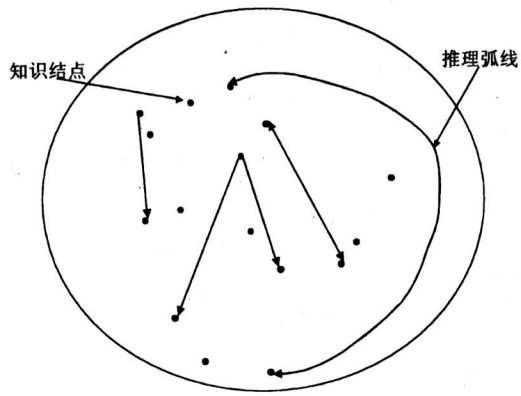

图 1　知识库结构示意图

Fig.1　Knowledge base structure sketch map

1.2.2　*软件实现结构*　在规定了简约始知识结点和简约终知识结点后，简约知识库可视为一个二维数组（或二维表），它的第一维是简约始知识结点；第二维是简约终知识结点；二维数组的每一个元素包含相应规则的信息。简约知识库的结构如表 1 所示。其中，二维数组 $R[1: m][1: n]$ 的每一个元素含有与该简约规则有关的信息。其中一个元素 $\mathrm{Rule}[i, j]$ 是一个数据结构，这个数据结构至少包含如下几方面的信息。

表1 简约知识库的结构

Table 1 The structure of reduced knowledge base

	1	2	3	…	n
1	Rule [1, 1]	Rule [1, 2]	Rule [1, 3]	…	Rule [1, n]
2	Rule [2, 1]	Rule [2, 2]	Rule [2, 3]	…	Rule [2, n]
3	Rule [3, 1]	Rule [3, 2]	Rule [3, 3]	…	Rule [3, n]
⋮	⋮	⋮	⋮		⋮
m	Rule [m, 1]	Rule [m, 2]	Rule [m, 3]	…	Rule [m, n]

1）该规则有无意义，无意义的规则包括恒成立的规则、恒不成立的规则和冗余的规则。

2）是否有领域基础知识能分解成这条规则。

3）表示对这条规则是否已进行了数据发掘。

4）若这条规则分别从基础知识和数据发掘中获得，它们之间有无相容性？

5）可信度；支持度；规则强度。

由于二维数组只是在程序运行中出现，而不能存储，所以我们要把知识库的信息存储到类关系数据库中去。类关系数据库中的一条记录便是一条规则，如表2所示。

表2 存放知识库规则的类关系数据库中一条记录的形式

Table 2 A record in similar relational database which store the rule of knowledge base

序号	始知识结点	终知识结点	是否发掘过	可信度	支持度	有无意义…

1.2.3 *规则的序* 在知识结点集中，要规定知识结点的序。首先，对于始知识结点：

1）用属性程度词的数目作为始知识结点的第一序。由于在简约始知识结点中，每一个属性中至多出现一个属性程度词，所以可以用知识结点中属性的数目作为第一序。

2）对属性程度词个数相同的始知识结点，用属性的字典序作为知识结点的第二序。

3）对属性程度词的数目相同并且属性的字典序相同的始知识结点，用同一属性中属性程度词的字典序作为始知识结点的第三序。

其次，对于终知识结点：

1）用属性程度词的数目作为第一序。

2）用全体属性程度词的字典序作为第二序。

1.3 简约知识库的初始化和基础知识入库

1.3.1 *初始化*

1）恒成立的规则：

a. 始知识结点中存在着某个属性程度词包含在终知识结点中；

b. 简约终知识结点中包含了某个属性所有的属性程度词。

2）恒不成立的规则：

a. 终知识结点只有一个属性程度词，并且始知识结点中包含一个和它属于同一属性的属性程度词；

b. 终知识结点的属性程度词来自于同一个属性，并且始知识结点中存在一个属性程度词来自于这个属性且与终知识结点中的属性程度词均不相同。

3）出现冗余：

a. 有规则（$m \rightarrow k$）和规则（$n \rightarrow k$），但 m 的属性程度词全部包含在 n 中。这时，规则（$n \rightarrow k$）便是冗余规则，可以去掉；

b. 有规则（$k \rightarrow m$）和规则（$k \rightarrow n$），但 n 中包含了 m 的所有属性程度词。这时，规则（$k \rightarrow m$）是冗余的，可以去掉。

1.3.2 *基础知识入库* 具体分为3个步骤：

1）基础规则进行简约分解；

2）查找分解后的简约规则在简约知识库中的位置；

3）对查到的位置上有关基础规则的信息加以调整。

2 构建供挖掘的数据子类结构库

2.1 产生数据子类结构集及其元素间的关系

定义6 称由单个属性程度词构成的知识结点为素知识结点；素知识结点对应的数据子类结构称为数据子类素结构。

产生所有的数据子类素结构。假设某一个素知识结点 n 仅含的一个属性程度词 A_{ij} 对应的数值子域 D_{ij} 是一个半开区间 [lower, upper)，则可以用SQL语句产生知识结点 n 对应的数据子类结构。

产生了数据子类素结构后，就可以繁衍出其他所有的简约知识结点对应的数据子类结构。可以产生两个数据子类结构的并交和一个数据子类结构的补。另外，数据子类结构之间有可达性关系（参见文献[1]的定义12）。可达关系的得出必须计算两个数据子类结构之间的可信度，而可信度的得出又必须首先产生两个数据子类结构的交，同时得到

交中元素的数目。

2.2 数据子类结构库的更新和维护

数据子类结构库是直接由关系数据库导出的。关系数据库的更新就要求数据子类结构库的更新。假设在关系数据库中追加了一条新的记录 u，u 的序号 NO 必须添加到与 u 有关的数据子类结构中去。

首先，u 与每一个属性有关，而且只与每一个属性的其中一个属性程度词有关。对于某一属性 $X[i]$，如果 $a(u)$ 属于第 j 个属性程度词 $A[i][j]$ 对应的数值子域，则把 $A[i][j]$ 记录下来，于是可以得到一个一维整数数组 Change $[s]$，记录了所有与新元组有关系的属性程度词。通过这个属性程度词，可以查找所有与这个新元组有关的数据子类素结构，而在其中末尾的位置添加这个元组。

3 实现双库协同机制的 2 个协调器

3.1 启发协调算法与启发型协调器

启发型协调器的主要目的是为系统的聚焦提供另一个途径。在经典 KDD 进程中，系统的聚焦通常是由用户提供感兴趣方向，KDD 沿此方向进行挖掘。但如果仅沿此方向进行，大量数据中的潜在的、也许会对用户有用的信息往往被忽略。为帮助 KDD 尽可能多的搜索到对用户有用的信息，以弥补用户或领域专家自身的局限性，提高机器的认知自主性，构造了启发型协调器。

当基础知识入库之后，简约知识库中始知识结点到终知识结点的推理弧线部分地建立起来，但对于许多的其他未建立的推理弧线，不能只靠专家的经验和感兴趣度对它们进行发掘，而应该建立一种搜索策略，让机器自身进行搜索知识的短缺，搜索的顺序（即定向挖掘的顺序）应该按照待挖掘规则的规则强度之大小（见 3.2 节）依序进行。

启发型协调器是通过启发协调算法来实现的，算法的奠基是在文献 [1] 中讨论的等价定理与结构对应定理；算法的流程图见图 2。

3.2 对规则强度的度量方法

3.2.1 规则强度

定义 7　假设论域 X 的知识库中的一个知识结点 n 对应的数据子类结构 $\langle n, \mathscr{R}(n)\rangle$ 中所含元组的数目为 $S(n)$，数据库中所含元组的总数为 $S(\mathscr{R})$，则对于一条规则 $r=(n\to k)$；称比值 $S(n)/S(\mathscr{R})$ 为规则 r 的前支持度，记为 $S_f(r)$；称比值 $S(k)/S(\mathscr{R})$ 为规则 r 的后支持度，记为 $S_b(R)$；称比值 $S(n\wedge k)/S(n)$ 为规则 r 的可信度，记为 $C(r)$。可信度与后支持度的比称为熵比，记为 $\gamma(r)$。

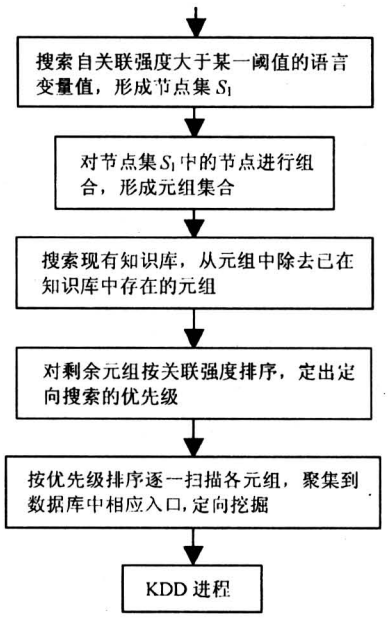

图 2　启发协调算法流程图

Fig.2　Heuristic coordinator algorithm flow chart

定义 8　对于给定论域 X，所谓规则强度函数是指从 X 的规则集 R 到非负实数集 $[0, +\infty)$ 的一个映射

$$\Gamma: R \to [0, +\infty),$$

这个映射具体为自变量 $S_f(r)$ 和 $\gamma(r)$ 到 $[0, +\infty)$ 的一个二元函数

$$\Gamma: [0, +\infty) \times [0, +\infty) \to [0, +\infty),$$

使得，

1) $\Gamma(S_f, 1) = 0$；

2) $\Gamma(S_f, 0) = \Gamma(S_f, +\infty) = \sup_\gamma \Gamma(S_f, \gamma)$；

3) γ 的函数 $\Gamma(S_f, \gamma)$ 在 $[0,1)$ 上单调减，并且当 $S_f \neq 0$ 时，为严格单调减；γ 的函数 $\Gamma(S_f, \gamma)$ 在 $[1, +\infty)$ 上单调增，并且当 $S_f \neq 0$ 时，为严格单调增。

4) S_f 的函数 $\Gamma(S_f, \gamma)$ 在 $[0,1]$ 上为单调增函数，并且当 $\gamma \neq 1$ 时为严格单调增函数；

任何一个满足定义 8 的要求的函数都可以作为规则强度函数，但应该选择一个表达和计算简单、光滑性比较好的函数，例如：

$$\Gamma(S_f, \gamma) = S_f R(\gamma, 0) = S_f \gamma e^{-\gamma^2/2}, \quad (1)$$

其中 $R(\gamma, 0)$ 为瑞利分布函数。

3.2.2 特殊情况的讨论 定义规则强度函数有重要的意义，这不但在规则提供给用户时给每条规则分配了明确的优先级，而且在数值子域的划分中具有重要的量化参考意义。

在给出定义8时，隐含了一个假设：后支持度不为0。但事实上，有些规则的后支持度是0。一旦这样的情况出现，就不能直接用定义8定义的规则强度函数。对此可以采取三种方案：

第一种方案基于信息扩散的假设 对于给定的规则 $r=(n\to k)$，如果 $S(k)=0$，则必有 $S(n\wedge k)=0$，从而 $S_f(r)$和 $C(r)$均为0。于是熵比 $\gamma(r)=C(r)/S_f(r)$为"0/0"型。因此，应该把其他空间中样本所含的信息扩散到这两个空间范围 $d=\psi(f^{-1}(k))$和 $e=\psi(f^{-1}(n\wedge k))$中去。为此建立如下的 s 维欧几里德空间 $(-\infty,+\infty)_s$ 上的抛物型偏微分方程：

$$\frac{\partial u}{\partial t}-a^2\Delta u=f, \tag{2}$$

$$\Delta u=\frac{\partial^2}{\partial x_1^2}+\frac{\partial^2 u}{\partial x_2^2}+\cdots+\frac{\partial^2 u}{\partial x_s^2}。$$

其中：$u(x_1,x_2,\cdots,x_s,t)$是温度函数（在这里是信息强度函数），Δu 为 s 维拉普拉斯算子，$a^2=k/c\rho$，$f=f_0/c$，k 为热传导常数，c 为比热常数，ρ 为物质密度，f 为热源（在这里是信息源）函数 $f(x_1,x_2,\cdots,x_s)$。

在确定了初、边值条件之后，就可以解上述偏微分方程，并得到解 $u(x_1,x_2,\cdots,x_s,t,\varepsilon)$；然后，令 $t\to+\infty$，可以得到一个空间上的函数

$$u_\infty(x_1,x_2,\cdots,x_s,\varepsilon)=\lim_{t\to\infty}u(x_1,x_2,\cdots,x_s,t,\varepsilon)。$$

然后，把函数 $u_\infty(x_1,x_2,\cdots,x_s,\varepsilon)$在 s 维空间 $d_1=\psi(f^{-1}(n\wedge k))$，$d_2=\psi(f^{-1}(n))$，$d_3=\psi(f^{-1}(\mathcal{R}))$，$d_4=\psi(f^{-1}(k))$ 上分别积分，即

$$I_i(\varepsilon)=\int_{d_i}u_\infty(x_1,x_2,\cdots,x_s,\varepsilon),\ i=1,2,3,4 \tag{3}$$

最后可以得出规则 $r=(n\to k)$的强度为

$$\Gamma(r)=\lim_{\varepsilon\to\infty}\frac{I_1(\varepsilon)\cdot I_3(\varepsilon)}{I_2(\varepsilon)\cdot I_4(\varepsilon)}。 \tag{4}$$

第二种方案是第一种方案的简化 把海量个热源转化为 n 个热源，使原方程大大简化。然后，再用与方案一类似的方法，就可以得到规则的强度。

第三种方案是一种简单化了的方案 把所有已知的规则的强度总平均，凡遇到规则强度为"0/0"型的情况，则规则强度一律为这个平均值乘上固定的系数。

规则强度函数从正、反两个方面说明了规则的重要性。如果规则的熵比大于1，那么规则为正规则；反之，如果规则的熵比小于1，那么为反规则；当规则的熵比等于1时，为平凡规则。

对于一个特定的规则，首先应该考虑它的强度，对强度大的规则保留，而把强度小的规则去掉；然后，查看规则的熵比，用熵比和1的关系确定规则是正规则、反规则还是平凡规则。在启发型协调算法中，如前述，据待挖掘规则的强度大小排序，依次自主地形成新聚焦以发现新知识。

3.3 中断协调算法与中断型协调器

中断型协调器是通过中断协调算法来实现的，其算法流程图见图3。

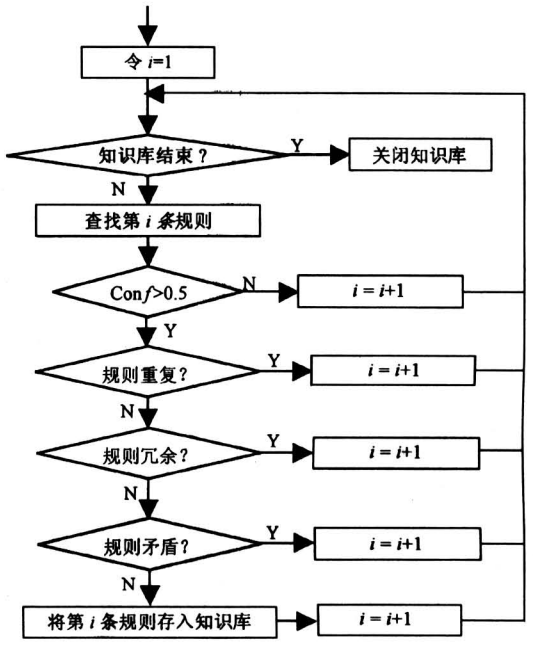

图3 中断协调算法流程图

Fig.3 Interruptive coordinator algorithm flow chat

由于中断型协调器对KDD过程的介入，可以在对于重复性、一致性、冗余性、从属性、循环性等给予准确定义的基础上，利用超图等理论工具，实时地、尽早地将重复、矛盾、冗余的知识淘汰掉，从而做到只对那些有可能成为新知识的假设进行评价，最大限度地减少了评价工作量。在实际的

专家系统中，最终成为新知识的假设占原假设的比例是很小的（发现新知识是困难的），大量假设会是重复和冗余的，因此中断型协调器的引入将提高 KDD 的效率。

4　KDD* 总体结构

4.1　KDD* 的总体结构

将双库协同机制融入 KDD 中，形成所提出的 KDD* 新系统的总体结构模型如图 4 所示。

4.2　KDD* 的特征

KDD* 相对于 KDD 而言，是 KDD 与双库协同机制相融合的一种知识发现的新结构，具有以下特征：

1）KDD* 有机地沟通与融合了 KDD* 新发现的知识与基础知识库中固有的知识，使它们成为一个有机的整体；即实现了用户的先验知识与先前发现的知识可以耦合到发现的过程中。

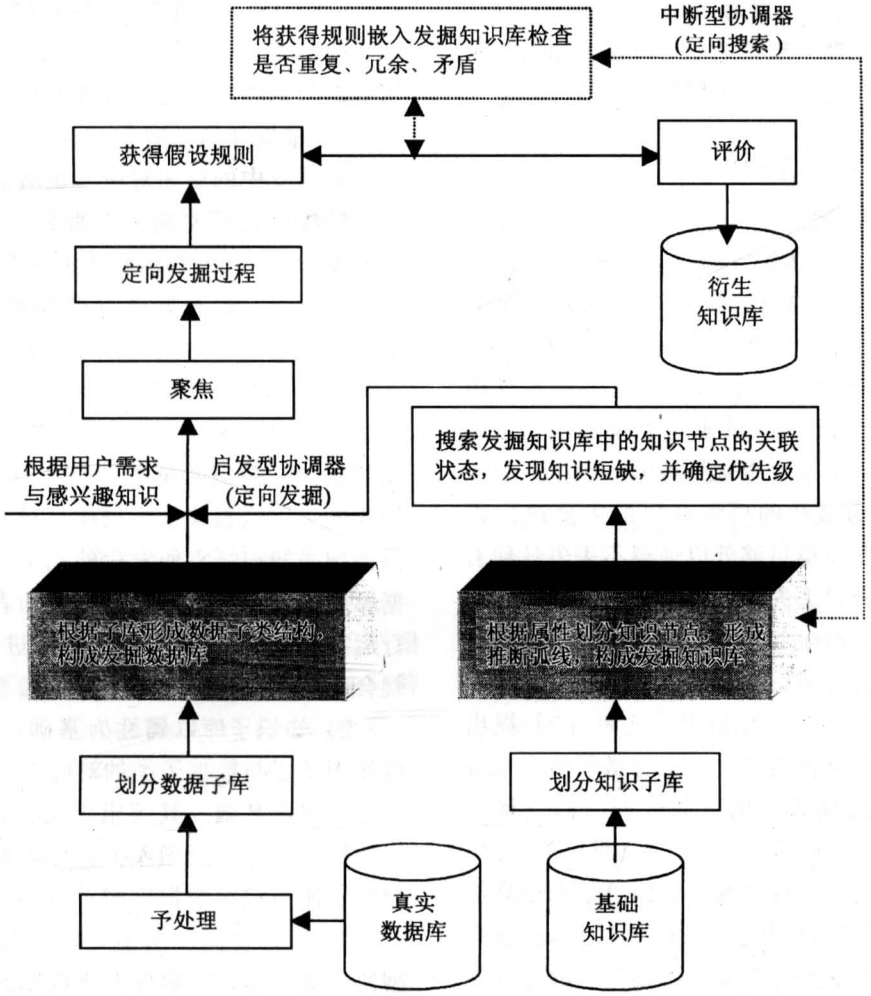

图 4　KDD* 的总体结构模型图

Fig.4　KDD* general structure model chart

2）在知识发现过程中，KDD* 对于冗余性的、重复性的、不相容的信息作出了实时处理，有效地减少了由于过程积累而造成的问题复杂性，同时为新旧知识的融合与合成提供了先决条件；实现了知识与数据库同步进化。

3）在数据库的数据积累过程中，虽然知识库的结构具有一定的稳定性，但它也是随着数据的积累而不断进化，并且这种进化的能力是双库协同机制本身所具有的、无须领域专家干预的。

4）KDD* 改变与优化了知识发现的过程与运行机制，实现了“多源头”聚焦与减少评价量。

从认知科学的角度看，KDD* 强化并提供了知识发现的智能化程度，提高了认知自主性（这将是今后相当长的一阶段内保持的研究基调），较有效

地克服领域专家的自身局限性，实现了采用领域知识辅助初始发现的聚焦。

5）作为 KDD* 的核心技术——双库协同机制的研究，揭示了在一定的建库原则下知识子库与数据子类结构之间的对应关系，为实现限制性的搜索，而减小搜索空间、提高发掘效率提供了有效的技术方法。

6）双库协同机制与其诱导的新结构模型 KDD* 对 KDD 主流的发展有着极大的作用。提出的关联规则与数据聚类规则的挖掘算法与目前实行的算法对比，具有更好的可扩展性与有效性。

5 关联规则的新挖掘算法

5.1 Apriori 算法描述

基于 Apriori 的大项集方法，即使进行了优化，仍存在一些无法克服的固有缺陷：**a**. 无法对稀有信息进行分析。由于大项集方法使用了参数最小支持度 minsup，所以就无法对小于 minsup 的事件进行分析；而如果将 minsup 设成一个很低的值，那么算法的效率就成了一个很难处理的问题。只要人们仍把支持度作为最初的项集产生的主要决定因素，要么把支持度设得足够低以使得不丢失任何有意义的规则，或者冒丢失一些重要规则的风险。对前一种情形可能会产生组合爆炸或计算效率问题，而后一种情形则有可能丢失从用户观点来看是有意义的规则问题。对于前一种情形，文献［2］提出了一种不用支持度阈值的方法，但它必须给定挖掘规则的结论，并且只能挖掘可信度非常高（接近1）的规则。为了弥补后一种情形，文献［3］提出了挖掘意外规则（即支持度都很低而可信度很高的规则），但它利用了常规规则与意外规则是一规则对的性质，故只能挖掘与常规规则相对应的意外规则，仍然会有丢失一些重要规则的风险。**b**. Apriori 的大项集方法需要对整个数据库进行多遍扫描，即使候选集中只有一个元素，也同样需要整个数据库，即它是一种全局扫描搜索。而且它没有考虑现在知识库中是否已经有这些知识，或已经可以从知识库中经过推理得出这些知识。**c**. 对于 Apriori 算法的第2步，也有必要研究。该算法产生的规则是非常多的，对 L_k 中任一元素，就可能产生 $C_k^1 + C_k^2 + \cdots + C_k^{k-1}$ 条规则，它一方面增加运算时间；另一方面对成千上万条规则，用户的评价量非常大，甚至会无从着手；固有这些知识的入库必须考虑是否与原知识库的知识产生矛盾、冗余、循环、从属、重复等问题。

笔者提到的 Maradbcm 算法，将从 KDD 内在机理研究入手，从双库协同机制，这一构建 KDD 过程中最重要的两个参与要素（数据库与知识库）本质联系的认知规律出发，利用新的知识发现结构模型 KDD*（特别是两个协调器），较好地解决上述存在的若干缺陷。

5.2 Maradbcm 算法的理论基础

Maradbcm 算法赖以产生的理论基础是双库协同机制与 KDD* 新结构模型。此处强调说明四点：

1）根据结构对应定理，知识库中的知识素结点与数据库中的层相对应，也就是和该素结点相应的属性程度词相对应，为此经过预处理[4]把真实数据库分成 n 个表（table），即 table 1，table 2，…，table n，n 为属性程度词的个数，而 table k 中的 k 对应了每个属性程度词的 ID 号。每个表的字段只有一个，用来存放真实数据库中的数据的 ID 号，该 ID 所对应的数据处于属性程度词 k 所描述的状态。挖掘数据库就是由这 n 个 Table 组成，这样就无需搜索整个数据库，对于每条短缺的知识只需扫描知识结点所对应的几个表。这对于大型数据库就显得尤为重要，这些小的表可以放入内存进行运算，而整个数据库就无法进行（Apriori 算法就会受到影响）。

2）知识子库以属性为基础，其特点是便于形成知识结点与数据子类的对应关系，从而为定向数据发掘奠定基础。其逻辑结构是在相应的论域内，以属性为基础将规则库类化为若干规则子库，每一规则子库与挖掘数据库相对应。

3）双库协同机制主要由启发型协调器和中断型协调器来实现。启发型协调器的功能是通过搜索知识库中知识结点的不关联态，以发现知识短缺，产生创见意象，从而启发与激活真实数据库中相应的数据类，以产生定向发掘进程，即完成了计算机自动聚焦。

中断型协调器的功能是，从真实数据库的大量数据中经聚焦而生成规则（知识）之后，使 KDD 进程产生“中断”，而去搜索知识库中对应位置有无此生成规则的重复、冗余、矛盾、从属、循环等。若有，则取消该生成规则或做相应处理后返回 KDD 的始端；若无，则继续 KDD 进程，即知识评价。

4）KDD* 的软件实现主要包括启发型协调器、KDD 过程和中断型协调器的功能实现。启发型协调器主要通过计算有向超图的可达矩阵来实现发现知识短缺，进而用规则强度阈值进行剪枝并形成聚焦；KDD 过程主要通过可信度阈值来实现（以挖掘关联规则为例）；而中断型协调器则用 SQL 语言或计算有向超图的可达矩阵来判断知识的重复、矛盾、冗余、循环和从属，并进行相应的处理。

5.3 Maradbcm 算法的算法的描述

设规则强度阈值为 minIntensity，支持度阈值为 minSup，可信度阈值为 minCon。限于篇幅，下面将根据 KDD* 的结构图 1 简要介绍所提出的新算法——Maradbcm 算法：

Step1　*数据预处理*　这里主要是用户选择真实数据库，对于多值属性利用文献［4］进行离散化，形成发掘数据库。

Step2　*发现知识短缺*　为此，首次提出了用有向超图 H 来表示知识库中的知识的新表示方法，并给出了有向超图的邻接矩阵 $\boldsymbol{A}(H)$ 表示，在此基础上提出了一种计算有向超图的可达矩阵 $\boldsymbol{P}(H)$ 新算法，可达矩阵 $\boldsymbol{P}(H)$ 中的 0 元素就是短缺的知识。

Step3　*产生 K_2*　设短缺知识集用 K 来表示，用 K_m 表示规则长度为 m 的短缺知识集，即 $K_m = \{r \mid \mathrm{Len}(r) = m\}$。因为 K 中的元素非常多，利用上面介绍的规则强度 Intensity (r_i) 对 K_2 进行剪枝，对 Intensity $(r_i) \square$ minIntensity (r_i) 的规则 r_i 进行聚焦。即对于短缺知识 $r_i: e_p \rightarrow e_q (r_i \in K_2)$，必须满足：支持度 sup (e_p)，sup $(e_q) \square$ minSup，而 Intensity (r_i) 中的 sup (r_i) = min (sup (e_p), sup (e_q))。

Step4　$m = 2$。

Step5　*对 K_m 产生假设规则*　对 K_m 中的短缺知识 $r_i: e_1 \wedge e_2 \wedge \cdots \wedge e_p \rightarrow e_q (r_i \in K_m)$，进行定向挖掘，即对数据表 table1，table2，…，table p，table q 进行挖掘，计算 Con (r_i) 和 Intensity (r_i)，如果 Con (r_i) minCon 且 Intensity $(r_i) \square$ minIntensity (r_i)，则转 Step6；否则，删除该规则。

Step6　*对规则 r_i 应用中断型协调器进行处理*　搜索基础知识库中对应位置有无此生成规则的重复、冗余、矛盾、从属、循环等。若有，则取消该生成规则或做相应处理，转 Step8；若无，则转 Step7。

Step7　*对规则 r_i 进行评价*　若评价通过，则入库，并对有向超图对应可达矩阵进行计算，对 K_m，进行调整；若评价没有通过，则删除该规则。

Step8　*K_m 是否结束*　若结束，转 Step9；若没有结束，转 Step5 进行下一条规则的处理。

Step9　$m = m + 1$，若 $K_m = \Phi$，转 Step10；否则，转 Step5。

Step10　显示新产生的规则。

Step11　结束。

对于 Maradbcm 算法，其中有些步骤的执行次序是可以调整的。

5.4 Maradbcm 算法的优越性

综上所述，Maradbcm 算法与 Apriori 算法的主要共同点是二者在本质上都是基于统计方法的；二者的主要区别在于以下 5 个方面：

1）基于的学术思想不同　Maradbcm 算法是基于内在机理研究，具体是基于知识短缺（利用有向超图）进行定向挖掘；而 Apriori 算法是基于组合论的数据库全局搜索。

2）基本流程（或基于的模型）不同 Maradbcm 算法的流程见 KDD * 结构模型图，它是短缺知识一条一条地挖掘；而 Apriori 算法是整体性一并挖掘。

3）基础不同　Maradbcm 算法是基于规则强度，它考虑了主观和客观两个方面，即考虑了用户的聚焦（感兴趣度），并涵盖了 Apriori 算法的支持度阈值。

4）发现知识的量不同　Maradbcm 算法考虑的知识库，从而能真正发现新颖的、用户感兴趣的知识，这正是符合了 KDD 定义；而 Apriori 算法是把满足条件的规则全部挖掘出来；另外，Maradbcm 算法克服了 Apriori 算法的两大缺点：遗漏重要的规则和数据库的全局搜索。

5）Maradbcm 算法可融入 KDD 中形成新的开放型的结构模型——KDD*，整个算法实现的运算背景是 KDD* 结构；而 Apriori 算法是原有的封闭系统 KDD。

5.5 实例运行

为了具有比较性，利用文献［5］所提供的蘑菇数据库（mushroom database）进行实验，该数据库有 8124 条纪录，记录了蘑菇的帽子形状、帽子的颜色、颈的形状、颈的颜色、气味、生存环境、是否有毒等 23 种属性，每种属性有 2～12 个枚举值。其中的属性 stalk-root 有 2480 个样本缺省，另

外 veil-type 属性有两个枚举值 partial 和 universal，而所有样本值均为 partial，故对这两个属性不予考虑。我们只对蘑菇是否有毒感兴趣，因此规则的后件均为是否有毒这一属性（即包括‘edible’、‘可以食用’和‘poisonous’、‘有毒’）。该算法是所采用的编程语言为 Delphi 5.0，数据库系统是微软的 SQL-Server 7.0，采用了 Client-Server 结构，计算机 CPU 为 Intel 的 766，内存为 256 MB。

首先，运行 Apriori 算法，设置阈值支持度 minSup=0.4，可信度 minCon=0.6，产生 19 条规则。其次，运行意外规则的挖掘算法。设置阈值支持度 minSup=0.14，可信度 minCon=0.8，另外产生 9 条与上述常规规则相对应的意外规则。最后，以上述的 28 条规则作为基础知识库，运行启发型协调器，为了可比性，设置阈值支持度 minSup=0.14，可信度 minCon=0.6，规则强度 minIntensity=0.45。另外产生 45 条规则，如图 5 所示（仅显示了其中的 12 条规则）。

显示发现规则

删除 入库 取消 关闭

序号	条件1	条件2	结果	支持度	可信度
3	stalkSurAbove silky		mushroom poisonous	0.27	0.94
4	stalkSurBelow silky		mushroom poisonous	0.27	0.94
5	ringType evanescent		mushroom poisonous	0.22	0.94
6	capSurface fibrous		mushroom edible	0.19	0.67
7	bruises no	ringType large	mushroom poisonous	0.16	1
8	odor foul	ringNumber one	mushroom poisonous	0.27	1
9	gillAttachment free	gillSize narrow	mushroom poisonous	0.27	0.89
10	gillSize broad	ringType large	mushroom poisonous	0.16	1
11	capshape convex	stalkSurBelow smooth	mushroom edible	0.19	0.7
12	capSurface fibrous	gillAttachment free	mushroom edible	0.19	0.67
13	bruises bruises	odor none	mushroom edible	0.24	0.96
14	bruises bruises	gillSize broad	mushroom edible	0.33	0.88

图 5 Maradbcm 算法产生的新规则

Fig.5 Maradbcm algorithm create new rule

6 挖掘聚类规则的新算法

6.1 评价函数

对知识库的某一个特定的级别，假设所有的规则框架集（因为还未知是正规则、反规则还是平凡规则，只知道它们的始知识结点和终知识结点）R 中含有 T 条规则。定义数值域划分的评价函数为所有规则强度的平均值。即对于给定的 T 条规则框架，数值域划分的评价函数是一个从划分集 P 到非负实数集的一个映射：

$$\eta: P \to [0, +\infty)$$

使得

$$\eta(p) = \sum_{i=1}^{T} \Gamma(r_i)。 \tag{5}$$

也就是说，对每一个划分 p，都有唯一的非负实数来评价这个划分，式（5）是评价一种划分优劣的依据。

6.2 编码、交叉和突变策略

采用遗传算法，既保证了分解离散块时的小粒度，又保证了求解最优解的可行性。自然地，把数值域划分的评价函数式（5）作为遗传算法中的适应度函数。下面着重解决遗传算法中的编码问题。

6.2.1 *编码* 为了避免边界点选取不合理的情况，采用一种适当限定的方法。即先给每一个分界点确定一个适当的范围，使这一个分界点只能在这一个范围内选取。如数值域的两个边界点固定在两端，而中间的 4 个边界点可以在各自的范围内自由选取。

如果数值域是一个连续的区间[a,b]，则可以把它等分成 2^m 块，然后给每一块一个二进制编码，这一块的二进制编码是一个 m 位的，并且块的集合与 m 位二进制数的集合形成了一一对应。数与块的对应关系是通常的二进制与十进制的对应关系。

如果数值域本身是离散的，它只有 M 个离散点，则取 m 为 ent lb M（向大的方向取整数）。但

这样生成的 2^m 个二进制编码与 M 个离散点可能是非一一对应的关系。一个二进制数只有一个离散点与之对应；但一个离散点可以有一个或两个二进制数与它对应。建立对应的策略是：建立一个长度为 2^m 个单位小区间的区间，把 M 个离散点均匀分布在区间上。当离散点向二进制代码映射时，如果离散点在某一个小区间之中，则把这个离散点对应于这个小区间；如果离散点在两个小区间的边界上，则把离散点对应于左边的小区间。当二进制代码（小区间）向离散点映射时，选择与小区间的中心最近的离散点；如果两个离散点同时与小区间中心位置距离最近，则选择左边的离散点。

于是得到了一条二进制数形成的编码，共分为 s 段，每一段上是对应属性数值域划分所对应的二进制代码。于是，对数值域的每一个划分映射到唯一一条二进制编码链；然而几条二进制编码链可能会映射到同一个数值域的划分，但这种情况出现时，这几条二进制编码链在各个段上的数值是很相近的（最多相差 1）。

6.2.2　交叉　假设二进制编码链（基因链）共有 G 位。对于给定的两个基因 g_1 和 g_2，其交叉方法：

1）在 1 到 G 之间（包括 1 和 G 本身）随机地取两个数 b 和 e，若 $b>1$，则 b、e 交换；

2）把基因 g_1 中 b 到 e 的位（包括 b 和 e 本身）平行地与基因 g_2 中 b 到 e 的位交换。

3）基因 g_1 成为下一代 g_1'，基因 g_2 成为下一代 g_2。

6.2.3　突变　对于给定的基因 g，它产生突变的方法：

1）在 1 到 G 之间（包括 1 和 G）随机地选取一个整数 b；

2）把第 b 位的编码取反（0 变 1，1 变 0）；

3）得到的新基因为下一代 g'。

6.3　基于双库协同机制的数值域划分算法（数据聚类算法）的描述

1）对每一个属性确定划分的边界点的可变范围；

2）对每一个属性的每一个范围分割成小的区间，并确定与它们对应的二进制编码长度；

3）随机产生 $7N$ 条基因链（N 一般取 40 左右）；

4）把所得 $7N$ 条基因链中每一条基因链分别映射到一种划分，并分别求出划分的评价函数值，把函数值从大到小排序，淘汰评价函数值最小的 $3N$ 个基因链；

5）找出评价函数最大的 N 条基因链，作为下一代的一部分；

6）在未被淘汰的 $4N$ 基因链中随机地挑选 $6N$ 条基因链，并以 95 % 的概率进行随机地交叉配对，以 5 % 的概率进行基因突变，得到下一代；

7）重复从第 4 开始执行，直到最优基因的评价函数值前后相差不超过给定误差值 ε，记录下最优基因链；

8）对最优基因链用梯度法再进行优化，得到最终基因链；

9）把最终基因链映射成一种划分，便得到最优（或次优）划分。

使用遗传算法是因为它具有全局搜索能力和极强的鲁棒性，一般不会陷入局部最优。但在若干代遗传、交叉、突变之后，它很可能在一个最优解的周围回旋而不能达到这个最优解；因此，在算法的最后用梯度下降法使它能更好地接近最优解。

7　结论

上述成果充分证实：知识发现系统内在机理之一——双库协同机制的研究，对 KDD 的主流发展起到了重要的推动作用。主要表现为：

1）阐明了作为 KDD 过程中的两个组成要素，（知识库和数据库）之间的关系构造了新的结构模型 KDD*；从而大大缩减了 KDD 的搜索与挖掘空间，使传统的 KDD 运行机制向着开放和智能化的方向拓展。

2）产生了一种知识库的实时维护机制。随着新知识的随时入库，知识库的重复、冗余、矛盾、从属、循环等随时进行检查。

3）充分体现了知识发现系统的认知自主性——KDD 的核心概念与研究基调，提高了发现空间的自动化程度。

4）通过这种机制的研究，可以对当前 KDD 着重于研究具有高效的可扩展性的挖掘算法这一主线，起着本质的强劲的拉动作用；可以派生出超越现有流行算法的新算法，如：Maradbcm 算法与数值域划分（数据聚类）算法等。

5）在哲学方面带来了许多新的思考，同时也反作用于 KDD 领域的研究。　（下转第 78 页）

Analysis on the Machined Surface Profile Error Using Fractals and Wavelet

Liao Xiaoyun, Tang Qian, Zhao Ying, Zheng Shize

(*College of Mechanical Engineering, Chongqing University, Chongqing* 400044, *China*)

[**Abstract**] An analysis method based on fractals and wavelet for machined surface profile error was presented. It can be employed to analyze the machined surface in details, and reconstruct the surface profile for doing some related work, such as performance-tolerance analysis and process quality control. The fractal dimension algorithms, the analysis and reconstruction algorithms of surface profile based on wavelets theories were developed. The given example showed that this method and related algorithms were very effective.

[**Key words**] machining; machined error analysis; fractal geometry; wavelet analysis

(上接第43页)

参考文献

[1] 杨炳儒，王建新．KDD中双库协同机制的研究(Ⅰ)[J]．中国工程科学，2002，4(1)：41～51

[2] Agrawal R, Srikant R. Fast algorithms for mining association rules [A]. Proc 20th Int'l Conf Very Large Database [C], Santiago, Chile, 1994. 487～499

[3] Savasere A, Omiecianski E, navathe S. An efficient algorithm for mining association rules [A]. Proc Int'l Conf Very Larger Databases [C], Zurich, 1995. 423～444

[4] Yang, Bingru. KD(D&K) and double-bases cooperating mechanism [J]. Journal of System Engineering and Electronics, 1999, 10(2): 48～54

[5] Brin S, Motwani R, Silverstein C. Beyond market baskets: Generalizing association rules to correlations [A]. Proc of the ACM SIGMOD [C], Montreal, Canada, 1996. 255～276

A Study on Double Bases Cooperating Mechanism in KDD (Ⅱ)

Yang Bingru, Wang Jianxin, Sun Haihong

(*Information and Engineering School, University of Science and Technology Beijing, Beijing* 100083, *China*)

[**Abstract**] On the basis of paper (Ⅰ), this paper realized the reduction of the knowledge base and constructed the two coordinators and a component in the double-bases cooperating mechanism. Furthermore, the KDD*, which is the combination of KDD and double-bases cooperating mechanism, is constructed as an open and optimized enlargement of KDD. In the same time, this paper proposes a new association rule mining algorithm —— Maradbcm which is based on the study on the inner mechanism, i.e. double base cooperating mechanism, of KDD and explains the algorithm of Maradbcm and Apriori respectively. Through the analysis of the algorithm and examples, it is shown that the main disadvantage of Apriori, overall database searching and deleting meaningful rules by minimum support threshold during creating large item set, can be overcome by Maradbcm algorithm. Then this paper proposed a new clustering method based on double-base cooperating mechanism. It is shown that the study of inner mechanism is important to the KDD mainstream development and it also provides a new path for the research in the area of knowledge discovery system.

[**Key words**] knowledge discovery; double-base cooperating mechanism; heuristic coordinator; interruptive coordinator; Maradbcm algorithm; numerical domain division algorithm