基于粒计算的增量式知识获取方法

张清华*①②　　幸禹可②　　周玉兰②
①(重庆邮电大学数理学院　重庆　400065)
②(重庆邮电大学计算机科学与技术研究所　重庆　400065)

摘　要：该文在研究粒计算理论的基础上，提出了一种基于粒计算的增量式知识获取方法。该方法通过建立决策信息系统原始的知识粒树，对新增数据，在原始知识粒树中查找相匹配的知识粒，并依据决策值更新知识粒树，实现快速高效地处理动态信息系统。算法分析及实验对比结果表明，该方法在动态信息系统知识获取方面优于 RGAGC 和 ID4 方法。

关键词：数据挖掘；知识粒树；粒计算；动态信息系统；增量式知识获取
中图分类号：TP301.6　　**文献标识码**：A　　**文章编号**：1009-5896(2011)02-0435-07
DOI: 10.3724/SP.J.1146.2010.00217

The Incremental Knowledge Acquisition Algorithm Based on Granular Computing

Zhang Qing-hua①②　　Xing Yu-ke②　　Zhou Yu-lan②
①(*College of Mathematics & Physics*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)
②(*Institute of Computer Science & Technology*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)

Abstract: A new incremental knowledge acquisition method based on granular computing theory is proposed. First, an original knowledge granule tree is established according to the decision-making information system. Then, for any new additional data, its matched knowledge granule in original knowledge granule tree is found at first, and then the original knowledge granule tree is updated according to the corresponding decision-making value. The new method is an efficient tool for processing dynamic data information. Both algorithm analysis and experiment results show that the new method for processing dynamic information systems and acquiring corresponding rules is superior to RGAGC and ID4 respectively.

Key words: Data mining; Knowledge granule tree; Granular computing; Dynamic information systems; Incremental knowledge acquisition

1　引言

信息技术的高度发展，使得数据呈几何式的增长，如何从大量、复杂的数据中提取出有用的知识是一个迫切待解决的问题[1,2]。由于 Web 上的数据是半结构化的，且动态更新，这就造成了数据环境的改变，即由静态数据环境转变为动态数据环境。因此，传统的静态数据环境下的数据挖掘方法越来越难以满足动态数据发展的需要[3]。对于动态数据的挖掘问题，文献[4]设计了 ID4 决策树归纳演算法，文献[5]在 ID4 基础上设计了 ID5 决策树算法。文献[6]提出了基于粗糙集的增量式约简算法，文献[7]提出了基于分明矩阵的增量约简算法，文献[8]提出了数据驱动的增量式约简算法，文献[9]提出了基于粗糙集增量式数据挖掘方法等。这些方法都从不同的侧面提出了动态数据的处理技巧，但它们都存在一些缺点或不足，如不能够快速查找知识，不够准确地找到需要的知识，或者当新数据加入时，不得不对整个数据进行重新操作，原来的知识无法有效利用，耗费的时间代价较大，这些问题制约了其应用与推广。粒计算作为一种动态的数据处理方法，其实质是通过选择合适的粒度，来寻找问题的一种较好的、近似的解决方案，降低问题求解的难度[1,10]。从粒计算的角度，动态数据环境下的数据挖掘方法可以理解为：原始的知识库可以构成一个原始的知识粒，该知识粒由多个知识子粒构成。当新数据加入到原始数据集时，寻找合适的知识子粒(或知识子粒集合)，对新数据进行判断和处理，并对各知识子粒(或

2010-03-11 收到，2010-09-17 改回
重庆邮电大学博士启动基金(A2010-06)和国家自然科学基金(60573068)资助课题
*通信作者：张清华　zhangqh@cqupt.edu.cn

知识子粒集)进行知识更新。当处理完所有新加数据后，对各知识子粒进行粒子合并，从而得到动态数据的新知识粒。其实质可以看成是将数据建立成分层递阶[11]的知识粒树，然后自顶向下逐渐更新知识粒以及得到新的知识粒。为此，本文提出了一种基于粒计算的动态数据知识获取方法，即基于粒计算的增量式知识获取算法，该算法从粒计算的角度，对原知识子粒的匹配、粒分解以及粒子的动态学习和更新等方面进行了研究，提出了增量式的知识获取算法，从而实现了对动态数据的高效处理。通过实验对比，说明了该算法优于 RGAGC[12]算法和 ID4[4]算法，这表明这种算法不仅具有理论意义，而且具有实用价值的。

2 相关基本概念

为了更好的描述问题，接下来本文首先介绍相关的基本概念。

定义 1 一个信息系统 $S=<U,R,V,F>$，$R=C\cup D$ 是属性集合，子集 C 和 D 分别称为条件属性集和决策属性集，$D\neq\varnothing$。$V_i=\cup_{r\in R_i}V_r$ 是属性值的集合，V_r 表示属性 $r\in R_i$ 的属性值范围，即属性 r 的值域，$f_i:U\times R_i\to V_i$ 是信息函数。

定义 2 (基本粒)在信息系统 $S=<U,R,V,F>$ 中，令 $a\in R$，$v\in V$，设 (a,v) 或 a_v 表示信息系统上的原子公式，令 $m(a_v)$ 表示 U 上所有满足 $f(x,a)=v$ 的对象集合，其中 m 表示意义函数。称二元序对 $\left(a_v,m(a_v)\right)$ 为信息系统 S 上的基本粒。

定义 3 (知识粒)给定信息系统 $S=<U,R,V,F>$，设 $\phi\Rightarrow\varphi$ 是信息系统 S 的一条决策规则，记为 (ϕ,φ)。称基本粒 $((\phi,m(\phi)),(\varphi,m(\varphi)))$ 为规则 $\phi\Rightarrow\varphi$ 对应的知识粒。

例 1 给出一个信息系统，如表 1 所示。

由表 1 可知，共 6 个样本，每个样本由 3 个条件属性和 1 个决策属性组成。可得到如下的原始规则：

$(\text{Hair,dark})\Rightarrow(\text{Class},-)$，$(\text{Eyes,brown})\Rightarrow(\text{Class},-)$

$(\text{Hair,red})\Rightarrow(\text{Class},+)$

$(\text{Hair},\text{blond})\wedge(\text{Eyes},\text{blue})\Rightarrow(\text{Class},+)$

由这些原始规则得到初始知识粒：$(((\text{Hair,dark}),\{O_4,O_5\}),((\text{Class},-),\{O_2,O_4,O_5\}))$，$(((\text{Eyes},\text{brown}),\{O_2\}),((\text{Class},-),\{O_2,O_4,O_5\}))$，$(((\text{Hair,red}),\{O_3\}),((\text{Class},+),\{O_1,O_3,O_6\}))$，和 $(((\text{Hair,blond})\wedge(\text{Eyes},\text{blue}),\{O_1,O_6\}),((\text{Class},+),\{O_1,O_3,O_6\}))$。

表 1 信息系统

对象	条件属性			决策属性
	Hair	Eyes	Height	Class
O_1	blond	blue	short	+
O_2	blond	brown	short	−
O_3	red	blue	tall	+
O_4	dark	blue	tall	−
O_5	dark	blue	tall	−
O_6	blond	blue	tall	+

给定信息系统 S，设 $M=\{((\phi_1,m(\phi_1)),(\varphi_1,m(\varphi_1))),\cdots,((\phi_n,m(\phi_n)),(\varphi_n,m(\varphi_n)))\}$ 是已有的知识粒集合，对于新样本 x，若存在一个知识粒 $((\phi_i,m(\phi_i)),(\varphi_i,m(\varphi_i)))\in M$，使得 x 在条件属性上与 ϕ_i 对应的原子公式取值相同，则称 x 与该知识粒相匹配；若 x 与该知识粒相匹配，且 x 在决策属性上与 φ_i 对应的原子公式取值相同，则称 x 与该知识粒完全匹配；若存在一个知识粒 $((\phi_i,m(\phi_i)),(\varphi_i,m(\varphi_i)))\in M$，使得 x 在条件属性上与 ϕ_i 对应的原子公式取值相同，而 x 在决策属性上与 φ_i 对应的原子公式取值不同，则称 x 与该知识粒相冲突。在数据增加的过程中，对于一个新样本 x，数据信息更新归纳起来主要有以下几种不同情况：(1) x 与原知识粒匹配；(2) x 与原知识粒完全匹配；(3) x 与原知识粒冲突。

定义 4(知识粒树) 知识粒树是由原始规则形成的知识粒集合构成的树型结构，它的每一层分枝中的非叶子结点都包含一个原子公式，知识粒树定义如下：

(1)树的根结点对应于整个样本空间；

(2)一个知识粒构成一条分枝；

(3)树的每一层都由某个条件属性确定，每个非叶结点都包含一个其所在层属性的原子公式，对知识粒集中所有条件粒在该层可能取的每一个原子公式用一个分枝引出到另一个结点，若某些条件粒中不包含该层属性 $a(a\in R)$ 对应的原子公式，则对应一个表示任意值原子公式 $(a,\#)$ 的分枝；

(4)每个叶子结点为该分枝对应的条件粒的意义集和决策粒的原子公式。

3 基于粒计算的增量式知识获取方法

对动态信息系统，从粒计算的角度建立原始信息系统的知识粒，该知识粒由多个知识子粒构成，当新增数据加入到原始数据集时，须寻找合适的知识子粒，对新增数据进行判断和处理，并对冲突的知识子粒(或知识子粒集)进行知识更新。当处理完所有新增数据后，对各知识子粒进行粒子合并，得到动态学习后的新知识粒。为了加快对原知识子粒的搜索速度，该算法利用原始规则构造了一棵知识粒树，通过逐层对知识粒树中粒的搜索，寻找与新

样本相匹配的知识粒。本文对不同的粒子做相应的更新操作，其主要操作如下：

(1)当新样本 x 与已有知识粒完全匹配，则将新样本 x 与匹配的粒子合并，并更新知识粒树；

(2)当新样本 x 找不到匹配的知识粒，则把新样本 x 作为一个新的知识粒加到知识粒树中，并由该样本生成新的粒子，更新知识粒树；

(3)当新样本 x 与已有知识粒相冲突，将新样本 x 加入到该粒中构成冲突粒，对冲突粒用粒计算的方法进行粒的更新，即将冲突粒分解成多个相容粒，并更新知识粒树。

该算法充分利用了原始知识粒，通过对知识粒树自顶向下的粒搜索方法，避免每次从庞大的原始信息表中重新开始比较，加快了知识粒的搜索速度，从而加快了知识粒的匹配速度。同时，由于算法的原始规则集是基于粒计算方法生成的，对冲突粒子的更新方法也是基于粒计算方法的分类思想，所得规则前件分布均匀。下面给出具体的算法描述。

算法 1　基于粒计算的增量式知识获取算法

输入：原始信息系统 $S=<U,R,V,F>$ 及其由 *RGAGC* 算法得到的规则集，新样本 $\{\text{Newobject}_1, \text{Newobject}_2,\cdots,\text{Newobject}_n\}$

输出：更新后的知识粒树

步骤 1　GT = KnowledgeGranuleTree( )；//根据原始知识粒集创建知识粒树

步骤 2　For $i=1$ to n //n 为新样本个数

(1) GT 根结点＝GT 根结点$\cup\{\text{Newobject}_i\}$；

(2) GranularNode ← GT 根结点；

(3) MatchGranulars=MatchGranular(GranularNode,Newobject_i)；//知识粒的匹配算法

(4) SGranular=SelectGranular(MatchGranulars)；//知识粒的选择

(5)如果 SGranular 存在且匹配知识粒的决策值和 Newobject_i 的决策值相同，则 GranularNode 叶结点＝GranularNode 叶结点$\cup\{\text{Newobject}_i\}$；

(6) 如果 SGranular 存在且匹配知识粒和 Newobject_i 的决策值不同，则 DecomposeGranular( )；//冲突知识粒的分解算法

(7)如果 SGranular 为空，则将新样本作为一个新的知识粒，并调用算法 Addknowledgegranule (Newobject_i) //知识粒的添加算法

步骤 3　算法结束。

算法 2　创建知识粒树的算法描述 KnowledgeGranuleTree( )

步骤 1　由原始信息表 $S=<U,R,V,F>$ 和原始规则集，得到原始规则的知识粒集合 RGS = $\{\text{RGS}_1,\cdots,\text{RGS}_n\}$ (n 为原始规则个数)，其中 RGS_i 为一个知识粒；

步骤 2　知识粒树的根结点为 $\{U\}$；

步骤 3　由条件属性依次作为知识粒树第 1 层到 $h=|C|$ 层的划分属性，每个非叶结点包含该属性的一个原子公式；

步骤 4　For $i=1$ to n // n 表示原始知识粒的个数调用算法 Addknowledgegranule(RGS_i)；//将知识粒添加到知识粒树中

步骤 5　生成知识粒树。

算法 3　知识粒树的添加算法描述 Addknowledgegranule(RGS)

步骤 1　GranularNode ← 知识粒树的根结点

步骤 2　For $j=1$ to h //h 表示知识粒树的层数

{如果 GranularNode 引出的分枝结点中存在一个表示知识粒 RGS 在属性 j 上的原子公式 (a_j,v_j)，则 GranularNode ← (a_j,v_j)；

否则创建 GranularNode 的一个分枝结点表示知识粒 RGS 在属性 j 上的原子公式 (a_j,v_j)，并 GranularNode ← (a_j,v_j)；}

步骤 3　将知识粒 RGS 中条件粒的 $\{m(\phi_1)\wedge,\cdots,m(\phi_i)\}$ 和决策粒的原子公式加入到 GranularNode 的叶子结点中。

算法 4　匹配知识子粒的算法描述 MatchGranular (GranularNode, Newobject)

步骤 1　For $i=1$ to GranularNode 的分枝数目

记 $M=0$ 表示成功匹配知识粒

{如果 GranularNode 在的分枝结点为叶子结点，则匹配成功，$M=M+1$，返回叶子结点及 M；//叶子结点即为匹配的知识子粒

如果 GranularNode 存在原子公式与新样本 Newobject 在属性 a_i 上属性值相等的分枝结点或 GranularNode 存在原子公式为任意值“#”的分枝结点 $(a_i,\#)$

{GranularNode ← 当前分枝结点；

MatchGranular(GranularNode,Newobject);}

否则，匹配失败，返回 M；}

算法 5　知识子粒的选择算法描述 SelectGranular( )

步骤 1　If $(M=0)$ //表示没有匹配的知识粒

{把 Newobject_i 作为一个新的知识粒，加入到知识粒树中}

else //表示存在匹配知识粒

{if $(M=1)$ //表示存在唯一匹配知识粒

{则选择该叶子节点作为匹配的知识粒}

else //表示存在多个匹配的知识粒

{if (存在多个相同覆盖度的知识粒)

//覆盖度表示知识粒中含有的样本的个数

{选择规则最短的知识粒，若存在多个最短规则的知识粒，则选择任意}

else

{选择覆盖度最高的知识粒}}

}

算法 6 冲突知识粒的更新算法描述 Decompose Granular( )

步骤 1 将匹配知识子粒中条件粒的意义集 $\{\text{object}_1,\text{object}_2,\cdots,\text{object}_n\}$ 和新样本 Newobject 组成一个新的样本空间 U'，$R'=C'\bigcup D'$，其中 C' 是原始知识空间中知识粒 RGS 约简掉的条件属性，D' 为决策属性；

步骤 2 令 $\text{RS}=\Phi$，$Gr_{\text{RS}}=\Phi$，$h=1$，RS 是规则集，Gr_{RS} 表示已经被规则集所包含的对象集，DFS 表示由决策属性 $D'=\{\text{d}\}$ 得到的原子公式集，h 表示分解 U' 的公式长度；

步骤 3 计算 U' 在 h 层的组合，$\Psi_s^h=\{\{m(\phi_1^h)\},\cdots,\{m(\phi_i^h)\},\cdots,\{m(\phi_n^h)\}\}$，$\phi_i^h$ 表示长度为 h 的原子公式，$i=1,\cdots,n$，n 是 Ψ_s^h 的粒子个数；令 $\text{CFS}^h=\{\phi_1^h,\cdots,\phi_i^h,\cdots,\phi_n^h\}$ 对于每一个 $\phi_i^h\in\text{CFS}^h$，$\varphi_j\in\text{DFS}$：

若 $\text{AS}(\phi_i^h\Rightarrow\varphi_j)==1$ 且 $m(\phi_i^h)-Gr_{\text{RS}}\neq\varnothing$，

则 $\text{RS}=\text{RS}\bigcup\{\phi_i^h\Rightarrow\varphi_j\mid\text{AS}(\phi_i^h\Rightarrow\varphi_j)==1\}$，$Gr_{\text{RS}}=Gr_{\text{RS}}\bigcup\{m(\phi_i^h\Rightarrow\varphi_j)\}$；

步骤 4 如果 Gr_{RS} 包含了 U' 中的所有样本，则转入步骤 5，否则 $h=h+1$，返回到步骤 3；

步骤 5 对于 RS 中的每一条规则 rule_i {由规则得到其对应的知识粒，并调用算法 AddKnowledge Granule( )}。

4 算法分析

设 $|U|$ 为原始记录数，$|C|$ 为条件属性个数，r 为条件属性中最大可能的属性值个数。当增量式学习一条记录时，传统的非增量式 RGAGC 算法在最坏情况下，其时间复杂度为：$O\left(\sum_{i=1}^{|C|}C_{|C|r}^{i}(|U|+1)^i)\cdot\log(|U|+1)\right)$。本文提出的算法首先需要与已有的知识子粒匹配，其时间复杂度为 $O(|C|r)$，当新样本与知识子粒相冲突时，则需要对冲突的知识子粒进行更新操作，若匹配知识子粒中记录的个数为 $|U'|$，知识子粒中被约简的属性个数为 m，则时间复杂度为 $O\left(\sum_{i=1}^{m}C_{mr}^{i}(|U'|+1)^i\log(|U'|+1)\right)$。显然，由于 $|U'|+1<<|U|+1$，$m<<|C|$，因此，本文提出的增量式算法优于传统的非增量式的 RGAGC 算法。

例如，一个原始决策信息系统如表 2 所示，其中 $U=\{1,2,3,4,5,6,7\}$ 为对象集，$C=\{\text{Outlook, Temperature,Humidity,Windy}\}$ 为条件属性集，$D=\{\text{d}\}$ 是决策属性集。

表 2 原始信息系统

对象	条件属性				决策属性
	Outlook	Temperature	Humidity	Windy	d
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

根据 RGAGC 算法得表 2 的决策规则：(((Outlook,Sunny,{1,2}),((d,N),{1,2,6})),(((Outlook,Overcast),{3,7}),((d,P),{3,4,5,7})),(((Temperature,Mild),{4}),((d,P),{3,4,5,7})),(((Temperature, Cool)∧(Windy,False){5}),((d,P),{3,4,5,7})) 和 (((Outlook,Rain) ∧ (Windy,Ture),{6})((d,N),{1,2,6}))。根据这些决策规则，我们来说明增量式知识获取算法的具体过程。

第 1 步 由原始规则得到初始知识粒，并构建原始知识粒树如图 1 所示。

第 2 步 逐渐加入了 3 个新的样本，如表 3 所示。

表 3 新增的 3 个新样本

8	Sunny	Hot	High	False	N
9	Sunny	Hot	Normal	False	P
10	Rain	Mild	Normal	True	P

下面将根据增量式学习的 3 种不同情况，考虑对新增样本的匹配和知识粒树更新。

(1)对于新样本 8，在第 1 层属性 Outlook 上有两个与之相匹配的原子公式 (Outlook,Sunny) 和任意值 (Outlook,#)，继续匹配由这两个原子公式所引出的下一分枝，发现 (Outlook,#) 这枝的 Temperature 属性并不匹配，而 (Outlook,Sunny) 这枝是引出叶子结点，匹配对象为 {1,2}，所以选择 {1,2} 为匹配知识粒，判断决策属性，发现它们的决策属性一致，则合并得到新的叶子结点 {1,2,8}。

(2)对于新样本 9，在第 1 层属性 Outlook 上有

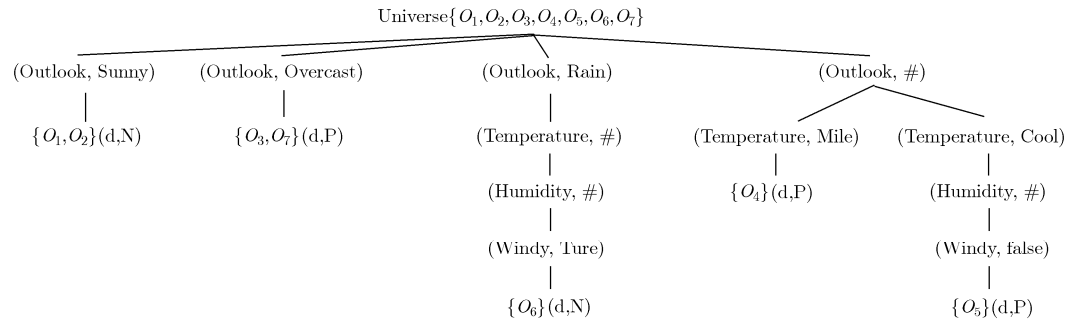

图 1 原始知识粒树

两个与之相匹配的原子公式 (Outlook,Sunny) 和任意值 (Outlook,#)，则继续匹配由这两个原子公式所引出分枝的下一层，发现 (Outlook,#) 这枝的 Temperature 属性并不匹配，而 (Outlook,Sunny) 这枝是引出叶子结点，匹配对象为 {1,2,8}，所以选择 {1,2,8} 为匹配知识粒，判断决策属性，发现它们的决策属性并不一致，产生冲突知识粒。将 {1,2,8,9} 作为一个新的样本空间，然后根据基于粒计算的分类思想对冲突粒进行更新，得到新的规则：

(((Humidity,High) ∧ (Windy,False),{1,8}),((d,N),{1,2,8})) (((Humidity,Normal),{9}),((d,P),{9})) 和 ((Windy,True),{2}),((d,N),{1,2,8}))。按照知识粒的添加规则，分别将由这两条规则得到的知识粒添加到知识粒树中。

(3)按照类似方法，对于新样本 10，选择与 {4} 匹配并合并得到新的叶子结点 {4,10}。

第 3 步　增加 3 个新样本后，更新后的知识粒树如图 2 所示。

5　实验对比分析

本文在内存为 512 MB，CPU 为 2.4 GHz，操作系统为 WindowsXP 环境下，用 VC6.0 编程实现了两组对比实验，第 1 组为基于粒计算的增量式知识获取算法与 RGAGC 算法的对比实验，第 2 组为基于粒计算的增量式知识获取算法与 ID4 增量式知识获取算法的对比实验。为了反映多次追加学习的效果，本文采用了基于训练集一定比例的数据作为新增样本进行多次增量学习，在原始数据集中随机抽取 α % 作为原始训练集，抽取 β % 作为增量学习集，最后对整个数据集进行测试，通过 5 次实验求平均值作为实验结果。

5.1　测试实验 1

主要步骤如下：

(1)用基于粒计算的规则获取算法 RGAGC 生成原始训练集的规则集；

(2)根据得到的规则集和增量学习集，运行本文提出的基本粒计算的增量式知识获取算法，生成规则，并记录运行时间；

(3)把生成的规则对原始数据集进行样本测试，输出正确的识别率；

(4)把原始训练集和增量学习集一起用非增量式的基于粒计算知识算法 RGAGC 生成规则，并记录运行时间，然后对原始数据集进行样本测试，输出正确识别率。

仿真结果如表 4 所示，显然该算法在保证正确识别率与 RGAGC 算法相当的情况下，其增量学习的时间比传统粒计算模型学习算法大大缩短。由于本文采用对知识粒树自顶向下逐层的粒搜索方法，避免每次从庞大的原始决策表开始重新比较，也比对原始规则集的进行搜索快，匹配速度大大降低，并且算法只是对部分冲突知识粒进行了基于粒计算思想的更新操作，从而加快更新速度，减小了计算量。

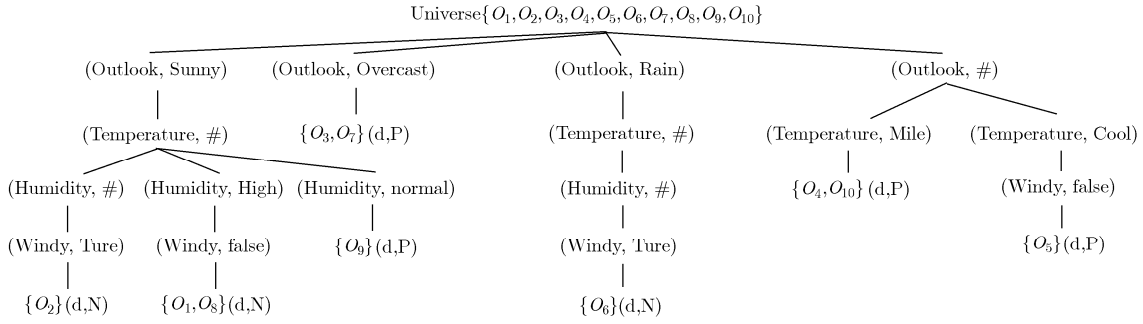

图 2 更新后的知识粒树

表 4 数据测试结果

数据集	α (%)	β (%)	本文算法				RGAGC 算法				ID4 算法			
			n	l	c(%)	t(ms)	n	l	c(%)	t(ms)	n	l	c(%)	t(ms)
BALANCE	40	10	160	3.21	80.2	30	147	3.10	78.7	78	173	3.58	50.6	<1
BALANCE	35	15	155	3.19	79.7	31	147	3.10	78.7	78	164	3.67	50.9	<1
BALANCE	30	20	160	3.23	77.9	31	147	3.10	78.7	78	193	3.67	46.5	10
Ecoli	40	10	77	2.75	87.5	15	76	2.63	87.5	187	93	5.42	76.5	<1
Ecoli	35	15	77	3.44	85.4	15	76	2.63	87.5	187	98	5.46	71.4	<1
Ecoli	30	20	78	3.03	86.3	15	76	2.63	87.5	187	96	5.39	63.7	<1
German	40	10	306	2.95	78.0	135	287	2.26	79.6	4125	361	6.65	63.4	10
German	35	15	316	2.84	77.2	156	287	2.26	79.6	4125	338	7.82	60.1	20
German	30	20	301	3.62	75.5	228	287	2.26	79.6	4125	361	9.17	53.2	20
Pima indians	40	10	221	3.51	81.8	63	188	3.15	82.9	3640	217	5.67	65.2	16
Pima indians	35	15	230	3.56	80.7	78	188	3.15	82.9	3640	221	5.80	69.4	16
Pima indians	30	20	234	3.58	79.4	78	188	3.15	82.9	3640	224	5.85	65.1	31

注：α：原始训练集比例，β：新增样本比例，n：规则的数目，c：正确识别率，l：规则平均长度，t：算法运行时间。

5.2 测试实验 2

主要步骤如下：

(1)用基于粒计算的规则获取算法(RGAGC)生成原始训练集的规则集；

(2)根据得到的规则集和增量学习集，运行本文提出的基本粒计算的增量式规则获取算法，生成规则，并记录运行时间；

(3)把生成的规则对原始数据集进行样本测试，输出正确的识别率；

(4)把原始训练集用 ID3 算法生成决策树，

(5)对增量学习集用 ID4 算法进行学习并生成新的决策树，记录运行时间，并对原始数据集进行样本测试，输出正确识别率。

通过本文算法与经典 ID4 算法的实验结果(表 4 所示)看出，该算法无论是在规则的数量还是在正确识别率方面都较 ID4 算法有较大优势，但其运行时间大于 ID4 算法。由于本文提出的算法在对冲突粒的更新操作上正是基于粒计算方法的思想，力争获得识别效率较高的规则(提高解的精确度)，因此花费的时间较 ID4 算法多，但获得规则数目与平均长度比 ID4 算法优越，并且正确识别率也比 ID4 算法高。这说明了本文提出的算法对动态数据的处理是可行的。

6 结束语

为了实现对动态数据的处理，本文从粒计算的角度，建立信息系统的知识粒树，并对知识粒树中的知识子粒进行分解、粒子的动态学习和更新、匹配知识子粒的判定条件等方面进行了研究，提出了基于粒计算的增量式知识获取算法，实现了高效地从动态信息系统中获取规则。仿真实验表明，该算法在保证正确识别率与传统粒计算模型的知识获取算法相当的情况下，其增量学习的时间比传统粒计算模型增量学习算法大大缩短。本文的研究工作是粒计算方法在处理动态信息系统的尝试与探索，希望这些研究结果对研究高效的知识获取方法有一定的借鉴意义，促进粒计算理论在数据挖掘领域中的应用研究。

参 考 文 献

[1] 苗夺谦, 王国胤, 刘清, 林早阳, 姚一豫. 粒计算: 过去、现在与展望[M]. 北京: 科学出版社, 2007: 142–178.
Miao Duo-qian, Wang Guo-yin, Liu Qing, Lin Zhao-yang, and Yao Yi-yu. Granular Computing: Past, Present and Future Prospects[M]. Beijing: Science Press, 2007: 142–178.

[2] 杨育彬, 李宁, 张瑶. 基于社会网络可视化分析的数据挖掘[J]. 软件学报, 2008, 19(8): 1980–1994.
Yang Yu-bin, Li Ning, and Zhang Yao. Networked data mining based on social network visualizations[J]. *Journal of Software*, 2008, 19(8): 1980–1994.

[3] 鲍宇, 曾国荪, 管红杰. Web 数据挖掘中的可信数据来源[J]. 计算机科学, 2009, 36(4): 211–214.
Bao Yu, Zeng Guo-sun, and Guan Hong-jie. Trusted data source in web data mining[J]. *Computer Science*, 2009, 36(4): 211–214.

[4] Schlimmer J C and Fisher D A. Case study of incremental concept induction[C]. Proceedings of the 5th National Conference on Artificial Intelligence, Philadephia, USA, 1986: 496–501.

[5] Utgoff P E. ID5: An incremental ID3[C]. Proceedings of International Conference on Machine Learning, San Mateo,

CA, 1988: 107–120.

[6] 刘宗田. 属性最小约简的增量式算法[J]. 电子学报, 1999, 27(11): 96–98.
Liu Zong-tian. An incremental arithmetic for the smallest reduction of attributes[J], *Acta Electronica Sinica*, 1999, 27(11): 96–98.

[7] Wang J. Reduction algorithms based on discernibility matrix: the order attributes method[J]. *Journal of Computer Science and Technology*, 2001, 16(6): 489–504.

[8] Wang Guo-yin, Wang Yan. 3DM: Domain-oriented data-driven data mining. *Fundamenta Informaticae*, 2009, 90(4): 395–426.

[9] 李鹏, 王晓龙, 关毅. 一种基于粗糙集增量式规则学习的问题分类方法研究[J]. 电子与信息学报, 2008, 30(5): 1127–1130.
Li Peng, Wang Xiao-long, and Guan Yi. Question classification with incremental rule learning algorithm based on rough set[J]. *Journal of Electronics & Information Technology*, 2008, 30(5): 1127–1130.

[10] 张铃, 张钹. 问题求解理论及应用 —— 商空间粒度计算理论及其应用(第2版) [M]. 北京: 清华大学出版社, 2007: 1–48.
Zhang Ling and Zhang Bo. The Theory and Applications of Problem Solving-Quotient Space Based Granular Computing (The Second Version) [M]. Beijing: Tsinghua University Press, 2007: 1–48.

[11] 张清华, 王国胤, 刘显全. 分层递阶的模糊商空间结构分析[J]. 模式识别与人工智能, 2008, 21(5): 627–634.
Zhang Qing-hua, Wang Guo-yin, and Liu Xian-quan. Analysis of the hierarchical quotient space structure of fuzzy quotient space[J]. *Pattern Recognition and Artificial Intelligence*, 2008, 21(5): 627–634.

[12] An J J, Wang G Y, Wu Y, and Gan Q. A rule generation algorithm based on granular computing[C]. Proceedings of 2005 IEEE International Conference on Granular Computing, Beijing, China, 2005: 102–107.

张清华: 男，1974 年生，副教授，博士，硕士生导师，研究方向为智能信息处理、粒计算等.

幸禹可: 男，1986 年生，硕士生，研究方向为粒计算理论及其应用.

周玉兰: 女，1983 年生，硕士生，研究方向为粒计算与知识获取.