含缺省属性值的数据中的规则发现算法*)

张师超　倪艾玲

（悉尼科技大学信息技术学院 澳大利亚悉尼）（广西师范大学计算机系　桂林 541004）

摘　要　根据用户定义的主观重视程度，综合考虑属性的缺省值情况以及信息系统中各属性的属性值个数，确定模式的综合重要度，进而得出模式的最小支持度。另外，提出剪枝技术剪除无意义的频繁项集，仅挖掘用户感兴趣的规则。实验证明该方法是有效的。

关键词　知识发现，数据挖掘，领域知识

Rule Discovery from Data with Default Attribute Values

ZHANG Shi-Chao　NI Ai-Ling

(Faculty of Information Technology, University of Technology Sydney, Australia)

(Department of Computer Science, Guangxi Normal University, Guilin 541004)

Abstract　Taking into account the interest given by the customers, the number of the value of each attribute and the default attribute value in the information system, this paper proposes an approach for hunting the minimum support by synthetical importance of patterns. Furthermore, an algorithm is developed for mining interesting rules from data with default attribute values. For efficiency, a pruning algorithm is designed for removing indifference frequent itemsets. We experimentally evaluate the proposed approach, and demonstrate it is efficient and promising.

Keywords　Knowledge discovery, Data mining, Domain knowledge

1　引言

随着信息系统的建立和快速发展，许多企业和组织积累了大量的数据。要充分利用这些数据，我们必须挖掘那些对决策有帮助的、隐藏在数据中的规律和规则。因此，从大量数据中自动发现新奇、有用和易于理解的规律、规则的数据挖掘技术是信息系统的一个重要研究课题。通过属性间的关联规则发现，我们可以找到属性之间的相互关联，从而能获得隐藏在属性背后的属性值之间的重要的关联信息。

在现实世界中，用户凭着对领域知识的了解会有主观的兴趣取向。而目前对规则的挖掘主要是对规则的客观关注程度的研究，也就是用支持度、置信度等作为有用规则的标准来挖掘关联规则[1]。文[1]对信息系统中的主观的观注程度问题作了研究，但只是在所挖掘出来的规则聚类，进行规则推荐时，考虑到了用户的主观观注程度，显然这是不够的；另一方面，由于现实条件的限制，有些属性值的来源不是十分的准确可靠的，或者由于种种原因有的属性值不是完备的，也就是有缺省值，当然它的重要程度不能与具有精确的完全值的属性相媲美；还有，虽然在文[2]中，在交易数据库中根据属性重要性讨论了关联规则的挖掘方法，而信息系统也是要转化成交易数据库的形式进行规则发现，但它们在很多方面还存在很大的差异，如在信息系统中每一个属性的类别的个数一般是不一样的，对于属性值个数较多的属性，一般其每一属性值的支持度较低，如果不加以处理的话，即使其观注程度高也难以得到有关该属性的关联规则。最后在信息系统中，有些属性值之间并不一定存在规则关系，但由于该属性的属性值较少而导致的偶合关系使它们的支持度很高。基于上述原因，本文根据各项集的不同重要度、缺省值情况以及属性值的个数提出了不同项集不同的支持度，并对频繁项集进行剪枝，去除无意义的频繁项集，提出了基于信息系统中模式综合重要度的关联规则发现方法。

2　基本概念

2.1　信息系统的定义

其为$\langle U,\Omega,V_q,f_q\rangle_{q\in\Omega}$，其中$U$是有限对象集，即$U=\{x_1,x_2,\cdots,x_n\}$。$U$中的每个$x_i$称为一个对象。$\Omega$是有限属性集，对于每一个$q\in\Omega$，$V_q$是属性$q$的值域，$f_q:U\to V_q$是信息函数[4]。

信息系统是一由“对象-属性”关系所构成的表格形式，下面给出一实例：

表 1

U	a	b	c	d	e	f	g	h
X1	1	1	3	1	2	2	1	2
X2	1	2	2	3	3	1	2	2
X3	2	3	3	2	2	1	1	2
X4	2	3	2	3	2	2	1	1

对此信息系统作如下解释：

对象$x1,\cdots\cdots,x4$是病例；

属性a：性别 b：土壤情况；c：居住地形；d：钩虫；e：鞭虫；f：蛔虫；g：结肠内阿米巴；h：大酵母菌

2.2　二进制信息系统[4]

*)基金资助项目：澳大利亚 ARC 项目(DP0559536)；国家自然科学基金项目(60463003)；广西自然科学基金项目。张师超　教授，博士，研究方向为数据挖掘，人工智能等；倪艾玲　硕士研究生。

如果值域 V_q 仅有两个值，属性 q 就称为二进制属性；如果每个属性均为二进制属性，则该信息系统称为二进制信息系统。二进制属性又分为对称二进制属性与非对称二进制属性。在非对称二进制属性中，其值 $v_q=\{0,1\}$：如果 $f_q(x)=1$，表明 q 属性出现，而如果 $f_q(x)=0$，表明我们不知道有关对象 x 的属性 q 的任何信息。在此我们转化成的是非对称二进制信息系统。

二进制信息系统表示为：$\langle U,\Omega^B,\{0,1\},f_q^B\rangle_{q\in\Omega^B}$

$\Omega^B:=\{\langle q,v\rangle:q\in\Omega,v\in ran(f_q)\}$，其中，$ran(f_q)$为 f_q 的值集。

信息函数 $f_q^B(x):=\begin{cases}1 & \text{if } f_q(x)=v\\ 0 & \text{otherwise}\end{cases}$

表 2 即为由表 1 转化而来的二进制信息系统。为了易读，在表 2 中用 q_i 代替$\langle q,v\rangle$作为属性的表示形式。

表 2

U	a		b			c			d			e			f		g		h	
	a1	a2	b1	b2	b3	c1	c2	c3	d1	d2	d3	e1	e2	e3	f1	f2	g1	g2	h1	h2
X1	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1
X2	1	0	0	1	0	0	1	0	0	0	1	0	0	1	1	0	0	1	0	1
X3	0	1	0	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1
X4	0	1	0	0	1	0	1	0	0	0	1	0	1	0	0	1	1	0	1	0

注："1"代表该属性出现，"0"代表该属性不出现。事实上，上表中的属性代表信息系统中的属性值，如：a1 代表信息系统中的属性 a 的值为"1"，而 a2 代表信息系统中的属性 a 的值为"2"的情况。

对于表 2，我们抽取属性值为"1"的属性记录下来，作为进一步分析的依据。也即成了交易数据库的形式，见表 3。

表 3

U	Items
x1	a1 , b1 ,c3 ,d1,e2 ,f2 ,g1 ,h2
x2	a1 ,b2 ,c2 ,d3 ,e3 ,f1 ,g2 ,h2
x3	a2 ,b3 ,c3 ,d2 ,e2 ,f1 ,g1 ,h2
x4	a2 ,b3 ,c2 ,d3 ,e2 ,f2 ,g1 ,h1

从表 3 可以看出，由信息系统转化来的交易数据库与普通交易数据库有几点不同：1)每条交易是等长的，也即每一条记录的属性个数；2)若某属性的值个数较多，则其每个值在数据库中出现的次数必然较少，如果取固定的支持度，则拥有该属性值的频繁项集必然少，也就是说很难得到关于该属性的频繁项集；3)当某属性值较少时，尽管它与其它的属性没有一定的关联，只是随机地成对出现，但它们很可能还会以频繁项集的形式存在，导致一种欺骗现象。

3 基于模式综合重要度的最小支持度

3.1 模式的综合重要度

定义 1 属性综合重要度定义为：

$$I'(a_i)=I(a_i)\times(n_{ai}/N_{attr})\times((|U|-|X|)/|U|)$$

其中，$a_i\in\Omega$，$I(a_i)$为根据领域知识由用户给定的属性 a_i 的重要度，n_{ai} 为属性 a_i 的属性值个数，N_{attr} 为所有属性的属性值个数之和。$|U|$是对象集 U 的个数，$X\subseteq U$ 且 X 中对象的属性 a_i 的值为空。如前所述，当把信息系统转化成交易型的数据进行关联规则挖掘时，各属性的属性值的个数一般不会完全一样，若某属性有较多的属性类别，则其各属性值出现的几率必然相对较小，每一个值的支持度会自然就较小。因而，在挖掘规则时，在同一支持度下，很难挖掘到有关属性类别较多的属性的规则，即使对它的主观观注程度高也一样无法得到有关该属性的规则。为此，在上面的公式中我们根据该属性值的个数，得到其属性综合重要度，最终目的是为了改变其最小支持度。另外，当属性 a_i 没有空缺值时，$(|U|-|X|)/|U|$的值为 1，对综合重度没有影响，如果属性 a_i 的空缺值越多，其重要度就越低，这与其能提供的信息量及可靠程度是一致的，也是合理的。

定义 2 模式 A 定义为合取式的形式：$c_{i1}\wedge c_{i2}\cdots\wedge c_{ip}$，其中 c_{is} 为 $a_{is}=v_{is}$，a_{is} 是属性，v_{is} 是属性 a_{is} 的值($s=1,2,\cdots,p$)，其中 p 称为模式 A 的长度，记为 $len(A)$。如"钩虫=1 鞭虫=0 居住地形=3"即为一模式，该模式的长度为 3。出现在模式 A 中的属性的集合$\{a_{i1},a_{i2}\cdots,a_{ip}\}$记为 A_{attr}(参考文[1])。

定义 3 模式 A 的综合重要度 IMP(A)定义为

$$\mathrm{IMP}(A)=\sum_{a_i\in Aattr} I'(a_i)/\mathrm{len}(A) \qquad (1)$$

其中，A 为定义 2 中的模式，len(A)为模式 A 的长度，a_i 为模式 A 中的属性，$I'(a_i)$为模式 A 中属性 a_i 的综合重要度。从式(1)来看，IMP(A)也就是该模式中各属性综合重要度的平均值，是合理的。但从另一方面看，用户对不同属性的重要度的设置也只是表明用户对各属性的重要性的一种排序以及重要性之间的距离。例如，由三个属性构成的属性组，对应的属性值个数为 7,7,4，其重要度设置为(1 ,1 ,0.8)和设置为(0.6,0.6,0.4)，虽然它们的绝对重要度有很大的差别，但在每一组重要度的设置中，它们之间的顺序没变，距离差别不大，也就是说它们的相对重要度相似。从用户的角度来说，这样的设置代表了他们近似的观点，即在这三个属性中，第一条和第二条属性是最受用户关注的，而第三条属性所受的关注程度低一些。但如果简单地用式(1)去定义重要度，明显地，两种不同的重要度设置会产生差别很大的模式重要度。从而在规则挖掘中，将产生很大的不同。根据需要，我们把属性重要度再进行处理如下：

$$I''(a_i)=(I'(a_i)/\sum_{a_j\in Aattr} I'(a_j))\times(1/\max(I'(a_j))$$

上式由两部分组成，第一部分 $I'(a_i)/\sum_{a_j\in Attr} I'(a_j)$对重要度进行规一化，即使所有重要度之和为 1。因此，重要度为(1,1,0.8)和(0.6,0.6,0.4)，属性个数分别为 7,7,4，则对应处理后的重要度为(0.407, 0.407, 0.186)和(0.42, 0.42, 0.16)，我们可以看出处理过的重要度更能反映用户的真实观点，近而挖掘出的规则也更能符合用户的关注程度。但当属性较多时，属性的综合重度变得很小且非常接近，误差会增大，因而就有了上面公式的第二部分 $1/\max(I'(a_j))$，其中 $\max(I'(a_j))$为属性综合重要度的最大值，目的是为了使属性综合重要度最大者扩大为 1，其它的属性重要度也相应地进

行扩大。前面举例的属性重度经第二部分处理后为(1,1,0.457)和(1,1,0.381)。相应地我们改式(1)为:

$$IMP(A)=\sum_{a_i\in Aattr} I''(a_i)/len(A) \quad (2)$$

显然,式(2)定义的模式综合重要度更为合理。

3.2 基于模式综合重要度的最小支持度

定义 4 设用户给定的最小支持度为 s_0,对于模式 A 的支持度定义为:

$$S_0=s_0/IMP(A) \quad (3)$$

由上定义可知,如果模式的重要度越大,则对应的最小支持度就越小,综合重要度最大的模式,其支持度即是 s_0,也就是说我们可以得到更多感兴趣的或可靠的规则;而对于重要度小的模式,其对应的最小支持度就越大(都大于 s_0),也就是说对于关注程度比较小的模式,只有当该模式的频繁程度达到较大的值时,才考虑该模式。

4 剪去无意义的频繁项集

在实验中我们发现经过上面的处理后,结果还不尽如人意。主要是因为当属性值较少时,存在很多偶然因素使得存在无意义但支持度很高的频繁项集。如在实验中,属性 I1 的属性值有两个即 1 和 2,其重要度为 0.1,属性 $I7$,$I8$,$I9$,$I10$,$I11$,$I12$ 的重要度均为 0.6,且 $I12$ 的属性值也有两个 1 和 2,从重要度的设置我们可以看出用户根据领域知识可以知道属性 I1 和属性 I12 之间并没有太大的关系,而用户非常关注属性 7,8,9,10,11,12 之间的关系。但由于属性 1 和属性 12 的属性值都只有 2 个,尽管它们之间没有多大的关系,但在生成频繁项集时,它们只有四种组合,因而出现的频繁度很高,又由于它们之间都是随机关系,四种组合的支持度很接近。反过来看如 $\sup(I1=1,I12=1)=0.321$,$\sup(I1=1,I12=2)=0.32$,从这两个频繁项集属性值和支持度来看,它们毫无意义,但支持度却很高,为此我们要删除这些毫无意义的频繁项集。

在等长度的频繁项集中,(1)若某个属性的属性值都存在于不同的频繁项集中,且除该属性外其它的项都一样,则我们考虑这几个频繁项集的支持度,若它们支持度非常接近,它们两两之间的差值都处于某个给定值 ε 内,则判别这几个频繁项集都为无意义的,是随机的,将它们一并删除;若它们之间的差值不都在给定值 ε 内,则取每个频繁项集的支持度与这几个频繁项集中支持度最大者进行比较,若差值大于给定值 ε,则将其删除。(2)否则,也就是说在生成频繁项集时,该属性已有属性值因不是频繁项而被去除了,那么我们只将每一频繁项集对应的支持度与这几项中最大的频繁项进行比较,若差值大于给定值 ε,则将支持度小的删除。例如,对于均只有两个属性值 1,2 的属性 $I1$ 和 $I12$,若取定 ε 为 0.002,则频繁项($I1=1,I12=1$)支持度为 0.321 和($I1=1,I12=2$)支持度为 0.32 均被删除,若取 ε 为 0.0005,则频繁项($I1=1,I12=2$)被删除。

但与前面讨论的一样,拥有不同的属性值的属性在挖掘频繁项时它们的支持度是不一样的,因此如果不加以区分的话,则无论怎么取 ε 的值,总是对部分频繁项集是不合理(过大或过小)的。因此,仿照前面处理问题的方法,仍旧从属性的重要度和属性的个数两个方面来考虑。根据不同模式的综合重要度,定义不同的 ε。

设给定的最小值为 ε',则 $\varepsilon=\varepsilon'(1-IMP(A))$

5 过程

1. 对于给定的信息系统,把它转化为二进制信息系统。根据属性值的情况,把每个属性拆分成多个属性,如前面的表 1。

2. 把二进制信息系统中对应的属性值为"1"的属性提取出来,转化为交易数据库的形式,见表 3。

从表 3 中可以看出,信息系统中的信息已经完全可以用交易数据库的形式表示。对于交易数据库,有很多成熟的关联规则的挖掘方法。在下一步的处理中,我们用 Aproiri 算法来实现二进制信息系统中的属性之间的关联规则挖掘。

3. 对转化成的交易数据库的形式,用挖掘关联规则的方法进行规则挖掘。

然而,对于不同的模式,由于根据属性的重要性得到的最小支持度并不相同,用 Aproiri 算法不能直接达到我们所要求的目的,但我们在 Aproiri 算法的基础上多加一步操作即可。下面给出本文的提出的方法可以用 Aproiri 算法来挖掘的定理和证明。

定理 1 任何模式的支持度均大于或等于重要度最大的属性的支持度。

证明:由模式的重要性的定义知,模式的重要性是单个属性重要性的平均值,所以任何模式的重要性一定小于或等于单个属性重要性中的最大值。又由模式支持度的定义知,模式的支持度与模式的重要度成反比,也就是说任何模式的最小支持度一定大于等于重要性最大的属性的支持度。证毕。

由以上定理 1 知,在挖掘频繁项集时,可以通过以下两步来实现:

- 根据重要度最大的属性的最小支持度,用 Aproiri 算法挖掘出频繁项集。
- 在生成的频繁项集中,用对应于属性重要性的最小支持度对频繁项集进行过滤,留下基于属性重要性的频繁项集。

Aproiri 算法是一个经典的算法,在此就不给出具体的算法了。

4. 对上面产生的频繁项集进行剪枝,除去无意义的频繁项集。下面给出基本算法:

```
输入:频繁项集 //因处理需要,用数组链表进行存储
输出:用户感兴趣的频繁项集
procedure Prune
{for(i=2;i<=numfreq;i++) //numfreq 为输入频繁项集中最大的频繁项数
{读入 i-频繁项集,并存入数组链表中,依次对每一个结点 do
for(j=i; j>0; j--)
{读取该结点的数组,遍历 i-频繁项集,查找与 j 项属同一属性且除 j 项以外都相同的频繁项集,读取它们的支持度;
if(第 j 个值所属性的值都出现)
  {if (其中最大的支持度与其它所有的支持度的差值<=ε)
     {删除该组所有的频繁项集;}
   else{删除差值大于 ε 的具有支持度较小的频繁项;}
  }//endif
else {删除差值大于 ε 的较小频繁项集}
}//endfor
}//endfor
}//end
```

6 实验

我们的实验数据库是寄生虫感染数据库中的数据。包括 12 条属性:性别(0,1;0.1),民族(0,1,2,3;0.2),职业(0,1,2,3,4,5,6;0.3),居住地形(0,1,2,3;0.4),土壤情况(0,1,2;0.1),施肥情况(0,1,2,3;0.2),钩虫(0,1,2,3,4,5;0.6),鞭

(下转第 153 页)

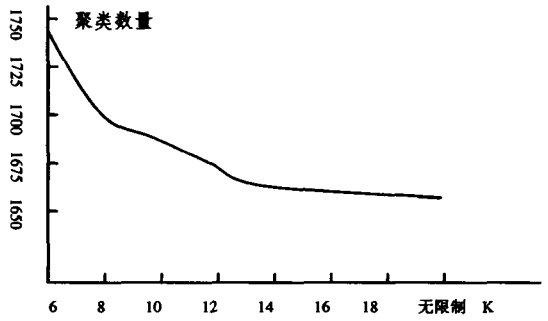

图 7 参数 K 对聚类数量的影响

由表 6 及图 7 可以看出，当 K 不太小时，使用该策略基本不影响聚类的效果。

将序列聚类之后，成功地在单机和网格系统中完成拼接，使用两个计算节点耗时 28 分钟拼接完成。同样的数据集，GiSA 系统执行时间超过 3 小时。而未经分组的原始数据在拼接过程中，由于内存不足而出错退出。

小结及未来工作 本文提出了一种基于最大频繁序列模式的聚类算法，能较准确地分组序列，同时给出了挖掘最大频繁序列模式的高效算法。在此基础上，我们实现了一个基因拼接网格系统，扩展计算能力，使序列聚类能够独立并行地得到处理。

未来的工作包括：基于硬盘的挖掘和聚类算法、挖掘模糊的最大频繁序列模式的算法以及将现有成果应用到日本血吸虫的基因组拼接。

参 考 文 献

1 http://www.phrap.org
2 http://www.ncbi.nlm.nih.gov/blast/
3 Wang J, Wang J, Yang HM, et al. RePS A: Sequence Assembler That Masks Exact Repeats Identified from the shotgun Data. Genome Research, 2002, 12: 824～831
4 Tang J, Huang D, Wang C, et al. GiSA: A Grid System for Genome Sequences Assembly(Industrial Full Paper). In: Proc. of 23th Intl. Conf. on Conceptual Modeling(ER'04), 2004
5 Jian P, Han JW, Morta-zavi-Asl B, et al. Mining Sequential Patterns by Prefix-Projected Growth. ICDE, 2001. 215～224
6 Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1998
7 OGSA(Open Grid Services Architecture) Documents. http://www.globus.org/ogsa
8 Globus: Research in Resource Management. http:// www.globus.org/research/
9 Foster I, Kesselman C. The globus project: A status report. In: Proc. The Heterogeneous Computing Workshop, 1998. 4～18
10 Mullikin J C, Ning Z. The Phusion Assembler. Genome Research, 2003, 13(1): 81～90
11 Wang JY, Han JW. BIDE: Efficient Mining of Frequent Closed Sequences. In: 20 Intl. Conf. on Date Engineering

(上接第 134 页)

虫(0,1,2;0.6)，蛔虫(0,1,2;0.6)，布氏嗜典阿米巴(0,1;0.6)，微小内蜒阿米(0,1;0.6)，巴人酵母菌(0,1;0.6)。属性后括号内的格式为(属性;重要度)，因而属性 1～12 的属性值个数分别为：2,4,7,4,3,4,6,3,3,2,2,2。

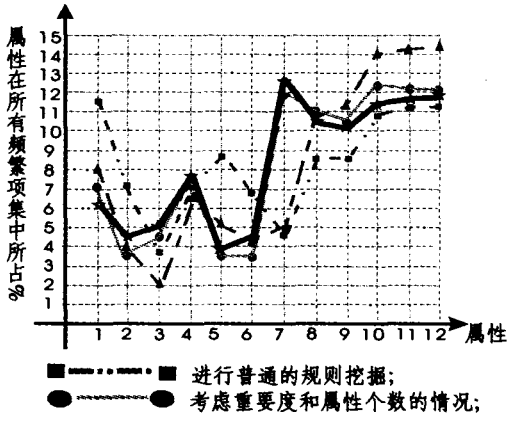

图 1 (supp(2%))

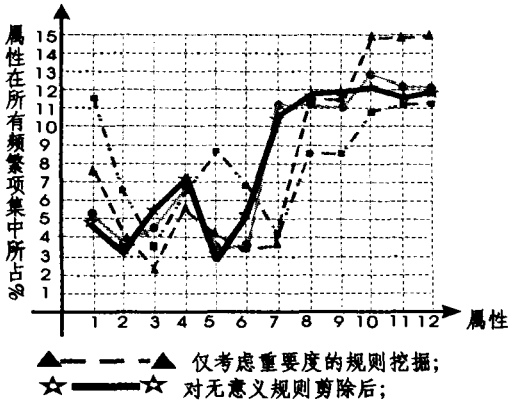

图 2 (supp(3%))

从上面两个图表可以看出，对于不同的支持度有类似的结果：如果不考虑属性重要度，则在挖出的频繁项集中，含属性值较多的属性很少，因而也就很难得到对应属性的规则；如果仅考虑属性重要度，可以看出，对于属性值较少重要度较高的属性可以得到很高的出现率，如属性 10,11,12，但对于属性值较多的属性尽管其重要度高也没有多大作用，如属性 7。因此仅对重要度的设置根本达不到预期的目的；当考虑属性的重要度兼属性个数时，从总体上来说，属性重要度大的属性其在频繁项集中出现的频度也较高。但我们看不出剪除无意义频繁项集的效果，因从百分比上看剪枝前后没有明显的差别。而从我们对 1000 条记录进行处理来看，支持度为 2%和 3%时，频繁项集中项的累计出现次数分别从 2702 下降至 1495 和从 1055 下降至 573。可见在频繁项集中无意义的项集占去了很大一部分，剪除无意义的频繁项集是必要的。

小结 本文开发了信息系统中根据用户的主观重视程度设计的规则挖掘方法，并归结了信息系统中规则发现的特点，提出综合属性值的个数和用户的主观关注程度的模式综合重要度，进而调整不同模式的最小支持度，从而得到合理的频繁项集。并通过实验证明该方法是可行并有效的。但本文只涉及到离散型的属性值，而在现实中很多数据是连续的，这是下一步要进行的工作。

参 考 文 献

1 程继华，郭建生，施鹏飞. 挖掘所关注规则的多策略方法研究. 计算机学报，2000(1)
2 陆建江. 加权关联规则挖掘算法的研究. 计算机研究与发展，2002(10)
3 武鹏程，袁兆山. 混合关联规则及其挖掘算法. 小型微型计算机系统，2003(5)
4 Duntsh I, Gediga G. Simple data filtering in rough set systems. International Journal of Approximate Reasoning, 1998, 18: 93～106