一种模糊对象的极大 co-location 模式挖掘算法

温佛生 肖 清 王丽珍 孔 兵
(云南大学信息学院 昆明 650091)

摘 要 空间 co-location 模式表示的是空间对象的实例在一个相同的区域内频繁地进行空间并置。人们已经对确定和不确定数据 co-location 模式挖掘做了很多工作,也有很多成果,但对极大 co-location 模式挖掘研究较少,特别是针对模糊对象的极大 co-location 模式挖掘研究还未见报道。提出 Mevent-tree 算法来挖掘模糊对象的极大 co-location 模式,首先为每个对象构建空间对象树,从而得到候选模式,然后为候选模式集构建 HUT 树,最后在 HUT 树中从阶数最大的候选模式开始到阶数 2 为止,深度优先搜索极大 co-location 模式并在得到极大模式后对 HUT 树剪枝。接着提出两个改进算法,包括预处理阶段模糊对象的剪枝算法和在构造 HUT 树之前 co-location 候选模式的剪枝算法。最后通过大量实验验证了 Mevent-tree 算法和改进算法的效果和效率。

关键词 模糊对象,极大 co-location 模式挖掘,模糊参与率
中图法分类号 TP311 **文献标识码** A

Algorithm of Mining Maximal Co-location Patterns for Fuzzy Objects

WEN Fo-sheng XIAO Qing WANG Li-zhen KONG Bing
(School of Information Science and Engineering, Yunnan University, Kunming 650091, China)

Abstract A spatial co-location pattern is a group of spatial objects whose instances are frequently located in the same region. There are lots of jobs and achievements of co-location patterns mining algorithms for certain and uncertain data, but less maximal co-location patterns mining algorithms, especially spatial maximal co-location patterns for fuzzy objects. Mevent-tree algorithm was proposed in this paper for mining maximal co-location patterns for fuzzy objects. Firstly, it builds a event tree which can get candidate patterns for each object, build the HUT tree of candidate patterns, and then depth-first searches maximal co-location patterns begining with maximal-size candidate patterns and ending to size-2 candidate patterns in HUT tree, as well pruning co-location candidate patterns after geting maximal co-location patterns. Then we put forward two improved strategies, including the pruning fuzzy objects during preprocessing and the pruning co-location candidates before creating HUT trees. Finally, extensive experiments show the effectiveness and efficiency of Mevent-tree algorithm and its improved methods.

Keywords Fuzzy objects, Maximal co-location pattern mining, Fuzzy participation ratio

1 引言

空间数据挖掘是从空间数据库中挖掘出隐含的知识、空间关系或存储在空间数据库中的其他模式等,空间数据挖掘的研究需要综合空间数据库和数据挖掘技术。空间数据挖掘在地理信息系统(GIS)、图像数据、医学图像处理、交通控制、导航等许多使用空间数据的领域中有着非常广泛的应用。空间 co-location 模式挖掘是空间数据挖掘的一个重要方向,空间 co-location 模式就是一组空间对象的子集,它们的实例在空间中频繁地关联。挖掘空间 co-location 模式就是在空间数据库中挖掘空间对象之间的关联关系。例如,西尼罗河病毒通常发生在蚊子泛滥、饲养家禽的区域;植物学家们发现"半湿润常绿阔叶林"生长的地方 80%会有"兰类"植物生长[1]。

现实世界中的模糊对象无处不在,例如"高山"、"老年人"等。目前模糊对象在许多领域有着十分广泛的应用,但在空间 co-location 模式上的研究还较少。因此,本文研究模糊对象上的极大 co-location 模式挖掘问题有一定的意义。例如:通过对云南三江并流保护区的植物分布数据集的极大 co-location 模式挖掘,植物学家可以找一个相对较好的区域,使得该区域范围虽少,但可以包含更多的植被,这样就可以节省很多的时间、人力以及物力,投入相对更少的成本来进行植物的研究。

本文第 2 节为相关工作;第 3 节为相关定义及性质;第 4、5 节为算法和实验;最后为结论。

到稿日期:2013-05-21 返修日期:2013-06-21 本文受国家自然科学基金资助项目(61063008,61272126,61262069),云南省应用基础研究基金项目(2010CD025),云南省教育厅基金项目(2012C103)资助。

温佛生(1988—),男,硕士生,主要研究方向为空间数据挖掘,E-mail:wenfosheng@126.com;肖 清(1975—),女,硕士生,讲师,主要研究方向为数据挖掘;王丽珍(1962—),女,博士,教授,博士生导师,主要研究方向为数据挖掘、计算机算法等,E-mail:lizhwang2005@126.com(通信作者);孔 兵(1968—),男,博士,副教授,主要研究方向为数据处理、人工智能。

2 相关工作

空间 co-location 模式挖掘问题是空间数据挖掘领域的一个重要研究方向。人们对确定数据的 co-location 模式挖掘问题进行了深入的研究，并提出了很多算法，比如 join-base 算法[2]、partial-join 算法[3]、join-1ess 算法[4]、CPI-tree 算法[5]、order-clique-based 算法[6]等。文献[2]给出了 co-location 模式挖掘相关的一些定义，包括邻近关系、空间 co-location 模式、行实例、表实例、参与率、参与度，以及 co-location 规则和条件概率等。近年来，对不确定数据上的 co-location 模式挖掘的研究也越来越多。文献[7]提出了不确定集上的 UJoin_based 算法。文献[8]研究了从区间数据表示的不确定对象中挖掘 co-location 模式。虽然目前空间 co-location 模式挖掘算法很多，但对象模糊的数据挖掘算法较少，特别是对象模糊的极大 co-location 模式挖掘算法还未见报道。通常，从现实事务集合中产生的频繁项集的数量庞大，为了提高算法挖掘效率，代替产生所有的频繁项集，而仅产生较少的、代表性的、可推导出所有频繁项集的极大频繁项集是非常有用的。模糊数据的研究目前主要集中在模糊对象的建模上，比如基本的类型和操作模型等。文献[10]研究了模糊对象的 K 最近邻(K-nearest neighbor，KNN)查找问题，提出了 AKNN(ad-hoc K-nearest neighbor)和 RKNN (range K-nearest neighbor) 算法。文献[11]研究了模糊对象的空间 co-location 模式挖掘问题。文献[12]研究了实例位置模糊的空间 co-location 模式挖掘。文献[13]研究了模糊对象的关联规则挖掘。

3 相关定义及性质

这一部分主要是介绍相关的定义和性质。首先对模糊对象、模糊概率阈值、空间距离、模糊参与率、模糊参与度、极大 co-location 模式、空间对象树、星型行实例等进行了定义。然后给出了模糊参与率及模糊参与度满足的一些性质。

3.1 相关定义

定义 1(模糊对象[11]) 本文中定义的模糊对象(见图 1)，表示为空间中离散点的集合，定义如下：$A=\{\langle a,\mu(a)\rangle \mid \mu(a)>0\}$，其中 A 表示模糊对象，a 表示实例，$\mu(a)$ 表示的是实例 a 属于模糊对象 A 的隶属度。

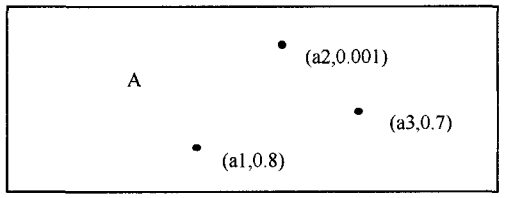


图 1 模糊对象

从图 1 中可以知道，实例 $a1$、$a2$、$a3$ 隶属于模糊对象 A 的隶属度分别为 0.8、0.001 和 0.7，其中实例 $a1$ 隶属于模糊对象 A 的隶属度最高，这是我们感兴趣的对象；而实例 $a2$ 隶属于模糊对象 A 的隶属度非常低，仅为 0.001，这个实例对研究模糊对象的贡献不大，我们希望在数据预处理阶段就先把该数据剔出。

定义 2(模糊度阈值) 模糊度阈值是用户给定的一个用户自定义的概率阈值 $f\_threshold$，集合 $A_{f\_threshold}=\{a \mid \mu(a)\geqslant f\_threshold\}$ 表示满足用户自定义模糊度阈值的实例集。

如图 1 中，$A_{0.7}=\{a1,a3\}$。

定义 3(空间邻近关系 R) 当两个实例间的欧几里德距离小于等于距离阈值 d 即 $d(a,b)\leqslant d$ 时，称这两个实例满足空间邻近关系 R，记为 $R(a,b)$，即它们是邻近的。在图中用线段将满足邻近关系 R 的实例连接起来。其中 a,b 表示两个不同的空间实例。

一个空间 co-location 模式表示的是一组空间对象的集合。co-location 模式的长度称为此 co-location 模式的阶，即 co-location 模式里空间对象的个数。

设有空间实例集 $I=\{i_1,i_2,\cdots,i_l\}$，如果有 $\{R(i_j,i_k) \mid 1\leqslant j\leqslant l,1\leqslant k\leqslant l\}$，则称 I 是一个团(clique)。如果 I 团包含了 co-location 模式 c 中的所有对象，并且 I 没有任何一个子集可以包含 c 中的所有对象，那么 I 是 co-location 模式 c 的一个行实例(称为 co-location 实例)。co-location 模式的所有行实例的集合称为表实例，记为 $table\_intstance(c)$。例如图 2 中，co-location$\{A,B,C\}$的表实例为$\{\{A2,B4,C2\},\{A3,B3,C1\}\}$。

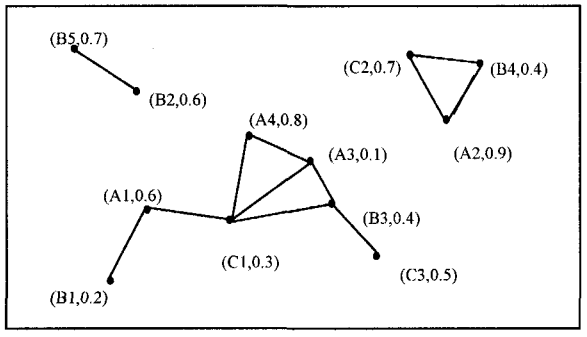


图 2 模糊对象参与度

定义 4(模糊参与率[11]) 设 f_i 为某个空间模糊对象，f_i 在 k 阶的 co-location 模式 c 中的模糊参与率表示为 $FPR(c,f_i)$，它是 f_i 的实例在空间 co-location 模式 c 的表实例中不重复出现的实例的隶属度之和与 f_i 中总实例个数的比率，公式如下：

$$FPR(c,f_i)=\frac{\sum\mu(a)}{|talbe\_instance(\{f_i\})|}$$

其中，$a\in\prod_{f_i}(talbe\_instance(c))$，而 $\prod$ 是关系投影操作。

定义 5(模糊参与度) 模糊对象的空间 co-location 模式 $c=\{f_1,f_2,\cdots,f_k\}$ 的模糊参与度表示为 $FPI(c)$，是模式中所有空间对象 FPR 值中的最小值，公式表示如下：

$$FPI=\min_{i=1}^{k}\{FPR(c,f_i)\}$$

设 min_prev 是用户给定的最小参与度阈值，当 $FPI(c)\geqslant min\_prev$ 时，称模糊对象的 co-location 模式 c 是频繁的。

例 1 如图 2 所示，在模糊对象空间集中有 A,B,C 3 个模糊对象。其中 $A=\{(A1,0.6),(A2,0.9),(A3,0.1),(A4,0.8)\}$；$B=\{(B1,0.2),(B2,0.6),(B3,0.4),(B4,0.4),(B5,0.7)\}$；$C=\{(C1,0.3),(C2,0.7),(C3,0.5)\}$。实例间的邻近关系 R 用实线连接表示。假设最小参与度阈值 min_prev 为 0.15，对 co-location 模式 $c=\{A,B,C\}$，有 $FPR(c,A)=(0.9+0.1)/4=0.25$，$FPR(c,B)=(0.4+0.4)/5=0.16$，$FPR(c,C)=(0.3+0.7)/3=1/3$。那么 $FPI(c)=\min\{FPR(c,A),$

$FPR(c,B)$，$FPR(c,C)\}=0.16$。由于 0.16 大于 0.15，因此可知 co-location 模式 $c=\{A,B,C\}$ 是频繁的。

定义 6（极大 co-location 模式） 如果 co-location 模式 c 是频繁的并且没有频繁的超集，则称 co-location 模式 c 是极大 co-location 模式。

定义 7（邻近实例集） 对于模糊对象集 S 中某个特定的实例 i_1，实例 i_1 的邻近实例集为 $\{i_2,\cdots,i_l\}$，其中实例 $i_2,\cdots,i_l$ 和实例 i_1 都满足空间邻近关系 R，并且 $i_1,i_2,\cdots,i_l$ 属于不同的空间对象。

例如，如图 3 所示，实例 $A1$ 的邻近实例集为 $\{C1,D2,E1\}$。

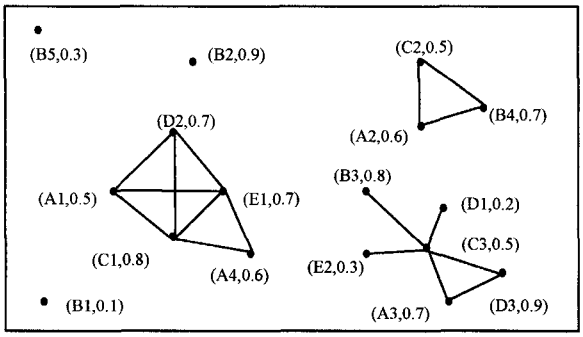


图 3 模糊空间对象集

图 4 描述了把图 3 中的模糊空间对象集转化成相应的邻近实例集和邻近对象集。从邻近实例集得到邻近对象集是轻松的，只需把邻近对象集中的所有实例换成实例所属对象，并把每个邻近对象集中重复的对象合并即可。

InsNO	邻近实例集	
1	A1	C1,D2,E1
2	A2	B4,C2
3	A3	C3,D3
4	A4	C1,E1
5	B1	
6	B2	
7	B3	C3
8	B4	A2,C2
9	B5	
10	C1	A1,A4,D2,E1
11	C2	A2,B4
12	C3	A3,B3,D1,D3,E2
13	D1	C3
14	D2	A1,C1,E1
15	D3	A3,C3
16	E1	A1,A4,C1,D2
17	E2	C3

InsNO	邻近对象集	
1	A	C,D,E
2	A	B,C
3	A	C,D
4	A	C,E
5	B	
6	B	
7	B	C
8	B	A,C
9	B	
10	C	A,D,E
11	C	A,B
12	C	A,B,D,E
13	D	C
14	D	A,C,E
15	D	A,C
16	E	A,C,D
17	E	C

图 4 图 3 对应的邻近实例集和邻近对象集

定义 8（空间对象树） 空间对象树是满足以下条件的树：

1. 参照对象作为根结点；

2. 每个邻近对象邻近集作为一个分支；子树中的结点必须与根结点满足邻近关系 R；

3. 对于各个分支，若能共享路径则共享，并且各结点记录共享次数。

例如，如图 5 所示，在对象 C 的邻近对象集中，对象 C 为参照对象，作为空间对象树的根结点。邻近对象集 A,D,E 作为第一个分支，A,B 作为第二个分支，A,B,D,E 作为第三个分支，由于 A,B 和 A,B,D,E 可以共享路径，最终构造的对象 C 的空间树如图 5 所示。

我们可以为每个空间对象创建对应的空间对象树。通过空间对象树集，利用 FP-Growth 算法[14] 可以得到候选模式集。图 6 中描述了从空间对象树集得到星型候选模式集（由于这些候选模式以某个对象为中心，模式中的其他对象都以中心对象为参考，因此形象地称这些模式为星型候选模式），从而筛选出 co-location 候选模式集的过程。

INS NO	Items（邻近对象）	
10	C	A, D, E
11	C	A, B
12	C	A, B, D, E

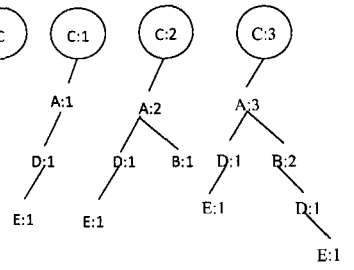
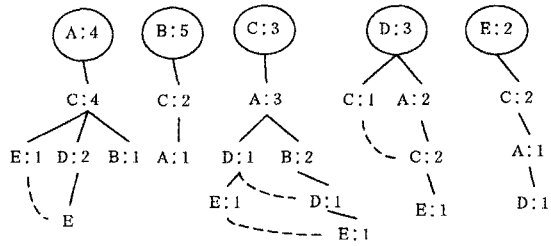


图 5 对象 C 的空间对象树



(a) 各空间对象对应的空间对象树

{A,E}:2/4	{B,A}:1/5	{C,E}:2/3	{D,C}:3/3	{E,D}:1/2
{A,D}:2/4	{B,C}:2/5	{C,D}:2/3	{D,E}:1/3	{E,A}:1/2
…	{B,A,C}:1/5	…	…	…
{A,B,C}:1/4		{C,A,B}:2/3	{D,A,C}:2/3	{E,A,D}:1/2
{A,C,D}:2/4		{C,B,E}:1/3	{D,A,E}:1/3	{E,C,D}:1/2
…		…	…	…
{A,C,D,E}:2/4		{C,A,D,E}:2/3	{D,A,C,E}:1/3	{E,A,C,D}

(b) 图(a)中各对象树对应的星型候选集

{A,E}:{2/4,1/2}:1/2

{D,E}:{1/3,1/2}:1/3…

{A,B,C}:{1/4,1/5,2/3}:1/5

{A,C,D}:{2/4,2/3,2/3}:1/2

…

{A,C,D,E}:{2/4,2/3,1/3,1/2}:1/2

(c)从图(b)中筛选出的 co-location 候选模式

图 6 co-location 候选模式的生产过程

星型候选模式并不是真正的 co-location 候选模式，必须通过再次的筛选后才能判断一个星型候选模式是否是 co-location 候选模式。以下方法可以判断出一个星型候选模式是否是一个 co-location 候选模式。

如图 6 所示，星型候选模式 $\{A,B,C\}$ 是一个 co-location 候选模式。这是因为在邻近对象集中存在 $\{A,B,C\}$、$\{B,A,C\}$ 和 $\{C,A,B\}$。同理，可以知道星型候选模式 $\{C,B,E\}$ 不是 co-location 候选模式。因为邻近对象集中虽然存在 $\{C,B,E\}$，但是不存在 $\{B,C,E\}$，所以星型候选模式 $\{C,B,E\}$ 不是 co-location 候选模式。

定义 9(星型行实例) 设 $I=\{I_1,\cdots,I_k\}$是一组空间实例集,$I\subseteq T_{I_1}$(T_{I_1}为参考实例 I_1 的实例邻近事务),I 中实例所属的对象为 co-location 候选模式 $C=\{O_1,\cdots,O_k\}$,则称实例集是候选模式 $C=\{O_1,\cdots,O_k\}$的星型行实例。

从邻近实例集中可以得到每个候选模式的星型实例集。图 7 描述了从邻近实例集得到候选模式的星型实例集的过程。以产生 co-location 候选模式$\{C,D,E\}$的星型实例集为例。从图中可以得出,co-location 候选模式$\{C,D,E\}$有 3 个星型实例,分别为$\{C1,D2,E1\}$,$\{C3,D1,E2\}$,$\{C3,D3,E2\}$。其中星型实例$\{C1,D2,E1\}$中的子实例$\{D2,E1\}$存在于模式$\{D,E\}$的实例中,即星型实例$\{C1,D2,E1\}$是 co-location 候选模式$\{C,D,E\}$的行实例。而星型实例$\{C3,D1,E2\}$和$\{C3,D3,E2\}$的子实例$\{D2,E1\}$,$\{D1,E2\}$不存在于模式$\{D,E\}$的实例集中,即星型实例$\{C3,D1,E2\}$,$\{C3,D3,E2\}$不是模式$\{C,D,E\}$的行实例。

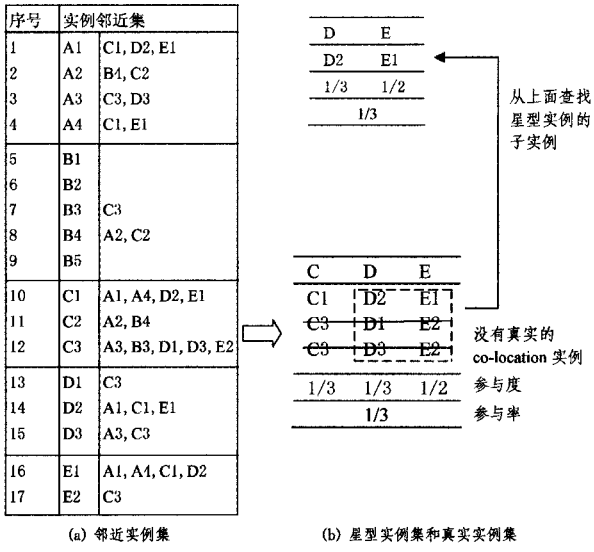


图 7 从星型实例集中产生 co-location 实例集

3.2 性质和定理

引理 1 模糊参与率(FPR)和模糊参与度(FPI)随着 co-location 模式阶的增大单调递减。

证明:假设某个模糊空间对象的实例包含在 co-location 模式 c 的实例中,那么当有 co-location 模式 $c1\subseteq c$ 时,这个模糊空间对象的实例也一定包含在模式 $c1$ 的实例中,反之则不然,所以模糊参与率是单调递减的。模式的模糊参与度取它包含的对象中的最小参与率值,当模式的阶增大时,由于模糊参与率是递减的,因此 co-location 模式的模糊参与度也是单调递减的。

定理 1 如果 k 阶 co-location 模式 c 是频繁的,那么它的任意 $k-1$ 阶的子 co-location 模式也是频繁的。

证明:根据引理 1,k 阶 co-location 模式 c 的模糊参与度要小于其 $k-1$ 阶子 co-location 模式,所以若 k 阶 co-location 模式 c 是频繁的,则 $k-1$ 阶的子 co-location 模式肯定也是频繁的。

定理 2 极大 co-location 模式的子模式不是极大 co-location 模式。

证明:假设极大 co-location 模式 $c=\{A,B,C\}$,其子模式 $c1=\{A,B\}$。若 $c1$ 也是极大 co-location 模式而 $c1$ 是 c 的子模式,这与定义 6 矛盾。因此,极大 co-location 模式的子模式不是极大 co-location 模式。

4 模糊对象极大 co-location 模式挖掘算法

首先给出一个模糊对象的极大 co-location 模式挖掘的基本算法——Mevent-tree 算法。接着提出了预处理阶段模糊对象的剪枝、在构造 HUT 树[15](head union tail)之前对 co-location 候选模式剪枝的改进算法。

4.1 Mevent-tree 算法

4.1.1 算法基本思想

根据极大 co-location 模式的定义,阶数最大的 co-location 模式很有可能成为极大 co-location 模式。基于这种思想,本文中的算法从阶数最大的 co-location 候选模式开始挖掘极大 co-location 模式,直到阶数为 2 的模式为止。

4.1.2 Mevent-tree 算法

输入:SF:空间模糊对象集,SFI:空间实例集,min_prev:参与度阈值,dis_threshold:距离阈值,f_threshold:模糊度阈值,F,FI:满足模糊度阈值条件的模糊对象与实例的集合(其中 $F=\{O_1,\cdots,O_n\}$)

输出:FP:极大 co-location 规则集

变量:ST:模糊对象邻近关系集,Treei:模糊对象 O_i 的空间对象树,l:co-location 模式的阶,C:候选模式集,Cl:阶为 l 的候选模式,pi:候选模式的真实参与度,lmax:候选模式集中阶的最大值,CIc:co-location 候选模式 c 的真实实例集,SIc:co-location 候选模式 c 的星型实例集,SIl:阶为 l 的 co-location 候选模式的星型实例集,R:极大的 co-location 模式,Rl:阶为 l 的极大 co-location 模式,HUT:HUT 树

步骤:

1. F,FI=gen_fdata(SF,SFI,f_threshold)
2. ST=gen_nei_tra(F,dis_thres-hold);
3. Treei[m]=build_ev_tre(O_i,ST,min_prev);
4. C =gen_candidates(Tree1,…,Treem);
5. HUT=buid_HUT(C);
6. for(l=lmax;l≥2 or Cl≠ϕ;l--)
   6.1. Cl=get_l_candidates(C,l);
   6.2. SIl=find_star _instances(Cl,ST);
   6.3. for each candidate c in Cl
      6.3.1. CIc=find_clique_instances(SIl,c);
      6.3.2. pi=calculate_pi(CIc);
   6.4. R=R∪Rl ; C=C−Cl ;
   6.5 SubsetPruning(Rl,C);

步骤(1)是根据文中定义的模糊度阈值得到满足条件的模糊对象和实例集;步骤(2)得到邻近实例集和邻近对象集;步骤(3)为每个空间对象建立空间对象树;步骤(4)从空间对象树中得到每个对象相应的星型候选模式集并且筛选得到 co-location 候选模式集;步骤(5)为 co-location 候选模式集构造 HUT 树;步骤(6)从阶最大的 co-location 候选模式集开始到阶为 2 的模式集为止,循环执行(6.1−6.2)寻找 co-location 模式的星型实例;(6.3)寻找 co-location 真实实例并计算参与度;(6.4)把满足大于参与度阈值的 co-location 模式放入结果集;(6.5)通过定理 2 对候选模式剪枝。

4.1.3 算法示例

从图 3 模糊对象集中得到如图 4 所示的邻近实例集合邻

近对象集。按照图5,为每个模糊对象构造对象空间树,并按照空间对象树生成每个模糊对象的星型候选集,通过进一步筛选得到 co-location 候选模式。

接下来把 co-location 候选模式构造成 HUT 树。构造方法如下:HUT 树以 NULL 为根,它由按照字典序排列的候选模式构成,树中的每个分支为一个候选模式,若候选模式有相同的前缀,则共享一条路径。例如,对于 co-location 候选模式集 $S=\{\{A,B,C\},\{A,C,D,E\},\{A,C,E\},\{A,D,E\},\{A,E\},\{B,C\},\{C,D,E\},\{C,E\},\{D,E\},\{E\}\}$,构造的 HUT 树如图8所示。以图中第二层中的节点 A 为 head,则 HUT 是 $\{A,B,C,D,E\}$。以 B 为 head,则 HUT 为 $\{B,C\}$。

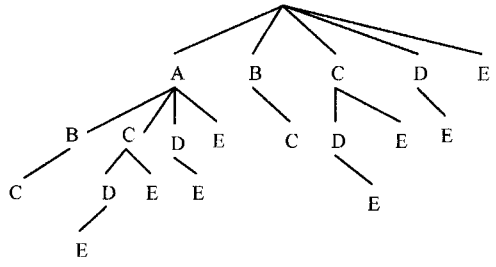


图8 co-location 候选模式的 HUT 树

算法首先深度优先搜索 HUT 树,从中找出阶数最大的 co-location 候选模式,本例中为 $\{A,C,D,E\}$。再计算阶数最大的模式的真实参与度,若阶数最大的模式是频繁的,则该模式就是极大 co-location 模式。假设 $\{A,C,D,E\}$ 是频繁的,即它是极大 co-location 模式。下一步按照定理2对 HUT 树广度优先搜索进行剪枝。剪枝过程如图9所示。对 HUT 树的第三层来说,对于模式 $\{A,B\}$,由于 $\{A,B\}$ 不是 $\{A,C,D,E\}$ 的子模式,因此不剪枝。对于模式 $\{A,C\}$,它由于是 $\{A,C,D,E\}$ 的子模式,因此被剪枝。同理,模式 $\{A,D\}$,$\{A,E\}$,$\{C,D\}$,$\{C,E\}$,$\{D,E\}$,$\{E\}$ 被剪枝。同理,最终得到剪枝后的 HUT 树。从图9中可以得出阶为4的极大 co-location 模式为 $\{A,C,D,E\}$,且 co-location 候选模式集中只剩下了 $\{A,B,C\}$ 和 $\{B,C\}$,下一步是要从 HUT 树中深度搜索找到阶为3的 co-location 模式。本例中只有 $\{A,B,C\}$,若 $\{A,B,C\}$ 是频繁的,则 $\{A,B,C\}$ 是极大 co-location 模式,模式 $\{B,C\}$ 被剪枝。若 $\{A,B,C\}$ 不频繁,那么候选模式集中只剩下 $\{B,C\}$。同理,可以计算出 $\{B,C\}$ 的参与度,从而判定 $\{B,C\}$ 是否是极大 co-location 模式。

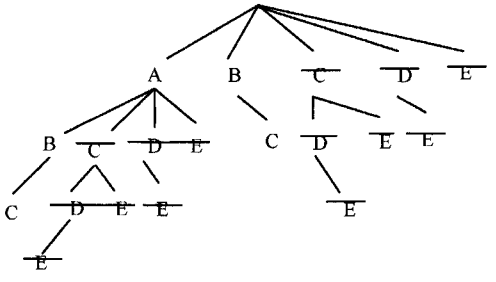


图9 剪枝后的 HUT 树

4.2 预处理过程的模糊对象剪枝

极大 co-location 模式挖掘需要对模糊对象的大量实例间的距离进行计算,而且从星型实例到真实实例需要大量的比对。若能够在数据预处理阶段就把那些不可能存在于极大 co-location 模式的对象剔除掉,将能在一定程度上减少算法的时间复杂度和空间复杂度。基于上述考虑,本文提出了一种有效的剪枝算法。

定理3 对于一个模糊对象 A,若它最大模糊参与率值小于给定的最小参与度阈值,则对象 A 不可能存在于任意的频繁 co-location 模式中。

证明:(反证法)假设模糊对象 A 存在于某个频繁的 co-location 模式 c 中,则我们可以得到 $FPR(c,A)\geqslant min\_prev$。模糊对象 A 在 c 中的最大参与率为它的所有实例均在 co-location 模式 c 的表实例中。根据模糊参与率的定义,最大模糊参与率值等于模糊对象 A 的所有实例的隶属度之和与对象 A 的实例数目的比率,由定理条件可知,它小于给定的最小支持度阈值,这时可以得到 $FPR(c,A)<min\_prev$,与假设矛盾,所以对象 A 不可能存在于任意的 co-location 模式中。

4.2.1 算法思想

在数据预处理阶段,依据定理3,计算每个空间模糊对象的最大模糊参与率,然后和给定的最小参与度阈值比较,把不满足条件的模糊对象剔除。

4.2.2 剪枝模糊对象算法

输入:F:模糊对象集;FI:实例集;min_prev:参与度阈值;T_PR:对象的最大参与率

输出:剪枝模糊对象后的 F,FI

步骤:

1. for all fuzzy object f∈F do
   1.1. 计算 f 的最大参与率 T_PR
   1.2. if T_PR<min_prev then
      1.2.1. F=F−{f};FI=FI−{a}(其中 a∈f);
2. return F,FI

4.2.3 算法示例

如图2中,设 $min\_prev=0.6$,假设模糊对象 C 的所有实例均在 co-location 模式的行实例中,B 的最大参与率值等于 $(0.7+0.6+0.2+0.4+0.4)/5=0.46<0.6$,由定理2可知,$B$ 不可能存在于任意的 co-location 模式中,这时把对象 B 剪枝掉。

4.3 在构造 HUT 树之前的 co-location 候选模式剪枝算法

按照 Mevent-tree 算法,当从星型候选模式得到了 co-location 候选模式,从星型实例得到了 co-location 候选模式的真实实例之后,要构建 HUT 来挖掘极大 co-location 模式。若能在构建 HUT 树之前就把一些不可能成为极大 co-location 模式的候选模式剪枝,将有利于降低算法的时间和空间复杂度。基于这种考虑,本文提出了在构造 HUT 树之前的 co-location 候选模式剪枝算法。

定理4 在 co-location 模式 c 中,假设模糊对象 $A\subseteq c$,如果 A 在 c 的表实例中的实例满足 $\max\{\mu(a)\}<min\_prev$,其中 a 是对象 A 的实例,则 co-location 模式 c 可以被剪枝掉。

证明:因为

$$FPR(c,A)=\frac{\sum_{i=1}^{n}\mu(a_i)}{|table\_instance(A)|}\leqslant\frac{n*\max\{\mu(a_i)\}}{|table\_instance(A)|}$$

$$=\frac{n*\max\{\mu(a_i)\}}{n}=\max\{\mu(a_i)\}$$

所以当 $\max\{\mu(a_i)\}<min\_prev$ 时,co-location 模式可以被剪枝(假设模糊对象 A 的实例数是 n)。

4.3.1 算法思想

从星型候选模式、星型实例得到 co-location 候选模式和 co-location 真实实例后，可以计算出候选模式中每个对象的最大模糊参与率。按照定理 4，可以把不可能成为极大 co-location 模式的候选模式剪枝。

4.3.2 在构造 HUT 树之前的 co-location 候选模式剪枝算法

输入：C'：co-location 候选模式集

CI'：co-location 候选模式实例集

Min_prev：最小参与度阈值

输出：C：有可能成为极大 co-location 模式的候选模式集

变量：c：计算过程中用来存放临时 co-location 候选模式

步骤：

1. for each co-location c candidate in do

1.1. get_instances(C',CI');

1.2. max{μ(a)}=calculate_u(c);

1.3. if max{μ(a)}<min_prev then

1.3.1. $C'=C'-c$;

2. End do

3. $C=C'$;

4. return C;

4.3.3 算法示例

如图 2 中，考虑 2 阶 co-location 模式 $c=\{B,C\}$，每个对象的实例序按隶属度非递增进行排序，假设 $min\_prev=0.6$，对象 B 与对象 C 实例的 R 关系为$((B3,0.4),(C1,0.3))$，$((B3,0.4),(C3,0.5)),((B4,0.4),(C2,0.7))$。其中 $B3$ 和 $B4$ 的模糊隶属度同为最大 0.4。由于 $0.4<min\_prev$，因此模式$\{B,C\}$可以被剪枝。

5 实验与分析

本节做了大量实验来验证 Mevent-tree 算法和改进算法的有效性，并将文中提出的算法与传统算法的挖掘结果进行了实验比较。所有算法均采用 C# 编写，并在 Intel(R) Core (TM) Duo T9600 2.8GHz cpu 和 4 GB memory 的计算机上运行。实验中采用“云南三江并流保护区”的植物分布数据集。每一种植物都有它自己的特征，但从图上来对植物种类进行判断时，很难百分之百确定一株植物所属的种类，这时我们用模糊隶属度来表示植物所属种类。模糊对象数目为 10，模糊隶属度值为 0 到 1，是由植物学家给出的。数据分布在 1000 * 1000 的空间里。

5.1 Mevent-tree 算法与其改进算法的性能比较

在这一小节中，本文将对模糊对象极大 co-location 挖掘算法(Mevent-tree)与其改进算法进行比较，改进算法包括预处理阶段模糊对象的剪枝算法(PO)和在 PO 基础上的在构造 HUT 树之前对 co-location 候选模式的剪枝算法(PO_RI)。

5.1.1 实例数目对算法的影响

参与度阈值 $min\_prev=0.1$，距离阈值 $dis\_threshold=100$，模糊度阈值 $f\_threshold=0$，实例数目从 2500 增加到 50000。从图 10 可以看出，3 种算法的执行时间随着实例数目的增加而增加。因为实例数增加了，那么得到的星型实例就会增加，从星型实例筛选出真实实例的时间也会增加。由于 PO 和 PO_RI 算法能够在算法的预处理阶段就把不满足条件的对象剪枝了，即对象的实例也被剔除，因此算法的执行时间上升幅度较小。

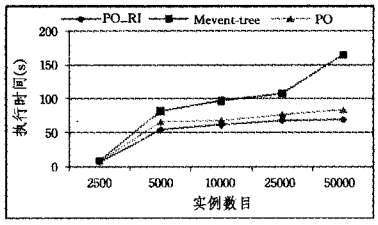


图 10 实例数目对算法的影响

5.1.2 参与度阈值对算法的影响

实例数为 12000，距离阈值 $dis\_threshold=100$，模糊度阈值 $f\_threshold=0$，参与度阈值 min_prev 从 0.15 变化到 0.3。从图 11 可以看出，3 种算法随着参与度阈值的增大，算法执行时间都变小。因为参与度阈值越大则满足条件的 co-location 候选模式就越少，算法执行时间就会越少。随着参与度阈值的增大，3 种算法的执行时间越来越接近，这是因为预处理阶段模糊对象的剪枝算法(PO)是依据参与度阈值的，当参与度阈值小时，被剪枝的模糊对象就很少，因此 3 种算法执行时间越来越接近。但随着参与度阈值的增大，PO 算法和 PO_RI 算法能够剪枝掉更多的模糊对象，因此执行时间会要比 Mevent-tree 算法有较明显的减少。

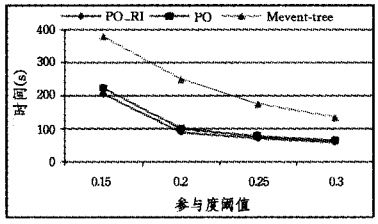


图 11 参与度阈值对算法的影响

5.1.3 距离阈值对算法的影响

实例数为 12000，模糊度阈值 $f\_threshold=0$，参与度阈值 $min\_prev=0.1$，距离阈值 $dis\_threshold$ 在 50 到 150 区间变化。如图 12 所示，3 种算法执行时间都随着距离阈值的增加而增加。随着距离阈值的增加，有更多的实例满足邻近关系，产生的星型实例会增加，从而执行时间增加。但距离阈值大于 100 后，Mevent-tree 算法的执行时间剧烈增长，而 PO 和 PO_RI 算法由于使用了模糊对象剪枝，使得算法执行时间缓慢地增长。

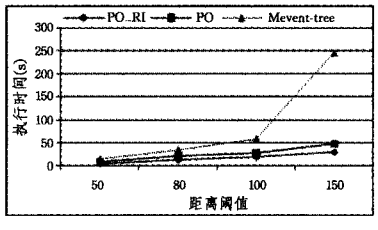


图 12 距离阈值对算法的影响

5.1.4 模糊度阈值对算法的影响

实例数为 12000，参与度阈值 $min\_pre=0.1$，距离阈值 $dis\_threshold=100$，模糊度阈值 $f\_threshold$ 在 0 变化到 0.5。如图 13 所示，当模糊度从 0 到 0.2 时，3 种算法的运行时间均上升，而在 0.2 以后，3 种算法的运行时间又开始下降，这是因为把模糊对象的那些不满足模糊度阈值的、具有低模糊度的实例剪去后，剪枝掉实例的对象的参与率值会增大，这时

满足参与度阈值条件的模式数目会增多，使得算法运行时间上升。但是随着模糊度阈值的不断增大，要剪去对象的实例也越多，这就意味着一个对象具有的实例数目越来越少，就会减少模式产生的数目，使得算法的运行时间下降。

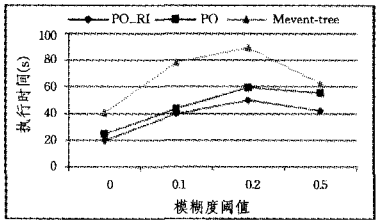


图 13　模糊度阈值对算法的影响

5.2　模糊对象算法与传统算法的比较

在这一节，将本文 PO_RI 算法与传统挖掘算法的结果进行实验比较。这里传统算法采用的是基于 co-location 挖掘算法中经典的 join_based 算法[2] 的 GeneralColoc＋Maximal 算法[15]。

5.2.1　模糊对象算法与传统算法产生候选 co-location 模式比较

实例数为 12000，参与度阈值 *min_pre*＝0.1，距离阈值 *dis_threshold*＝80，模糊度阈值 *f_threshold* 为 0。如图 14 所示，候选模式的阶从 2 到 7，PO_RI 算法所产生的候选模式都要少于 GenralColoc＋Maximal 算法，这是由于 PO_RI 算法在预处理时进行了模糊对象剪枝，在构造 HUT 树之前也对候选模式剪枝。而最终两种算法得出的极大 co-location 模式一样，即 PO_RI 算法的效率要比 GeneralColoc＋Maximal 高。

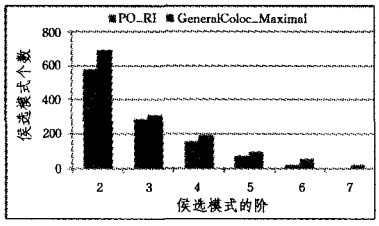


图 14　两种算法得到的候选 co-location 模式数量

5.2.2　参与度阈值对算法的影响

实例数为 12000，距离阈值 *dis_threshold*＝80，模糊度阈值 *f_threshold*＝0，参与度阈值 *min_prev* 从 0.15 变化到 0.3。从图 15 能够看出，两种算法的执行时间都随参与度阈值的增加而减少。相同参与度阈值条件下 PO_RI 算法执行时间要比 GeneralColoc＋Maximal 算法少。

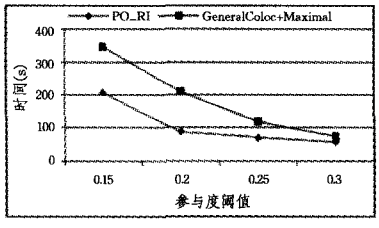


图 15　参与度对算法的影响

5.2.3　距离阈值对算法的影响

实例数为 12000，模糊度阈值 *f_threshold*＝0，参与度阈值 *min_pre*＝0.1，距离阈值 *dis_threshold* 在 50 到 150 区间变化。如图 16 所示，两种算法随着距离阈值的增大，算法的执行时间也增大。距离阈值为 50 到 80 区间时，两种算法的执行时间基本相同。距离阈值从 80 开始，GeneralColoc ＋Maximal 算法执行时间急剧上升，而 PO_RI 算法执行时间较缓慢增加，因为 PO_RI 算法有模糊对象剪枝和候选 co-location 模式剪枝。

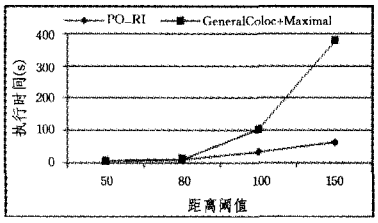


图 16　距离阈值对算法的影响

5.2.4　实例数对算法的影响

参与度阈值 *min_prev*＝0.1，距离阈值 *dis_threshold*＝80，模糊度阈值 *f_threshold*＝0，实例数目从 2500 增加到 47000。如图 17 所示，两种算法的执行时间随实例数的增加而增加。实例数从 5000 开始，可以发现 PO_RI 由于有剪枝，算法执行时间随实例数的增加平缓地增长，而 GeneralColoc ＋Maximal 算法执行时间的增长速度较 PO_RI 快很多。

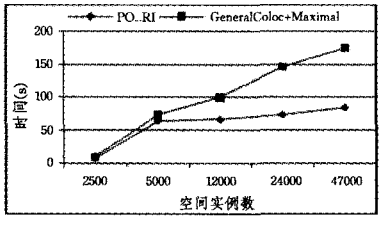


图 17　空间实例数对算法的影响

结束语　尽管空间 co-location 模式挖掘是一种非常有价值的空间挖掘，而且模糊对象也经常出现在许多重要的应用中，但是目前对于模糊对象的极大 co-location 模式挖掘的研究较少。本文针对模糊对象的空间极大 co-location 模式挖掘问题，提出了一种基本挖掘算法——Mevent-tree 算法。为了提高算法的挖掘效率，文中提出了 2 种改进算法，即预处理阶段模糊对象的剪枝、在构造 HUT 树之前对 co-location 候选模式的剪枝。通过大量的实验表明，本文提出的算法及改进算法是非常有效的。下一步的工作将在此基础上，考虑模糊度阈值在一个范围内变化时的空间极大 co-location 模式挖掘问题。

参 考 文 献

[1]　王丽珍，周丽华，陈红梅，等. 数据仓库与数据挖掘原理及应用(第二版)[M]. 北京：科学出版社，2009

[2]　Huang Y, Shekhar S, Xiong H. A general approach Discovering colocation patterns from spatial data sets[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(12): 1472-1485

[3]　Yoo J S, Shekhar S. A partial join approach for mining co-location patterns[C]// ACM International Symposium on Advances in Geographic Information Systems (ACMGIS). Washington, USA, 2004: 241-249

[4]　Yoo J S, Shekhar S, Celik M. A join-less approach for co-location pattern mining[J]. A Summary of Results IEEE International Conference on Data Mining(ICDM), Houston, USA, 2005: 813-816

[5] Wang Li-zhen, Bao Yu-zhen, Lu J, et al. A new join-less approach for co-location pattern mining[C]// Proceedings of the IEEE 8th International Conference on Computer and Information Technology (CIT 2008). Sydney, Australia, 2008: 197-202

[6] Wang Li-zhen, Zhou Li-hua, Lu J, et al. An order-clique-based approach for mining maximal co-locations[J]. Information Sciences, 2009, 179(19): 3370-3382

[7] 陆叶,王丽珍,张晓峰. 从不确定数据集中挖掘频繁 Co-location 模式[J]. 计算机科学与探索, 2009, 3(6): 656-664

[8] Wang Li-zhen, Chen Hong-mei, Zhao Li-hong, et al. Efficiently mining co-location rules on interval data[C]// Proceedings of the 6th Int Conf on Advanced Data Mining and Applications (ADMA 2010). Chongqing, China, Part I, LNCS 6440, 2010: 477-488

[9] Altman D. Fuzzy set theoretic approaches for handling imprecision in spatial analysis[J]. International Journal of Geographical Information Science, 1994, 8(3): 271-289

[10] Zheng Kai, Fung Pui-cheong, Zhou Xiao-fang. K-nearest neighbor search for fuzzy objects[C]// Proceedings of the Special Interest Group on Management of Data (SIGMOD'10). Indiana, USA, 2010: 699-710

[11] 欧阳志平,王丽珍,陈红梅. 模糊对象的空间 co-location 模式挖掘研究[J]. 计算机学报, 2011, 34(10): 1947-1955

[12] 欧阳志平,王丽珍,周丽华. 实例位置模糊的空间 co-location 模式挖掘研究[J]. 计算机科学与探索, 2012, 6(12): 1144-1152

[13] 吴萍萍. 模糊 Co-Location 模式挖掘[D]. 昆明:云南大学, 2012

[14] Hirate Y, Iwahashi E, Yamana H. An Efficient Algorithm for Mining Frequent Patterns without any Thresholds[C]// Proc. of Workshop on Alternative Techniques for Date Mining and Knowledge Discovery. 2004

[15] Bow M. Compact Co-location Pattern Mining [D]. Indiana: Indiana University, 2011

(上接第 125 页)

参 考 文 献

[1] Zacks J M, Tversky B. Event Structure in Perception and Conception [J]. Psychological Bulletin, 2001, 127(1): 3-21

[2] Nelson K, Gruendel J. Event knowledge: Structure and function in development [M]. Hilladale, NJ: Erlbaum, 1986

[3] ACE (Automatic Content Extraction). Chinese Annotation Guidelines for Events[S]. National Institute of Standards and Technology, 2005

[4] 刘宗田,黄美丽,周文,等. 面向事件的本体研究[J]. 计算机科学, 2009, 6(11): 189-192

[5] 戈也挺,朱朝晖,陈世福. 行动推理中若干问题的研究[J]. 计算机科学, 2000, 27: 85-89

[6] 黄智生. 关于行动的推理[J]. 计算机科学, 1993, 20: 7-13

[7] McCarthy J. Situations, actions, and causal laws[R]. Stanford, California: Stanford University Artificial Intelligence Project, 1963

[8] Shanahan M. The event calculus explained [M]. Artificial Intelligence Today, Spring Berlin Heidelberg, 1999: 409-430

[9] 史忠植,董明楷,蒋运承,等. 语义 Web 的逻辑基础[J]. 中国科学:E 辑, 2004, 34(10): 1123-1138

[10] 常亮,史忠植,陈立民,等. 一类扩展的动态描述逻辑[J]. 软件学报, 2010, 21(1): 1-13

[11] Chang Liang, Shi Zhong-zhi, Qiu Li-rong, et al. Dynamic description logic: embracing actions into description logic[C]// Proc. of the 20th International Workshop on Description Logics (DL'07). Italy: 2007: 243-253

[12] Baader F, Lippmann M, Liu Hong-kai. Using causal relationships to deal with the ramification problem in action formalisms based on description logics[C]// Logic for Programming, Artificial Intelligence, and Reasoning. Springer Berlin Heidelberg, 2010, 6397: 82-96

[13] Liu Wei, Xu Wen-jie, Wang Dong, et al. An extending description logic for action formalism in event ontology [J]. International Journal of Computational Science and Engineering, 2010, 6104: 471-481

[14] Martinez DC, Hitzler P. Extending Description Logic Rules[C]// Proceedings of 9th Extended Semantic Web Conference. Heraklion, 2012: 345-359

[15] Grosof B N, Horrocks I, Volz R, et al. Description logic programs: Combining logic programs with description logic[C]// Proceedings of the 12th International Conference on World Wide Web. ACM, 2003: 48-57

[16] Horrocks I, Patel-SchneiderP F. A proposal for an OWL rules language[C]// Proceedings of the 13th international conference on World Wide Web. ACM, 2004: 723-731

[17] Donini F M, Lenzerini M, Nardi D, et al. AL-log: Integrating datalog and description logics [J]. Journal of Intelligent Information Systems, 1998, 10(3): 227-252

[18] Levy A Y, Rousset M C. CARIN: a representation language combining horn rules and description logics[C]// ECAI. Pitman, 1996: 323-327

[19] Rosati R. Towards expressive KR systems integrating Datalog and description logics: Preliminary report [C]// Proc. of DL'99. 1999: 160-164

[20] Rosati R. On the decidability and complexity of integrating ontologies and rules [J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2005, 3(1): 61-73

[21] Mei Jing, Lin Zuo-quan, Boley H. ALCP^{u}: An Integration of Description Logic and General Rules[C]// Proceedings of the 1st International Conference on Web Reasoning and Rule Systems. Heidelberg: Springer-verlag Berlin, 2007: 163-177

[22] Schmidt-Schauβ M. Subsumption in KL-ONE is Undecidable. Principles of Knowledge Representation [M]. Stanford, CA, US: CSLI Publications, 1989: 421-431

[23] Yehia W, Liu H K, Lippmann M, et al. Experimental results on solving the projection problem in action formalisms based on description logics[C]// Proc. of the 25th Intern. Workshop on Description Logics. 2012

[24] Coelho H. Verifying properties of infinite sequences of description logic actions[C]// ECAI 2010: 19th European Conference on Artificial Intelligence. IOS Press, Incorporated, 2010: 53-58

[25] Drabent W, Eiter T, Ianni G, et al. Hybrid reasoning with rules and ontologies [M]. Semantic techniques for the Web, Springer Berlin Heidelberg, 2009: 1-49

[26] Allen J F. Temporal reasoning and planning[C]// Reasoning about Plans. Morgan Kaufmann, 1991: 1-67