软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2014,25(Suppl.(2)):189–200

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 空间 co-location 模式增量挖掘及演化分析*

芦俊丽[1,2], 王丽珍[1], 肖 清[1], 王 新[2]

[1](云南大学 信息学院 计算机科学与工程系,云南 昆明 650091)

[2](云南民族大学 数学与计算机科学学院,云南 昆明 650031)

通讯作者: 王丽珍, E-mail: lzhwang@ynu.edu.cn

**摘 要**: 空间 co-location 模式挖掘是空间数据挖掘的一个重要研究方向.空间 co-location 模式是空间对象的一个子集,它们的实例在空间中频繁关联.到目前为止,空间 co-location 模式挖掘都只关注某一个时刻的空间 co-location 模式.然而,在实际应用中,数据库中的数据是随着时间改变的,所以高效地增量挖掘空间 co-location 模式是非常必要的;空间 co-location 模式演化分析可以发现空间 co-location 模式的变化规律,预测特定事件的发生,但是对这些问题的研究并未见诸报道.研究了高效的空间 co-location 模式增量挖掘及空间 co-location 模式的演化分析,首先,提出了高效的空间 co-location 模式增量挖掘基本算法及剪枝算法.其次,在多个随时间变化的真实数据集上挖掘 co-location 演化模式.再次,证明了空间 co-location 模式增量挖掘基本算法及剪枝算法是正确的和完备的.最后,在"模拟+真实"的数据集上用充分的实验验证了增量挖掘基本算法的性能以及剪枝算法的剪枝效果.此外,把空间 co-location 增量挖掘基本算法、剪枝算法及演化模式挖掘算法应用到三江并流区域珍稀植物数据集上,增量挖掘出空间 co-location 模式及演化模式,预测了 co-location 模式的演化规律,更好地实现了对珍稀植物的动态跟踪和保护.

**关键词**: 空间 co-location 模式;增量挖掘算法;剪枝算法;演化分析;三江并流区域

中文引用格式: 芦俊丽,王丽珍,肖清,王新.空间 co-location 模式增量挖掘及演化分析.软件学报,2014,25(Suppl.(2)):189–200. http://www.jos.org.cn/1000-9825/14037.htm

英文引用格式: Lu JL, Wang LZ, Xiao Q, Wang X. Incremental mining and evolutional analysis of co-locations. Ruan Jian Xue Bao/Journal of Software, 2014,25(Suppl.(2)):189–200 (in Chinese). http://www.jos.org.cn/1000-9825/14037.htm

## Incremental Mining and Evolutional Analysis of Co-Locations

LU Jun-Li[1,2], WANG Li-Zhen[1+], XIAO Qing[1], WANG Xin[2]

(Department of Computer Science and Engineering, School of Information Science and Engineering, Yunnan University, Kunming 650091, China)

(School of Mathematics and Computer Science, Yunnan Minzu University, Kunming 650031, China)

Corresponding author: WANG Li-Zhen, E-mail: lzhwang@ynu.edu.cn

**Abstract**: Spatial co-locations mining is an important research domain in spatial data mining. Spatial co-locations represent the subsets of spatial features which are frequently located together in geographic space. Up to present, all the existing co-location mining algorithms only focus on discovering ordinary co-location patterns or co-location rules. However, in real-world applications, the data in a database do not usually remain a stable condition, making efficient incremental mining for co-locations very indispensable and interesting. The evolutionary analysis of co-locations can discover the development rules of co-locations, and predict the particular event happened in future. However, no results have yet been reported from these researches. This paper studies the incremental mining for co-locations and the evolutionary analysis of co-locations. Firstly, an efficient basic algorithm and a prune algorithm for incremental mining are proposed. Secondly, evolutionary co-locations are discovered based on several real datasets. Thirdly, both the basic algorithm and prune algorithm

* 基金项目: 国家自然科学基金(61472346, 61272126); 云南省教育厅基金(2012C103)
收稿时间: 2014-05-07; 定稿时间: 2014-08-19

are proved correct and complete. Fourth, extensive experiments are performed to verify the performance and effectiveness of the basic algorithm and prune algorithm. At last, the basic algorithm and prune algorithm for incremental mining in conjunction with the evolutionary co-locations mining algorithm are applied to the Three Parallel Rivers of Yunnan protected Areas plant database to predict the development rules of co-locations, and dynamically track and protect the rare plants of this area.

**Key words**: spatial co-locations; incremental mining algorithm; pruning algorithm; evolutionary analysis; Three Parallel Rivers of Yunnan Protected Areas

空间数据挖掘是从空间数据库中发现潜藏的有趣模式的过程.由于空间数据类型,空间关系和空间自相关性等复杂性,从空间数据中挖掘有趣的模式远比从事务型数据库中要困难得多[1].

空间co-location模式代表了一组空间对象的子集,它们的实例在空间中频繁关联.挖掘空间co-location模式就是在空间数据库中发现和挖掘空间对象之间的关联关系.由于空间co-location模式挖掘的重要性,现已有很多研究方法[2−18].

现有的 co-location 挖掘算法都比较关注挖掘普通的空间 co-location 模式或规则.然而,在实际应用中,数据库中的基础数据量很大,而数据又是随着时间改变的,数据更新的速度非常快,而更新的数据量相对于基础数据量却不大,因此,高效的空间 co-location 模式增量挖掘显得非常必要;空间 co-location 模式演化分析可以发现空间 co-location 模式的变化规律,预测特定事件的发生.但是,对这些问题的研究并未见报道.

例如,在 2005 年的珍稀植物数据上挖掘空间 co-location 模式,显示长苞冷衫和松茸是一频繁模式,而在 2006 年的数据上挖掘时,这一模式没有出现在挖掘结果中.调查其原因,是由于大量砍伐导致了长苞冷衫减少.挖掘频繁模式的变化,可以对珍稀植物保护发出预警,及时采取措施,改善环境,避免大量砍伐和采摘,进行相同环境的大量种植等.

农作物会受病虫害的影响,病虫害的种类会随着气候、季节、农作物的特性、农作物间的传播而发生改变.对随着时间变化的多个数据集挖掘空间 co-location 演化模式,可以预测模式的演化规律 .例如,在每年的四、五月水稻病虫害数据集中,我们都挖掘到了同一模式(水稻,二化螟 ),这一发现可以指导我们在下一年春季对二化螟的预防.其他的应用包括地球科学、公共卫生、生态学、交通运输等.实际上凡是普通的空间 co-location 模式挖掘的应用领域,都需要增量挖掘及演化模式挖掘来进行深入的研究.

空间 co-location 模式增量挖掘的研究还未见报道,传统的方法就是对当前数据集重新挖掘.本文是从原始数据集和当前数据集中找到变化的数据集,利用原始数据集中已经挖掘到的模式的信息和变化的数据集中得到的变化信息,来生成当前数据集中模式的信息.由于变化的数据集要比原始数据集或当前数据集小得多,从而提高了效率.

## 1 相关工作

空间co-location模式挖掘的方法[2−17]很多,最早由Huang等人在文献[2]中提出最小参与率概念,由于最小参与率概念的自然和具有类 Apriori 性质,此类挖掘算法被广泛研究.文献[2]在最小参与率概念的基础上进而提出了基于完全连接(join-based)的方法,该方法以 Apriori-like 的形式,基于 $k$ 阶模式产生 $k$+1 阶候选模式,基于 $k$ 阶表实例连接产生 $k$+1 阶表实例.该方法能够产生完整的和正确的 co-location 模式集合,可是当数据集比较稠密或 co-location 模式长度增大时,连接操作开销变得巨大.于是,文献[3]提出基于部分连接(partial-join)的方法,其核心思想是先把连续空间中的实例分割为不相交的块(划分),实例连接就变成了块内实例的连接和块间实例的连接,从而减少连接中的计算量.这个方法的关键在于能否划分出尽量大的块,产生尽量少的块间邻近关系.

与传统关联规则挖掘的 FP-Growth 研究相对应,文献[4]提出一种基于投影的 co-location 模式挖掘(FP-CM)算法,该算法首先将空间数据集转换为传统的事务集,然后基于 FP-Growth 求出最大频繁模式,通过最大频繁模式组合求出所有的频繁模式.该方法将空间数据集转换为传统事务集后,出现大量的冗余信息,且表示不自然.

文献[5]提出一种基于星型邻居扩展的无连接(join-less)算法以解决产生表实例的连接开销问题,在稠密型数据库中,它的效率比 join-based 算法高,但当候选 co-location 的星型实例中存在很多非 co-location 实例的时候,

过滤步骤将会很耗时.因此,在大型空间数据集中,算法改进效果不明显.

无连接挖掘方法中,3 种基于前缀树的方法值得关注:① CPI-tree(co-location pattern instances tree)算法[6].CPI-tree 以树结构物化空间实例间的邻近关系,所有的 co-location 表实例能够通过 CPI-tree 快速生成,因此该算法的时间性能超过了 join-less 算法,不过,随着数据集的急剧增长,存储和遍历 CPI-tree 变得困难,另外,在稠密型数据集中,该算法的时间代价仍然高.② iCPI-tree 算法[7].该算法综合 Apriori 性质剪枝和 CPI-tree 树结构,进一步优化了挖掘过程.③ Order-Clique-Based 算法[8].在进一步优化的前缀树结构下,文献[8]讨论了挖掘最大频繁 co-location 模式的问题,提出基于有序团(Order-Clique-Based)的最大 co-location 模式挖掘算法.该算法能够高效地生成候选最大 co-location 模式和表实例,与传统的基于 Apriori-like 的算法相比,该算法避免了大量表实例的存储,具有较高的效率.

针对实际应用中数据的特殊性,研究人员展开了广泛地研究,空间 co-location 模式挖掘技术从而得到了迅速扩展.在最小参与率度量下,具有很少实例的空间对象(稀有空间对象)常常因不频繁而被忽略,结果丢失了一些有趣模式,文献[9]引入最大参与率概念,提出一个有趣的最大参与率的弱单调性质,从而解决稀有空间对象的 co-location 模式挖掘问题.此方法消除了稀有对象和普通对象不平等出现的矛盾,能在有稀有对象存在的数据集合中找到频繁的 co-location 模式.然而,由于最大参与率方法并没有考虑某些模式中没有稀有对象存在和稀有对象参与率小于普通对象参与率的情况,因此一些并不频繁的模式被挖了出来[10].针对这一情况,文献[10]又提出最小加权参与率的概念,不但可以挖掘出带稀有对象的频繁 co-location 模式,而且可以排除不频繁的模式.针对模糊数据,文献[11]首次提出了模糊参与率及模糊参与度概念,并提出了基本的挖掘算法和 3 个有效的剪枝算法来挖掘模糊数据的空间 co-location 模式.文献[12]针对带模糊属性的空间数据集进行 co- location 模式挖掘.文献[13]对任意形状的数据簇进行区域挖掘 co-location 模式.近几年来,不确定数据受到广泛关注,不确定数据的 co-location 模式挖掘也随之兴起[14−18].文献[14]从区间数据中挖掘空间 co-location 模式,文献[15]从位置不确定的数据中挖掘 co-location 模式,数据是以概率密度函数(PDF)形式描述其位置的不确定性.文献[16]在空间不确定数据集中挖掘概率频繁的空间 co-location 模式.在概率频繁性度量上,论文不仅考虑了传统的期望参与度,还考虑了与参与度相关的置信度,使挖掘出的模式更合理.文献[17]从带概率区间的空间不确定数据中挖掘 co-location 模式.

空间 co-location 模式增量挖掘的研究至今未见报道,但是已经有很多关于关联规则增量挖掘的研究[18−20].在真实世界中,数据库中的记录时刻在变化.一些新的规则可能被生成,而旧的规则变得无效[18].为了在动态数据库中进行高效的增量挖掘,更新已发现的规则,文献[19]提出了快速更新(FUP)增量挖掘算法.其频繁模式的度量仍采用传统的支持度-置信度框架,虽然 FUP 增量挖掘算法可以利用原有挖掘结果和新增数据对当前的据集实现高效地增量挖掘,但此度量方法在一些实际应用中是不合适的.针对这一情况,文献[20]基于 FUP 概念提出了高效用关联规则的增量挖掘算法,其度量不再只考虑模式的频繁性,而是根据用户的偏好考虑模式的价值、利润等其他因素,因此更符合实际需要.

## 2 空间 co-location 模式增量挖掘

本节将介绍空间 co-location 模式增量挖掘,先介绍一些相关概念,再介绍基本挖掘算法及剪枝算法.相关概念主要有变化的实例集、变化的 co-location 模式及实例、候选模式的当前表实例以及变化的空间 co-location 频繁模式.

### 2.1 相关定义

**定义 1(变化的实例集 *S_change*)**. 设存在两个空间实例集:原始实例集 *S_old* 和当前实例集 *S_new*,比较 *S_old* 和 *S_new*,可得到 *S_old* 中消失的实例及团关系,*S_new* 中新增的实例及团关系,对于这样发生变化的实例及受牵连的团关系集,我们称之为变化的实例集 *S_change*.

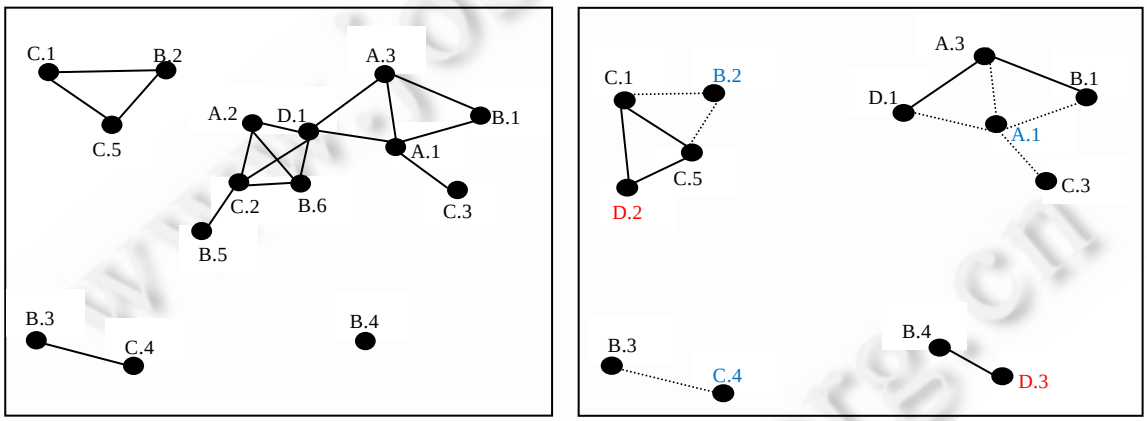
由于可利用的实例集较多,*S_old*,*S_new*,*S_change*.本文在计算某一 co-location 模式的实例时,会注明是在哪一个实例集上的 co-location 实例.

**定义 2(变化的 co-location 模式及实例)**. 给定一个空间对象集 *F*,*S_old*,*S_new*,*S_change*.*S_old*,*S_new* 以及 *S_change* 上的邻近关系 *R*,变化的 co-location 模式 ***c*** 为 *F* 的一个子集,*c* 的 co-location 实例为 *S_change* 上的实例利用邻近关系 *R* 形成的团.变化的 co-location 模式 *c* 的实例是在 *S_change* 中识别的实例,它包含 co-location 模式 *c* 的所有对象,且形成一个团关系,也可称之为 co-location 模式 *c* 在 *S_change* 中的实例.

从上述定义可以看出,一个变化的 co-location 实例 *I* 仍然满足一般的实例的特点,即包含了模式中所有对象,且形成一个团关系,不同的是只从变化的实例集中识别变化的 co-location 实例.

图 1 是一个具体例子,图 1(a)为 *S_old*,对象集 *F*={A,B,C,D},各对象的实例总数分别为 $n_A$=3,$n_B$=6,$n_C$=5,$n_D$=1. 图 1(b)为 *S_change*,其中消失的实例用蓝色标识,因消失的实例导致的消失的邻近关系用虚线连接,增加的实例导致的增加的团关系仍用实线连接.以模式{B,C}为例,其变化的实例有(B.2,C.1),(B.2,C.5), (B.3,C.4),分别是由于 C.4 和 B.2 的消失导致的这些变化的实例.

增加了实例,可能会导致某一模式的实例增加,也可能不增加,比如图 1(b)中的实例 D.3,对于模式{B,D},就增加了一个实例,而对于模式 {A,B,C}或其他模式,其实例数就没有变化.同理,实例 B.2 的消失导致模式{B,C}减少了两个实例,而模式{A,B ,C}或其他模式的实例数就没有变化.对于每个模式 *c*,其增加的 co-location 实例和减少的 co-location 实例构成变化的表实例.




(a) 原始实例集 (b) 变化实例集

图 1 一个例子

增量挖掘 *S_new* 的频繁模式时,候选模式为 *S_change* 中的对象构成的模式,即可能发生变化的模式,其表实例定义如下.

**定义 3(候选模式的当前表实例)**. 对于某一 *k* 阶 co-location 候选模式 *c*,*c* 在 *S_old* 中的表实例,去掉变化的表实例中减少的表实例,加上变化的表实例中增加的表实例,就是候选模式 *c* 的当前表实例,即候选模式 *c* 在 *S_new* 中的表实例.

**定义 4(变化的 co-location 频繁模式)**. 若一模式 *c* 在 *S_old* 中是频繁的(不频繁的),而在 *S_new* 中是不频繁的(频繁的),则 *c* 为变化的 co-location 频繁模式.

为区分某一模式 *c* 在 S_old 和 S_new 中的参与度,我们记 $PR_O(c)$为 *c* 在 S_old 中的参与度,$PR_N(c)$为 *c* 在 S_new 中的参与度.图 1 的例子中,若 *min_prev*=0.5,$PR_O(\{B,C\})=\min\{PI_O(\{B,C\},B),PI_O(\{B,C\},C)\}=\{4/6,3/5\}=3/5>min\_prev$,在 S_new 中的参与度为 $PR_N=\min\{PI_N(\{B,C\},B), PI_N(\{B,C\},C)\}=\{2/5,1/4\}=1/4<min\_prev$,所以模式{B,C}是变化的频繁模式,并且是消失的频繁模式.

### 2.2 空间co-location模式增量挖掘算法

#### 2.2.1 基本算法

空间 co-location 模式增量挖掘基本算法描述如下:

**算法 1**. 空间 co-location 模式增量挖掘基本算法.

输入:空间对象集 *F*, *S_old*,*S_new*,参与度阈值 *min_prev*,距离阈值 *dist_threshold*;

输出:*S_new* 中的 co-location 频繁模式集 EC.

变量:*k*:co-location 模式的阶,$C\_N_k$:*k* 阶 co-location 候选模式集,$T\_CH_k$:候选模式在 *S_change* 中的变化表实例,$T\_O_k$:候选模式在 *S_old* 中的 *k* 阶表实例集,$T\_N_k$:候选模式在 *S_new* 中的 *k* 阶表实例集,$P\_O_k$:*S_old* 中 *k* 阶的 co-location 频繁模式集,$P\_N_k$:候选模式中得到的 *k* 阶 co-location 频繁模式集,$P\_RN_k$:*S_new* 中 *k* 阶的 co-location 频繁模式集.

步骤:

1. 从 *S_old* 与 *S_new* 中去掉位置相同的实例,得到变化的实例集 *S_change*, *S_change* 中的对象集为 *FCH*.

2. $P\_N_1=FCH$,$T\_CH_1=S\_change$,EC=∅

3. for ($k$=2; $P\_{k-1}\neq\varnothing$;$k$++)

3.1. $C\_N_k$=gen_candidate_co-location($k$,$P\_N_{k-1}$);

3.2. $T\_CH_k$=gen_table_instance($C\_N_k$,$T\_CH_{k-1}$);

3.3. $T\_N_k$=gen_cur_table_instance($T\_CH_k$,$T\_O_k$); //由*S_old*的表实例和变化表实例生成当前表实例

3.4. $P\_N_k$ =sel_prev_co-location($C\_N_k$,$T\_N_k$,*min_prev*); //生成 *S_new* 中可能变化的频繁模式集

3.5. $P\_RN_k$ = cur_prev_co-location($P\_O_k$,$P\_N_k$);

3.6. EC=EC∪$P\_RN_k$;

算法第 2 步初始化 $P\_N_1$ 和 $T\_CH_1$,模式 $P\_N_1$ 初值是从 *S_change* 中的对象产生,以保证第 3.1 步的候选模式均从 *S_change* 中产生;而变化的表实例初值 $T\_CH_1$ 为 *S_change*.

第 3 步迭代地利用模式在 *S_old* 中的表实例和变化的表实例得到模式的当前表实例(3.3),进而得到模式在 *S_new* 中的频繁性,求解出模式的频繁性(3.4),与模式在 *S_old* 中的频繁性对照,得到 *S_new* 中 *k* 阶的所有 co-location 频繁模式(3.5),并到 EC 集合中.

第 3.1 步是从 *S_new* 中的上一阶频繁的模式生成下一阶的候选模式,因为在 *S_new* 中,空间 co-location 模式满足向下闭合性质,可以进行模式剪枝.

第 3.2 步,上一阶变化的表实例连接生成下一阶变化表实例.由于减少的表实例已经包含在原始表实例中,就不再重新生成.因此,实际操作时,变化的表实例中就只有增加的表实例.在计算当前表实例时,直接从原始表实例中去掉减少的表实例,再加上增加的表实例,增加的表实例需要用上一阶连接生成下一阶.

算法 1 在求解变化的 co-location 频繁模式时,需已知模式在 *S_old* 中的表实例,因此在计算 *S_old* 中的 co-location 频繁模式时,需要保存每个模式的表实例.由于剪枝关系,可能 *S_old* 中的一些模式没有计算表实例,此时,需重新计算其在 *S_old* 中的表实例,可以由已有的最近阶表实例生成.因为变化的数据量不大,在实际运算时,这样的情况是很少的.

### 2.2.2 剪枝算法

剪枝算法快速剪掉那些不会发生变化的模式,其剪枝思路是基于算法 1 中存在的问题提出的.问题阐述为:

在挖掘空间 co-location 模式时,模式的参与度是个统计值.空间 co-location 模式增量挖掘时,*S_change* 因包含了被牵连的团关系,可能覆盖所有对象,就会生成所有的候选模式.但是,实例的少量变化可能不会对模式的参与度产生很大影响,因此不会影响模式的频繁性.若模式为频繁的,则要继续求解其高阶模式,这样无疑浪费了大量时间.到底多少实例的变化(即量的变化),会导致模式频繁性的变化(即性的改变),这是我们进一步要细致研究的问题.

事实上,增量挖掘只需对变化的候选模式进行重新挖掘.非变化的候选模式的频繁性和它在 *S_old* 中的频繁性相同.那么,什么样的 co-location 模式是变化的候选模式?判断原则如定理 1 所述,为阐述定理 1,介绍两个概念.

**定义 5(模式的对象在 *S_new* 中的参与率上限)**. 设一个 *k* 阶 co-location 模式 $c$,$c=(f_1,f_2,\ldots,f_k)$,*OPR*

$(c,f_i)=\frac{io(c,f_i)}{n_{fi}(f_i)}$ 为模式 $c$ 的对象 $f_i$ 在 *S_old* 中的参与率.设在 *S_change* 中,对象 $f_i$ 增加的实例数为 $n_{incre}(f_i)$,减少的实例数为 $n_{decre}(f_i)$,则模式 $c$ 的对象 $f_i$ 在 *S_new* 中的参与率上限为 $CPR^{\max}(c,f_i)$.$CPR^{\max}(c,f_i)$的取值与 $OPR(c,f_i)$和 *min_prev* 之间的关系相关.若 $OPR(c,f_i)<min\_prev$,则

$$CPR^{\max}(c,f_i)=\frac{io(c,f_i)+n_{incre}(f_i)}{n_{fi}(f_i)+n_{incre}(f_i)-n_{decre}(f_i)},$$

若 $OPR(c,f_i)>min\_prev$,则

$$CPR^{\max}(c,f_i)=\frac{io(c,f_i)-n_{decre}(f_i)}{n_{fi}(f_i)+n_{incre}(f_i)-n_{decre}(f_i)}.$$

**定义 6(模式在 *S_new* 中的参与度上限)**. 设一个 $k$ 阶 co-location 模式 $c=(f_1,f_2,\ldots,f_k)$,则模式 $c$ 的参与度上限 $CPI^{\max}(c)=\min_{i=1}^{k}\{CPR^{\max}(c,f_i)\}$.

**定理 1**. 若 $OPR(c,f_i)<min\_prev$,$CPR^{\max}(c,f_i)<min\_prev$,则此模式在 *S_new* 中仍为不频繁的,其高阶也不会频繁,若 $CPR^{\max}(c,f_i)>min\_prev$,则此模式是变化的候选模式(可能由不频繁变为频繁);若 $OPR(c,f_i)>min\_prev$,$CPI^{\max}(c)>min\_prev$,则此模式在 *S_new* 中仍为频繁,若 $CPR^{\max}(c,f_i)<min\_prev$,则此模式也是变化的候选模式(可能由频繁变为不频繁).

证明:若 $OPR(c,f_i)<min\_prev$,$CPR^{\max}(c,f_i)<min\_prev$,说明 $OPR(c,f_i)$距离 min_*prev* 很远,对象 $f_i$ 又没有大量的实例增加能让 $OPR(c,f_i)$超过 *min_prev*,此模式在 *S_new* 中仍为不频繁的,因模式 $c$ 的对象 $f_i$ 在 *S_new* 中的参与率上限随着阶的增加而递减,则它的高阶也不会频繁.若 $CPR^{\max}(c,f_i)>min\_prev$,说明 $OPR(c,f_i)$距离 min_*prev* 很近,或者对象 $f_i$ 大量的实例增加使得 $CPR^{\max}(c,f_i)$超过 min_*prev*,此模式是变化的候选模式,需要计算其精确的参与率以断定其频繁性;若 $OPR(c,f_i)>min\_prev$,$CPI^{\max}(c)>min\_prev$,说明 $OPR(c,f_i)$距离 min_*prev* 很远,每个对象都没有大量的实例减少能让 $CPI^{\max}(c)<min\_prev$,此模式在 *S_new* 中仍为频繁,不必做表连接操作计算精确参与率.若 $CPR^{\max}(c,f_i)<min\_prev$,说明 $OPR(c,f_i)$距离 min_*prev* 很近,或者对象 $f_i$ 有大量的实例减少使得 $CPR^{\max}(c,f_i)$小于 min_*prev*,此模式是变化的候选模式,需要计算其精确的参与率以断定其频繁性.

利用定理 1,可以设计剪枝算法,对 *S_change* 中的每个模式进行考察,判断它是否为变化的候选模式.剪枝算法见算法 2.

**算法 2**. 剪枝算法.

输入:*S_change*,*S_change* 上的对象集 *FCH*,*FCH* 中的模式中各对象在 *S_old* 中的参与率,参与度阈值 *min_prev*,距离阈值 *dist_threshold*;

输出:变化的候选 co-location 模式集 PC.

步骤:

1. 对于 *FCH* 中的每个模式 $c$
2. 计算 $CPR^{\max}(c,f_i)$,

   若 $OPR(c,f_i)<min\_prev$,

   $CPR^{\max}(c,f_i)>min\_prev$,$PC=PC\cup c$;

   若 $OPR(c,f_i)>min\_prev$,

   $CPR^{\max}(c,f_i)<min\_prev$,$PC=PC\cup c$;

算法 2 得到变化的候选模式集 PC,接下来只对这些候选模式重新计算其参与度,判断其频繁性.

## 3 变化的空间 co-location 模式演化分析

空间 co-location 模式增量挖掘算法的高效性体现在用 *S_change* 和已有的 *S_old* 中的表实例挖掘 *S_new* 中的频繁模式.在挖掘空间 co-location 演化模式时,我们希望对第一个原始实例集挖掘一次,以后均用前一个实例集的表实例及变化的实例集挖掘下一个实例集的 co-location 频繁模式,连续做下去,即得到了变化的空间

co-location 演化模式,挖掘过程如图 2 所示.

**算法 3**. 空间 co-location 演化模式挖掘算法.

输入:空间对象集 $F$,$n$ 个实例集 $S_1,S_2,\ldots,S_n$,参与度阈值 min_*prev*,距离阈值 *dist_threshold* ;

输出:空间 co-location 演化频繁模式 CL.

步骤:

1. 对第 1 个实例集 $S_1$,使用全连接算法挖掘其 co-location 频繁模式集;

2. $CL_1=BA(S_1,S_2)$ //调用增量挖掘算法,得到 $S_2$ 中的模式

3. For ($i$=3;$i$<=$n$;$i$++)

{

3.1 保存实例集 $S_{i-1}$ 中模式的表实例;

3.2 $CL_{i-1}=BA(S_{i-1},S_i)$

}

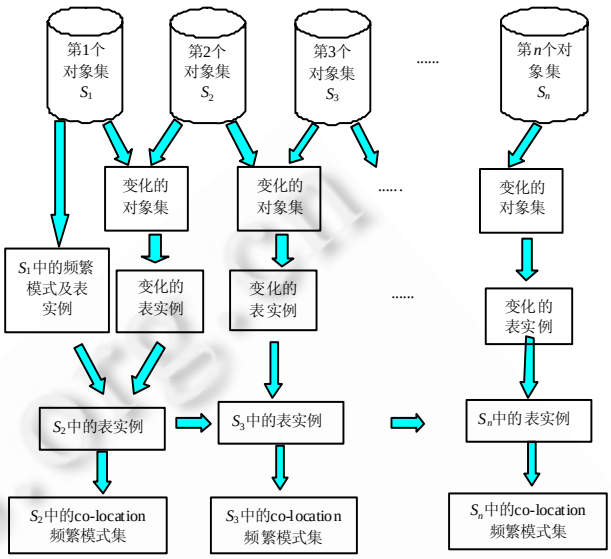



图 2 变化的数据集挖掘空间 co-location 演化模式

## 4 算法的正确性和完备性证明

完备性意味着空间 co-location 模式增量挖掘基本算法及剪枝算法可以挖掘到所有在 *S_new* 中满足参与度阈值的频繁模式.正确性意味着空间 co-location 模式增量挖掘基本算法及剪枝算法挖掘到的所有空间 co-location 模式均为 *S_new* 中满足参与度阈值的频繁模式.首先证明完备性,先给出一些引理.

**引理 1**. *S_change* 中不会缺少变化的实例及被牵连的团关系.

证明:由定义 1 可知,*S_change* 是由变化的实例以及被牵连的团关系构成. □

**引理 2**. 设 $c$ 是一个 $k$ 阶 co-location 模式,$c=\{f_1,f_2,\ldots,f_k\}$.$T\_O_k(c)$, $T\_CH_k(c)$, $T_N_k(c)$分别为 $c$ 在 *S_old*,*S_change* 及 *S_new* 中的表实例,则由 $T_O_k(c)$和 $T_CH_k(c)$获得的表实例 $T_TN_k(c)$不小于 $T\_N_k(c)$.

证明:$T_CH_k(c)$中包括增加的表实例和减少的表实例,增加或减少的 co-location 实例中必须含有增加或减少的实例.但是为生成 $T_CH_{k+1}(c)$时不丢失 $k$+1 阶实例,不含增加或减少实例的实例也在 $T_CH_k(c)$中.例如 $T_CH_k(c)$中的一个 3 阶实例 $t_1=\{f_1.1,f_2.1,f_3.2\}$,其中 $f_1.1,f_2.1,f_3.2$ 均为 *S_change* 中被牵连的实例,不属真正增加或减少的实例,但是实例 $t_1$ 仍在 $T_CH_3(c)$中,为了在 $T_CH_3(c)$中不丢失像这样的实例 $t_2=\{f_1.1,f_2.1,f_3.2,f_4.5\}$,其中 $f_4.5$ 为对象 $f_4$ 中减少的实例.考虑到上述因素后,从 $T_O_k(c)$中去掉 $T_CH_k(c)$中含有减少实例的表实例,加上 $T_CH_k(c)$中含有增加实例的表实例,得到的表实例 $T_TN_k(c)$不小于 $T\_N_k(c)$. □

**定理 2**. 空间 co-location 模式增量挖掘基本算法是完备的.

证明: 由引理 1 可知,*S_change* 中不会缺少变化的实例及被牵连的团关系,因此由 *S_change* 生成的 $T_CH_k(c)$是完备的.由引理 2 可知,由 $T_O_k(c)$和 $T_CH_k(c)$获得的表实例 $T_TN_k(c)$不小于 $T\_N_k(c)$.因此,计算得到的模式 $c$ 的参与度不小于模式 $c$ 在 $T_N_k(c)$中的参与度.若模式 $c$ 在 $T_TN_k(c)$中频繁,在 $T_N_k(c)$中也会频繁,空间 co-location 模式增量挖掘基本算法不会漏掉 *S_new* 中的频繁模式,其挖掘结果是完备的. □

**引理 3**. 空间 co-location 模式增量挖掘剪枝算法没有剪掉变化的模式.

证明:根据定义 5、定义 6 和定理 1 可知,计算 $CPR^{max}(c,f_i)$及可以确定它是变化的候选模式.只要是变化的候选模式,都要精确计算其参与度,并没有被剪掉. □

**引理 4**. 空间 co-location 模式增量挖掘剪枝算法没有丢失 *S_new* 中的频繁模式.

证明:由引理 3,剪枝算法剪掉的模式均为频繁性不可能发生变化的模式,其频繁性与 $c$ 在 *S_old* 中的频繁性相同.虽然没有重新计算,但是模式 $c$ 在 *S_new* 中的频繁性可以由在 *S_old* 中的频繁性得到,因此不会丢失 *S_new*

中的频繁模式. □

**定理 3**. 空间 co-location 模式增量挖掘剪枝算法是完备的.

证明:由定理 2 和引理 4 可知,空间 co-location 模式增量挖掘剪枝算法是完备的.

正确性意味着空间 co-location 模式增量挖掘基本算法及剪枝算法挖掘到的所有空间 co-location 模式均为 *S_new* 中满足参与度阈值的频繁模式. □

**定理 4**. 空间 co-location 模式增量挖掘基本算法是正确的.

证明:空间 co-location 模式增量挖掘基本算法的正确性由第 1 步和第 3.2 步,第 3.3 步保证.第 1 步正确地获取了 *S_change*,其中包含了被牵连的团关系.引理 2 的证明中已经介绍第 3.2 步如何正确地从低阶变化的表实例生成高阶变化表实例,3.3 步如何正确地由原始表实例和变化表实例生成当前表实例. □

**定理 5**. 空间 co-location 模式增量挖掘剪枝算法是正确的.

证明:由定义 5、定义 6 和定理 1 可知,剪枝算法利用 $CPR^{\max}(c,f_i)$和 $CPI^{\max}(c)$剪掉了非变化的候选模式,其频繁性可由其在 *S_old* 中的频繁性获得,不影响挖掘结果.因此,空间 co-location 模式增量挖掘剪枝算法是正确的. □

## 5 实 验

我们将在模拟数据和真实数据上详细地验证算法的效率和效果.实验评估主要从以下几方面进行:空间 co-location 模式增量挖掘基本算法的效率和剪枝算法的剪枝效果,空间 co-location 演化模式挖掘算法在真实数据上的应用.所有算法均在奔腾IV,2.4 GHz CPU,4GB 内存的 PC 机上用 C#语言实现.

由于空间 co-location 模式增量挖掘的研究还未见报道,传统的方法就是 *S_new* 重新挖掘空间 co-location 模式.增量挖掘的实验主要是用传统方法与本文提出的基本算法、剪枝算法进行比较.

### 5.1 数据集

为评估算法性能,我们使用了拥有不同空间对象和实例的多个数据集,表 1 显示了各数据集的大小、对象数、实例的分布区域以及数据集的来源,其中 Data-sets6 是一组不同年份的相同规格的真实数据集,用来验证演化模式挖掘算法.

**表 1** 数据集

| 数据集 | 数据集大小 | 对象数 | 分布区域 | 数据来源 |
|---|---|---|---|---|
| Data-set1 | 10 000 | 6 | 500×500 | 真实数据 |
| Data-set2 | 30 000 | 6 | 1000×1000 | 模拟数据 |
| Data-set3 | 50 000 | 8 | 1000×1000 | 模拟数据 |
| Data-set4 | 60 000 | 10 | 1000×1000 | 模拟数据 |
| Data-set5 | 70 000 | 15 | 1000×1000 | 模拟数据 |
| Data-sets6 | 10 000 | 6 | 500×500 | 真实数据 |

### 5.2 空间co-location模式增量挖掘基本算法的效率及剪枝算法的效果验证

#### 5.2.1 基本算法和剪枝算法的性能验证

我们将通过改变距离阈值、参与度阈值、数据集大小来详细而彻底地比较传统算法、基本算法和剪枝算法的性能和可扩展性.

##### 5.2.1.1 距离阈值 *d* 对传统算法、基本算法和剪枝算法的影响

本实验是在 Data-set1 真实数据集上进行,*S_change* 大约 *S_new* 的 20%,参与度阈值 min_*prev*= 0.6.图 3 显示了 3 种算法的性能对比结果.从图 3 可以看出:① 3 种算法的性能均随着 *d* 的增大而下降.因为 *d* 增大时,co-location 实例会增加,表实例迅速壮大导致算法性能降低;② 基本算法的性能并没有一直像我们预想的那么好,当 *d* 增大到一定程度时,其性能下降的更快,比传统算法还要差.这是因为 *d* 的增大会导致 *S_change* 中被牵连的实例快速增多(当 *d*=30 时,*S_change* 约占 *S_new* 的 40%),而基本算法对于 *S_change* 求解以及当前表实例的生成都需要耗费时间.那么到底 *S_change* 占 *S_new* 的多少比例,基本算法才会表现得比传统算法更好呢?第 5.2.2

中的实验会给出指导性的建议;③ 相比之下,剪枝算法的效果非常好.通过跟踪剪枝算法,我们发现剪枝算法在二阶模式中很见效.因为模式本身的向下剪枝,导致二阶模式是所有阶模式中数量最多的,即二阶的频繁模式最多.而二阶模式中,很多模式又是不可能发生变化的模式,因此剪枝算法就起到很大的作用.需要重新计算频繁性的二阶模式少了,生成的高阶模式自然就少了;④ 随着 *d* 的增大,剪枝算法没有像传统算法和基本算法性能下降的那么快.

5.2.1.2 参与度阈值 min_*prev* 对传统算法、基本算法和剪枝算法的影响

本实验仍在 Data-set1 真实数据集上进行,*S_change* 大约占 *S_new* 的 20%,距离阈值 *d*=20.图 4 显示了实验结果.从图中可以看出:① 3 个算法的性能均随着 min_*prev* 的减小而下降.因为 min_*prev* 减小时,满足参与度阈值要求的模式增多,导致性能下降;② 由于保证 *S_change* 相对 *S_new* 的比例(20%),所以基本算法的性能明显优于传统算法,剪枝算法的效率更优.

5.2.1.3 不同规模的数据集对传统算法、基本算法和剪枝算法的影响

本实验在数据集 Data-set1~Data-set5 上进行,*S_change* 均约占 *S_new* 的 20%.图 5 显示了实验结果.从图 5 中可以看出,随着数据集的增大,3 种算法的性能均下降,这一点是很明显的结论.此外,在 Data-set5 上实验时,传统算法和基本算法不能在允许的时间内得出结果.因此,剪枝算法的可扩展性更强,可适应大数据环境下的应用.

5.2.1.4 S_change 大小对传统算法、基本算法和剪枝算法的影响

理论分析和前面的实验均表明,基本算法和剪枝算法是基于 *S_change* 求解 *S_new* 的 co-location 频繁模式,因此会对 *S_change* 的大小比较敏感.为进一步验证基本算法和剪枝算法的性能,我们又对传统算法、基本算法和剪枝算法在 *S_change* 增大时进行了实验.到底 *S_change* 占 *S_new* 的多少比例,基本算法才会表现得比传统算法更好?

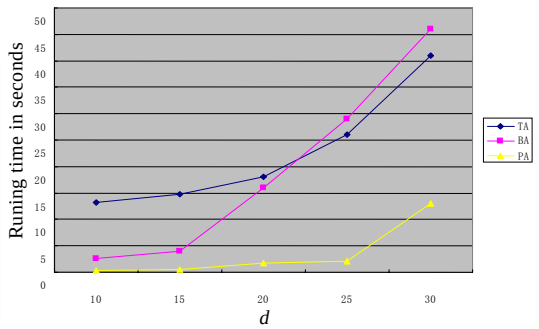



图 3 距离阈值 *d* 对 3 种算法的影响

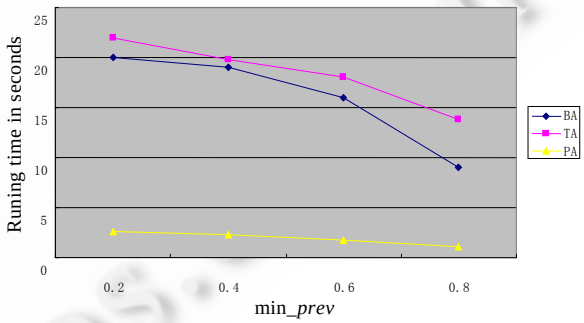



图 4 3 种算法在 min_*prev* 改变时的性能比较

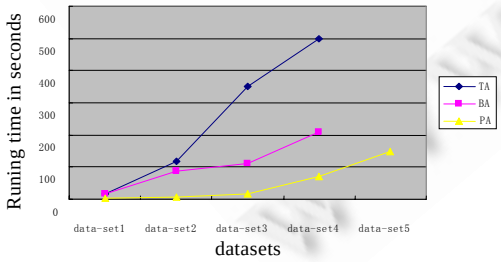



图 5 不同规模数据集对 3 种算法的影响

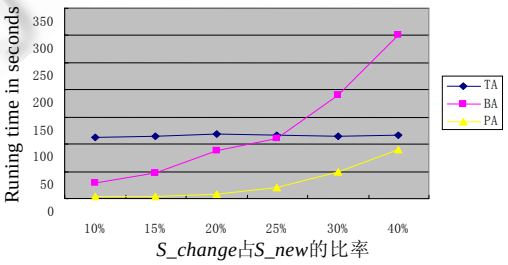



图 6 *S_change* 大小对 3 种算法的影响

实验在 Data-set2 模拟数据集上进行,随机改变 *S_change* 的大小分别为 *S_old* 的 10%,15%,20%,25%,30%,40%.实验结果如图 6 所示.从图中可以看出:随着 *S_change* 的增大,传统算法因对 *S_new* 重新挖掘,其性能不受影响;基本算法的性能下降得很快,从 30%开始,基本算法的性能低于传统算法,并随着 *S_change* 的增大继续恶

化;剪枝算法虽受 *S_change* 的增大影响,但是由于剪枝的高效率,其性能仍远远好于传统算法.

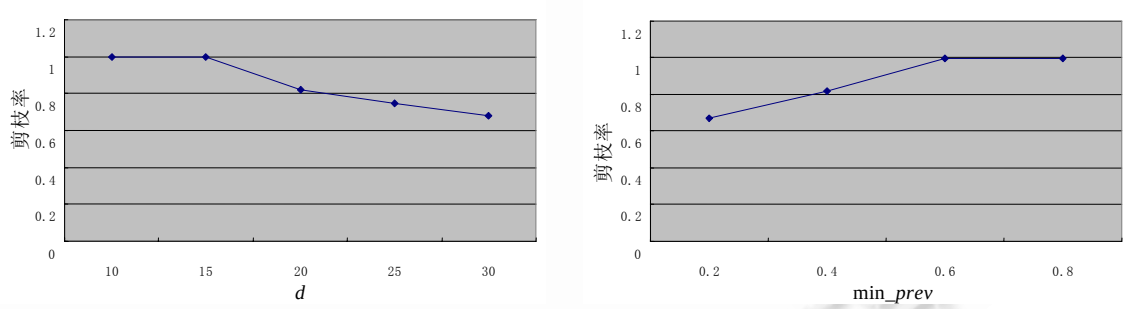
5.2.2 剪枝算法的剪枝率

我们用剪枝率来表示剪枝算法的效果,剪枝率被定义为剪枝算法剪掉的模式数与基本算法需要计算的总模式数的比值.图 7(a)和图 7(b)分别显示了剪枝算法在 *d* 和 min_*prev* 发生变化时的剪枝率.从图中我们可以看出:① 剪枝算法的总体剪枝效果很好,第 5.2.1 节的诸多实验也验证了这一点;② 剪枝率随着 *d* 的增大而下降;③ 剪枝率随着 min_*prev* 的增大而升高.

通过对比 *S_old* 和 *S_new* 上挖掘出来的 co-location 模式,我们得到了变化的模式.例如,在 Data-set1 数据集上,*d*=20,min_*prev*=0.4 时,我们挖掘到了减少的模式,如(长苞冷杉,松茸)、(川八角莲,雪兔子,绵参).

### 5.3 演化模式挖掘算法在真实数据上的应用

Data-sets6 是梅里自然保护区从 2005 年~2008 年的 4 个数据集.有代表性的演化模式挖掘算法的结果显示在图 8 中.A、B、C、D、E、F 代表 6 个对象,连线表示对象间的 co-location 频繁模式.从 2005 的数据中挖掘到模式(B,C,D,E),在 2006 年的挖掘结果中,模式(B,C)、(B,D)、(B,E)消失,原模式变为(C,D,E).2008 年模式(C,D)和(C,E)消失,只剩模式(D,E).原来 2006 年由于大量砍伐导致长苞冷杉(对象 B)减少,之后,与长苞冷杉共生的松茸(对象 C)开始减少,到 2008 年,松茸的大量减少导致模式(C,D)和(C,E)消失.这是一个非常有用的挖掘结果,它显示了由于一种生物被破坏,影响了生物间相互依赖的共生环境,从而导致生物链的改变.




(a) *d* 变化对剪枝率的影响 (b) min_*prev* 变化对剪枝率的影响

图 7 剪枝率

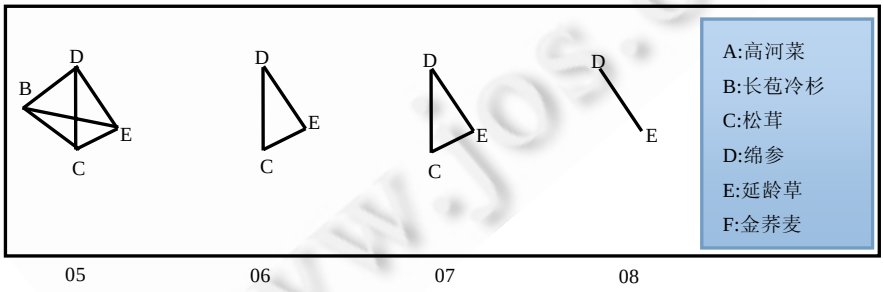



图 8 演化模式挖掘算法在真实数据上的挖掘结果

## 6 总结及未来的工作

本文根据空间 co-location 模式在实际应用中的需要,对空间 co-location 模式进行增量挖掘和演化分析.首先提出了高效的空间 co-location 模式增量挖掘基本算法及剪枝算法.其次,在多个随时间变化的真实数据集上挖掘 co-location 演化模式.最后,证明了空间 co-location 模式增量挖掘基本算法及剪枝算法是正确的和完备的,并用充分的实验验证了增量挖掘算法的性能,剪枝算法的效果.此外,本文还将提出的算法应用到三江并流区域珍稀植物数据集上,增量挖掘出空间 co-location 模式及演化模式,预测了 co-location 模式的演化规律.未来的工

作包括更符合实际应用的高效用空间 co-location 模式挖掘,高效用空间 co-location 模式的演化分析,常变 co-location 模式的挖掘等.

**References**:


[1] Han JW, Kamber M. Data mining: Concepts and techniques. 2nd ed., Beijing: China Machine Press, 2006. 269−274.

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

[3] Yoo JS, Shekhar S. A partial join approach for mining co-location patterns. In: Pfoser D, Cruz IF, Ronthaler M, eds. Proc. the 12th ACM Int'l Workshop on Geographic Information Systems (GIS 04). New York: ACM, 2004. 241−249.

[4] Huang Y, Zhang LQ, Yu P. Can we apply projection based frequent pattern mining paradigm to spatial co-location mining? In: Ho TB, Cheung DW, Liu H, eds. Proc. of the 9th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining (PAKDD 2005). Berlin, Heidelberg: Springer-Verlag, 2005. 719−725.

[5] Yoo JS, Shekhar S, Celik M. A join-less approach for co-location pattern mining: A summary of results. In: Proc. of the 5th IEEE Int. Conf. on Data Mining (ICDM 2005). Washington: IEEE Computer Society, 2005. 813−816.

[6] Wang LZ, Bao YZ, Lu J, Yip J. A new join-less approach for co-location pattern mining. In: He XJ, Nguyen QV, ed. Proc. of the IEEE 8th Int'l Conf. on Computer and Information Technology (CIT2008). Washington: IEEE Computer Society, 2008. 197−202.

[7] Wang LZ, Bao YZ, Lu ZY. Efficient discovery of spatial co-location patterns using the iCPI-tree. The Open Information Systems Journal, 2009,3(1):69−80.

[8] Wang LZ, Zhou LH, Lu J, Yip J. An order-clique-based approach for mining maximal co-locations. Information Sciences, 2009,179(19):3370−3382.

[9] Huang Y, Pei J, Xiong H. Mining co-location patterns with rare events from spatial data sets. GeoInformatica, 2006,10(3): 239−260.

[10] Feng L, Wang LZ, Gao SJ. A new approach of mining co-locaiton patterns in spatial datasets with rare feature. Journal of Nanjing University (Natural Sciences), 2012,48 (1):99−107 (in Chinese with English abstract).

[11] Ouyang ZP, Wang LZ, Chen HM. Mining spatial co-location patterns for fuzzy objects. Chinese Journal of Computers, 2011, 34(10):1947−1955 (in Chinese with English abstract).

[12] Wu PP, Wang LZ, Zhou YH. Discovering co-location from spatial data sets with fuzzy attributes. Journal of Frontiers of Computer and Techology, 2013,7(4):348−358 (in Chinese with English abstract).

[13] Wang S, Huang Y, Wang XY. Regional co-locations of arbitrary shapes. In: Nascimento MA, Sellis T, Cheng R, eds. Advances in Spatial and Temporal Databases-13th Int'l Symp. (SSTD 2013), Berlin, Heidelberg: Springer-Verlag, 2013. 19−37.

[14] Wang LZ, Chen HM, Zhao LH, Zhou LH. Efficiently mining co-location rules on interval data. In: Cao LB, Feng Y, Zhong J, eds. Proc. of the 6th Int'l Conf. on Advanced Data Mining and Applications (ADMA 2010). Berlin, Heidelberg: Springer-Verlag, 2010. 477−488.

[15] Lu Y, Wang LZ, Zhang XF. Mining frequent co-location patterns from uncertain data. Journal of Frontiers of Computer Science and Technology, 2009,3(6):656−664 (in Chinese with English abstract).

[16] Wang LZ, Wu PP, Chen HM. Finding probabilistic prevalent co-locations in spatially uncertain data sets. IEEE Trans. on Knowledge and Data Engineering, 2013,25(4):790−804.

[17] Wang LZ, Guan P, Chen HM, Zhao LH. Mining co-locations from spatially uncertain data with probability intervals. In: Meng XF, Liu H, Kitagawa H, eds. Proc. of the 14th Int'l Conf. on Web-Age Information Management (WAIM 2013). Berlin, Springer-Verlag, 2013. 301−314.

[18] Hong TP, Lin CW, Wu YL. Incrementally fast updated frequent pattern trees. Expert Systems with Applications, 2008,34, 2424−2435.

[19] Cheung DW, Han JW, Ng VT, Wong CY. Maintenance of discovered association rules in large databases: An incremental updating technique. In: Proc. of the 12th Int'l Conf. on Data Engineering. New Orleans: IEEE Computer Society, 1996. 106−114.

[20] Lin CW, Lan GC, Hong TP. An incremental mining algorithm for high utility itemsets. Expert Systems with Applications, 2012,39: 7173−7180.


**附中文参考文献**:


[10] 冯岭,王丽珍,高世健.一种带稀有特征的空间 Co-location 模式挖掘新方法.南京大学学报(自然科学),2012,48(1):99−107.

[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.计算机学报,2011,34(10):1947−1955.

[12] 吴萍萍,王丽珍,周永恒.带模糊属性的空间 Co-Location 模式挖掘研究.计算机科学与探索,2013,7(4):348−358.

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

**芦俊丽**(1982－),女,黑龙江安达人,博士生,副教授,主要研究领域为空间数据挖掘与算法;
E-mail: ljl1982_3_6@126.com

**王丽珍**(1962－),女,教授,博士生导师,主要研究领域为数据库,数据挖掘,计算机算法.
E-mail: lzhwang@ynu.edu.cn

**肖清**(1975－),女,讲师,主要研究领域为空间数据挖掘.
E-mail: xiaoqing@ynu.edu.cn

**王新**(1963－),男,教授,主要研究领域为数据挖掘,软件工程.
E-mail: wxkmyn@163.com