基于特征效用参与率的空间高效用 co-location 模式挖掘方法

王晓璇 王丽珍 陈红梅 方 圆 杨培忠

(云南大学信息学院 昆明 650504)

摘 要 空间 co-location 模式是指其实例在空间邻域内频繁一起出现的空间特征子集.与传统的空间 co-location 模式挖掘不同,在空间高效用 co-location 模式挖掘中,不再将参与度(PI)作为有趣模式的度量指标,而是将效用值作为挖掘有趣模式的兴趣度量指标.现有的空间高效用 co-location 模式挖掘方法分为特征带效用和实例带效用两类.特征带效用的现有方法没有考虑不同特征效用之间的差异,挖掘的结果往往包含了许多不尽合理的"高效用"模式;而实例带效用的现有方法,则考虑了不同特征对模式效用的影响,但没有客观地度量这种影响.该文提出了一种确定特征在模式中的效用权重 $\omega(f_i,c)$ 的方法,定义了更为合理的空间高效用 co-location 模式概念,设计了一个有效的挖掘算法.大量的实验表明提出的高效用 co-location 模式度量方法和相应的挖掘算法能够处理特征效用差异性和特征间的相互影响问题,能更有效地挖掘到空间高效用 co-location 模式.

关键词 空间数据挖掘;空间 co-location 模式;高效用;效用权重;数据挖掘

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2019.01721

Mining Spatial High Utility Co-location Patterns Based on Feature Utility Ratio

WANG Xiao-Xuan WANG Li-Zhen CHEN Hong-Mei FANG Yuan YANG Pei-Zhong

(*School of Information Science and Engineering, Yunnan University, Kunming* 650504)

Abstract A spatial co-location pattern is a subset of spatial features which shows frequent association relationships based on the spatial neighborhood. Different from the participation index (PI) which is regarded as a measure of interests in traditional spatial co-location pattern mining, the utility of co-location pattern is considered as the measure of interests in the spatial high utility co-location pattern mining. The purpose of the spatial high utility co-location pattern mining is to compensate for the knowledge omission in traditional co-location pattern mining, and the high utility co-location can reflect the interactions between different spatial features or different spatial instances. It is noteworthy that there are lots of differences between the traditional high utility pattern mining and the spatial high utility co-location pattern mining. Firstly, due to the specificity of different spatial features, it is irrational to measure the utilities of different spatial instances using a unified standard. Second, although the high utility pattern mining technology in traditional databases is very mature, these techniques cannot be directly applied to spatial high utility co-location pattern mining, because the prevalence of spatial co-location patterns, which is completely different from the itemsets in transaction database, is measured by the cluster relations formed by the proximity relationships. So far, the existing methods of the spatial high utility co-location pattern

收稿日期:2017-09-27;在线出版日期:2018-04-19.本课题得到国家自然科学基金项目(61472346,61662086)、云南省自然科学基金项目(2016FA026,2015FB114)、云南省创新团队项目资助.**王晓璇**,博士研究生,主要研究方向为空间数据库、数据挖掘. E-mail:wangxiaoxuan1037@163.com. **王丽珍**(通信作者),博士,教授,博士生导师,主要研究领域为数据库、空间数据挖掘,计算机算法. E-mail:lzhwang2005@126.com. **陈红梅**,博士,副教授,主要研究方向为数据库、空间数据挖掘. **方 圆**,博士研究生,主要研究方向为空间数据库、数据挖掘. **杨培忠**,硕士研究生,主要研究方向为空间数据库、大数据.

mining can be classified into two classes: spatial features with utilities and spatial instances with utilities. The spatial features with utilities method considers the utilities of different features, which will weaken the value of the features that are highly participated but relatively low-valued in the co-location pattern. The interactions of the features in a co-location pattern are very important to evaluate a co-location, but the method of spatial features with utilities has not considered this important factor. Moreover, this method has also not considered the different utility measure standards of different spatial features, the utilities of spatial features are added directly. In the method of spatial instances with utilities, the utility participation ratio (UPR) is calculated and used to evaluate the features' utility value in a co-location pattern. At the same time, it considers the influences between the different features. However, the method requires that the UPR of all features in a co-location pattern is greater than a specified utility threshold, which leads to some high utility co-location patterns are missed, so the method can not measure the spatial high utility co-location patterns objectively. In summary, the former does not consider the difference of different features' utilities which lead to mine some unreasonable patterns, and the latter considers the influence of different features, but the different influence has not been measured objectively. This paper proposes an efficient method to measure the utility weight $\omega(f_i, c)$ of different features in a spatial co-location pattern, which can reflect the interaction of features in the co-location pattern more correctly, and define a more reasonable concept of spatial high utility co-location pattern. A basic mining algorithm and two efficient pruning algorithms are designed. Experimental results show that the method proposed in this paper can help users to mine the useful, reasonable and interesting spatial high utility co-location patterns.

Keywords spatial data mining; spatial co-location pattern; high utility; utility weight; data mining

1 引 言

空间数据挖掘技术旨在从海量的空间数据中获取有效的、具有指导性和预测性的知识. 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 模式是空间特征的一个子集,它们的实例在地理空间频繁地邻近,例如在公共交通领域,交通拥堵、车祸现场和警察的出现频繁并置(co-located). 在频繁 co-location 模式挖掘体系中,一般以特征实例参与并置的程度度量 co-location 模式的有趣性. 它对特征实例内部的效用价值并没有进行区别和研究. 这种传统的频繁 co-location 模式挖掘方法容易忽视那些不频繁出现,但是却至关重要的模式.

高效用模式的提出旨在弥补传统模式挖掘中的知识遗漏,它能找出更有价值的模式来指导科学决策,提取隐含的预测性信息. 例如传统事务模式{钻石,项链}可能在某个数据集中并不是一个频繁的模式,但它的经济价值要高于频繁模式{牙膏,牙刷}. 在研究销售价值贡献时,如果仍然挖掘频繁模式的话,{钻石,项链}这种具有较大效用价值的模式将会被忽略. 同样,空间高效用 co-location 模式挖掘的提出,也在很大程度上弥补了频繁 co-location 模式挖掘所造成的信息遗漏. 空间高效用 co-location 模式体现出的是多种空间特征或实例聚集在一起时的影响与作用,反映出的是一种高质量的聚集规律. 所以,空间高效用 co-location 模式挖掘可以为城市规划、商业建筑群设计、经济作物种植规划等提供决策支持.

与经典的事务数据库中的高效用模式不同,由于空间特征的特异性,使用统一的效用度量标准来衡量不同空间实例的效用价值,并不是在任何的空间 co-location 模式中都适用. 例如,在事务数据库中{面包,牛奶}可以通过价格这个度量标准来对其效用进行分析,但在 co-location 模式{小区,学校}中,小区与学校是两种具有不同社会功能的空间特征,如果通过占地面积来统一衡量他们的效用显然是不合理,因为学校更适合用教学质量或等级来衡量其价值. 其次,虽然高效用模式的挖掘在事务数据库中已经较为成熟,但由于空间 co-location 模式是通过邻近关系形成的团关系来度量的,与事务数据库

中的项集大不相同，所以现有的事务高效用模式挖掘方法并不适用于空间高效用 co-location 模式的挖掘. 因此空间高效用 co-location 模式挖掘还需要不断的探索与研究. 目前空间高效用 co-location 模式挖掘主要分为以下两种方法：

(1) 特征带效用. 在文献[1]中提出的方法，通过将模式 c 中的各个特征的价值(同一特征的实例价值相同)$V(f_i)$相加得到整个模式 c 的效用值，表示为 $u(c)=\sum_{f_i\in c}V(f_i)$，通过计算 $u(c)$与整个数据库的总效用 $U(S)$的比值来衡量模式 c 是否是一个高效用模式；

(2) 实例带效用. 在文献[2]中考虑了每个实例的不同价值，并且考虑模式中特征之间的相互影响. 文献[2]利用内部效用率 $IntraUR(f_i,c)=\dfrac{u(f_i,c)}{u(f_i)}$来反映特征 f_i自身的效用参与程度，然后利用外部效用率 $InterUR(f_i,c)=\dfrac{\sum_{f_i\in c,j\neq i}u(f_j,c)}{\sum_{f_i\in c,j\neq i}u(f_j)}$来反映在模式 c 中其它特征对 f_i的效用影响. 定义特征效用参与率 $UPR(f_i,c)=\omega_1\times IntraUR(f_i,c)+\omega_2\times InterUR(f_i,c)$($\omega_1,\omega_2$由人为指定)，最后通过计算特征效用参与度 UPI 来衡量高效用 co-location 模式的兴趣度，$UPI(c)=\min\{UPR(f_i,c),f_i\in c\}$，当 $UPI(c)$大于给定的效用阈值时，模式 c 可判定为高效用模式.

空间模式的效用价值如何来度量是一个非常值得思考的问题. 尽管上述两种方法都给出了相应的度量方式，但仍然存在一些问题. 首先，方法(1)没有考虑特征之间的效用值相差极大时，简单的累加各个特征参与的效用会削弱参与度极高但自身价值相对较低的特征在模式中的价值作用. 例如，假设枫香在模式中的总价值为 100 000，而高山松的总价值却仅为 100，若直接相加，模式{枫香，高山松}的效用值为 100 100. 假设枫香树的实例数是 5(棵)，在模式中只参与了 2(棵)，高山松的 10 个实例(10 棵)都参与到了模式中. 虽然高山松在模式中的价值相对较小，但是它的所有实例都参与到了模式中，应用方法(1)计算模式效用时，忽略了高山松的高参与度. 问题出现的原因在于简单的对不同特征的效用值进行相加，忽视了特征效用值间的差异性以及特征实例的参与程度. 其次，方法(1)中还存在的一个问题就是当两个特征的价值单位不同，直接把效用值相加将更加的不合理，例如 co-location 模式{小区，公园}，假设居住小区的价值单位是房价，而公园的价值单位是面积. 这样直接相加计算模式的效用是完全不可思议的. 而且在一个模式中，由于特征价值和特征参与度的不同，肯定存在不同的效用贡献度，也就是应该有不同的效用权重，然而方法(1)并没有对模式中不同特征的效用权重进行计算和区分，所以特征对模式的效用贡献度都是相同的，使得所计算得到的模式效用不够合理有效.

方法(2)首先存在的一个问题就是在计算外部效用率 $InterUR(f_i,c)=\dfrac{\sum_{f_i\in c,j\neq i}u(f_j,c)}{\sum_{f_i\in c,j\neq i}u(f_j)}$时，仍然将每个特征的效用直接加和，这和方法(1)中存在的不同单位不可加和的问题是一样. 即使单位相同，也仍然存在效用相差极大而造成的问题. 其次，在方法(2)中虽然考虑特征效用的参与程度，但它的问题是过度地要求高的特征效用参与度，因为该方法要求模式中所有特征的效用参与率 $UPR(f_i,c)$都必须大于给定的阈值，这样的模式才是高效用模式. 另外，在方法(2)中，尽管在计算某个特征的效用参与率时，考虑了其它特征对该特征的影响，但影响的权重是人为设定的，且是不随特征的改变而改变的，并不能较好地反映模式中各特征效用的真实情况，例如 co-location 模式{小区，公园，学校}，{公园，学校}对小区的影响程度与{小区，公园}对学校的影响程度是不同的. {公园，学校}的邻近对住房小区的房价有很大的影响，而{小区，公园}对学校教学质量的影响是非常小的. 虽然特征之间的效用影响是必须的考虑的，但简单的由用户给定权重来表示特征的效用贡献，对于挖掘高效用模式是不够的，在某些情况下还可能会流失一部分有趣的高效用模式. 所以本文提出了一种基于特征效用参与率的特征效用权重计算方法，有效改善了上述两种挖掘方法存在的问题. 主要贡献总结如下：

(1) 提出了一种更为合理的空间高效用 co-location 模式的度量方法. 该度量通过计算特征在模式中的效用参与率，确定特征在模式中的权重，模式的效用是特征的加权效用和，这样不仅能够考虑特征本身，还能将特征之间的影响也反映在模式的效用中；

(2) 提出了基于最小特征效用参与率剪枝算法和基于最大特征效用参与率剪枝算法，讨论了基于最小特征效用参与率剪枝在密集数据集上失效的原因，并通过引入团实例邻居集计算方法，进一步优化了基于最大特征效用参与率剪枝算法；

(3) 在模拟数据集和真实数据集上验证了所提

出的度量方法的合理性,证明了提出的挖掘算法的正确性和有效性,同时在密集与稀疏的数据集上讨论了剪枝算法的效果.

2 相关工作

自 Han 等人奠定了事务数据库中频繁模式挖掘的基础[3-4]以及提出较好的数据处理方法[5]后,数据挖掘就衍生了不同种类不同类别的研究方向,如链接预测[6]、序列数据挖掘[7-9]以及空间数据挖掘[10-11]等.空间 co-location 模式挖掘是空间数据挖掘领域的一个重要研究方向.最早在文献[10]中对实例频繁邻近问题进行了定义.文献[11]提出了 co-location 模式的有趣性度量指标最小参与率,即参与度(PI).由于 PI 满足向下闭合性质,可应用类似 Apriori 的方法进行有效挖掘,各种基于 PI 的 co-location 模式挖掘算法被提出,如文献[11]的基于连接操作的 join-based 算法、文献[12]的基于星型邻居的 join-less 算法等.文献[13]基于密度加权来度量空间实例间的邻近关系,从而挖掘不同的 co-location 模式.文献[14] 基于模式间语义距离概念形式化定义了非冗余 co-location 模式.文献[15]对模糊 co-location 模式进行挖掘,并分别从剪枝对象,减少实例间连接,改进剪枝步等几个方面提出了剪枝算法.文献[16]提出隐含在空间 co-location 模式中的竞争关系概念及挖掘方法.文献[17]使用了 MapReduce 框架并行地来挖掘空间频繁模式,以应对庞大的空间数据量.而当空间模式挖掘应用于城市规划时,由于实例存在可达或不可达等特点,文献[18]提出了一种新的频繁性度量.为有效地减少计算量以提高空间模式挖掘效率,文献[19]提出了一种快速空间存取算法以提高极大 co-location 模式挖掘效率.文献[20]通过查询团的方式改进原有 join-less 算法的效率.

高效用模式挖掘是将事务或对象的价值作为有趣性度量指标.目前已经有很多针对事务数据库的高效用模式挖掘方法[21-26].文献[21]首次提出了高效用的概念.由于高效用模式的挖掘不满足频繁模式挖掘中的向下闭合性质,所以基于效用值的挖掘比频繁模式挖掘更具挑战性.文献[22]提出允许用户使用效用值来量化项目集的偏好度,来衡量项目集的效用价值,文献[23]通过定义一个统一的效用公式,提出了统一的事务数据库效用挖掘框架.文献[25]提出了一种基于树形结构的高效用模式挖掘算法.此外还有很多关于高效用挖掘的算法:如 UP-growth 算法[26]以及 two-phase 算法[24]等.文献[27]讨论了在动态的事务数据库中如何挖掘高效用模式.由于高效用模式挖掘的计算复杂性较高,所以目前的研究主要通过以下几个方面来提高高效用模式挖掘的效率:(1)减少候选模式的生成[28];(2)提出较好的剪枝策略对候选模式进行剪枝[29];(3)设计较好的数据结构,便于效用的计算以及候选剪枝,例如 UP-Hist tree 算法[30].

空间高效用 co-location 模式挖掘将 co-location 模式挖掘理论与事务数据库高效用模式挖掘理论相互结合,对频繁邻近出现且效用值较高的模式进行挖掘.目前的研究成果可分为两类:基于特征带效用和基于实例带效用的高效用 co-location 模式挖掘.文献[1]首次提出了基于特征带效用的空间高效用 co-location 模式挖掘,通过计算每个模式效用与整个数据库效用的比率来确定该模式是否是高效用模式.基于文献[1]的理论体系,文献[31]和文献[32]在动态空间数据库中研究了高效用 co-location 模式挖掘.文献[2]提出了基于实例带效用的高效用 co-location 模式挖掘,并且考虑了特征间的相互影响,定义了效用参与率和效用参与度等概念.在上述两类方法中,文献[1]中的方法首先固定每个特征的价值,特征在模式中的效用就是特征价值与参与模式的实例个数的乘积,然后将这些特征的效用相加得到模式的总效用.这种简单地将各个特征在模式中的效用相加的方法,没有考虑到特征间的特异性和可加性.而文献[2]中的方法注重每个实例的参与性,度量方法沿用了传统频繁 co-location 模式参与度的度量形式,忽略了一些效用值很小,但对模式的效用贡献很大(参与度大)的特征.本文提出一种新的高效用 co-location 模式挖掘新概念,较好地解决上述两类方法中存在的问题.

3 传统 co-location 模式挖掘相关概念

在空间数据库中,不同的空间特征代表了不同类型的空间对象,通常用 $F=\{f_1,f_2,\cdots,f_n\}$ 来表示空间数据库中出现的特征集合.在空间数据库中所挖掘到的空间 co-location 模式 c 是 F 的一个子集,$c\subseteq F$.一个模式 c 的长度称为模式 c 的阶,例如 {A,B,C}是一个 3 阶 co-location 模式.在空间数据库中,每个特征都包含了许多属于该特征的空间实例,他们分布在空间中不同的位置.如图 1 中 A.1 是空间特征 A 的一个实例.通常用一个三元组〈特征名,实例编号,地理位置〉表示一个空间实例.如果两

个空间实例的欧几里得距离满足用户给定的距离阈值 d，那么就称这两个空间实例满足邻近关系，在图中用实线连接，如图 1 中 A.2 和 B.1.

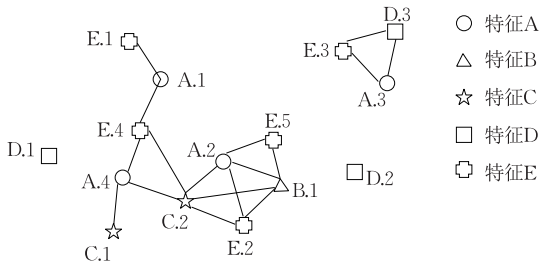


图 1　空间特征及其实例示例

设有一个空间实例集 $I=\{o_1,o_2,\cdots,o_m\}$，如果 I 中两两实例都满足邻近关系，那么就称 I 是一个团实例. 如果团实例 I 包含了 co-location 模式 c 中的所有特征，并且 I 中没有任何一个子集包含 c 中所有的特征，那么 I 被称为模式 c 的一个行实例，co-location 模式 c 的所有行实例的集合称为 co-location 模式 c 的表实例，记为 *table_instance*(c).

如图 1 中，模式{A,B,E}的表实例为 *table_instance*({A,B,E})={{A.2,B.1,E.2},{A.2,B.1,E.5}}，那么特征 A 参与在 co-location 模式{A,B,E}中的实例就是{A.2}，特征 B 为{B.1}，特征 E 为{E.2,E.5}. 表实例给出了模式中空间特征并置(共存)的状态.

传统 co-location 模式挖掘使用参与度 *PI* (Participation Index)来度量一个模式的有趣性，而 *PI* 是参与率 *PR*(Participation Ratio)的最小值. 设 f_i 为某个空间特征，f_i 在 co-location 模式 c 中的参与率定义为 f_i 的实例在 c 的表实例中不重复出现的个数与 f_i 总实例个数的比率，表示为

$$PR(f_i,c)=\frac{\left|\prod_{f_i} table\_instance(c)\right|}{f_i\text{的实例总数}},$$

co-location 模式 c 的参与度 *PI* 定义为模式 c 的所有空间特征的 *PR* 值中的最小值：$PI(c)=\min_{i=1}^{k}\{PR(f_i,c)\}$，当 $PI(c)$ 大于用户所给的最小参与度阈值 *min_prev*，那么这个模式就被称为频繁模式. 在图 1 中，模式{A,E}的参与度 $PI(\{A,E\})=\min\{2/4,2/2\}=2/4$，若参与度阈值为 0.2，则模式{$A$,$E$}是一个频繁模式.

特征在模式中的参与率给出了这个特征实例与模式中其它特征实例并置的概率，而模式的参与度则给出了模式中任何特征实例出现的近邻内其它特征实例出现的概率.

4　高效用 co-location 模式相关定义及性质

本节首先对带效用值的空间实例，模式中特征的效用参与率，特征的效用权重和模式的效用值等进行定义，然后给出特征效用权重满足的性质.

4.1　特征效用参与率 *FUR* 及效用权重

定义 1.　带效用的空间实例[2]. 带有效用 v 的空间特征 f_i 的第 j 个实例记为 $f_i.j^v$，或将实例 $f_i.j$ 的效用记为 $u(f_i.j)=v$.

例如，图 2 中带效用的实例 $A.4^5$，它的效用值为 $u(A.4)=5$.

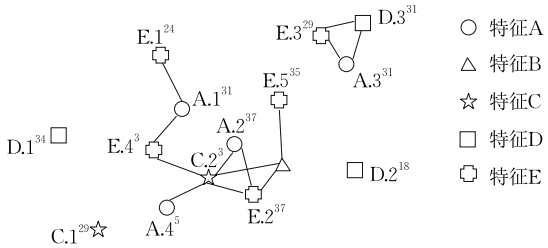


图 2　带效用值的空间实例示例 1

定义 2.　特征效用参与率(Feature Utility Ratio). 给定一个 k 阶 co-location 模式 $c=\{f_1,f_2,\cdots,f_k\}$，特征 f_i 在模式 c 中的效用参与率 $FUR(f_i,c)$ 定义为 f_i 的实例在模式 c 的表实例中不重复出现的实例效用值之和与 f_i 中所有实例的总效用的比率，即(其中 $\prod$ 是投影操作)：

$$FUR(f_i,c)=\frac{\sum_{f_i.j\in\prod_{f_i} table\_instance(c)} u(f_i.j)}{U(f_i,S)},$$

$U(f_i,S)=\sum_{f_i.j\in S} u(f_i.j)$ 表示数据库中 f_i 的所有实例的效用总和.

例如，图 2 中，模式{A,C}的表实例 *table_instance*({A,C})={{$A.4^5$,$C.2^3$},{$A.2^{37}$,$C.2^3$}}，所以特征 A 的特征效用参与率：$FUR(A,\{A,C\})=(5+37)/(31+37+31+5)=0.4$.

FUR 的计算是借鉴频繁模式 co-location 挖掘中的特征参与率的定义，这样的计算方法的优点有两个：(1) 消除了不同特征不同价值单位所造成的效用不可加的问题；(2) 在考虑特征效用的同时，也考虑了特征的参与度，可以综合地来评判特征在模式中综合的效用价值. *FUR* 与 *PR* 的不同点在于：*PR* 反映特征实例的参与程度，而 *FUR* 反映的是特征实例的效用参与程度.

引理 1. 特征效用参与率的不增性：特征 f_i 的效用参与率 FUR 随着模式 c 的阶数的增大而不增.

证明. 根据定义 2 有 $FUR(f_i,c)=\dfrac{\sum_{f_i,j\in\prod_{f_i}table\_instance(c)}u(f_i,j)}{U(f_i,S)}$，如果 $c'\supset c$，由表实例的定义可知：$\prod_{f_i}(table\_instance(c'))\subseteq\prod_{f_i}(table\_instance(c))$，所以 $\sum_{f_i,j\in\prod_{f_i}table\_instance(c)}u(f_i.j)\geq\sum_{f_i,j\in\prod_{f_i}table\_instance(c')}u(f_i.j)$，由于 $U(f_i,S')$ 不变，所以 $FUR(f_i,c)\geq FUR(f_i,c')$. 证毕.

文献[2]中考虑了不同特征间的相互影响，即一个特征在模式中的效用不能只考虑自身的效用，同时还要考虑其它特征效用对该特征的贡献. 对于一个模式来说，不同特征的效用参与率是不同的，也就是说不同特征在一个模式中的效用贡献是不同的，在文献[2]中这个贡献度是人为设定的，本文提出一种较为简单且合理的确定特征贡献度的计算方法以替换人为给定.

定义 3. 特征效用权重. 给定一个 co-location 模式：$c=\{f_1,f_2,\cdots,f_k\}$，特征 f_i 在模式 c 中的效用权重(效用贡献度)表示为 $\omega(f_i,c)$，定义如下：

首先，计算特征 $f_i(i=1,\cdots,k)$ 与 c 中其它特征 f_j 的效用参与率比值 $RF(f_i,f_j,c)=\dfrac{FUR(f_j,c)}{FUR(f_i,c)}$，然后，用下面的公式来计算特征 f_i 在模式 c 中的效用权重：

$$\omega(f_i,c)=\frac{RF(f_i,f_i,c)}{\sum_{f_j\in c}RF(f_i,f_j,c)},$$

其中 $RF(f_i,f_i,c)=\dfrac{FUR(f_i,c)}{FUR(f_i,c)}=1$. 根据效用参与率比值，效用权重的公式可以进行如下的化简：

$$\omega(f_i,c)=\frac{\frac{FUR(f_i,c)}{FUR(f_i,c)}}{\sum_{f_j\in c}\frac{FUR(f_j,c)}{FUR(f_i,c)}}=\frac{\frac{FUR(f_i,c)}{FUR(f_i,c)}}{\frac{\sum_{f_j\in c}FUR(f_j,c)}{FUR(f_i,c)}}=\frac{FUR(f_i,c)}{\sum_{f_j\in c}FUR(f_j,c)}.$$

例如，对于图 2 中的模式{A,C}，可计算得到 A 的权重为

$$\omega(\mathrm{A},\{\mathrm{A},\mathrm{C}\})=\frac{FUR(\mathrm{A},\{\mathrm{A},\mathrm{C}\})}{FUR(\mathrm{A},\{\mathrm{A},\mathrm{C}\})+FUR(C,\{\mathrm{A},\mathrm{C}\})}=\frac{0.4}{0.4+0.094}=0.81;$$

C 的权重为

$$\omega(\mathrm{C},\{\mathrm{A},\mathrm{C}\})=\frac{FUR(\mathrm{C},\{\mathrm{A},\mathrm{C}\})}{FUR(\mathrm{A},\{\mathrm{A},\mathrm{C}\})+FUR(\mathrm{C},\{\mathrm{A},\mathrm{C}\})}=\frac{0.094}{0.4+0.094}=0.19.$$

引理 2. 在一个 k 阶 co-location 模式 c 中，所有特征的权重相加为 1，即：

$$\omega(f_1,c)+\omega(f_2,c)+\cdots+\omega(f_k,c)=1.$$

证明. 根据特征权重的计算公式可以得到：

$$\omega(f_i,c)=\frac{RF(f_i,f_i,c)}{\sum_{f_j\in c}RF(f_i,f_j,c)}=\frac{\frac{FUR(f_i,c)}{FUR(f_i,c)}}{\sum_{f_j\in c}\frac{FUR(f_j,c)}{FUR(f_i,c)}}=\frac{FUR(f_i,c)}{\sum_{f_j\in c}FUR(f_j,c)},$$

所以有：

$$\omega(f_1,c)+\omega(f_2,c)+\cdots+\omega(f_k,c)=\frac{FUR(f_1,c)}{\sum_{f_j\in c}FUR(f_j,c)}+\frac{FUR(f_2,c)}{\sum_{f_j\in c}FUR(f_j,c)}+\cdots+\frac{FUR(f_k,c)}{\sum_{f_j\in c}FUR(f_j,c)}=\frac{\sum_{f_j\in c}FUR(f_j,c)}{\sum_{f_j\in c}PFR(f_j,c)}=1.$$ 证毕.

4.2 模式效用度 *PUI*

有了特征效用率和效用权重，就可以来定义空间高效用的效用计算公式.

定义 4. 模式效用度(Pattern Utility Index). 一个 co-location 模式 $c=\{f_1,f_2,\cdots,f_k\}$ 的模式效用度表示为 $PUI(c)$，是模式中所有特征的效用参与率 FUR 与权重乘积的总和，公式表示如下：

$$PUI(c)=\sum_{f_i\in c}\omega(f_i,c)\times FUR(f_i,c).$$

给定一个效用度阈值 ξ，当模式 c 的模式效用度 $PUI(c)\geq\xi$，那么称这个模式 c 就是一个高效用模式.

例如，图 2 中的模式{A,C}的模式效用度为

$$PUI(\{\mathrm{A},\mathrm{C}\})=\omega(\mathrm{A},\{\mathrm{A},\mathrm{C}\})\times FUR(\mathrm{A},\{\mathrm{A},\mathrm{C}\})+\omega(\mathrm{C},\{\mathrm{A},\mathrm{C}\})\times FUR(\mathrm{C},\{\mathrm{A},\mathrm{C}\})=0.8\times0.4+0.2\times0.094=0.32+0.0188=0.3388.$$

若 $\xi=0.3$，模式{A,C}就是一个高效用模式.

在此首先说明与事务数据库中的高效用挖掘类似，我们定义的模式效用度 PUI 不满足向下闭合性质.

引理 3. 模式效用度 PUI 不满足向下闭合

性.即 k 阶模式 c 的 $PUI(c)$ 不一定大于它的所有超模式的 $PUI(c')$(c'是 c 的超模式).

证明.　举反例.假设效用度阈值为 0.5,从图 3 中我们可以看到:模式{C,E}不是高效用模式,因为 $PUI(\{C,E\})=0.26$,但它的高阶模式{C,B,E}是高效用模式,因为 $PUI(\{B,C,E\})=0.7896$.可见,模式效用度 PUI 并不满足向下闭合性.　　证毕.

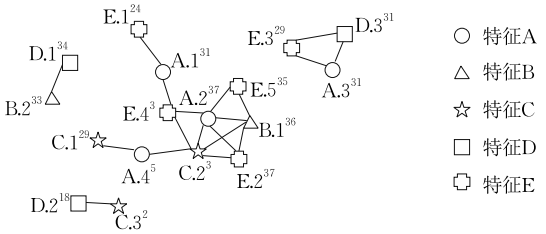


图 3　带效用值的空间实例示例 2

定理 1.　当模式 $c=\{f_1,f_2,\cdots,f_k\}$ 中的所有特征的效用参与率都小于 ξ 时,该模式一定不是高效用模式,即如果 $\forall f_i\in c$,都有 $FUR(f_i,c)<\xi$,那么模式 c 一定不是高效用模式.

证明.　若 $FUR(f_i,c)<\xi$,那么 $\omega(f_i,c)\times FUR(f_i,c)<\omega(f_i,c)\times\xi$,则有:

$$PUI(c)=\sum_{f_i\in c}\omega(f_i,c)\times FUR(f_i,c)$$

$$=\omega(f_1,c)\times FUR(f_1,c)+\cdots+\omega(f_k,c)\times FUR(f_k,c)$$

$$<\omega(f_1,c)\times\xi+\cdots+\omega(f_k,c)\times\xi$$

$$=(\omega(f_1,c)+\cdots+\omega(f_k,c))\times\xi$$

$$=\xi.$$

所以,当模式 c 中所有特征的效用参与率都小于 ξ 时,该模式的效用值一定小于 ξ,模式 c 一定不是高效用模式.　　证毕.

例如,图 2 中的模式{A,D},如果 $\xi=0.4$,$FUR(A,\{A,D\})=31/104=0.3$,$FUR(D,\{A,D\})=31/83=0.37$,所以模式{A,D}不是高效用模式.

表 1 总结了本文所提效用度量方法(PUI)对原有方法(文献[1]和文献[2]方法)存在缺陷的弥补情况及优势.

表 1　效用度量方法对比

	特征带效用(文献[1])	实例带效用(文献[2])
计算方式缺陷	1. 没有考虑特征间的相互影响 2. 不能体现不同特征在模式中不同的重要性	1. 人为规定特征的权重 2. 特征间相互影响的考虑过于笼统
PUI 是否弥补	是	是
PUI 的优势	1. 引入特征权重概念及计算方法,解决了人为给定权重的问题 2. 模式效用度 PUI 有效地考虑了一个模式中特征间的相互影响,解决了文献[1]和[2]中存在的问题	

5　挖掘算法

在传统的频繁 co-location 模式挖掘中,由于 co-location 模式满足先验原理,所以当模式 c 不是频繁模式时,它的所有超模式都将不是频繁模式,很快就能完成剪枝.剪枝的目的是希望能够减少不必要的计算,如果在计算效用模式度之前,就可以判断模式 c'不是高效用模式,那么以后的计算中可以不再对模式 c'进行计算(c'为模式 c 的超模式).但由于模式效用度 PUI 不满足向下闭合性,因此如何有效地剪枝以提高空间高效用 co-location 模式挖掘的效率是算法设计的关键.

本节中,根据 co-location 模式的挖掘特性以及高效用的性质,提出了基于特征效用率的空间高效用模式挖掘基础算法 HUCMA(High Utility Co-location Ming Algorithm).由于空间高效用 co-location 模式不满足向下闭合性,在本节中又给出了一个最小特征效用参与率剪枝算法 MinFURA(Min Feature Utility Ratio Algorithm).接着讨论了最小特征效用参与率剪枝的失效以及失效原因,然后提出了一个最大特征效用参与率剪枝算法 MaxFURA(Max Feature Utility Ratio Algorithm).

5.1　基于特征效用率的空间高效用模式挖掘基础算法(High Utility Co-location Ming Algorithm,HUCMA)

算法 1.　基于特征效用率的空间高效用模式挖掘基础算法(HUCMA).

输入:空间特征集 F,带效用实例集 I,距离阈值 d,效用度阈值 ξ

输出:高效用模式集 U_{high};

步骤:

BEGIN

1. $k=2$,基于距离阈值 d 计算 2 阶模式的表实例,所有 2 阶模式形成 2 阶候选 C_k.
2. WHILE $C_k<>$NULL
   - 2.1 FOR EACH $c\in C_k$
     - 2.1.1 计算每个特征的效用值,计算特征的效用参与率 $FUR(f_i,c)$;
     - 2.1.2 通过效用参与率确定每个特征的效用权重 $\omega(f_i,c)$;
     - 2.1.3 计算模式 c 的模式效用度 $PUI(c)$;
   - 2.2 生成 $k+1$ 阶的模式 C_{k+1}.
   - 2.3 $k=k+1$;
3. 输出高效用集合 U_{high}.

END

由于高效用模式不满足向下闭合性,在基础算

法中需要对所有的模式都进行效用的计算与判断，所以基础算法虽然保证了高效用模式挖掘的正确性，但也造成了很大的计算量.为了减少计算量，提高挖掘效率，提出了最小特征效用参与率的剪枝算法(MinFURA).

5.2 最小特征效用参与率剪枝算法(Min Feature Utility Ratio Algorithm，MinFURA)

定义 5. 相关模式集 $S_c(c)$. k 阶模式 c 的相关模式集 $S_c(c)$由与 c 的交集特征数等于 $k-1$ 的所有 k 阶模式组成.同时称特征 f_i 为模式 c 的相关特征，如果 $f_i \in c'$，$(c' \in S_c(c))$且 $f_i \notin c$.

例如，图 2 中模式{A，D}的相关模式集为：$S_c(\{A,D\})=\{\{A,E\},\{A,C\},\{D,E\}\}$(还有{A，B}和{B，D}，但因为它们的表实例为空，故而丢弃)，那么相关特征就是 E 和 C.

定理 2. 如果一个 k 阶模式 c 中 $\forall f_i \in c$，都有 $FUR(f_i,c)<\xi$，且 c 的相关特征在 $S_c(c)$中的效用参与率的最小值中的最大值小于 ξ，那么，模式 c 的所有高阶模式不可能是高效用模式，可以直接剪枝.

证明. 假设 $k+1$ 阶模式 $c'=c\cup\{f\}$，f 是 c 的相关特征中在 $S_c(c)$中的效用参与率的最小值中是最大的且小于 ξ，则有：

$$\begin{aligned}PUI(c')&=\sum_{f_j\in c'}\omega(f_j,c')\times FUR(f_j,c')\\&=\Big[\sum_{f_i\in c\wedge f_i\in c'}\omega(f_i,c')\times FUR(f_i,c')\Big]+\\&\quad\omega(f,c')\times FUR(f,c').\end{aligned}$$

根据引理 2，一个特征的效用参与率只会随着模式阶数的升高非递增，且 $\forall f_i \in c$，都有 $FUR(f_i,c)<\xi$.所以有：

$$\begin{aligned}PUI(c')&\leq\sum_{f_i\in c\wedge f_i\in c'}\omega(f_i,c')\times FUR(f_i,c)+\\&\quad\omega(f,c')\times FUR(f,c')\\&<(1-\omega(f,c'))\times\xi+\omega(f,c')\times FUR(f,c').\end{aligned}$$

根据定理 2 的条件，满足 $FUR(f,c)<\xi$，故有：

$$PUR(c')<(1-\omega(f,c'))\times\xi+\omega(f,c')\times\xi=\xi.$$

基于特征效用参与率的不增性，我们可以类似地证明模式 c 的所有超集都是非高效用的. 证毕.

例如：若效用阈值为 0.5，在图 2 中模式{A，D}不是高效用模式.根据定义 5 可以得到{A，D}的相关模式集：$S_c(\{A,D\})=\{\{A,B\},\{A,C\},\{A,E\},\{B,D\},\{D,E\}\}$，由此可得相关特征集为{C，E}，计算得：$\max\{\min_C\{3/32\},\min_E\{93/128,29/128\}\}=29/128$，由于相关特征 B 的特征效用参与率为 0，所以不再考虑，那么就有 $29/128<0.5$，所以{A，D}及其高阶模式都不是高效用模式.

算法 2. 最小特征效用参与率剪枝算法(MinFURA).

输入：空间特征集 F，带效用实例集 I，距离阈值 d，效用度阈值 ξ

输出：高效用模式集 U_{high}

BEGIN

1. $k=2$，基于距离阈值 d 计算 2 阶模式的表实例，所有 2 阶模式形成 2 阶候选 C_k.
2. WHILE $C_k<>$NULL
   - 2.1 FOR EACH $c\in C_k$
     - 2.1.1 计算每个特征的效用值，计算特征的效用参与率 $FUR(f_i,c)$；
     - 2.1.2 通过效用参与率确定每个特征的效用权重 $\omega(f_i,c)$；
     - 2.1.3 计算模式 c 的模式效用度 $PUI(c)$；
     - 2.1.4 若 $PUI(c)<\xi$，使用定理 2 判定模式 c 的高阶模式是否可以剪枝，若满足剪枝条件就有：$P_p=P_p\cup\{c\}$；若 $PUI(c)\geq\xi$，则 $U_{high}=U_{high}\cup\{c\}$；($P_p$ 是满足减枝条件的模式集合)
   - 2.2 生成 $k+1$ 阶的模式 C_{k+1}，且基于 P_p，应用定理 2 对 C_{k+1} 实施剪枝；
   - 2.3 $k=k+1$；
3. 输出高效用集合 U_{high}

END

算法 MinFURA 主要利用了定理 2 的最小特征效用参与率剪枝策略，对满足条件的模式进行剪枝，在一定程度上提高了算法的效率.因为计算模式的表实例和效用参与度是十分耗时的.

5.3 最小特征效用参与率剪枝的失效及原因分析

我们在 1000×1000 的网格数据范围内进行了最小特征效用参与率剪枝算法的实验，发现当空间数据越来越稠密时，最小特征效用参与率剪枝效果急剧下降.如图 4 所示，在数据越来越密集时，使用剪枝和未使用剪枝的时间消耗在逐渐逼近，在空间数据点为 18 000 时，时间消耗几乎重合.最小特征

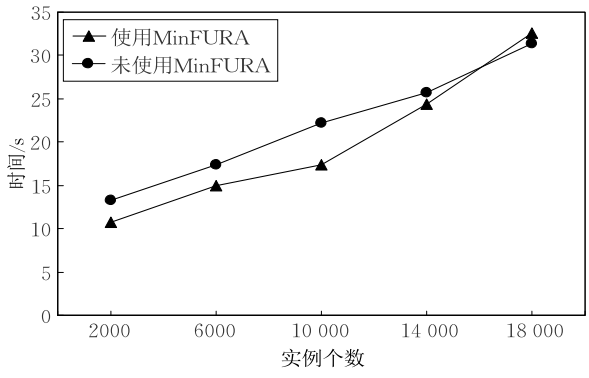


图 4 最小特征效用参与率剪枝的失效问题

效用参与率剪枝算法失效的原因在于随着数据密集性增大，实例与实例间的邻近关系增多，构成团的几率增大. 所以每个特征随阶数增大时，特征效用参与率的减少程度削弱，最小特征效用参与率剪枝的条件难以达到，从而出现了最小特征效用参与率剪枝失效情况.

5.4　最大特征效用参与率剪枝算法(Max Feature Utility Ratio Algorithm，MaxFURA)

5.3 节讨论了最小特征效用参与率剪枝的失效问题，为了在失效的情况下算法仍然具有高效性，本文进一步提出了基于最大特征效用参与率的剪枝方法. 下面给出相关定义.

定义 6.　特征的高阶参与实例集. 给定一个 co-location 模式 c，模式 c 的相关模式集 $S_c(c)=\{c_1,c_2,\cdots,c_s\}$，相关特征 f_i 是 $S_c(c)$ 中模式的特征且 $f_i\notin c$，$If_i(f_i,c)$ 是 $S_c(c)$ 中包含特征 f_i 的模式的表实例中 f_i 的实例集的交集，称为特征 f_i 的高阶参与实例集，即 $If_i(f_i,c)$ 中的实例是特征 f_i 可能出现在 c 的高阶模式的表实例中的实例，$If_i(f_i,c)$ 计算公式如下：

$$If_i(f_i,c)=\bigcap_{f_i\in c_i\wedge f_i\notin c\wedge c_i\in S_c(c)}\left(\prod_{f_i}table\_instance(c_i)\right).$$

定义 7.　最大特征效用参与率. 设 f_i 是模式 c 的相关特征，其高阶参与实例集为 $If_i(f_i,c)$，特征 f_i 在 c 的高阶模式中的最大效用参与率 $RUR(f_i,c)$ 是 $If_i(f_i,c)$ 中所有实例的效用总值与特征 f_i 的效用总值的比值，公式如下：

$$RUR(f_i,c)=\frac{\sum_{f_i.j\in If_i(f_i,c)}u(f_i.j)}{U(f_i,S)}.$$

定理 3.　给定一个 co-location 模式 c，如果 $\forall f_i\in c$，都有 $FUR(f_i,c)<\xi$，那么，当 c 的相关特征的最大效用参与率 RUR 小于效用阈值 ξ 时，c 的所有高阶模式都不是高效用模式.

证明.　根据条件，模式 c 不是一个高效用模式，$PUI(c)<\xi$，根据定义 7 计算所有相关特征的最大效用参与率后，假设 $f_{\max}$ 是最大效用参与率中的最大特征，对于 $c'=c\cup\{f_{\max}\}$，我们有：

$$\begin{aligned}PUI(c')&=\sum_{f_i\in c'}\omega(f_i,c')\times FUR(f_i,c')\\&\le\sum_{f_i\in c\wedge f_i\in c'}\omega(f_i,c')\times FUR(f_i,c)+\\&\quad\omega(f_{\max},c')\times RUR(f_{\max},c').\end{aligned}$$

当 $RUR(f_{\max},c')<\xi$，且存在 $\forall f_i\in c$，都有 $FUR(f_i,c)<\xi$ 时，有：

$$\begin{aligned}PUI(c')&<\sum_{f_i\in c\wedge f_i\in c'}\omega(f_i,c')\times\xi+\omega(f_{\max},c')\times\\&\quad RUR(f_{\max},c')\\&<\xi.\end{aligned}$$

可以类似地证明，模式 c 的所有高阶模式一定不是高效用模式.　　证毕.

算法 3.　最大特征效用参与率剪枝算法(MaxFURA).

输入：空间特征集 F，带效用实例集 I，距离阈值 d，效用度阈值 ξ

输出：高效用模式集 U_{high}

步骤：

BEGIN

1. $k=2$，基于距离阈值 d 计算 2 阶模式的表实例，所有 2 阶模式形成 2 阶候选 C_k.
2. WHILE $C_k<>$NULL
   2.1 FOR EACH $c\in C_k$
      2.1.1 计算每个特征的效用值，计算特征的效用参与率 $FUR(f_i,c)$；
      2.1.2 通过效用参与率确定每个特征的效用权重 $\omega(f_i,c)$；
      2.1.3 计算模式 c 的模式效用度 $PUI(c)$；若 $PUI(c)<\xi$，获取模式 c 相关模式与相关特征. 求每个相关特征在各个相关模式中的交集，并计算相关特征的最大特征参与率 $RUR(f_i,c)$；若 RUR 的最大值$<\xi$，使用相关特征效用参与率判定模式 c 的高阶模式是否可以剪枝，若满足剪枝条件就有：$P_p=P_p\cup c$；若 $PUI(c)\ge\xi$，则 $U_{high}=U_{high}\cup c$；
3. 生成 $k+1$ 阶模式 C_{k+1}，且 C_{k+1} 不包含 P_p 中模式的 $k+1$ 阶高阶模式；
4. $k=k+1$；
5. 输出高效用集合 U_{high}

END

例如，我们以图 3 中的模式{A，D}为例，模式{A，D}的相关模式集是：{{A，B}，{A，C}，{A，E}，{B，D}，{C，D}，{D，E}}，相关特征集是{B，C，E}，假设效用度阈值为 0.4，则通过最小特征效用参与率剪枝策略无法剪枝，因为 $\max\{\min_B\{36/69,33/69\},\min_C\{1,29/34\},\min_E\{1,29/128\}\}=29/34>\xi$，然而可以发现真正参与到{A，D}高阶模式的实例只有{E.3}，所以求各个相关特征的最大参与实例集：

$If_i(C,\{A,D\})=\{C.1,C.2\}_{AC}\cap\{C.3\}_{CD}=\varnothing$；

$If_i(B,\{A,D\})=\{B.1\}_{AB}\cap\{B.2\}_{BD}=\varnothing$；

$If_i(E,\{A,D\})=\{E.1,E.2,E.3,E.4,E.5\}_{AE}\cap\{E.3\}_{DE}=\{E.3\}$；

计算可得 $RUR(E,\{A,D\})=29/128<\xi$，所以模式{A，D}的所有高阶模式可以被剪枝.

5.5 进一步的优化策略

优化策略 1. 快速得到模式的高阶参与实例.

在上述 MaxFURA 算法中,对每一个非高效用模式都需要计算相关特征的高阶参与实例. 为减少这一过程产生的计算消耗,本文引入了邻居集(Neighbor Instances List)的概念,其定义如下.

定义 8. 团实例邻居集. 给定一个 k 阶 co-location实例 $I=\{o_1,o_2,\cdots,o_k\}$,该实例 I 的邻居实例集 $NIL(I)$定义如下:

$NIL(I)=\{o_i\,|\,o_j\in I\wedge distance(o_i,o_j)\leq d\wedge f_{o_i}>f_{o_j}\}$.

例如,图 3 中,2 阶 co-location 实例{A.2,B.1}的邻居实例集如表 2 中所示:$NIL(\{A.2,B.1\})=\{C.2,E.2,E.5\}$,C.2,E.2,E.5 与 A.2,B.1 都存在邻近关系.

表 2 co-location 实例 I 的邻居实例集 $NC(I)$

Size	Co-location instances I	Neighbor instances list	Patterns
2	{A.2,B.1}	C.2,E.2,E.5	{A,B}
	{A.2,C.2}	E.2,E.4	{A,C}
	{A.4,C.1}	∅	
	{A.4,C.2}	∅	
	{A.3,D.3}	E.3	{A,D}
	{A.1,E.1}	∅	{A,E}
	{A.1,E.4}	∅	
	{A.2,E.2}	∅	
	{A.2,E.4}	∅	
	{A.2,E.5}	∅	
	{B.1,C.2}	E.2	{B,C}
	{B.2,D.1}	∅	{B,D}
	{B.1,E.2}	∅	{B,E}
	{B.1,E.5}	∅	
	{C.3,D.2}	∅	{C,D}
	{C.2,E.2}	∅	{C,E}
	{C.2,E.4}	∅	
	{D.3,E.3}	∅	
3	{A.2,B.1,C.2}	E.2	{A,B,C}
	{A.2,B.1,E.2}	∅	{A,B,E}
	{A.2,B.1,E.5}	∅	
	{A.2,C.2,E.2}	∅	{A,C,E}
	{A.2,C.2,E.4}	∅	
	{A.3,D.3,E.3}	∅	{A,D,E}
4	{A.2,B.1,C.2,E.2}	∅	{A,B,C,E}

通过邻居实例集,可以快速的收集模式 c 的高阶参与实例,例如模式{A,B}的表实例为{A.2,B.1},那么它的高阶参与实例一定存在于它的邻居实例集{C.2,E.2,E.5}中,不需要再对模式{A,B}的相关模式进行遍历和求取交集.

优化策略 2. 可直接计算模式 c 的 $k+1$ 阶超模式的效用度.

显然,可通过模式{A,B}的团实例邻居集快速得到其三阶超模式{A,B,C},{A,B,E}的表实例,这样就可以计算{A,B,C}和{A,B,E}的效用度,即可判断其三阶超模式是否为高效用模式,无须再通过连接生成三阶模式. 同样,根据定理 3 以及团实例邻居集,可计算模式 c 的相关特征的最大效用参与率 RUR,可快速的判断模式 c 的高阶超模式是否是高效用模式. 由优化 2 可知,得到 k 阶模式后可直接计算其 $k+1$ 阶的超模式的效用值,所以在生成候选模式时,可以直接生成 $k+2$ 阶超模式. 当然并不是所有的 $k+2$ 模式都需要生成. 根据优化 1,可以直接计算最大特征效用参与率 RUR,例如模式{A,B},可计算 RUR(C)和 RUR(E),且满足定理 3 的要求,RUR(C)和 RUR(E)都小于效用阈值,那么可直接判断模式{A,B}的四阶模式{A,B,C,E}也不是高效用模式.

5.6 两种剪枝算法的比较

剪枝算法 MinFURA 与 MaxFURA 的区别在于:

MinFURA 以计算相关特征参与率 FUR 在相关模式中的最小值,选取这些特征参与率最小值中的最大值来实施简单的候选剪枝,而 MaxFURA 则通过求相关特征在相关模式中实例集的交集来估算每个相关特征在高阶模式中的最大参与率,所以剪枝计算代价要高于 MinFURA 算法. 但是,MaxFURA 能剪枝 MinFURA 不能剪枝的候选模式. 两个算法的计算效率分析比较如下:由于计算特征参与率 FUR 与计算最大特征参与率 RUR 的复杂度基本相同,所以 MaxFURA 比 MinFURA 多出的计算部分是相关特征的高阶参与实例集的求解. 假设数据集中一共有 m 个特征,对于一个 k 阶模式 c,它的相关特征最多有 $m-k$ 个,如果记每个特征参与高阶模式实例的平均个数为 n_{avg},则求取高阶参与实例集的复杂度分析如下:

若不对相关特征的实例进行排序,那么对模式 c 而言,求交集运算的复杂度为

$$(m-k)(k-1)n_{avg}^2.$$

若事先对相关特征的实例进行排序,那么相应的复杂度为

$$(m-k)(k-1)n_{avg}\log_2 n_{avg}.$$

假设在连接生成候选时,每个特征的参与实例的平均数为 N_{avg},则在 MinFURA 中由于无法精确剪枝而造成的候选模式计算的复杂度分析如下:

由于模式 c 未能剪枝,那么模式 c 至多需要和 $k(m-k)$个模式进行链接生成候选模式以及表实例,那么生成候选计算的复杂度 T_{min} 至少是:

$$T_{min}=k(m-k)N_{avg}^2.$$

由于一般有 $N_{avg}>n_{avg}$,所以,MaxFURA 由于

精确剪枝增加了的计算复杂度，不会超过候选计算部分的复杂度，所以 MaxFURA 的计算效率高于 MinFURA.

MaxFURA 通过引入团实例邻居集，简化了高阶参与实例的计算过程，在进一步优化的剪枝算法 MaxFURA-NIL 中通过一个并集操作就可以得到高阶参与，那么对模式 c 而言，其表实例长度为 l，每一条行实例的团实例邻居集平均长度为 n_c，那么求并集的复杂度就为 $O(l\times n_c)$，所以其复杂度要低于 MaxFURA.

6　实验结果与分析

在本节中，我们做了大量实验来验证所提算法的效果和效率，并将本文提出的算法与文献[1]与文献[2]中的算法进行了比较. 实验中所涉及的算法均用 C# 语言实现. 硬件环境为 Intel Core i7 CPU、8 GB 内存；运行环境为 Windows 10、Visual Studio 2013.

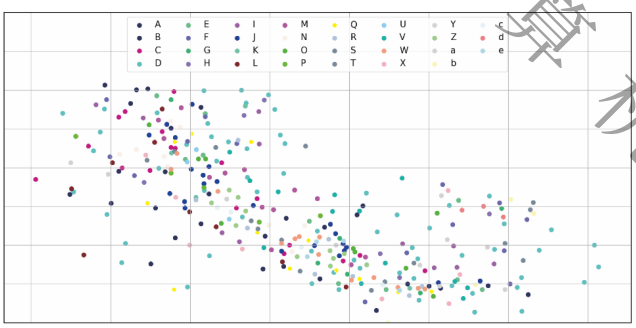
6.1　实验数据集

本文采用的实验数据为 3 个人工合成数据集和 3 个真实的数据集. 人工合成的数据分别在 300×300，1000×1000，2000×2000 范围内随机产生特征数与实例数. 在不同的生成范围内，其数据的密度有所不同. 真实数据的相关信息如表 3 所示.

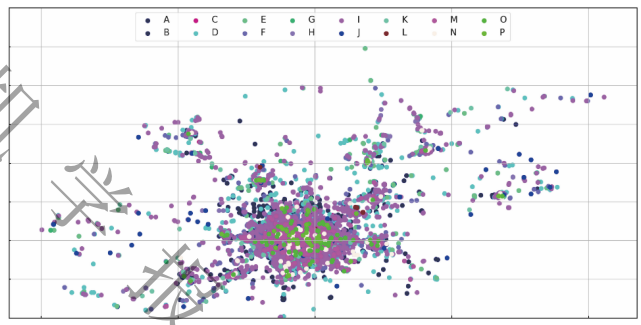
表 3　真实数据集

数据集	特征数	实例个数	范围
plantdata-1	32	353	(5000,130000)
plantdata-2	25	13349	(5000,230000)
Beijing-POI	16	23025	(15,23000)

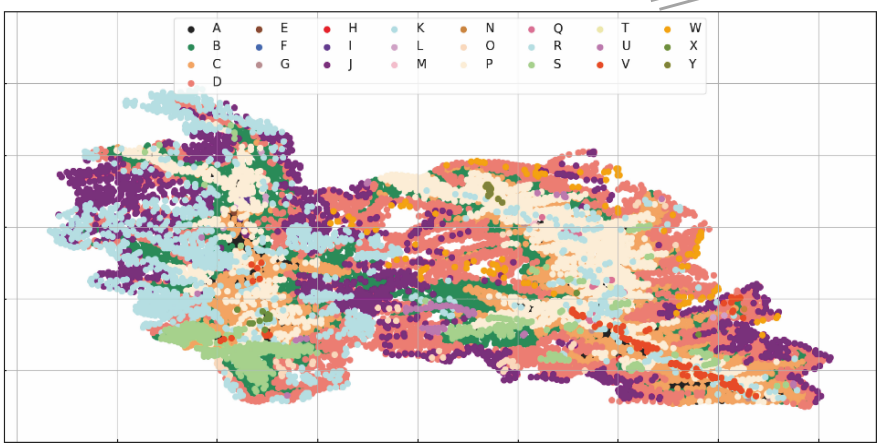
如表 3 所示，plantdata-1 是一个包含 32 种珍惜植物，353 个实例的“三江并流”区域珍惜植物分布数据集，其分布形状为带状(见图 5(a)). plantdata-2 是一个包含 25 种植被，13349 个实例的高黎贡山植被数据集，成簇状分布(见图 5(b)). Beijing-POI 是一个包含 23025 个数据点的北京市 POI 数据集，分布形状见图 5(c).



(a) plantdata-1



(b) plantdata-2



(c) Beijing-POI

图 5　真实数据集分布

6.2　高效用模式合理性分析

6.2.1　模式数量评估

图 6 和图 7 分别在模拟数据集与真实数据集上比较了 EPA(文献[1]中的算法)和 UPI(文献[2]中的算法)与本文所提出的效用度量方法 PUI 在同一个数据集中所挖到的高效用模式个数. 实验模拟数

据设置:范围为 1000×1000,特征数目为 15,距离阈值为 20,效用阈值为 0.4. 真实数据为 plantdata-1,plantdata-2,Beijing-POI. 在模拟数据集上,通过图 6 可以发现,无论数据量的增加还是减少,可以发现始终基本满足 EPA>PUI>UPI. 在真实数据集上,我们选择 0.4~0.75 多个效用阈值,从图 7 中可以发现,在不同的效用阈值情况下,模式数量上也始终基本满足 EPA>PUI>UPI.

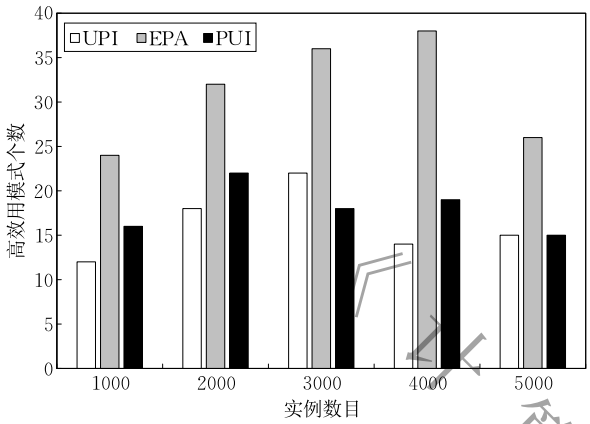


图 6 模拟数据集下的高效用模式数

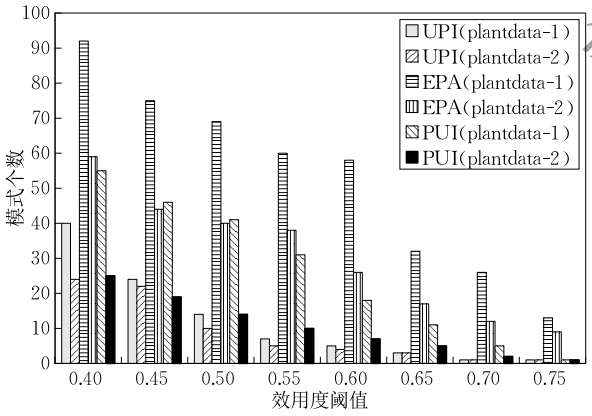


图 7 真实数据集下的高效用模式数

出现这些结果的原因是 EPA 挖掘的是特征带效用的高效用模式,由于该方法没有考虑特征间效用的差异性,简单的将效用值相加,并没有考虑效用的参与率,盲目于效用值较大的特征,而忽略了模式中效用较小的特征的贡献,导致了大量的模式被判定为高效用模式. 而 UPI 是人为设定特征的效用贡献度(权重),且用最小效用参与率衡量模式的效用,使得一些本应该是高效用的模式被剔除,导致了模式数量比我们的方法 PUI 少. 因此,就模式数量而言,PUI 挖掘到的模式数量介于 EPA 和 UPI 之间.

6.2.2 效用占比评估

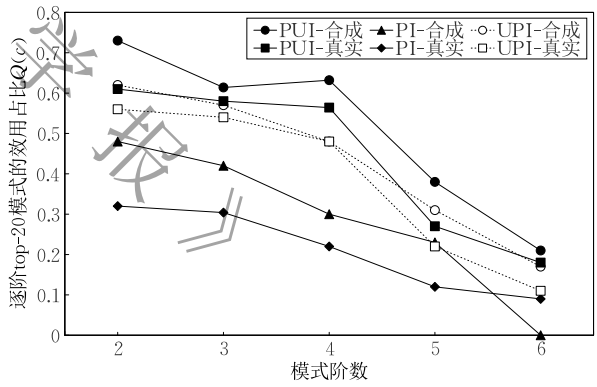
空间高效用模式旨在找出那些空间上邻近且要求每个模式的效用度都大于效用度阈值的模式. 它对模式的有趣度的度量与传统的频繁模式挖掘有所不同,为此,提出模式 c 的效用占比 $Q(c)$ 度量所挖掘到模式的合理性,公式如下:

$$Q(c)=\frac{\sum_{f\in c}u(f,c)}{\sum_{f\in c}U(f,S)},$$

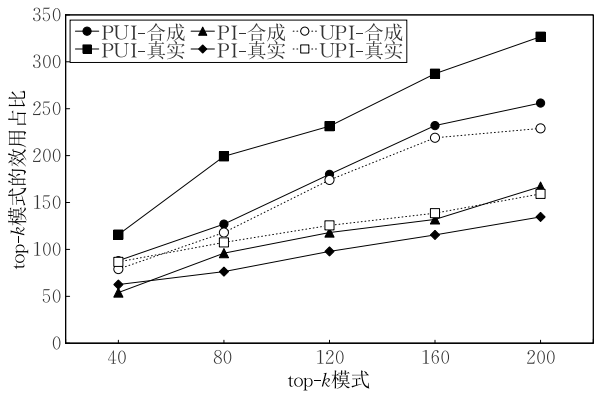
其中 $u(f,c)$ 表示特征 f 在模式 c 中的效用总和,$U(f,S)$ 表示特征 f 在数据集 S 中的效用总值.

由于 EPA 是特征带效用的算法,而文献[2]所提出的度量标准 UPI 是基于实例带效用的算法,所以在这部分的实验主要对 PUI、UPI 和 PI 算法进行了比较. 为了验证本文提出 PUI 要比 UPI[2],PI[11] 更为有效和合理,分别从下面两个方面来进行了验证,实验模拟数据设置:特征数目为 15,距离阈值为 20,实例数为 13 000. 在此实验中为了验证效用占比,模拟数据中将价值设定在同一价值标准下,使其可以不同特征间相加. 真实数据选用 plantdata-2.

不同阶数下的效用占比:通过图 8(a)可以看到,在三种不同的度量指标下挖掘得到的 top-20 模式中不同阶数的效用占比 $Q(c)$,结果表明,本文提



(a) 在不同阶数下的效用占比$Q(c)$



(b) top-k的效用总和

图 8 三种指标的效用占比评估

出的 PUI 度量挖掘到的模式效用占比要高于传统的频繁模式挖掘方法的 PI 度量方法和文献[2]的 UPI 度量.

Top-k 模式的效用总和：

从图 8(b)可以看到，通过计算三种方法下的 top-k 模式的效用之和，可以发现，PI 方法所挖掘得到的 top-k 的效用总和始终是低于 UPI 和 PUI，且 PUI 始终高于 UPI. 可见 PUI 挖掘高效用模式的结果要优于 UPI 算法.

6.2.3　频繁性评估

模式的频繁性反映的是一个模式在数据集中的普遍程度，具有一定的可用性. PI 是最经典的频繁性的度量方法，那么高效用的度量方法 UPI 和本文提出的 PUI 算法在频繁性上的表现如何呢？我们使用频繁性占比 $P(c)$来评估三个指标的频繁性，公式如下：

$$P(c)=\frac{k\text{ 阶模式中的频繁模式}}{k\text{ 阶模式的总数}}.$$

实验模拟数据设置：特征数目为 15，距离阈值为 20，实例数为 13 000. 真实数据选择 plantdata-1. 频繁度参与率设为 0.4. 如图 9 所示，在频繁度的评估上 PI 始终是要高于 UPI 和 PUI，但同时我们也可以得到相较于 UPI，PUI 挖掘到的高效用模式的频繁度要比 UPI 高. 所以利用 PUI，我们可以挖掘到效用高且相对频繁的高效用模式，不再是单纯的高效用模式或是频繁模式. 接着，我们在合成数据上对不同特征的参与率进行了统计，并对 PUI 和 UPI 这两个基于实例带效用的算法进行了对比，从图 10 中可以发现本文所提方法 PUI 挖掘到的模式中特征的参与率要高于 UPI，弥补了 UPI 高效用模式丢失的问题.

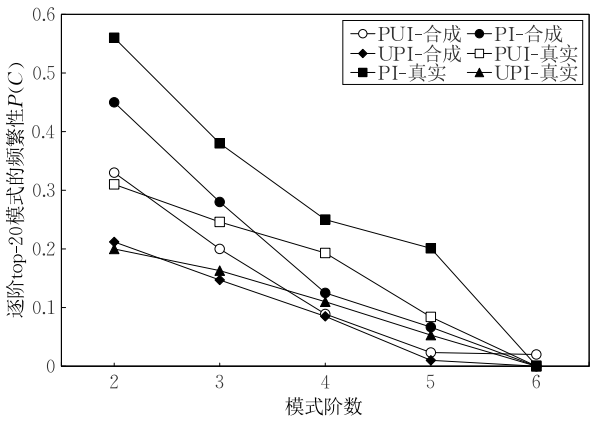


图 9　三种指标的频繁性占比评估

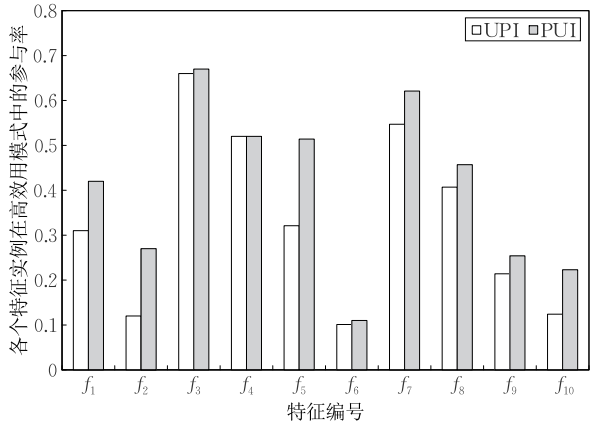


图 10　Top-100 高效用模式的特征的参与率

综上，实验结果表明本文所提的高效用度量方法 PUI 在效用占比和频繁性方面均优于现有的 UPI 度量.

6.3　剪枝算法有效性评估

本节实验主要研究本文所提剪枝算法 MinFURA，MaxFURA，以及 MaxFURA 的优化算法 MaxFURA-NIL 在不同数据密集程度下的剪枝有效性和执行时间，同时讨论了数据分布对算法有效性的影响.

6.3.1　剪枝率分析

本节实验对本文提出的三个算法 MinFURA、MaxFURA 和 MaxFURA-NIL 进行了剪枝率分析，其中：

剪枝率 = 被减模式个数/模式总个数.

不同数据密集程度下的剪枝率变化：

实验通过合成不同密集程度(1000×1000、2000×2000)的数据来验证不同剪枝算法的剪枝效率. 从图 11 可以发现随着数据量的增加，MinFURA 的剪枝率在不断降低，数据越密集，下降程度越剧烈. 而

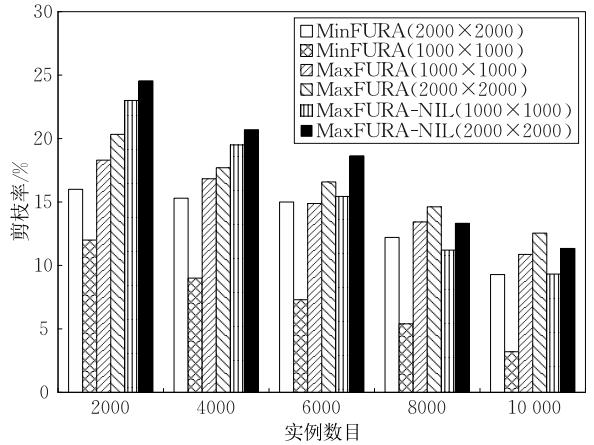


图 11　剪枝算法在不同数据密度下的剪枝率

MaxFURA 和 MaxFURA-NIL 的剪枝率随着数据密集程度的增加有所降低，但始终能保持在 10%以上，有较好的剪枝效率.

真实数据集下的剪枝情况：

我们选用数据集 Beijing-POI 对剪枝率进行实验. 从图 12 可以看出，随着距离阈值的增大，MaxFURA 的剪枝率基本维持在一个水平上，但是 MinFUPA 的剪枝率略有下滑. 在这个真实数据集中，可以看到 MaxFURA 和 MaxFURA-NIL 的剪枝效果要优于 MinFURA，而 MaxFURA-NIL 的剪枝效果要略优于 MaxFURA.

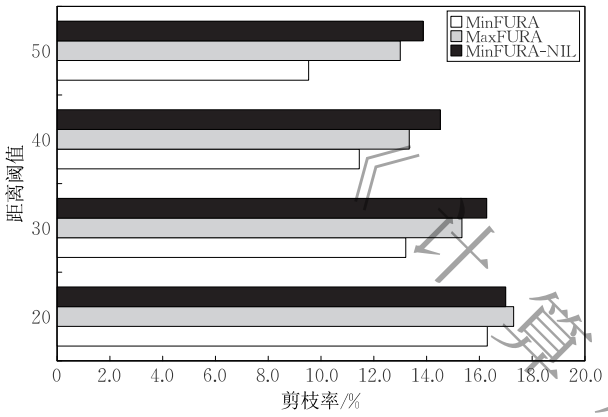


图 12 三个剪枝算法在不同距离阈值下的剪枝率

6.3.2 剪枝算法的时间效率分析

将 MinFURA 分别在稀疏 2000×2000 的模拟数据集上进行实验，由图 13 可发现，当数据量增大且越来越密集时，MinFURA 的执行时间不断升高. 而 MaxFURA 和 MaxFURA-NIL 的时间消耗要比只使用 MinFURA 低很多，且 MaxFURA 消耗的时间要比 MaxFURA-NIL 多，可见 MaxFURA-NIL 的优化效果较好.

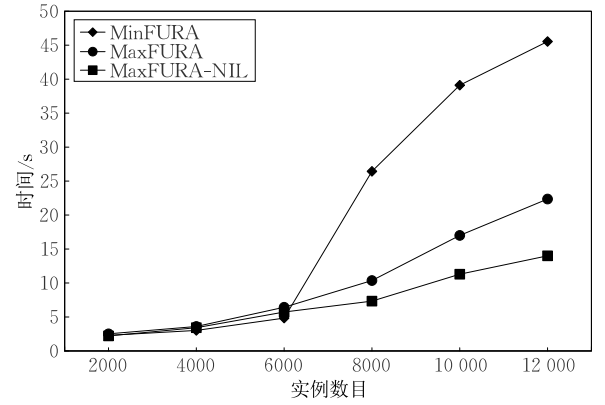


图 13 在不同数据密集程度下的时间效率分析

6.4 算法的可扩展性评估

本节实验主要研究本文所提算法 MinFURA(MaxFURA)在改变效用阈值，距离阈值，以及实例数时的可扩展性. 同样，由于 EPA 是特征带效用的算法，而文献[2]所提出的度量标准 UPI 是基于实例带效用的算法，所以在这部分的实验主要对 MinFURA 算法、MaxFURA 算法和 UPI 算法进行了比较. 且主要在模拟数据上进行了可扩展性实验.

6.4.1 不同数据分布对算法的影响

实验模拟数据在 300×300 和 1000×1000 两个范围内生成，300×300 的数据集较为密集，1000×1000 的数据集较为稀疏，距离阈值为 10，效用阈值为 0.4. 从图 14 中可以看到随着实例数目的增加，三个算法的执行时间都在上升. 在密集数据集中，MinFURA 的执行时间有时超过了 UPI 算法，这是因为随着数据越来越密集，MinFURA 的剪枝失效了. 但无论密集还是稀疏，MaxFURA 算法始终保持着较好的执行效率. 实验中，当数据量增大到 10 万个实例时，MaxFURA 的执行时间为 5 h 26 min 48 s，而其它算法的执行时间已经超过 7 h.

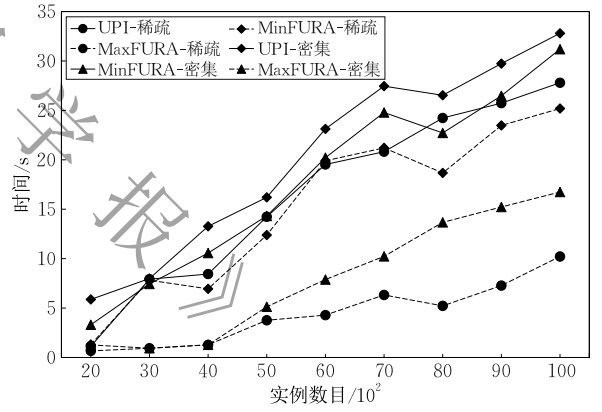


图 14 三个算法执行效率对比

6.4.2 效用阈值对算法的影响

在效用度阈值的实验中(图 15)，模拟数据参数设置为：特征数目为 20，距离阈值为 10，实例数目为 13 000. 真实数据选择 plantdata-2. 随着效用度阈值的增加可以发现，无论在合成数据集还是在实际数据集上，三个算法的执行时间都在降低，这是因为满足效用度阈值的模式越来越少. 同时可以发现本文提出的两个算法 MinFURA(MaxFURA)总是优于 UPI 算法. 因为 UPI 需要计算每个特征的效用参与度，从而选取最小值作为模式的效用值，但 MinFURA

(MaxFURA)都只需要计算一个效用值，计算复杂性要小于 UPI，且两个算法都用了有效的剪枝策略，使得执行效率优于 UPI.

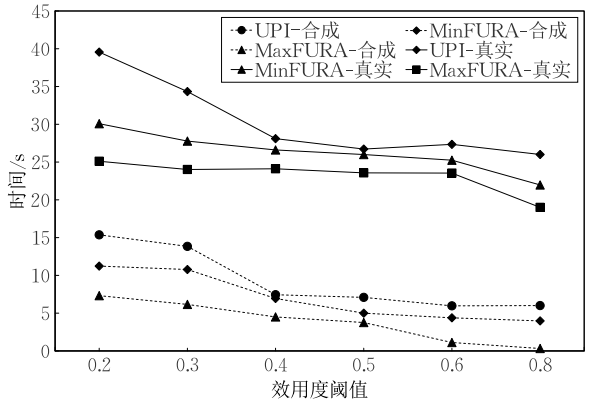


图 15　不同效用度阈值下的时间效率

6.4.3　距离阈值对算法的影响

在距离阈值的实验中(图 16)，参数值设置为：特征数目为 20，效用阈值为 0.45，实例数目为 23 000. 真实数据选择 plantdata-2. 随着距离阈值的增加可以发现，无论在人工数据集还是在实际数据集上，三个算法的执行时间逐渐升高，这是因为满足邻近关系且构成团的实例也越来越多. 而当距离增加到一定程度时，出现 MinFURA 算法执行时间几乎与 UPI 重合，这是由于剪枝失效. 同时可以发现 MaxFURA 算法始终保持着一个较好的执行效率.

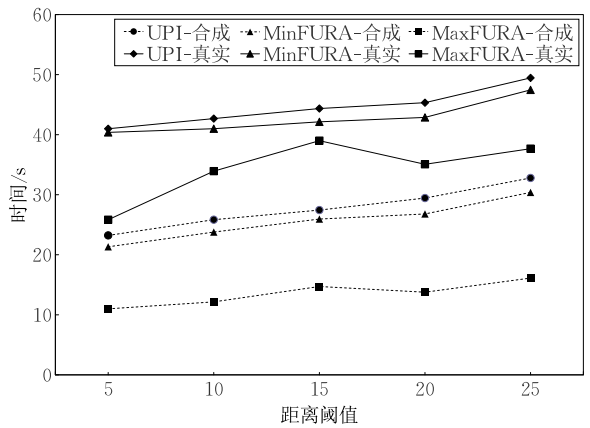


图 16　不同算法在不同距离阈值下的时间效率

6.5　真实数据挖掘结果分析

对于同样的数据集，我们发现需要根据不同的应用需求来评估效用，下面根据植物的食用价值和经济价值来对植物分布数据集上挖掘到的高效用模式进行分析.

(1) 食用价值——高效用模式分析

在表 4 中标有“▲”标记的植物是在 25 种植物中具有食用价值的植物，分别是板栗、核桃、花椒、马桑、铁核桃. 挖掘得到的基于食用价值的高效用模式入表 5 所示，表 5 中对几个特殊的高效模式的效用值与频繁度进行对比：

表 4　各个优势树种数量

树种	数量	树种	数量
板栗▲	16	落叶松	56
杜鹃	110	马桑▲	10
枫香	1	桤木	802
高山松	10	其它阔叶林	2072
灌竹	12	其它竹	109
核桃▲	10	乔松	15
花椒▲	7	软阔	1
华山松	10	杉木	1
桦木	10	铁核桃▲	11
箭竹	1288	铁杉	2176
冷杉	2866	云南松	1959
栎类	471	杂灌	1282
云杉	53		

表 5　食用价值评估下的高效用模式分析

模式	PUI	PI
{核桃，马桑}	0.82	0.10
{花椒，铁核桃}	0.56	0.18
效用阈值为：0.5；参与度阈值为：0.3； 距离阈值为：20；挖掘算法：MaxFURA-NIL		

可以发现，{核桃，马桑}，{花椒，铁核桃}两个模式都不是频繁模式，但却都是高效用模式. 并且找到的模式在食用价值的指标下是有效的.

(2) 经济价值——高效用模式分析

该实验对植物种类进行经济价值的分析，表 6 中给出了 5 个经济价值较高的高效用模式，对不同方式计算的模式效用值以及频繁度进行了对比：

表 6　经济价值评估下的高效用模式分析

模式	PUI	UPI	PI
{枫香，冷杉}▲	0.95	0.43	0.0028
{枫香，冷杉，铁杉}▲	0.833	0.3101	0.0027
{核桃，马桑}□	0.82	0.77	0.3
{核桃，花椒，桤木}▲	0.63	0.247	0.0049
{铁杉，杂灌}◇	0.457	0.407	0.357
效用阈值：0.5；参与度阈值为：0.3 距离阈值：15；挖掘算法：MaxFURA-NIL			

从表 6 可以发现，标有“▲”的三个模式都不是频繁模式，且使用 UPI 来度量效用值时仍然不能被挖掘到，但是就经济价值而言，这 3 个模式均有较高的经济价值. 例如模式{枫香，冷杉}，由于其树种在

药用以及家具制造业有较高的价值，就如同事务数据库中的{钻石，项链}模式一样，虽然在数据集中数量较少，但其价值不可忽视，所以利用 PUI 来评价该模式的价值更为合理. 而标有“□”的模式{核桃，马桑}，由于参与的数量比例相当，所以两种方法 PUI 和 UPI 都能将其挖掘到，且也为频繁模式. 标有“◇”的模式{铁杉，杂灌}虽然是频繁模式，但该模式无论用何种度量方法都不被视为该数据集中的经济价值——高效用模式.

7 总 结

由于空间高效用 co-location 模式挖掘的复杂性，导致现有高效用度量方法存在不同的问题，在本文中，我们详细分析了现有空间高效用 co-location 模式度量方法的缺陷，提出了一种空间高效用 co-location 模式度量新方法，利用特征的效用参与率，定义了特征在模式中的效用权重，区分了不同特征在模式中效用贡献度. 由于向下闭合性在挖掘过程中的失效，本文设计了两个高效的剪枝算法，并且对两个剪枝算法的有效性和适用性进行了对比分析. 通过与原有算法和不同数据密度下的实验结果对比，证明了算法的正确性和有效性. 最后根据不同的效用兴趣，在真实的植物数据集上进行了分析.

在未来工作中将会考虑以下几个方面：(1) 实例效用的不确定性，考虑用区间数表示实例效用，或基于模糊集论方法研究空间高效用模式的挖掘等；(2) 类似于文献[21]，考虑空间实例间的可达性，使得挖掘结果更加的实用与有效.

参 考 文 献

[1] Yang S, Wang L, Bao X, Lu J. A framework for mining spatial high utility co-location patterns//Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'15). Zhangjiajie, China, 2015: 631-637

[2] Wang L, Jiang W, Chen H, Fang Y. Efficiently mining high utility co-location patterns from spatial data sets with instance-specific utilities//Proceedings of the 22nd International Conference on Database System for Advanced Applications. Suzhou, China, 2017: 458-474

[3] Chen M, Han J, Yu P. Data mining: An overview from a database perspective. IEEE Transactions on Knowledge and Data Engineering, 2002, 8(6): 866-883

[4] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation//Proceedings of the ACM SIGMOD International Conference on Management of Data. New York, USA, 2000: 1-12

[5] Zaki M, Hsiao C. Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(4): 462-478

[6] Wang D, Pedreschi D, Song C, et al. Human mobility, social ties, and link prediction//Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, USA, 2011: 1100-1108

[7] Chiu D Y, Wu Y H, Chen A L P. An efficient algorithm for mining frequent sequences by a new strategy without support counting//Proceedings of the 20th International Conference on Data Engineering. Boston, USA, 2004: 375-386

[8] Yin J, Zheng Z, Cao L. USpan: An efficient algorithm for mining high utility sequential patterns//Proceedings of the KDD2012. Beijing, China, 2012: 660-668

[9] Pei J, Han J, Pinto H, et al. PrefixSpan: Mining sequential patterns efficiently by prefixprojected pattern growth//Proceedings of the 17th International Conference on Data Engineering. Los Angeles, USA, 2001: 215-224

[10] Morimoto Y. Mining frequent neighboring class sets in spatial databases//Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA, 2001: 353-358

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

[12] Yoo J S, Shekhar S. A join-less approach for co-location pattern mining: A summary of results. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(10): 1323-1337

[13] Yao X, Chen L, Peng L, Chi T. A co-location pattern-mining algorithm with a density-weighted distance thresholding consideration. Journal of Information Sciences, 2017, 396: 144-161

[14] Wang L, Bao X, Zhou L. Redundancy reduction for prevalent co-location patterns. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(1): 142-155

[15] Ouyang Zhi-Ping, Wang Li-Zhen, Chen Hong-Mei. Mining spatial Co-location patterns for fuzzy objects. Chinese Journal of Computers, 2011, 34(10): 1947-1955(in Chinese)
(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)

[16] Lu J, Wang L, Fang Y, Li M. Mining competitive pairs hidden in co-location patterns from dynamic spatial databases//Proceedings of the Pacifica-Asia Conference on Knowledge Discovery and Data Mining. Jeju, South Korea, 2017: 467-480

[17] Sheshikala M, Rao D R, Rajanala V P. Join-less approach for finding co-location patterns-using map-reduce framework. Journal of Theoretical and Applied Information Technology, 2016, 87(2): 355-365

[18] Yu W, Ai T, He Y, Shao S. Spatial co-location pattern mining of facility points-of-interest improved by network neighborhood and distance decay effects. International Journal of Geographical Information Science, 2016, 31(2): 280-296

[19] Yao X, Peng L, Yang L, Chi T. A fast space-saving algorithm for maximal co-location pattern mining. Expert Systems with Applications, 2016, 63(C): 310-323

[20] Lin Z, Lim S. Fast spatial co-location mining without cliqueness checking//Proceedings of the 17th ACM International Conference on Information and Knowledge Management. Napa Valley, USA, 2008: 1461-1462

[21] Yao H, Hamilton H J, Butz C J. A foundational approach to mining Item set utilities from database//Proceedings of the 4th SIAM International Conference on Data Mining. Orlando, USA, 2004: 215-221

[22] Yao H, Hamilton H. Mining itemset utilities from transaction databases. Data & Knowledge Engineering, 2006, 59(3): 603-626

[23] Yao H, Hamilton H, Geng L. A unified framework for utility-based measures for mining itemsets//Proceedings of the ACM SIGKDD 2nd Workshop Utility-Based Data Mining. Philadelphia, USA, 2006: 28-37

[24] Liu Y, Liao W, Choudhary A. A two-phase algorithm for fast discovery of high utility item sets//Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Hanoi, Vietnam, 2005: 689-695

[25] Ahmed C F, Tanbeer S K, Jeong B S, et al. Efficient tree structures for high utility pattern mining in incremental databases. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(12): 1708-1721

[26] Tseng V S, Wu C W, Shie B E. UP-growth: An efficient algorithm for high utility itemset mining//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA, 2010: 253-262

[27] Lin C, Lan G, Hong T. An incremental mining algorithm for high utility item sets. Expert Systems with Applications, 2012, 39(8): 7173-7180

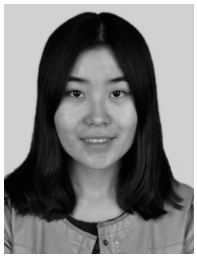
[28] Liu J, Wang K, Fung B. Mining high utility patterns in one phase without generating candidates. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(5): 1245-1257

[29] Tseng V, Shie B, Wu C, Yu P. Efficient algorithms for mining high utility itemsets from transactional databases. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(8): 1772-1786

[30] Dawar S, Goyal V. UP-Hist tree: An efficient data structure for mining high utility patterns from transaction databases//Proceedings of the 19th International Database Engineering Application Symposium. Yokohama, Japan, 2015: 56-61

[31] Wang X, Wang L, Lu J, Zhou L. Effectively updating high utility co-location from spatial database//Proceedings of the 17th Web-Age Information Management. Nanchang, China, 2016: 67-79

[32] Wang X, Wang L. Incremental mining of high utility co-locations from spatial database//Proceedings of the IEEE International Conference on Big Data and Smart Computing. Jeju, South Korea, 2017: 215-222



WANG Xiao-Xuan, Ph.D. candidate. Her current research interests include spatial database and data mining.

WANG Li-Zhen, Ph. D., professor, Ph. D. supervisor. Her research interests include database, spatial data mining, and computer algorithm.

CHEN Hong-Mei, Ph. D., associate professor. Her research interests include database and spatial data mining.

FANG Yuan, Ph. D. candidate. Her research interests include spatial database and Data mining.

YANG Pei-Zhong, M. D. His research interests include spatial database and big data.

Background

With the fast development of spatial data collection and storage technology, many organizations have accumulated mass of spatial data. How to find the useful knowledge from spatial data becomes more and more important. Spatial data mining is a technology which combined the traditional analytical methods and the complex spatial algorithms. And spatial data mining refers to finding the interaction between spatial objects, spatial-dependent and causal relationship patterns of spatial data. Traditional spatial co-location mining aims to find the corresponding subsets of spatial features whose objects or events are often located in close spatial proximity. And many efficient algorithms have been proposed to mine

the frequent co-location patterns, such as join-based and join-less algorithm.

The technology of mining high utility item-sets or patterns from the transaction database has been very mature, many algorithms have been proposed. And the researchers have proved high utility pattern doesn't satisfy the downward closure. So, the two most important points of high utility mining are: utility function and reasonable pruning algorithms. In this paper, we study the problem of mining high utility co-locations from spatial database. The traditional metrics for spatial co-location mining do not take account of the utility of different spatial objects belongs to different features. Compared with the traditional high utility mining, high utility co-locations mining faces more challenges. Because the characteristic of spatial data points, the traditional utility function can't evaluate the utility of co-locations. Therefore, we proposed an utility function to more reasonably evaluate the utility of co-locations, which considers the utility weight of different features in the co-locations and uses the Feature Utility Ratio to eliminate the large utility gap between different features. At the same time, the pruning algorithms are also very important to mine high utility co-location. In this paper, we proposed two efficient pruning algorithms to improve the efficiency of high utility mining.

This work is supported by the National Natural Science Foundation of China (No. 61472346 and No. 61662086), the Natural Science Foundation of Yunnan Province (No. 2016FA026 and No. 2015FB114), and the Project of Innovation Research Team of Yunnan Province (No. 20181tc019).

In recent years, the authors have devoted to the researches of spatial co-location mining, and have gained achievements on spatial data mining and trajectory data mining. In this work, we focus on spatial high utility co-location mining, and propose the related concepts and the efficient algorithms to mine spatial high utility co-locations.