



Mining High Utility Co-location Patterns Based on Importance of Spatial Region

Jiasong Zhao^{1,2}, Lizhen Wang^{1(✉)}, Peizhong Yang¹,
and Hongmei Chen¹

¹ Department of Computer Science and Engineering, Yunnan University,
Kunming, China

zhaojs75@163.com, lzhwang@ynu.edu.cn

² Department of Electronic and Information Engineering,
Yunnan Agricultural University, Kunming, China

Abstract. Co-location pattern mining aims at finding the subsets of spatial features whose instances are frequently located together in geographic space. Most studies mainly focus on whether spatial feature instances are frequently located together. However, the utilities of spatial instances in different space regions are different. Based on the importance of spatial a region, the utility value of the region is determined, and then a utility participation index of co-location patterns as a new interestingness measure is defined. We present a basic high utility co-location pattern mining algorithm. To reduce the computational cost, an improved mining algorithm with pruning strategy is developed by cutting down the search space. The experiments on synthetic and real world datasets show that the proposed methods are effective and efficient.

Keywords: Spatial data mining · Co-location pattern · High utility
Spatial region

1 Introduction

Owing to the rapid generation of spatial data and wide use of spatial databases, it's essential to discover spatial knowledge automatically from spatial datasets. Spatial co-location mining is to discover a set of spatial features of which instances frequently appear in a spatial neighborhood of each other. Mining spatial co-location patterns is an important spatial data mining task with broad applications [1, 2].

The participation index is used to measure the prevalence (or frequency) of a co-location pattern in the traditional approaches of co-location pattern mining. It mainly focused on whether spatial feature instances are frequently located together. However, the utility of the feature instances in different spatial regions is different. For example, in the study of plant symbiosis, the symbiotic plants that grow in the poor ecological environment (soil, humidity, altitude, etc.) are more general than those that grow in the good ecological environment. Another example, in the research field of public safety, social effect of criminal cases occurring in the downtown is absolutely different from ones occurring in the school. Therefore, it is necessary to establish the importance of spatial region in mining spatial co-location patterns. After setting of

region utility values based on regional importance, the utility value of the instance is the utility value of the region where the instance is located. By the way of mining high utility co-location patterns similar to the literature [3], the method of discovery high utility co-location patterns can not only detect some low-frequency but high-utility (interesting) patterns, but also remove some common-sense patterns (high-frequency but low-utility).

Here, we use a specific example to illustrate this problem. Figure 1 shows a sample spatial dataset consisting of four features, A, B, C, and D. A.2 represents the second instance of spatial feature A. Two instances are connected by edges (shown as solid lines) if they have a spatial neighbor relationship, and the whole space is divided into four equal regions. Suppose the utility values of these four regions are 8, 5, 2, and 1 respectively. According to the traditional co-location pattern mining method, the participation index of co-location {C, D} is 1/3. If the prevalence threshold is not less than 0.6, {C, D} would be regarded as a non-prevalent co-location. However, two row instances of {C, D}, {C.5, D.5} and {C.6, D.6}, all appear in Region 1 which shows the highest utility value. So the utility of each feature in {C, D} accounts for a large proportion of its total utility. {C, D} may be interesting to users. As to the pattern {B, C}, in contrast, the participation index of {B, C} is 2/3, the utility participation index of {B, C} is lower. So {B, C} may be non-interesting to users.

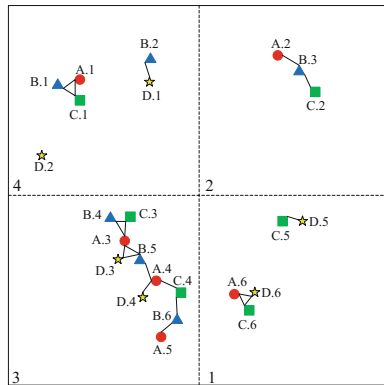


Fig. 1. A sample spatial dataset

Therefore, the real interesting co-location patterns cannot be found by traditional approach because the importance of different regions is ignored. In the paper, we focus on high utility co-location mining based on importance of spatial regions.

1.1 Related Work

Morimoto et al. [4] first defined the problem of finding frequent neighboring co-locations in spatial databases. Huang et al. [5] proposed a general approach for mining co-location patterns, the join-based approach, which defined the participation index to measure the prevalence of a co-location. After that, different algorithms were

proposed to improve the efficiency of the mining process, such as join-less algorithm [6] proposed by Yoo et al., and density based algorithm [2] proposed by Xiao et al. Wang et al. also presented a new join-less algorithm based on the CPI-tree [7] and an approach based on order-clique for discovering maximal co-locations [8]. Previous approaches focus on discovering global co-locations in a spatial dataset. The results of these approaches are usually not able to capture or represent the characteristics of different zones. Celik et al. [9] defined the problem of zonal co-location mining. He proposed an index structure based on the Quad-tree to support dynamic parameters for zonal co-location mining. Dai et al. [10] developed a new index structure and algorithm to mine zonal co-locations more efficiently. Sengstock et al. [11] introduced a new general class of interestingness measures that are based on the spatial distribution of co-location patterns. A new measurement using an evenness coefficient of the feature distribution was introduced, and a novel algorithm for discovering prevalent and evenly distributional co-location patterns was proposed in [12].

Much works on high utility pattern mining have presented different approaches in the transactional data sets [13]. A two-phase algorithm to efficiently prune down the number of candidates, through transaction-weighted downward closure property was presented in [14]. UP-Growth proposed in [15] enhances the mining performance in utility mining through maintaining the information of high utility itemsets by UP-tree. There are just a very limited number of studies on high utility co-location pattern mining. A framework for mining high utility co-locations was proposed in [16]. Wang et al. [17] discussed a problem of incremental mining high utility co-locations on spatial databases which are constantly changed with added and disappeared data. Recently Wang et al. [3] presented a method of mining high utility co-locations from spatial data sets with instance-specific utilities.

1.2 Contribution

The main contributions of this paper are as follows.

First, based on the importance of spatial regions, we propose a new measure called the utility participation index for high utility co-location pattern mining.

Second, we present a basic algorithm to discover high utility co-locations. In order to reduce the computational cost, an improved algorithm is given.

Finally, we evaluate our algorithms with experiments on both synthetic and real world data sets.

The remainder of this paper is organized as follows. Section 2 introduces basic concepts and definitions of high utility co-location pattern mining based on the importance of spatial regions. A basic algorithm and an improved algorithm with a pruning strategy are presented in Sect. 3. Experimental results are shown in Sect. 4. The last section is conclusions.

2 Basic Concepts and Problem Definition

2.1 Basic Concepts

A set of spatial features represents a collection of different kinds of items in space, as $F = \{f_1, f_2, \dots, f_k\}$. An object with a specific spatial position is called a spatial instance, and a set of instances, as $S = S_1 \cup S_2 \cup \dots \cup S_k$, in which $S_i (1 \leq i \leq k)$ is an instance collection that corresponds to spatial feature. Each instance denoted by the feature type and a numeric id value e.g. B.1. If the Euclidean distance of two instances is not greater than the given threshold d , which means the two instances meet neighbor relationship R . A co-location pattern c is a set of spatial features, in which $c \subseteq F$. The instance set I is a row instance of co-location pattern c , if (1) I contains all features of c , and no proper subset of I does so; (2) any two instances, i_i and i_j , $i_i, i_j \in I$, $R(i_i, i_j)$, that is, i_i and i_j are neighbors. The set of all row instances of c is the table instance of c , denoted as $T(c)$.

When the participation index of a co-location pattern is not less than the given threshold, it is referred as a prevalent co-location pattern. The participation index of co-location pattern c is expressed as $PI(c)$, which is the minimum of participation ratios among all spatial features of c , defining as follows:

$$PI(c) = \min_{f_i \in c} \{Pr(c, f_i)\} \quad (1)$$

The participation ratio of spatial feature f_i in co-location pattern c is expressed as $Pr(c, f_i)$, defined as:

$$Pr(c, f_i) = \frac{\text{Number of distinct instance of } f_i \text{ in any row instance of } c}{\text{Number of instance of } f_i} \quad (2)$$

2.2 Problem Definition

In practical applications, the effect of feature instances in different space regions is different. Therefore, we will set spatial region utility values based on the importance of each region, and then the region utilities will be transformed into the instances. In this section, some related definitions about mining high utility co-location patterns based on the region importance will be given.

Definition 1 (spatial region utility). *The spatial region utility is used to describe the importance of different spatial regions. We denote the spatial region utility of Region i as u_i .*

Definition 2 (spatial instance utility). Let spatial instance $f_{i,j}$ be the j -th instance of feature f_i . We denote the utility of spatial instance $f_{i,j}$ as $u(f_{i,j})$, which is the utility of region where $f_{i,j}$ is located in it.

For example, in Fig. 1, A.2 is located in Region 2. $u(A.2) = u_2 = 5$.

Definition 3 (total utility of feature). The total utility of a feature f_i is the sum of utilities of its instances, denoted as $u(f_i) = \sum_{f_{i,j} \in S_i} u(f_{i,j})$, where S_i is the a set of instances belonging to f_i .

For example, the total utility of feature A in Fig. 1 is $u(A) = u(A.1) + u(A.2) + u(A.3) + u(A.4) + u(A.5) + u(A.6) = 1 + 5 + 2 \times 3 + 8 = 20$.

Definition 4 (utility of feature in co-location). Given a size- k co-location pattern $c = \{f_1, f_2, \dots, f_k\}$, the utility of f_i in c is defined as the sum of utilities of instances belonging to feature $f_i \in c$ in table instance $T(c)$. It is denoted as $u(c, f_i) = \sum_{f_{i,j} \in \pi_{f_i}(T(c))} u(f_{i,j})$, where π is the relational projection operation with duplication elimination.

For example, in Fig. 1, considering the size-2 co-location pattern $c = \{C, D\}$, $T(c) = \{\{C.5, D.5\}, \{C.6, D.6\}\}$. The utility of D in c is $u(c, D) = u(D.5) + u(D.6) = 8 + 8 = 16$.

Definition 5 (Utility Participation Ratio, UPR). The utility participation ratio of feature f_i in co-location pattern c is defined as the proportion of f_i 's utility in c to its total utility. The utility participation ratio can be computed as:

$$UPR(c, f_i) = \frac{u(c, f_i)}{u(f_i)} \quad (3)$$

For example, in Fig. 1, for pattern $c = \{C, D\}$, the utility participation ratio of each feature in c is computed as

$$UPR(c, C) = \frac{u(C.5) + u(C.6)}{u(C)} = \frac{16}{26}, UPR(c, D) = \frac{u(D.5) + u(D.6)}{u(D)} = \frac{16}{22}.$$

Definition 6 (Utility Participation Index, UPI). The utility participation index $UPI(c)$ of a co-location pattern $c = \{f_1, f_2, \dots, f_k\}$ is the minimum in all $UPR(c, f_i)$ of co-location c :

$$UPI(c) = \min_{i=1}^k \{UPR(c, f_i)\} \quad (4)$$

Definition 7 (high utility co-location pattern, UCP). Given a minimum UPI threshold θ , a co-location pattern c is a high utility co-location pattern if $UPI(c) \geq \theta$ holds.

The prevalent patterns may not be high utility patterns and the high utility patterns may not be prevalent as well. For example, for patterns $\{B, C\}$ and $\{C, D\}$ in Fig. 1, $PI(\{B, C\}) = 0.67$ and $UPI(\{B, C\}) = 0.38$, while $PI(\{C, D\}) = 0.33$ and $UPI(\{C, D\}) = 0.62$. If both of minimum PI threshold and minimum UPI threshold are 0.6, we can identify $\{B, C\}$ is prevalent easily, but $\{C, D\}$ is not. At the same time, $\{C, D\}$ is a high utility co-location pattern, but $\{B, C\}$ is not.

Lemma 1. *The utility participation ratio and the utility participation index are anti-monotone (monotonically non-increasing) as the size of the co-location increases.*

Proof: The utility participation ratio is antimonic because a spatial feature instance that participates in a row instance of a co-location c also participates in a row instance of a co-location c' , where $c' \subseteq c$. Because the utility participation index is the minimum in all $UPR(c, f_i)$ of co-location c , when the size of the co-location increases, the utility participation index is decreases, so it is also monotonically non-increasing.

Theorem 1. *If a size- k co-location pattern $c_k = \{f_1, f_2, \dots, f_k\}$ is the high utility co-location pattern, each size- $(k-1)$ co-location pattern $c_{k-1} \subset c_k$ must be the high utility co-location.*

Proof: According to lemma 1, it is easy to be proved.

3 UCP Mining

3.1 Basic Algorithm

The basic UCP mining algorithm uses the generate-and-test methods, that is, generate candidates, and test each candidate to identify whether it is a high utility co-location pattern. Algorithm 1 shows the pseudocode of the basic UPC mining.

Algorithm 1. Basic UCP Mining

Input: F : a set of spatial features, S : a set of spatial instances, d : a distance threshold, θ : a utility participation index threshold.

Output: A set of high utility co-locations.

Variables: k : co-location size, C_k : a set of size- k candidates, T_k : a set of table instances of C_k , U_k : a set of size- k high utility co-location patterns

Methods:

1. the data space is divided into m regions;
2. set the utility value for the region, respectively, u_1, u_2, \dots, u_m ;
3. $U_1 = F$; $T_1 = S$;
4. for ($k = 2$; $U_{k-1} \neq \emptyset$; $k++$) do
5. $C_k = \text{gen_candidate_colocations}(k, U_{k-1})$;
6. $T_k = \text{gen_table_instances}(C_k, T_{k-1}, d)$;
7. Calculate $UPI(C_k, T_k)$;
8. $U_k = \text{select_high_utility_colocations}(C_k, T_k, \theta)$;
9. end for
10. Return $\cup(U_2, U_3, \dots, U_k)$

Initialization (Step 1–3): First of all, according to the importance of the regions, the data space is divided into m regions, and the utility value for these regions is set, respectively, u_1, u_2, \dots, u_m . And then size-1 high utility co-location patterns sets and size-1 table instance sets are initialized.

Generating Candidate Co-locations (Step 4–6): Given a spatial data set and a distance threshold, all neighboring instance pairs can be found by using a geometric method such as plane sweep. Meanwhile, the set of size-2 candidates and the set of table instances of C_2 can be generated from the neighbor instance pairs. For $k > 2$, size- k candidate co-locations and table instances of them are generated from size- $(k-1)$ high utility co-location patterns. Here, we use the antimonotone of utility participation index to prune the set of generated candidate patterns.

Discovering High Utility Co-locations (Step 7–8): First, we calculate the UPI of each candidate co-location according to the Definition 6. Then, we identify high utility co-location patterns by the UPIs of candidate co-locations and the given UPI threshold θ .

Steps 4–9 are repeated with the increment of size k . Note that the row instances which lie across two regions are ignored in Algorithm 1.

3.2 Improved Algorithm

The basic UCP mining algorithm can efficiently discover utility co-location patterns due to the antimonotone property of UPI. In this section, we present an improved UCP mining algorithm for reducing the computational cost.

As shown in Fig. 1, the instances of feature B and C in pattern $\{B, C\}$ exist in a high utility region, but there is no row instance of $\{B, C\}$. Therefore, the utility participation index of $\{B, C\}$ is low. Through this observation, we get the improvement of basic algorithm: m regions are sorted from high to low according to the utility values, and the regional number is $1, 2, \dots, m$ in turn. And then we mine high utility co-location pattern with the strategy of generate-and-test based on the sorting regions.

Definition 7 (utility of feature within a region). *Within region l , suppose S_i^l is a set of instances belonging to a feature f_i . The utility of f_i within region l is denoted as $u_l(f_i) = \sum_{f_i, j \in S_i^l} u(f_i, j)$.*

For example, in Fig. 1, the utility of feature A within Region 1 is $u_1(A) = u(A, 6) = 8$.

Definition 8 (utility of feature in co-location within a region). *Within region l , given a size- k co-location pattern $c = \{f_1, f_2, \dots, f_k\}$, the utility of f_i in c is defined as the sum of utilities of instances belonging to feature $f_i \in c$ in $R^l(c)$, where $R^l(c)$ is the set of row instance of c within region l . It is denoted as $u_l(c, f_i) = \sum_{f_i, j \in \pi_{f_i}(R^l(c))} u(f_i, j)$, where π is the relational projection operation with duplication elimination.*

For example, in Fig. 1, within Region 1, considering the size-2 co-location pattern $c = \{A, C\}$, $R^1(c) = \{A.6, C.6\}$. The utility of A in c within Region 1 is $u_1(c, A) = u(A.6) = 8$.

Lemma 2. *Let m regions be sorted from high to low according to the utility value, and the regional number is from 1, 2, ..., m in turn. After the related utility values of the first l regions are calculated, the maximum utility participation ratio of feature f_i is $UPR(c, f_i)_{\max l}$.*

$$UPR(c, f_i)_{\max l} = \frac{u(f_i) - \sum_{r=1}^l u_r(f_i) + \sum_{r=1}^l u_r(c, f_i)}{u(f_i)} \quad (5)$$

Proof:

$$\begin{aligned} UPR(c, f_i) &= \frac{\sum_{r=1}^l u_r(c, f_i) + \sum_{r=l+1}^m u_r(c, f_i)}{u(f_i)} \leq \frac{\sum_{r=1}^l u_r(c, f_i) + \sum_{r=l+1}^m u_r(f_i)}{u(f_i)} \\ UPR(c, f_i) &\leq \frac{\sum_{r=1}^l u_r(c, f_i) + u(f_i) - \sum_{r=1}^l u_r(f_i)}{u(f_i)} \end{aligned}$$

Theorem 2. *Given a size- k co-location pattern $c = \{f_1, f_2, \dots, f_k\}$, if $UPR(c, f_i)_{\max l} < \theta$ holds, then c must be a non-high utility co-location pattern, i.e., pattern c can be pruned.*

Proof: According to Definition 6 and Lemma 2, we have $UPI(c) \leq UPR(c, f_i)_{\max l}$. So, when $UPR(c, f_i)_{\max l} < \theta$ holds, pattern c can be pruned.

For example, in Fig. 1, for pattern $c = \{B, C\}$, we have

$$\begin{aligned} UPR(c, B)_{\max l} &= \frac{u(B) - u_1(B) + u_1(c, B)}{u(B)} = \frac{13 - 0 + 0}{13} = 1 \\ UPR(c, C)_{\max l} &= \frac{u(C) - u_1(C) + u_1(c, C)}{u(C)} = \frac{26 - 16 + 0}{26} \approx 0.38 \end{aligned}$$

If minimum UPI threshold θ is 0.6, pattern {B, C} can be pruned by Theorem 2.

Algorithm 2. Improved UCP Mining

Input: F : a set of spatial features, S : a set of spatial instances, d : a distance threshold, θ : a utility participation index threshold.

Output: A set of high utility co-locations.

Variables:

k : co-location size

C_k : a set of size- k candidates

T_k^l : a set of table instances of C_k within region l

U_k : a set of size- k high utility co-location patterns

Methods:

1. The data space is divided into m regions;
2. sort regions and make the utility values of region 1, 2, ..., m meet $u_1 > u_2 > \dots > u_m$;
3. $U_1 = F$; $T_1 = S$;
4. for ($k = 2$; $U_{k-1} \neq \emptyset$; $k++$) do
5. $C_k = \text{gen_candidate_colocations}(k, U_{k-1})$;
6. for each c in C_k
7. for $l = 1$ to m do
8. $t_c = \text{gen_table_instances}(c, T_{k-1}^l, d)$;
9. Calculate $UPR_{\max l}(c, t_c)$;
10. if $UPR(c, f_i)_{\max l} < \theta$ then
11. remove c from C_k ; break;
12. end if
13. append t_c to T_k^l ;
14. end for
15. end for
16. $U_k \leftarrow C_k$;
17. end for
18. Return $\cup(U_2, U_3, \dots, U_k)$

According to Theorem 2, the basic mining algorithm can be optimized. Instead of scanning the entire dataset, candidate patterns and their table instances can be generated and UPI can be calculated. In particular, the initialization is similar to the basic algorithm, but the m regions should be sorted from high to low (step 2) according to their utility values. Then, we use size- $(k-1)$ high utility co-location patterns to generate size- k candidate co-locations (step 5). And then, the table instance t_c is generated according to the regional order, one by one. Finally, the $UPR(c, f_i)_{\max l}$ is calculated. The unpruned candidate pattern C_k is the high utility pattern U_k (step 6–17). In the process of generating the table instance t_c of each pattern c , the time consumption for the improved algorithm will be reduced as a result of reducing the scanning scope of the dataset. Algorithm 2 shows the pseudocode of the improved UPC mining.

4 Experimental Evaluation

In this section, we perform a series of experiments to verify the effect and the efficiency of the basic UCP (BUCP) algorithm and the improved UCP (IUCP) algorithm on synthetic and real datasets.

The algorithms are implemented in JAVA and are memory-based algorithms. All the experiments were performed on an Intel core i7-6700 3.4 GHz PC with 8G MB main memory, running on Microsoft Windows10.

4.1 Datasets

For the synthetic dataset, we generate instances and randomly distribute them into a 1000×1000 space, and the utilities of regions are assigned randomly between 1 and 10. For the real dataset, we use a plant dataset of the “Three Parallel Rivers of Yunnan Protected Areas”, we set the utilities of regions by an expert judge method.

4.2 Quality of Mining Results

The criterion $Q(c) = \frac{\sum_{f \in c} u(c, f)}{\sum_{f \in c} u(f)}$ is used to evaluate the quality of a mined co-location pattern c , and the bigger $Q(c)$ is, c has the higher utilities [3]. We compare the quality of mining results identified by the traditional participation index (PI) and the UPI proposed in our paper.

Mining results on the synthetic dataset. The number of spatial features we take is 20; the total number of instances n is 15 K; the distance threshold d is 25; and the number of regions m is 9. Figure 2(a) shows the sum of quality of top- k interesting co-location patterns identified by the measure PI and UPI respectively. The results show that UPI measure can discover higher quality co-location patterns.

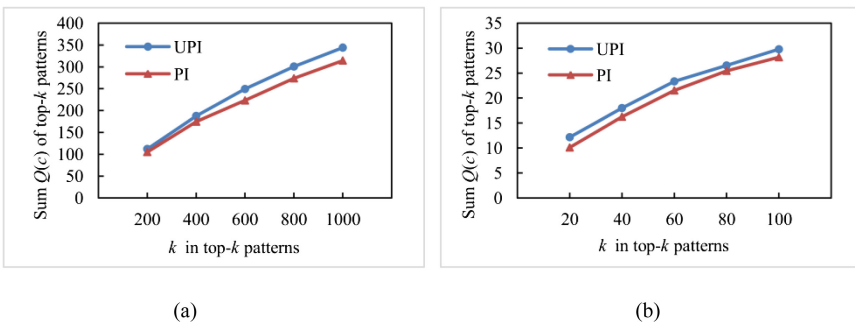


Fig. 2. The quality of mining results (a) on the synthetic dataset, (b) on the real dataset

Mining results on the real dataset. In Fig. 2(b), we use a real-world plant distribution dataset which the number of plant species (spatial features) is 25 and the number of instances is 13348 in a $90 \text{ km} \times 90 \text{ km}$ area. According to the soil properties, this

space is divided into 7 regions. With the same results of the synthetic dataset, the quality of mining results by UPI measure is still better than the quality of mining results by PI measure.

4.3 UCP Mining Performance Evaluation

In this subsection, we compare running time of BUCP algorithm and IUCP algorithm on synthetic datasets.

Effect of the number of instances n . The number of spatial features we set is 20; the distance threshold d is 10; the number of regions m is 25; and the utility participation index threshold θ is 0.3, the running time of two algorithms by increasing the number of instances n is shown in Fig. 3(a). We can see that with the increase of n , the running time of the two algorithms is increased. This is because that the increase of n will produce more clique instances, leading to more time consumption. When $n > 70$ K, the algorithm IUCP is obviously superior to that of BUCP, because the IUCP algorithm can prune non-high utility candidate patterns through scanning parts of data space.

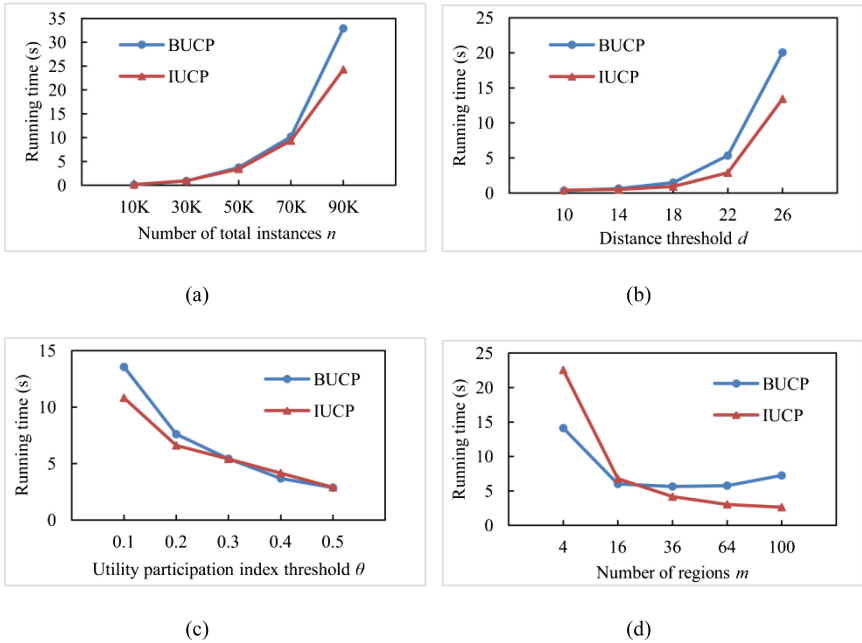


Fig. 3. Running time of UCP mining (a) by number of instances, (b) by distance threshold, (c) by utility participation index threshold, (d) by number of regions

Effect of the distance threshold d . In Fig. 3(b), the number of spatial features is 20, $n = 20$ K, $m = 25$, $\theta = 0.3$. Similarly, the running time is increased with the increase of d , because a larger value of d means more instances which could form cliques to bring

more join operations. The performance of IUCP algorithm is still better than BUCP algorithm, as there is a pruning strategy in IUCP algorithm.

Effect of the utility participation index threshold θ . In Fig. 3(c), the number of spatial features is 20, $n = 60$ K, $m = 25$, $d = 10$. As θ becomes higher, more patterns dissatisfy the condition of the high utility patterns, naturally, the performance of two algorithms meliorate. When $\theta < 0.3$, the running time of IUCP algorithm is less than BUCP. When $\theta > 0.3$, the running time of the two algorithms is almost equal. This is because that when θ value is larger, BUCP algorithm has greatly reduced the running time by using antimonotone property of UPI.

Effect of number of regions m . In Fig. 3(d), the number of spatial features is 20, $n = 60$ K, $d = 10$, $\theta = 0.3$. When $m > 25$, the running time of IUCP algorithm is less than that of BUCP algorithm. Besides, with the increase of m , the pruning utility of IUCP algorithm is better and its running time is less, while the UPI calculation amount of the algorithm BUCP is increased and its running time is slightly increased. When $m < 25$, with the increase of m , the running time of the two algorithms is reduced. However, the running time of IUCP algorithm is more than BUCP, because the pruning strategy of IUCP algorithm is invalid and it scans the entire datasets when identifying high utility patterns. In this way, the table instance is generated in each region and UPI is calculated, thus, increasing the running time.

5 Conclusions

In this paper, we present a method to find high utility co-location patterns based on importance of spatial regions. We use the utility participation index of the patterns as interestingness measure and present BUCP algorithm and IUCP algorithm with a pruning strategy. The experimental results show that we can effectively mine the high utility co-location patterns. IUCP algorithm achieves superior performance than to BUCP algorithm. In the future, we plan to validate our method with different types of datasets.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61472346, 61662086, 61762090), the Natural Science Foundation of Yunnan Province (2015FB114,2016FA026), the Project of Innovative Research Team of Yunnan Province (XT412011), and the Spectrum Sensing and Borderlands Security Key Laboratory of Universities in Yunnan (C6165903).

References

1. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: a summary of results. In: Proceedings of 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD), pp. 236–256 (2001)
2. Xiao, X., Xie, X., Luo, Q., Ma, W.: Density based co-location pattern discovery. In: Proceedings of 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–10 (2008)

3. Wang, L., Jiang, W., Chen, H., Fang, Y.: Efficiently mining high utility co-location patterns from spatial data sets with instance-specific utilities. In: Proceedings of 22nd International Conference on Database Systems for Advanced Applications, pp. 458–474 (2017)
4. Morimoto, Y.: Mining frequent neighboring class sets in spatial databases. In: Proceedings of 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–358 (2001)
5. Huang, Y., Shekhar, S., Xiong, H.: Discovering colocation patterns from spatial data sets: a general approach. *IEEE Trans. Knowl. Data Eng.* **16**(12), 1472–1485 (2004)
6. Yoo, J.S., Shekhar, S.: A joinless approach for mining spatial colocation patterns. *IEEE Trans. Knowl. Data Eng.* **18**(10), 1323–1337 (2006)
7. Wang, L., Bao, Y., Lu, J., Yip, J.: A new join-less approach for co-location pattern mining. In: Proceedings of 8th IEEE International Conference on Computer and Information Technology (CIT2008), pp. 197–202 (2008)
8. Wang, L., Zhou, L., Lu, J., Yip, J.: An order-clique-based approach for mining maximal co-locations. *Inf. Sci.* **179**(19), 3370–3382 (2009)
9. Celik, M., Kang, J.M., Shekhar, S.: Zonal co-location pattern discovery with dynamic parameters. In: Proceedings of 7th IEEE International Conference on Data Mining, pp. 433–438 (2007)
10. Dai, B.R., Lin, M.Y.: Efficiently mining dynamic zonal co-location patterns based on maximal co-locations. In: Proceedings of 11th IEEE International Conference on Data Mining Workshops, pp. 861–868 (2011)
11. Sengstock, C., Gertz, M., Canh, T.V.: Spatial interestingness measures for co-location pattern mining. In: Proceedings of 12th IEEE International Conference on Data Mining Workshops, pp. 821–826 (2012)
12. Zhao, J., Wang, L., Bao, X., Tan, Y.: Mining co-location patterns with spatial distribution characteristics. In: International Conference on Computer, Information and Telecommunication Systems, pp. 26–30 (2016)
13. Yao, H., Hamilton, H.J., Butz, C.J.: A foundational approach to mining itemset utilities from databases. In: Proceedings of 4th SIAM International Conference on Data Mining, pp. 482–486 (2004)
14. Liu, Y., Liao, W., Choudhary, A.: A two-phase algorithm for fast discovery of high utility itemsets. In: Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pp. 689–695 (2005)
15. Tseng, V.S., Wu, C.W., Shie, B.E., Yu, P.S.: UP-Growth: an efficient algorithm for high utility itemset mining. In: Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 253–262 (2010)
16. Yang, S., Wang, L., Bao, X., Lu, J.: A framework for mining spatial high utility co-location patterns. In: Proceedings 12th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 595–601 (2015)
17. Wang, X., Wang, L.: Incremental mining of high utility co-locations from spatial database. In: IEEE International Conference on Big Data and Smart Computing, pp. 215–222 (2017)