Fuzzy Systems and Data Mining VIII
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220400

Spatial Feature Evolutionary Relationship Discovery Based on Tree Structure Pattern Mining and Spatial Co-Location Pattern Mining in Spatial Cliques

Kaishuo LIN [a], Lizhen WANG [b,1] and Vanha TRAN [c]

[a] *School of Information Science and Engineering, Yunnan University, Dongwaihuan SouthRoad, Kunming 650500, China*
[b] *Dianchi College of Yunnan University, Kunming 650228, China*
[c] *FPT University, Hanoi 155514, Vietnam*

Abstract. Feature evolutionary relationship discovery based on tree structure pattern mining and spatial co-location pattern mining in spatial cliques aims to combine spatial co-location pattern mining algorithm and tree structure pattern mining algorithm to find spatial cliques in spatial data and evolutionary relationships of spatial features in these spatial cliques. This allows for a deeper analysis of possible causal relationships between spatial features, predicting the occurrence of other spatial features based on the occurrence of certain spatial features, or summarizing general laws. For a spatial clique, the appearance of one spatial feature may lead to the appearance of another spatial feature, which is called evolutionary relationship. Previous spatial co-location pattern mining algorithms (such as join-based) focus on discovering prevalent co-occur spatial features and prevalent co-location patterns. We further discovered evolutionary relationships between prevalent co-occur spatial features in spatial cliques through tree structure pattern mining algorithm.

Keywords. Spatial co-location pattern mining, Tree structure pattern mining, Evolutionary relationship

1. Introduction

Nowadays, all kinds of data are growing rapidly, including spatial data. More and more spatial data mining techniques are being developed to discover hidden patterns of interest. Spatial co-location pattern is one of patterns that people are interested in. Many related research efforts for spatial co-location patterns mining has been proposed, such as the joinless [1] approach, it can obtain all prevalent spatial co-location patterns. When discovering spatial co-location patterns, prevalent co-occur and adjacent spatial feature instances (referred to here as spatial cliques) are preserved, which are used to generate

[1] Corresponding author: Lizhen Wang, Dianchi College of Yunnan University, Kunming 650228, China. E-mail: lzhwang@ynu.edu.cn

prevalent spatial co-location patterns. For spatial cliques, prevalent co-occur and adjacent spatial feature instances symbolize prevalent co-occur and adjacent of spatial features.

One spatial feature emerges may cause another spatial feature emerges(referred to here as evolutionary relationship). For example, for a region, decrease of temperature will cause migration of wild geese, and migration of wild geese will cause increase of local crop storage. So, In this region, decrease of temperature, migration of wild geese and increase of local crop storage constitute a spatial feature evolutionary relationship.

At present, in the field of spatial co-location pattern mining, most studies focus on mining patterns that people are interested in, few people involve in mining the evolutionary relationship of spatial features. The reason is that spatial co-location mining is based on spatial data, which usually does not contain attribute of time, while concept of evolutionary relationship has a great relationship with attribute of time. By adding attribute of time, a framework based on tree structure pattern mining and spatial co-location pattern mining is proposed to discover the evolutionary relationship in spatial cliques.

The main contributions of this paper are as follows:

(1) On the basic of spatial co-location patterns mining algorithm, we combined with tree tructure mining algorithm to find evolutionary relationship in spatial cliques.

(2) The framework overcomes the difficulty that the spatial co-location pattern can only find the interested patterns on the spatial data without attribute of time.

(3) By adjusting the time threshold and distance threshold, the evolutionary relationship between spatial features at different scales can be found.

The remainder of this paper is organized as follows. Section 2 reviews the related work. The basic concepts are noted in Section 3. The main mining process and framework are described in Section 4. The experimental results are demonstrated in Section 5. Lastly, the conclusion and future work are discussed in Section 6.

2. Related work

Spatial data mining [2] is a popular research target for researchers. Mining spatial co-location patterns is to discover spatial patterns from spatial data which spatial features prevalently co-occured. For mining of co-location patterns efficiently, many approaches have been proposed, such as join-based [3] and mapreduce-based algorithms [4].

To process different types of spatial data, Ouyang [5] researched how to discover co-location patterns from fuzzy spatial data, Wang addressed the problem of mining co-location patterns from uncertain spatial data [6]. In recent years, some cencepts have been added to spatial co-location pattern mining to obtain potential and valuable patterns. Wang [8] mining spatial co-location pattern by incorporating fuzzy theory. Wu [9] mining prevalent co-location spatial patterns by a maximal ordered ego-clique based approach and Hu [10] mining prevalent co-location spatial patterns by utilizing fuzzy grid cliques.

Many spatial co-location pattern mining algorithms are focused on spatial data, which does not include the dimension of time. In order to mine the spatio-temporal data, it is necessary to do some expansion and innovation on original mining algorithms.

Tree structure pattern mining is one of the important directions of data mining, sub-tree mining has been used in Web mining and other aspects. The existing sub-tree mining algorithms are divided into three categories: (1) apriori series algorithms, such as

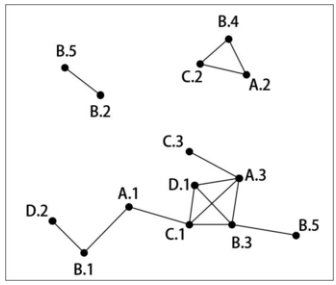


Figure 1. Spatial features and their spatial instances' distribution

TreeMiner [11]; (2) algorithms of generating candidate sub-trees by enumeration trees, such as FREQT [12]; (3) algorithms based on the principle of pattern growth.

The hierarchical relationship between tree structures can be related to the property of time. Therefore, the tree structure pattern mining algorithm is applied on the spatial co-location pattern mining algorithm, which can well extend the application field from two-dimensional spatial data to three-dimensional spatio-temporal data.

3. Basic concepts

In this section, concepts such as spatial co-location pattern, spatial clique, evolutionary relationship, and prevalent unordered embedded sub-tree will be described.

3.1. Spatial co-location pattern, spatial clique and evolutionary relationship

Definition 1. *Spatial feature*

Spatial features represent different kinds of things in spatial. A set of spatial features represent a set of different kinds of things in spatial, denoted as $F = \{f_1, f_2, ..., f_n\}$.

Definition 2. *Spatial instance*

An object at a specific location is called a spatial instance. The set of instances is called the instance set, denoted as $S = S_1 \cup S_2 \cup ...S_n$, *where* $S_i(1 \leq i \leq n)$ *is the instance set of corresponding spatial feature* f_i. *In general, spatial instances do not contain the property of time, but in order to discover evolutionary relationships, in this work, the spatial instance refers to the object at a specific time and a specific location, represented by a quad* $<$ *instance attribute, instance number, spatial location, occurrence time* $>$.

Definition 3. *Spatial co-location pattern*

A spatial co-location pattern is a set of spatial features c, where $c \subseteq F$. *In Figure 1,* $\{A, B, C\}$ *is a spatial co-location pattern.*

Definition 4. *Spatial clique*

Given the spatial instance set $I = \{i_1, i_2, ..., i_m\}$, *if for* $1 \leq j \leq m, 1 \leq k \leq m$, *the condition* $0 \leq |distance(i_j, i_k)| \leq D - threshold$ *is satisfied, then I is said to be a spatial clique. A spatial clique is a subset of the complete set formed by adjacent spatial instances in spatial. In Figure 1,* $\{A.3, B.3, C.1, D.1\}$ *is a spatial clique.*

Lemma 1. *If a N-size clique is a spatial clique, then any of its N-1-size sub-cliques satisfies the definition of the spatial clique.*
Proof. Given a 2-size spatial clique $I = \{i, j\}$, *and a spatial instance k, if and only if between i and k is satisfied the condition* $0 \leq |distance(i,k)| \leq D-threshold$, *and between j and k is satisfied the condition* $0 \leq |distance(j,k)| \leq D-threshold$. *Both of clique* $\{j,k\}$ *and clique* $\{i,k\}$ *are spatial clique, then the clique* $\{i,j,k\}$ *is the spatial clique.*

Definition 5. *Dependency relationship*

For two spatital instances i_1 *and* i_2, *if they have a dependency relationship, the following conditions must be met: (1)* $0 \leq |distance(i_1,i_2)| \leq D-threshold$; *(2)* $0 \leq |i_1.time - i_2.time| \leq T-threshold$, *denoted as* $Dep(i_1,i_2)$. $D-threshold$ *is the distance threshold given by the user and* $T-threshold$ *is the time threshold given by the user.*

Definition 6. *Parent-child relationship*

For two spatial instances i_1 *and* i_2, i_1 *is parent entity of* i_2 *if* $Dep(i_1,i_2)$ *and* i_1 *occurs before* i_2. *Parent-child relationship is a manifestation of evolutionary relationship.*

Definition 7. *Evolutionary relationship*

If in all spatial instances, the corresponding instances of spatial feature f_1 *and* f_2 *always appear together and the corresponding spatial instance of* f_1 *always occur before the corresponding instance of* f_2, *then* f_1 *and* f_2 *are said to have an evolutionary relationship and evolve from* f_1 *to* f_2. *For example, without considering the support degree, given the spatial clique* $I_1 = \{i_{f_1}, i_{f_2}, i_{f_3}\}$ *and* $I_2 = \{i_{f_1}, i_{f_2}, i_{f_4}\}$, i_{f_1} *is the spatial instance corresponding to the spatial feature* f_1, i_{f_2} *is the spatial instance corresponding to the spatial feature* f_2. *If* i_{f_1} *and* i_{f_2} *satisfy* $Dep(i_{f_1}, i_{f_2})$, *and* i_{f_1} *occurs before* i_{f_2}, *then* f_1 *and* f_2 *are said to have an evolutionary relationship, and the evolution from* f_1 *to* f_2.

3.2. Prevalent unordered embedded sub-tree

Definition 8. *Tree and forest*

Tree can be expressed as a quintuple $T = (V, v_0, E, \Sigma, L)$, V *is a given set of nodes;* v_0 *is the root of the tree;* E *is the edge set of the tree;* Σ *is a set of identifiers;* L *is a mapping from* V *to* Σ. *For* $(v_1, v_2) \in E$, $v_1, v_2 \in V$, v_1 *is a parent element of* v_2, *and* v_2 *is the child of* v_1. *The collection of trees is called a forest.*

Definition 9. *Unordered tree*

For a parent element in a tree, if the children are unordered, the tree is called an unordered tree, otherwise it is called a order tree.

Definition 10. *Embedded sub-tree*

Given a tree $T = (V, E)$, *and a sub-tree* $S = (V', E')$, *where* $V' \subset V$ *and* $E' \subset E$. *A sub-tree S is said to be an embedded sub-tree of T if for each edge* $e = (v_a, v_b) \in E'$, v_a *is an ancestor (and not necessarily the parent) of* v_b *in T. If for each edge* $e = (v_a, v_b) \in E'$, v_a *is a parent of* v_b *in T, then the sub-tree S is an induced sub-tree of T.*

Unordered trees contain more valuable information than ordered trees. Embedded sub-trees contain more valuable information than induced sub-trees. In Figure 2, if we mine induced sub-trees, no prevalent trees of size more than one are found. If we mine ordered embedded sub-trees, no prevalent trees of size more than one are found. But if

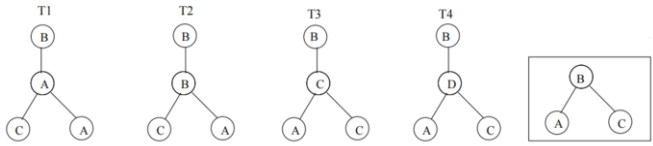
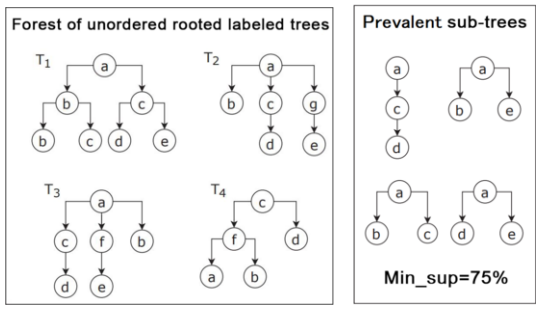
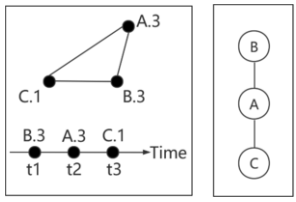


Figure 2. Unordered embedded sub-tree



(a) (b)

Figure 3. (a) A forest, (b) Four of prevalent embedded occurred sub-trees with a minimum support 75%



(a) (b)

Figure 4. (a) A spatial clique and its spatial instance occurrence time, (b) The tree after building

we mine unordered embedded sub-trees, the tree in the box is a prevalent sub-tree in forest. In Figure 3(a), it shows the forest composed of several examples of tree structure. In order to discover the evolutionary relationship between spatial features, in this work, every node in the tree is a spatial feature. Serveral prevalent unordered embedded sub-trees with a minimum support are shown in Figure 3(b).

A spatial clique prevalently appear is represented as a tree. In a tree, the dependency relationship is the edge of the tree, and the parent-child relationship is determined by the occurrence time between spatial instances which satisfying the dependency relationship in the spatial clique. For example, in Figure 4(a), given a spatial clique and its instances occurrence time, the tree after building is shown in Figure 4(b).

Thus, given a forest $F = \{T_1, T_2, ..., T_k\}$, the discovery of evolutionary relationship of spatial features is finding unordered embedded sub-trees with high prevalency in F. For a sub-tree S and tree T we define $support(S,T)$ by Eq.(1) and $support(S,F)$ by Eq.(2):

$$support(S,T) = \begin{cases} 1, \text{if } S \text{ is a sub-tree of } T \\ 0, \text{otherwise} \end{cases} \quad (1)$$

$$support(S,F) = \frac{\Sigma_{T \in F} support(S,T)}{|F|} \quad (2)$$

For a sub-tree S, if $support(S,F) \geq min\_sup$, where min_sup is a minimum support threshold specified by users, then we say S is a prevalent embedded sub-tree in F.

4. Main mining process

The algorithm framework is based on joinless [1] and SLEUTH [7].

4.1. Joinless [1]

The joinless [1] algorithm can detect prevalent spatial co-location patterns and spatial cliques. Compared with the join-based [3] algorithm, the joinless [1] algorithm uses the star partition model to represent spatial neighbor relationships and can find spatial cliques faster. The joinless [1] algorithm can find spatial cliques and prevalent spatial co-location patterns. Since this work is to discover the evolutionary relationship of spatial features in spatial cliques, we use the joinless [1] algorithm to discover all spatial cliques, and the content of discovering prevalent spatial co-location patterns is simplified in this work.

4.2. SLEUTH [7]

The SLEUTH [7] algorithm can generate candidate sub-trees by enumeration trees. It uses equivalence class extension scheme and scope-list join to mine prevalent unordered embedded sub-trees. The contents of equivalence class extension scheme and scope-list are described in article SLEUTH [7], and will not be repeated here .

4.3. Mining process and framework

We use a multi-step process to discover evolutionary relationships: 1) finding all spatial cliques; 2) finding dependency relationships; 3) building all trees by parent-child relationships; 4) finding all prevalent unordered embeded sub-trees.

•**Finding all spatial cliques:** The first step is to find all spatial cliques in spatial data, using the spatial clique defined in Section 3 and the joinless [1] algorithm. In Figure 1, the 3-size spatial cliques that can be found are shown in Figure 5.

•**Finding dependency relationships:** Finding all spatial instance pairs that satisfy dependency relationship from all the spatial cliques. All the dependent pairs of spatial instances to define the parent-child relationship for each pair.

•**Building all trees by parent-child relationships:** In all pairs of spatial instances satisfying the dependency relationship, the spatial instance of feature occurring earlier is parent node of the spatial instance of feature occurring later. In this step, we find all spa-

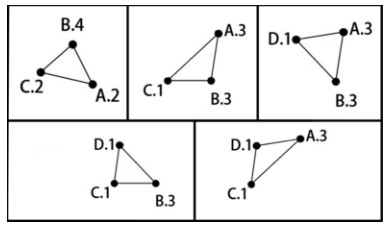


Figure 5. The 3-size spatial cliques

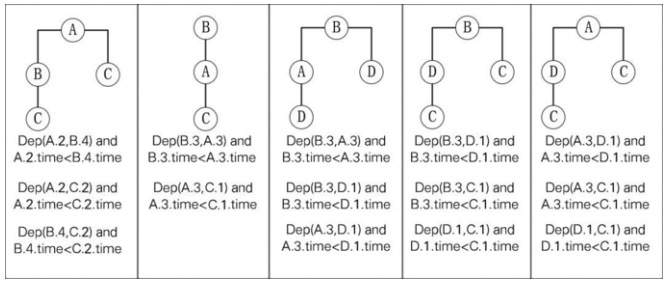


Figure 6. A forest that may be generated by cliques

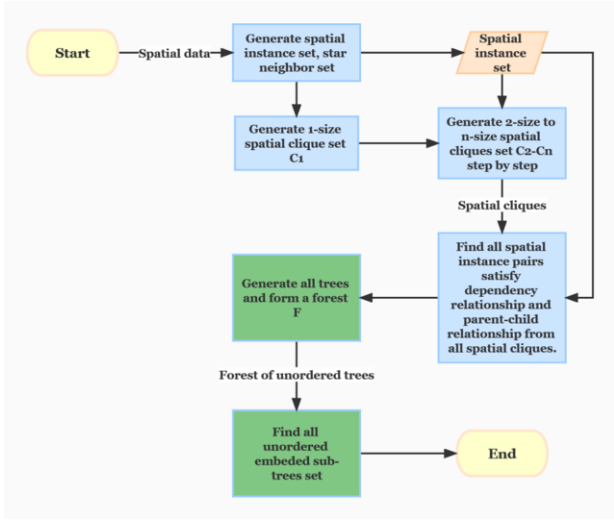


Figure 7. The mining framework

tial instance pairs with parent-child relationship and build all trees to form a forest. The forest that may be generated by the 3-size cliques in Figure 5 is shown in Figure 6.

•**Finding all prevalent unordered embeded sub-trees:** The last step is to perform tree structure pattern mining. The goal is to find all unordered embedded sub-trees in all trees that satisfy the minimum support threshold given by users.

Figure 7 shows the flow chart of the mining framework. Step 1 is to process spatial data. In Step 2, 1-size spatial clique set C_1 is generated. Step 3 is to generate 2-size to

n-size spatial cliques set $C_2 - C_n$ iteratively. Step 4 is to find all spatial instance pairs satisfy dependency relationship and parent-child relationship. Step 5 is to generate all unordered trees and from a forest F. Step 6 is to mine all unordered embedded sub-trees from the forest. Step 1 to Step 3 are shown in Algorithm 1, Step 4 is shown in Algorithm 2, and Step 5 and Step 6 are shown in Algorithm 3.

Algorithm 1

```
Input: S = {s1, s2, ..., sn}: a set of spatial instance;
       distance: D-threshold;
instanceSet = buildInstanceSet(S);
starNeighborSet = buildStarNeighborSet(instanceSet,distance);
c_curSet = c1 = build1SizeSpatialClique(instanceSet);
c2_cNSet = null, c_cur = null;
while c_curSet != null do
    newc_curSet = null;
    for c_cur in c_curSet do
        for i in instanceSet do
            if !c_cur.contains(i) then
                canJoinSpatialClique = true;
                for j in c_cur do
                    if !starNeighborSet.getNeighbor(j).contains(i) then
                        canJoinSpatialClique = false; break;
                    end if
                end for
                if canJoinSpatialClique then
                    newc_curSet.add(c_cur.clone().add(i));
                end if
            end if
        end for
    end for
    if newc_curSet != null then
        c_curSet = newc_curSet; c2_cNSet.add(c_curSet);
    end if
end while
```

In Algorithm 1, Step 1 is to process spatial data and form instanceSet. Step 2 is to generate star neighbor set defined in joinless [1]. Step 3 until the end of Algorithm 1 is to generate all spatial cliques from low size to high size iteratively. The core idea is to find a spatial instance that does not exist in the current spatial clique, and if it neighbors with each spatial instance of the spatial clique, then join the spatial clique.

Algorithm 2

```
Input: c2_cNSet: all spatial cliques;
       instanceSet: all instances;
       time: T-threshold;
spatialCliqueSetToBuildTree = null;
for c_curSet in c2_cNSet do
    for c_cur in c_curSet do
        i = c_cur.start();
```

```
5:          while i != c_cur.end() do
6:              j = c_cur.get(i+1);
7:              while j != c_cur.end() do
8:                  if Dep(i,j,time) then
9:                      spatialCliqueSetToBuildTree.add(c_cur); break;
10:                 end if
11:                 j = c_cur.get(j+1);
12:             end while
13:             i = c_cur.get(i+1);
14:         end while
15:     end for
16: end for
```

In Algorithm 2, we mainly find out spatial cliques which can satisfy the time threshold. As long as there is a spatial instance pair of the spatial clique satisfy dependency relationship and parent-child relationship, this spatial clique can form a tree with at least 2 nodes and will bepreserved. Step 4 to Step 14 is to check whether the spatial instance pairs formed by pairwise spatial instances within the spatial clique satisfy the dependency relationship.

Algorithm 3

Input: *spatialCliqueSetToBuildTree*: spatial clique set which can build trees;
minSTSup: minimum prevalent unordered embedded sub-tree support;
Output: *UESTSet*: prevalent unordered embedded sub-trees that satisfy minSTSup;

```
1: forest = null; UESTSet = null;
2: for sC in spatialCliqueSetToBuildTree do
3:     forest.add(buildTree(sC));
4: end for
5: eC = forest.buildEquivalenceClass();
6: while true do
7:     newEC = equivalenceClass.buildNewEquivalenceClass(minSTSup);
8:     if newEC != null then
9:         for subTree in newEC do
10:            UESTSet.add(subTree);
11:        end for
12:        eC = newEC;
13:    end if
14: end while
15: return UESTSet;
```

Algorithm 3 is to form forests and use the method of equivalence class extension used in SLEUTH [7] algorithm. The 1-size equivalence class is generated according to forests, all equivalence classes are generated iteratively, and all trees that satisfy minimal support degree are lifted out and returned. Step 2 to 5 are to generate forests, Step 6 is to create 1-size equivalence class according to the forest, Step 8 to Step 16 are to iteratively generating all equivalence classes and extracting unordered embedded sub-trees.

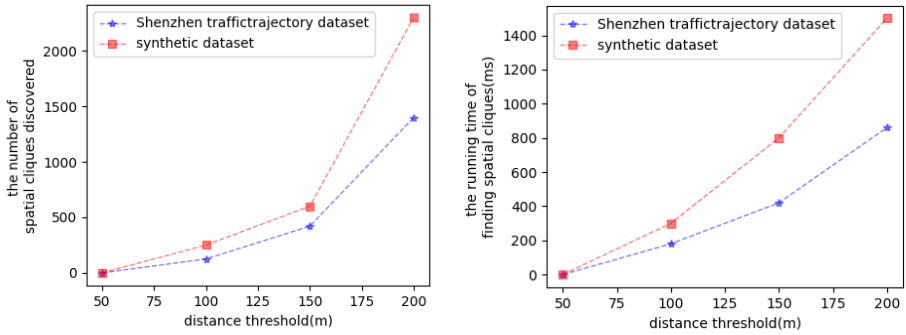


Figure 8. The number of spatial cliques and the efficiency of finding spatial cliques

5. Experimental results

In this section, we implement our algorithm framework by *Java* and experiment it with two datasets. The experimental environment is as follow: cpu with Intel i5-6500 @ 3.20GHz, memory with 8GB. We conducted experiments and evaluate the performance of our algorithm framework on a traffic trajectory dataset and a synthetic dataset. Two datasets used in the experiment are described in Table 1.

Table 1. The description of data sets

Spatial dataset	Number of instances	Number of features
Shenzhen traffic trajectory dataset	10000	100
Synthetic dataset	10000	50

5.1. Distance threshold

We evaluate the impact of different distance thresholds on the results, including the number of spatial cliques discovered and the efficiency of finding spatial cliques. In Figure 8, synthetic dataset spatial clique number more than Shenzhen traffic trajectory dataset, its reason is that spatial instances distribution of the Shenzhen traffic trajectroy dataset is more dispersed, which causes size of spatial cliques found in Shenzhen traffic trajectory dataset is less than synthetic dataset. With the increase of distance threshold, the number of space cliques increases, which leads to the increase of time consumption.

5.2. Time threshold

We evaluate the impact of time threshold in the experiment, including the number of retained spatial cliques, the total amount of generated trees and the efficiency of the algorithm framework. Since different minimum prevalent unordered embedded sub-tree support degree will affect results, we set the support degree as 0.15 and 0.3, the distance threshold as 200m to carry out the experiment. In Figure 9, the sight fluctuation of the time threshold will make the number of retained spatial cliques increase sharply, and because the number of spatial cliques is directly proportional to the total amount of trees,

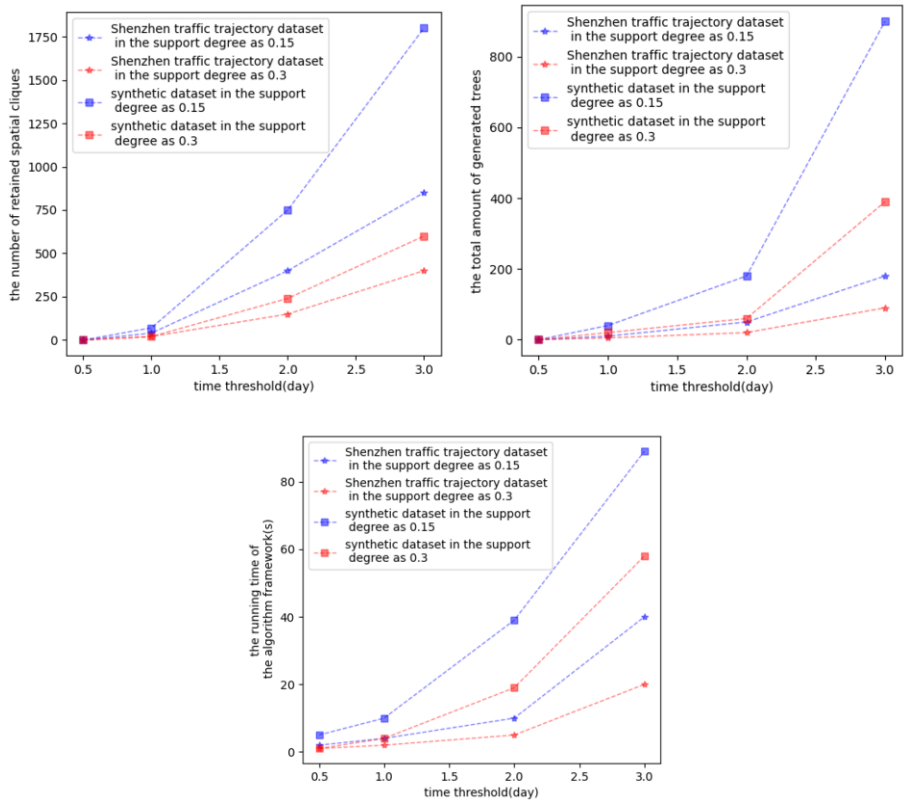


Figure 9. The number of retained spatial cliques, the total amount of generated trees and the efficiency of the algorithm framework

the total amount of trees will become larger, resulting in the lower efficiency of the algorithm framework. With the increase of the time threshold, the number of retained space cliques increases, and the number of forests also increases. Then the time consumption of our algorithm framework increases rapidly.

5.3. Spatial feature evolutionary relationship

The number of sub-trees we mine is the number of spatial feature evolutionary relationships. In this subsection, we evaluate the influence of different minimum prevalent unordered embedded sub-tree support degrees on the number of spatial feature evolutionary relationships. In Figure 10, since both distance threshold and time threshold will affect the final results, we adopt the distance threshold is 200m and the time threshold is 2days.

6. Conclusion and the future work

Previous spatial co-location pattern mining algorithms usually only in the spatial data without time attribute, we through combined with the tree structure pattern mining algo-

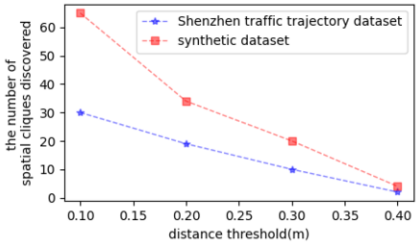


Figure 10. The number of spatial feature evolutionary relationships

rithm, make the spatial co-location pattern mining algorithms to use in spatial data with time attribute, and get the spatial feature evolutionary relationships in the spatial cliques. This makes it possible to discover more potentially useful patterns or relationships.

The combination also has some problems. FThe spatial data is generally large, and the tree structure consumes a lot of memory. As a result, once the amount of data is increased by one magnitude, the number of trees generated will increase sharply, which leads to the reduction of time efficiency and the increase of space occupation, and this also makes our algorithm framework sensitve to the setting of the thresholds. In the future work, we will try to solve these problems.

Acknowledgements This work is supported by the National Natural Science Foundation of China (62276227, 61966036), the Project of Innovative Research Team of Yunnan Province (2018HC019), Yunnan Fundamental Research Projects (202201AS070015), and Yunnan University Postgraduate Technological Innovation Project (2021Y175).

References

[1] L. Wang, Y. Bao, J. Lu, J. Yip: A new join-less approach for co-location pattern mining. In: 2008 8th IEEE International Conference on Computer and Information Technology. pp. 197–202. IEEE (2008)

[2] D. Li, S. Wang, D. Li: Spatial data mining. Springer (2015)

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

[4] J.S. Yoo, D. Boulware, D. Kimmey: A parallel spatial co-location mining algorithm based on mapreduce. In: 2014 IEEE International Congress on Big Data. pp. 25–31. IEEE (2014)

[5] Z. Ouyang, L. Wang, P. Wu: Spatial co-location pattern discovery from fuzzy objects. International Journal on Artificial Intelligence Tools 26(02), 1750003 (2017)

[6] L. Wang, P. Wu, H. Chen: Finding probabilistic prevalent colocations in spatially uncertain data sets. IEEE Transactions on Knowledge and Data Engineering 25(4), 790–804 (2011)

[7] M. J. Zaki. Efficiently mining frequent embedded unordered trees. Fundamenta Informaticae, 65(1/2):1–20, 2005.

[8] Xiaoxuan Wang, Le Lei, Lizhen Wang*, Peizhong Yang, Hongmei Chen. Spatial Co-location Pattern Discovery Incorporating Fuzzy Theory, IEEE Transactions on Fuzzy Systems (TFS), 2022, 30(6): 2055-2072.

[9] Pingping Wu, Lizhen Wang*, Muquan Zou. A Maximal Ordered Ego-clique Based Approach for Prevalent Co-location Pattern Mining, Information Sciences, 608 (2022) 630–654.

[10] Zisong Hu, Lizhen Wang*, Vanha Tran, Hongmei Chen. Efficiently mining spatial co-location patterns utilizing fuzzy grid cliques. Information Sciences, 592 (2022) 361-388

[11] MJ Zaki. Efficiently mining frequent trees in a forest. In Proceedings of the 8th ACM SIGKDD on Knowledge Discovery and Data Mining, 2002, 71-80

[12] T. Asai, K. Abe, S. Kawasoe etc. Efficient substructure discovery from large semi-structured data. IEICE TRANSACTIONS on Information and Systems, 2004, 87 (12): 2754-2763