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

doi:10.3233/FAIA200708

The Coupling Co-Location Pattern: A New Spatial Pattern for Spatial Data Sets

Shiran Zhou, Lizhen Wang[1] and Pingping Wu
School of Information Science and Engineering, Yunnan University, Yunnan, China

Abstract. There is a variety of interesting knowledge in spatial data sets. Spatial co-location pattern mining can discover sets of different features that are co-located. However, this type of pattern only lists the features that appear together without any consideration of the quantity ratio, which can cause confusion. For example, the co-location pattern {church, restaurants} shows that churches and restaurants are often close to each other, but information such as how many restaurants are near a church is usually not displayed. Also, in real spatial data sets, there is a mutual influence between spatial features, that is, a coupling relationship between different features or the same features. Thus, this paper proposes a novel spatial pattern called a coupling co-location pattern. First, we discuss the properties of the coupling phenomenon between spatial features, and then the concept of coupling co-location patterns is defined formally. Second, the measurement of support and mining framework for coupling co-location patterns are proposed. Finally, we conduct experiments on both real and synthetic data sets, and the results verify the practical significance of coupling co-location patterns.

Keywords. Spatial data mining, Coupling co-location pattern (CCP), Maximal clique

1. Introduction

Co-location pattern mining aims to discover subsets of spatial features that are frequently located together, where a spatial feature represents a type of cataloged object in a space. As an example, snack bar shops and beauty salon shops are often located near each other, which form a co-location pattern. In addition, “located together” usually means located in geographic proximity (e.g., measured by Euclidean distance). With the wide application of spatial positioning services, spatial data sets, which record cataloged objects and their locations (e.g., geographic mapping data or point-of-interest data), are generated very quickly. As a result, spatial co-location pattern mining has been applied in many areas such as mobile commerce, earth science, biology, public health, and transportation [1].

Fig. 1 is an example of a spatial data set, where a triangle, square, and circle represent different spatial features. There are four instances of triangles, circles, and squares, respectively, and each instance uses a number as an identifier. For example, ▲$_1$ represents the first instance of the triangle feature. We draw a line to connect two instances if they satisfy the spatial proximity relationship. For the example in Fig. 1, we

[1] Corresponding author: Lizhen Wang, School of Information Science and Engineering, Yunnan University, Dongwaihuan South Road, Kunming, Yunnan 650091, China. E-mail: lzhwang@ynu.edu.cn

obtain the four candidate (potential) co-location patterns {▲, ■}, {▲, ●}, {■, ●}, and {▲, ■, ●}. When a candidate co-location pattern satisfies the support measure, we say it is a prevalent co-location pattern.

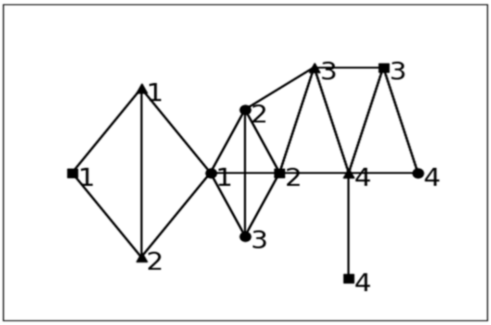

Fig. 1. An example spatial data set.

Several support measures have been proposed, e.g., partitioning-based by S. Shekhar et al [2], construction-based by Y. Morimoto et al [3], enumeration-based by Y. Huang et al [4] and [5] and participation-based by J. S. Yoo [6] and [7]. A partitioning-based measure divides the space into small squares by a set of horizontal lines and vertical lines, counting the number of potential patterns in each small square as the support. Construction-based approaches find candidate patterns heuristically. An enumeration-based measure counts the number of row instances as the support of a pattern, where an instance set of the pattern is said to be a row instance if they are neighbors of each other. For a participation-based measure, the participation rate and participation index are two key indicators. Through the projection of row instances, the participation rates of each feature in a pattern are calculated, and the minimum participation rate of features is considered to be the participation index of the pattern. In addition to this, a new support measurement method called Fraction-Score was proposed by Chen et al [8], where the main idea is to consider instances in row instances as fractional units and then all fractional units are aggregated to form a support of patterns.

A co-location pattern reveals a kind of relationship between spatial features. Consider the following examples: A restaurant in a normal urban area that is located close to a parking area can boost both incomes. If a restaurant is near to a garbage collection station, then the income of the restaurant will be reduced. In nature, the Peony flower and Cosmos are both insect-pollinated flowers. One of them will be pollinated when another attracts insects such as bees, butterflies, moths, and ants to pollinate, if they locate closely. This is a mutually beneficial relationship in nature. That inter-planting beans in a corn field can produce beans without reducing the yield of corn is also an example of this. From these examples, we can see the practical significance of co-location pattern mining.

The co-location relationship of spatial features indicated by co-location patterns looks like a coupling relationship [9]; however, the concept of coupling generally means that there exists a mutual effect between objects, and that this is in a dynamically stable state. Further, there are two aspects of coupling relationships. The first aspect is **intra-coupling**, which means the mutual effect between objects of the same type. For example, consider a restaurant in a certain street and the environmental factor supporting it is people who order food here daily. If the flow of people can support another restaurant here, the two restaurants will compete for turnover, and this reflects a mutual repulsion relationship. But when people know that they can get food here, this can lead to more customers and the turnover of the two restaurants will increase, which reflects a mutually beneficial relationship. Attaining a balance between promotion and repulsion is a coupling relationship between the same type of objects (intra-coupling). The second aspect is **inter-coupling**, which means the coupling relationship between different objects (features). We need to consider both intra-coupling and inter-coupling.

Let us consider a mutual promotion case for snack bars and beverage shops. Unlike traditional co-location patterns [10], we need to confirm whether two snack bars (intra-coupling) nearby one beverage shop (inter-coupling) are more popular than one snack bar near one beverage shop. We are interested in the proportion of the number of these two instances when the intra-coupling and inter-coupling are balanced, not just knowing that they will appear together. Take a tea shop as another example: imagine that there are many tea shops in a city square mall, and each of them is doing good business. In another city, the number of tea shops may well be different because of the difference in the flow of people, the environment, and other factors. What kind of intra-coupling and inter-coupling of different features are prevalent in a certain city is an interesting question and useful for decision-making. In order to find out the coupling relationships, in this paper we propose a new concept that considers both feature combinations (inter-coupling) and quantity combinations (intra-coupling) simultaneously, namely coupling co-location patterns.

We note that the instances that support coupling co-location patterns have a dynamic stability. Assuming that feature A has two instances and feature B has one instance, if the surrounding could support another instance of feature B, it will appear soon. If the surrounding can only support one instance of feature A, then one instance of A will disappear soon. In spatial data sets, instances satisfying the neighbor relationship form a clique; when a limit is reached of the balance of coupling (intra-coupling and inter-coupling) instances form a maximal clique (or a cluster). From this perspective, when extracting coupling co-location patterns from spatial data sets, we can consider reading candidate patterns from clusters [11]. But a better way is that we should consider maximal cliques as co-location instances that support coupling co-location patterns rather than traditional cliques [12] in co-location pattern mining. Considering the data set in Fig. 1 again, we can see that there are eight supporting instances (maximal cliques): $\{\blacktriangle_1, \blacktriangle_2, \blacksquare_1\}$, $\{\blacktriangle_1, \blacktriangle_2, \bullet_1\}$, $\{\bullet_1, \bullet_2, \bullet_{3,} \blacksquare_1\}$, $\{\blacktriangle_3, \blacksquare_2, \bullet_2\}$, $\{\blacktriangle_3, \blacktriangle_4, \blacksquare_2\}$, $\{\blacktriangle_3, \blacktriangle_4, \blacksquare_3\}$, $\{\blacktriangle_4, \blacksquare_3, \bullet_4\}$, and $\{\blacktriangle_4, \blacksquare_4\}$. Obviously, co-location patterns $\{\blacktriangle, \blacksquare\}$, $\{\blacktriangle, \bullet\}$, and $\{\blacksquare, \bullet\}$ have the same feature combinations with coupling co-location patterns $\{\blacktriangle^2, \blacksquare\}$, $\{\blacktriangle^2, \bullet\}$ and $\{\blacksquare, \bullet^3\}$, but the latter contain richer information and are more useful for decision-making.

In summary, our main contributions in this paper include: In section 1, a discussion of the coupling relationship of spatial features, and its practical significance; In section 2, the concept of a coupling co-location pattern is defined formally based on maximal cliques over the spatial data set; In section 3, The coupling co-location pattern mining framework is proposed, and a maximal clique generating algorithm is introduced; In section 4, we conduct extensive experiments on both real and synthetic data sets. The results verify that our proposed coupling co-location pattern concept is useful and has practical application. For example, we find that in Beijing, hotels and clinics frequently appear together, and the ratio of their numbers is 2 to 1.

2. Formal definitions

In this section, we define coupling co-location patterns and associated measures. Let F be a set of spatial features and I be a set of instances of spatial features [13], $i \in I$ is an instance and $i.f$ represents the feature of instance i. Let $R(i_1, i_2)$ denote that instances i_1 and i_2 satisfy a spatial neighbor relationship R, and let MC be the set of maximal cliques formed by instances under the neighbor relationship R. $mc \in MC$ is a maximal clique in MC.

Definition 1: *Coupling co-location pattern* CCP. Let F be a set of spatial features, a coupling co-location pattern is a collection of spatial features with numbers, i.e., $c = \{f_1^{n1}, f_2^{n2}, \dots, f_k^{nk}\}$, where ni denotes the number of instances where f_i appears; $f_i \in F, ni \in N^+$. In particular, ni is omitted when ni is 1.

Suppose $F(c/mc)$ returns the set of features in a coupling co-location pattern (CCP) c or returns a set of features contained in a maximal clique mc, and $N_f(c/mc)$ returns the number of instances of feature f in c/mc.

Definition 2: *Row instance and table instance*. We say a maximal clique mc is a row instance of a CCP c if $F(c)$ equals $F(mc)$ and $N_f(mc)$ equals $N_f(c)$ for any $f \in F(mc)$. All row instances of c form the table instance of c, denoted as $table\_instance(c)$.

In Fig. 1, let pattern $c = \{\blacktriangle^2, \blacksquare\}$, then the table instance is $table\_instance(c) = \{\{\blacktriangle_1, \blacktriangle_2, \blacksquare_1\}, \{\blacktriangle_3, \blacktriangle_4, \blacksquare_2\}, \{\blacktriangle_3, \blacktriangle_4, \blacksquare_3\}\}$.

The next issue is how to quantify the prevalence and stability of a CCP. We consider two basic principles. For a CCP c, the more instances of features in c that participate in its table instance, the higher the prevalence of c; the more row instances of c, the more stability it has.

Definition 3: *The coupling participation index* (*CPI*). The coupling participation index of a CCP c is based on the first of the above two basic principles, which is the same as the calculation of the participation index for traditional co-location pattern mining. That is:

$$CPI(c) = min_{f \in F(c)}\{{}^{|\pi_f(table\_instance(c))|}/_{TN_f}\}, \tag{1}$$

where π_f is a projection operation, and TN_f means the total number of instances of feature f.

Note that CPI has a basic property: in its table instance, the more overlapping instances (i.e., an instance appears in multiple row instances), the smaller the value of CPI. For example, in Fig. 1, $\{\blacktriangle_3, \blacktriangle_4, \blacksquare_2\}$ and $\{\blacktriangle_3, \blacktriangle_4, \blacksquare_3\}$ are both row instances of CCP $\{\blacktriangle^2, \blacksquare\}$. There would be four instances of feature $\blacktriangle$ participate in the two row instances if there are no overlapping instances; however, due to the overlapping instances $\blacktriangle_3$ and $\blacktriangle_4$, there are just two instances that participate in the two row instances.

Lemma 1. The CPI value of any CCP c is larger than 0 and smaller than or equal to 1, i.e., $0 < CPI(c) \leq 1$.

Proof: Obviously, $TN_f > 0$, and there is a maximal clique to be a candidate CCP, thus $|\pi_f(table\_instance(c))| > 0$. In addition, $|\pi_f(table\_instance(c))| \leq TN_f$, so $0 < CPI(c) \leq 1$.

Definition 4: *The coupling stability* (*CS*). The coupling stability of a CCP c is based on the second principle mentioned previously. Which is defined as follows:

$$CS(c) = {}^{|table\_instance(c)|}/_{|\Omega|}, \tag{2}$$

where Ω is the set of row instances of all *CCPs*, that is Ω is the set of all maximal cliques in the data set. For example, for $c = \{\blacktriangle^2, \blacksquare\}$ in Fig. 1, $|table\_instance(c)| = 3$, $|\Omega| = 8$, thus $CS(c)$=3/8.

For the coupling stability measure *CS*, we have the following lemma.

Lemma 2. The value range of CS is greater than 0 and less than or equal to 1, i.e., $0 < CS \leq 1$.

Proof: If c is a CCP, then c has at least one row instance, that is, $|table\_instance(c)| > 0$. In general, $|table\_instance(c)| < |\Omega|$ thus $0 < CS < 1$. If and only if all maximal cliques are row instances of c, then $|table\_instance(c)| = |\Omega|, CS = 1$. So, $0 < CS \leq 1$ holds.

In Fig. 1, for $c = \{\blacktriangle^2, \blacksquare\}$, there are three squares and all of them participate in the table instance of c. There are four triangles and three of them participate in the table instance of c, so $CPI(c) = min\{{}^3/_3, {}^3/_4\} = 0.75$. The total number of maximal cliques in Fig. 1 is eight, and three of them are row instances of c, so $CS(c) = 0.375$.

It is worth mentioning that CPI is calculated at the instances level, and CS is calculated at the maximal clique level. The projection operation in CPI considers the number of instances participating in the pattern and chooses the minimum of participation ratio of features in pattern to ensure the frequency. CS considers the number of occurrences of each CCP, and the bigger the number of occurrences, the more stable the CCP is.

Below, a new support measure that considers the participation index and stability simultaneously for a CCP is defined.

Definition 5: *The support (Sup)*. The support of a CCP *c* is defined as follows:

$$Sup(c) = \alpha * CPI(c) + (1 - \alpha) * CS(c), \tag{3}$$

where α is a weighting factor that balances stability and prevalence, and $0 \leq \alpha \leq 1$.

Lemma 3. The support measure does not satisfy anti-monotonicity.

Proof: From Fig. 1, assuming $\alpha = 0.5$, we can get $\mathrm{Sup}(\{\blacktriangle, \blacksquare, \bullet\}) = 0.75$, $\mathrm{Sup}(\{\blacktriangle^2, \bullet\}) = 0.625$, $\mathrm{Sup}(\{\blacktriangle, \blacksquare\}) = 0.625$, and $\mathrm{Sup}(\{\blacktriangle^2, \blacksquare\}) = 0.75$; therefore, at the feature level, we find that $\mathrm{Sup}(\{\blacktriangle, \blacksquare, \bullet\}) > \mathrm{Sup}(\{\blacktriangle, \blacksquare\})$, at the quantity level, $\mathrm{Sup}(\{\blacktriangle^2, \blacksquare\}) > \mathrm{Sup}(\{\blacktriangle, \blacksquare\})$.

Lemma 4. The value range of *Sup* is greater than 0 and less than or equal to 1, i.e., $0 < Sup \leq 1$.

Proof: According to Lemmas 2 and 3, we know that $0 < CPI \leq 1$ and $0 < CS \leq 1$. Thus, $0 < \alpha * PI \leq \alpha$, $0 < (1 - \alpha) * CS \leq (1 - \alpha)$, then $0 + 0 < \alpha * CPI + (1 - \alpha) * CS \leq \alpha + (1 - \alpha) = 1$.

Definition 6: *Strong coupling co-location pattern SCCP*. A coupling co-location pattern c is called a strong coupling co-location pattern if and only if $Sup(c) \geq min\_sup$, where min_sup is a support threshold specified by users.

3. Framework for mining SCCPs

As shown in Fig. 2, the SCCP mining framework [14] involves: (1) finding all the maximal cliques from a spatial data set; then (2) converting all the maximal cliques into candidate SCCPs; and finally (3) obtaining all SCCPs according to Definitions 2-6.

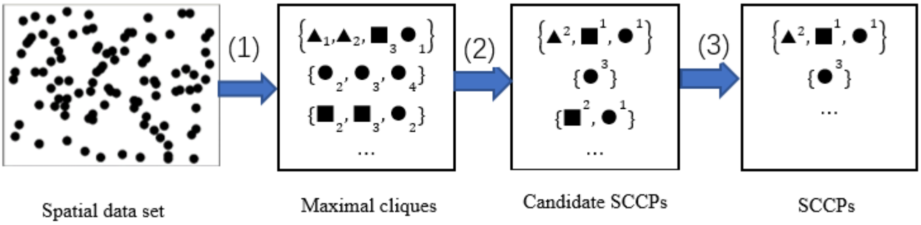

Fig. 2. A framework for SCCP mining

In the first step of SCCP mining, the aim is to enumerate all maximal cliques effectively for a given spatial instance set and a distance threshold. This issue has attracted the attention of some researchers, and we apply the latest work of C. Zhang et al [15] to resolve this issue. The main idea of the algorithm in [15] is turning the enumeration of maximal cliques into an enumeration of convex polygons. First, we assume that the instance set has been sorted based on the *x*-coordinate values. Second, we select an instance *i* in order, and enumerate all maximal constraint convex polygons in which *i* is the leftmost instance. Third, in order to improve the efficiency of the algorithm, pruning considering the geometric properties of the maximal cliques is implemented.

For a finite set of instances in a two-dimensional area, if the instances on the line segment with any two instances as the endpoints belong to the set, then the set is called a **convex set**, and a polygon formed by connecting the outermost instances in a convex set is called a **convex polygon**. The **convex hull** of a point (instance) set *S* is the minimum convex set containing the set *S*. The **maximal constraint convex polygon (MCCP)** is a convex polygon such that the distance of any two extreme instances satisfies the distance constraint and cannot be further expanded by adding one more instance without violating the distance constraint.

The basic operation of the algorithm is to find all convex hulls; the well-known Graham's algorithm [16] is briefly introduced in Algorithm 1 for this purpose. First, an anchor instance is selected from the instance set *I* (Line 1). Then, these instances are sorted according to the polar angle between the instance and *i.* Instances with the same polar angle are sorted by the distance between the instance and *i* (Line 2). Finally, the result array *Ret* is initialized (Lines 3-5) and every instance scanned to check whether it is a convex hull (Lines 6-8). The while loop removes the points found that are not vertices of the convex hull because when traversing the convex hull counter-clockwise we should turn left at each vertex. If the while loop finds that there is no left turn at a vertex, it removes the vertex. Otherwise, it is temporarily added to the result *Ret* (Line 8).

Algorithm 1: Graham's algorithm (*I*)

```
Input: A spatial instance set I
Output: All convex hulls
Select a rightmost or leftmost instance i as an anchor instance;
Sort these points according to the polar angle of each point relative to i, I={i1,i2,i3,…};
j=0;
Push i in Ret[ j++ ]
Push i1 in Ret[ j++ ];
For each instance i′ in I do
    while (multiply(i', Ret[j], Ret[j-1])>0) j--;
      Ret[++j]=i';
Return Ret
```

The pseudocode of the maximal cliques mining algorithm is given in Algorithm 2, and Algorithm 3 is a subprocess called in Algorithm 2. Algorithm 2 outlines the framework of the enumeration of the maximal cliques. In Line 1, the instances are sorted based on their *x*-coordinate values. For each instance i, setting three sets *M*, *C*, and *E*. *M* is the instances chosen for the current clique, and the initial value is i. *C* contains the candidate instances for the current clique. The initial value of set *C* is the neighbors that behind *i,* that is, the right-side neighbors of *i. E* is the excluded instances, which contains instances that are close to i and before i (Lines 3-4). Some related information concerning *C* is calculated for pruning strategies in Lines 5-6. Algorithm 1 is called to calculate the convex hull layer of instances. The subprocess Enum(Ø, *i*, C, E) is called to find all maximal cliques in which *i* is the leftmost instance.

Algorithm 2: Enumerate maximal spatial cliques (*I*, *d*)

```
Input: A spatial instance set I, and a distance threshold d
Output: All maximal spatial cliques
1.  Sort instances in I based on their x-coordinate value;
2.  For each instance i in I do
3.      Put right side neighbors of i in C
4.      Put left side neighbors of i in E
5.      Compute convex layers on C ∪ {i}
6.      Sort instances in C by their polar angles w. r. t i;
7.      Compute pivot convex polygons;
8.      Enum(Ø, i, C, E);
```

The time complexity is considered next, where n is the number of instances in I. Sorting (Lines 1, 6) takes $O(n * log_n)$ time. The convex layer calculation (Line 5) takes $O(v * n * log_n)$ time where v is the number of convex layers. The pivot convex polygon calculation takes $O(kn)$ time where k is the number of pivot convex polygons (Line 7).

Algorithm 3: Enum(M, v, C, E)

Input: Instances in the partial solution M, the instance v to be added to M, candidate set C, excluded set E.

Output: All MCCPs and maximal cliques

```
1.   Put v into M;
2.   If v is an outer instance put v into $P(M)$
3.   Else put v into $I(M)$
4.   Reduce the size of set C
5.   If $E = \emptyset$ and $C = \emptyset$ then
6.       $P(M)$ is a maximal constraint convex polygon;
7.       M is a maximal clique;
8.   $L = C$
9.   Reduce the size of L
10.  Sort instances in L by their convex layers;
11.  For each $v \in L$ do
12.      Enum(M, v, $C \cap NB(v)$, $E \cap NB(v)$)
13.      $E \leftarrow E \backslash \{v\}$
```

The Enum algorithm is described in Algorithm 3 and is the pseudocode of the enumeration of maximal cliques (represented by Maximal Constraint Convex Polygons, MCCPs) for each anchor point v. Line 1 adds a newly added instance v to M. Lines 2 and 3 determine whether v belongs to the constraint convex polygon set $P(M)$ or the internal instances set $I(M)$. Line 4 is a pruning technique such that if a candidate instance $c \in C$ is contained by the convex polygon P(M), we can move c to I(M). For Lines 5-7, if both E and C are empty, the instances in P(M) are instances of the MCCP and the instances in P(M) ∪ I(M) are the corresponding maximal clique. At Line 9, the algorithm reduces the size of set L; the main idea is that not all instances in C need to be considered, only instances in $C - M$ are checked. Line 10 sorts the instances in L by their convex layers to make sure that the outside instances will be accessed first. Lines 11-13 call the Enum process recursively by choosing an instance from L for the current partial solution M.

The time complexity of the reduction of the size of set C is $O(n)$ (Lines 1-4). The reduction of the size of L takes $O(kn)$ (Line 9) where k is the number of pivot convex layers.

After executing Algorithm 2 on the data set shown in Fig. 1, Table 1 lists all maximal cliques that are obtained. The advantage of this algorithm is that it specifically solves the problem of maximal clique enumeration in a two-dimensional space. Many pruning strategies in the algorithm are based on geometric properties in two-dimensional space. Therefore, this algorithm cannot be used when the proximity relationship is not Euclidean, which is a disadvantage.

Table 1. All maximal cliques of the data set in Fig. 1

{▲$_1$,▲$_2$,■$_1$}	{▲$_1$,▲$_2$,●$_1$}	{●$_1$,●$_2$,●$_3$,■$_2$}
{▲$_3$,■$_2$,●$_1$}	{▲$_3$,▲$_4$,■$_2$}	{▲$_3$,▲$_4$,■$_3$}
{▲$_4$,■$_3$,●$_4$}	{▲$_4$,■$_4$}	

In step (2) of the mining framework in Fig. 2, we extract candidate SCCPs from maximal cliques. Let MC be the set of all maximal cliques. The process of converting MCs to candidate SCCPs is given in Algorithm 4. For each $c \in MC$, all the features it contains is listed and the number of occurrences of each feature is recorded as a superscript. If the pattern has been recorded before, mc need added to the row instances set of the pattern; if not, the pattern needs added as a new item and put mc into the row instance set of the pattern.

Algorithm 4: Convert MCs to candidate SCCPs

Input: A maximal clique set MC

Output: A set of all candidate patterns and their row instances

```
For each $mc \in MC$:
    Convert $mc$ to candidate pattern $p$
    If $p$ has been recorded:
        Add $mc$ to its row instances set
    Else:
        Add a new pattern $p$, and put $mc$ in its row instances set
```

The time complexity of Algorithm 4 is related to the data structure used to record patterns. Suppose that there are $\acute{n}$ maximal cliques that require conversion; for each maximal clique, the conversion process has constant time complexity $O(1)$. Let $t(\overline{m})$ be the average time complexity for checking whether a pattern already exists, where m represents the total number of patterns, then the total time complexity is $O(\acute{n} * t(\overline{m}))$. If patterns are given an order, the time for each checking process becomes $O(log_2\overline{m})$, thus it becomes $O(\acute{n} * log_2\overline{m})$. If a hash table is used to record patterns, the time complexity of the checking process is $O(1)$, and thus, the total time complexity is $O(\acute{n})$. Suppose there are k candidate patterns in total, and the average memory cost for storing a pattern is x bytes, the space complexity will be $O(k * x)$.

Based on the result of Algorithm 4, we can calculate the support for each candidate pattern by Definitions 2-6 in step (3) of the mining framework in Fig. 2, and all SCCPs can be obtained.

4. Experimental Studies

We conducted experimentation with the SCCPs mining algorithm using both real and synthetic data sets on a computer with an Intel i7 3.2 GHz CPU and 16 GB RAM running Windows 10. All code is implemented using Visual Studio Code. The compiler is MingW. The real data set used contains the POIs (point of information) of the city of Beijing. We selected 20493 instances of 20 different features. With a neighboring relationship threshold of 100 meters, 5354 candidate patterns were obtained.

In the results there are many interesting patterns. We have selected four relatively highly supported and interesting patterns in Table 2. The pattern $\{hotel^2, clinic\}$ is interesting because we did not find the pattern $\{hotel, clinic\}$ in the results, which shows that the ratio of hotel and clinic clustered together is not 1:1 but 2:1. This pattern also proves the practical significance of the coupling phenomenon related to the quantity relationship. The pattern $\{hotel^2, bar\}$ is similar to the pattern $\{hotel^2, clinic\}$. The pattern $\{pharmacy^2\}$ is an intra-coupling pattern. We also find patterns such as $\{pharmacy^3\}$, $\{pharmacy^4\}$, and $\{pharmacy^5\}$. This means that pharmacies are stores that have a strong intra-coupling relationship and tend to group together. The pattern $\{hotel^3, pharmacy^2, cinema, laundry^2\}$ is a long pattern that can be used in city planning.

Table 2. Some interesting mined patterns

Pattern	CPI	CS	α=0.5
$\{hotel^2, clinic\}$	0.0238441	0.00252071	0.013182405
$\{pharmacy^2\}$	0.0359053	0.00438123	0.020143256
$\{hotel^2, \text{bar}\}$	0.0136668	0.0014404	0.0075536
$\{hotel^3, pharmacy^2, cinema, laundry^2\}$	0.000729661	0.000120034	0.000424848

We also use the same data set for mining traditional co-location patterns. After comparing the mining results of the two different algorithms, the following conclusions can be drawn.

(1) For traditional co-location pattern mining, there are patterns with the same feature combination as SCCP. For example, we find a size-2 pattern $\{hotel, clinic\}$ in the results for co-location pattern mining with a support threshold of 0.5. Fig. 3 shows the spatial distribution of instances of hotels and clinics, and Fig. 4 shows the instances of $\{hotel^2, clinic\}$ in SCCP. The difference is that Fig. 3 lists all instances of hotels and clinics, and Fig. 4 lists the instances forming maximal cliques of two hotels and one clinic. The differences of pattern definition and the differences of the way that candidate patterns are selected give us a different view of city planning.

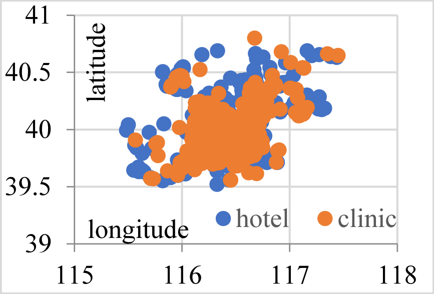

Fig. 3. Instance distribution.

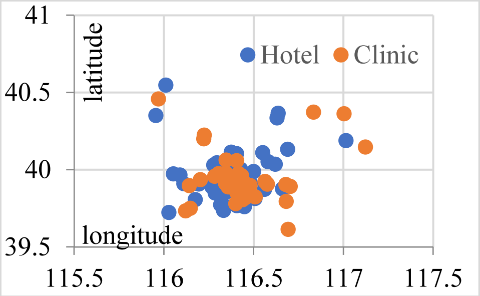

Fig. 4. Support instance of $\{hotel^2, clinic\}$.

(2) The *PI* in traditional co-location pattern mining is a cumulative value, because all instances with the same feature combination in different cliques will be grouped together. However, the *CPI* in SCCP is a non-cumulative value, because instances of the same feature combination are scattered into different maximal cliques. To look deeper into the support, we are interested in knowing how the *CPIs* have been dispersed. We consider two feature combinations $\{clinic, pharmacy\}$; all candidate SCCPs that contain this combination with different quantity combinations are shown in Fig. 5 (where C1P1 means the pattern that contains one clinic and one pharmacy). The y-axis is the number of instances of clinics and pharmacies in patterns that contain $\{clinic, pharmacy\}$. We can see that the support instances of a certain feature have been divided by different patterns.

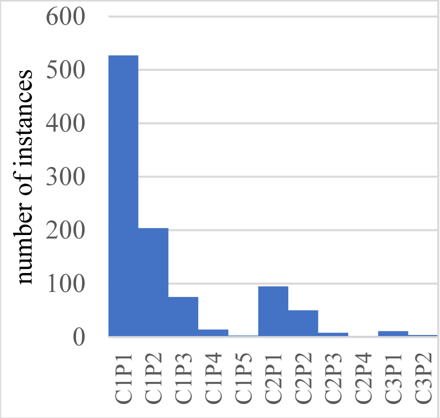

Fig. 5. Number of support instances.

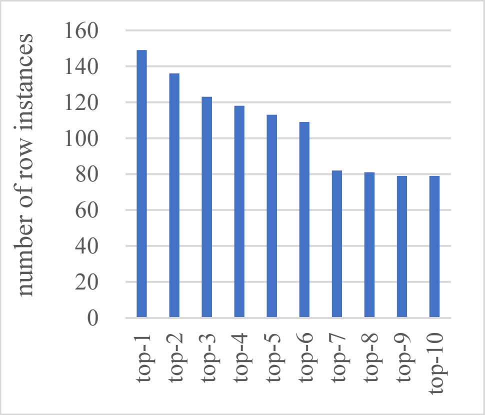

Fig. 6. Number of row instances.

(3) The CS value is a supplementary value that represents the relative frequency at which a pattern repeatedly appears. By Definition 3, the denominator is the number of all maximal cliques in the data set, and the numerator is the number of maximal cliques corresponding to a candidate pattern. Fig. 6 shows the top-10 CS values in the SCCP mining results. We can see that those values are much smaller than the total number of maximal cliques. When considering the frequency, if all values are small, a relatively large value corresponds to a relatively frequent pattern.

(4) The balance factor α can significantly influence the support value. Fig. 7 shows the changes of support value of the top-10 patterns with α. We can see that as

the value of α increases, the weight of CS decreases and the weight of *CPI* increases. When α = 0.3, the biggest support is the support of pattern 1; when α = 0.7, the biggest support is the support of pattern 2.

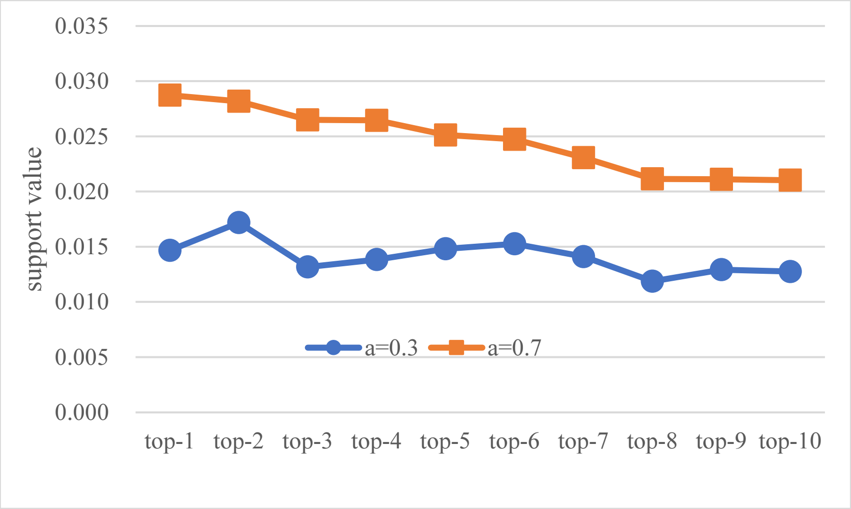

Fig. 7. Support value changes with α.

Table 3. Synthetic data sets

Data size	Distance threshold	Number of maximal cliques
5000	450	17332
12000	450	8860
20000	450	2784

In addition, we tested the maximal clique enumeration algorithm and Algorithm 4 with synthetic data sets. Table 3 shows the related information of the synthetic data sets. We mainly tested the runtime of the algorithm. Fig. 8 is the runtime of the maximal clique enumeration algorithm; the *x*-axis is the size of the data set, and the *y*-axis is the runtime (in seconds). Fig. 9 shows the runtime ratio of the two algorithms on the same data sets.

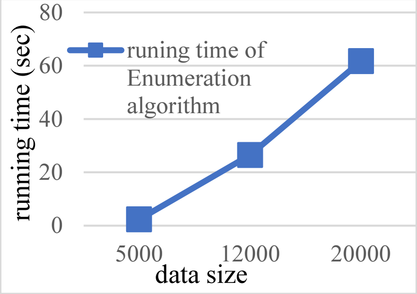

Fig. 8. Runtime.

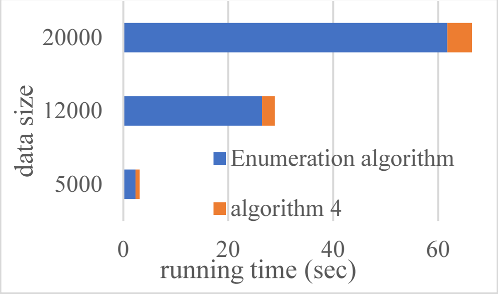

Fig. 9. The runtime ratio.

We can see that the enumeration algorithm used in the paper is a very efficient algorithm. The process of enumerating maximal cliques takes the most of the runtime, and the conversion process takes very little time. Also, experiments show that $\{hotel, clinic\}$ is a prevalent pattern and the new pattern tells us the ratio of hotels to clinics is 2:1, which reflects the significance of the proposed pattern.

5. Conclusion

In this paper, we studied the coupling co-location pattern mining problem which considers the coupling relationship (including intra-coupling and inter-coupling) between spatial features. A mining framework for discovering coupling co-location patterns has been developed. We conducted experiments on both real and synthetic data sets, which verified that the proposed coupling co-location pattern has practical meaning. The managerial implication of the new pattern is that it can augment co-location patterns in urban planning and other decision making. One limitation of the proposed pattern is that the support measure does not satisfy anti-monotonicity, which means that it cannot be pruned through the minimum support threshold like co-location patterns. In the future, we plan to design a new support measure approach for coupling co-location patterns and a new maximal clique mining algorithm to improve the efficiency of the mining process.

Acknowledgments

We would like to thank C. Zhang et al. for their outstanding work in maximal clique enumeration. This work is supported by the National Natural Science Foundation of China (61966036, 61662086) and the Project of Innovative Research Team of Yunnan Province (2018HC019).

References

[1] L. Wang, X. Bao, L. Zhou and H. Chen. Mining maximal sub-prevalent co-location patterns. *World Wide Web*, 2019, 22(5): 1971-1997

[2] S. Shekhar and Y. Huang. Discovering spatial co-location patterns: a summary of results. SSTD, 2001: 236-256

[3] Y. Morimoto. Mining frequent neighboring class sets in spatial databases. KDD, ACM, 2001: 353-358.

[4] Y. Huang, S. Shekhar and H. Xiong. Discovering co-location patterns from spatial data sets: a general approach. IEEE Trans. Knowl. Data Eng, 2004, 16(12): 1472-1385

[5] Y. Huang, S. Shekhar, H. Xiong and J. Pei. Mining confident co-location rules without a support threshold. In Proceedings, ACM, 2003: 497-501

[6] J. S. Yoo and S. Shashi. A join-less approach for mining spatial colocation patterns. IEEE TKDE, 2006, 18(10): 1323-1337

[7] J. S. Yoo, S. Shekhar, J. Smith, and J. P. Kumquat. A part join approach for mining co-location patterns. ACM, 2004: 241-249

[8] H. K. Chan, C. Long, D. Yan, and R. C. Wong. Fraction-score: A new support measure for co-location pattern mining. ICDE 2019, IEEE, 2019: 1514-1525.

[9] C. Wang, Z. She and L. Cao. Coupled Clustering Ensemble: Incorporating coupling relationships both between base clusterings and objects. ICDE 2013, IEEE, 2013: 374–385.

[10] P. Yang, L. Wang and X. Wang. A mapreduce approach for spatial co-location pattern mining via ordered-clique-growth. Distribution and Parallel Databases, 2019, 38: 531-560

[11] Y. Huang and P. Zhang. On the relationships between clustering and spatial co-location pattern mining. IEEE, 2006: 513-522.

[12] X. Bao and L. Wang. A clique-based approach for co-location pattern mining. Information Sciences, 2019: 244-264

[13] P. Wu, L. Wang and M. Zou. Vector-degree: A General similarity measure for co-location patterns. IEEE 2019, ICBK, 2019: 281-288

[14] H. Xiong, S, Shekhar, Y. Huang, V. Kumar, X. Ma and J. S. Yoo. A framework for discovering co-location patterns in data sets with extended spatial objects. SDM, 2004: 1-12.

[15] C. Zhang, Y. Zhang, W. Zhang et al. Efficient maximal spatial clique enumeration. ICDE 2019, IEEE, 2019: 878-889.

[16] F. P. Preparata and M. I. Shamos. Computational geometry: an introduction. Springer. 1985.