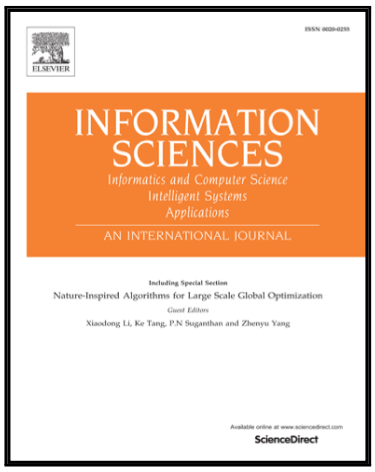



# Effective Lossless Condensed Representation and Discovery of Spatial Co-location Patterns

Lizhen Wang , Xuguang Bao , Hongmei Chen , Longbing Cao

PII: S0020-0255(18)30014-8
DOI: 10.1016/j.ins.2018.01.011
Reference: INS 13365

To appear in: *Information Sciences*

Received date: 24 April 2017
Revised date: 19 October 2017
Accepted date: 7 January 2018

Please cite this article as: Lizhen Wang , Xuguang Bao , Hongmei Chen , Longbing Cao , Effective Lossless Condensed Representation and Discovery of Spatial Co-location Patterns, *Information Sciences* (2018), doi: 10.1016/j.ins.2018.01.011

# Effective Lossless Condensed Representation and Discovery of Spatial Co-location Patterns

Lizhen Wang[a], Xuguang Bao[a], Hongmei Chen[a*], Longbing Cao[b]
[a]School of Information Science and Engineering, Yunnan University, Kunming, China
[b]Advanced Analytics Institute, University of Technology Sydney, Sydney, Australia

**ABSTRACT:** A spatial co-location pattern is a set of spatial features frequently co-occuring in nearby geographic spaces. Similar to *closed frequent itemset mining*, *closed co-location pattern (CCP) mining* was proposed for losslessly condensing large collections of prevalent co-location patterns. However, the state-of-the-art condensation methods in mining CCP are inspired by closed frequent itemset mining and do not consider the intrinsic characteristics of spatial co-locations, e.g., the participation index and ratio in spatial feature interactions, thus causing serious containment issues in CCP mining. In this paper, we propose a novel *lossless condensed representation* of prevalent co-location patterns, *Super Participation Index-closed* (*SPI-closed*) *co-location*. An efficient *SPI-closed Miner* is also proposed to effectively capture the nature of spatial co-location patterns, alongside the development of three additional pruning strategies to make the SPI-closed Miner efficient. This method captures richer feature interactions in spatial co-locations and solves the containment issues in existing CCP methods. A performance evaluation conducted on both synthetic and real-life data sets shows that SPI-closed Miner reduces the number of CCPs by up to 50%, and runs much faster than the baseline CCP mining algorithm described in the literature.

**Keywords:** Spatial data mining, Spatial co-location patterns, SPI-closed co-location patterns, Lossless condensed representation.

* Corresponding author.
E-mail addresses: lzhwang@ynu.edu.cn (L. Wang), bbaaooxx@163.com (X. Bao), hmchen@ynu.edu.cn (H. Chen), longbing.cao@uts.edu.au (L. Cao)

## 1. Introduction

Application areas such as earth science [20], public health [10], public transportation [2,35], environmental management [1], social media services [19,38], location services [7,28], multimedia [6,14,40-43], and so on produce large and rich spatial data. Potentially valuable knowledge is embedded in such data in various spatial features, and spatial co-location pattern mining has been proposed to identify interesting but hidden relationships between spatial features [18,8].

A spatial co-location pattern represents a set of spatial features that frequently co-occur in spatial proximity [18]. For example, West Nile Virus often appears in the regions with poor mosquito control and the presence of birds. Spatial co-location patterns yield important insight for various applications such as urban facility distribution analysis [35], e-commerce [39] and ecology [24]. A common framework of spatial co-location pattern mining uses the frequencies of a set of spatial features participating in a co-location to measure the prevalence (known as participation index [8], or PI for short) and requires a user-specified minimum PI threshold $M$ to find interesting co-location patterns. $M$ determines the level of prevalence of identified co-location patterns and $M$ may have to be small to avoid overlooking co-locations. However, a smaller $M$ might induce a larger number of co-locations with lower actionability [4,5,9], so it is difficult for a user to determine a suitable value for $M$.

Fig. 1(a) shows an example spatial data set. There are four different spatial features $F$ = {A, B, C, D} with each instance denoted by a feature type and a numeric ID value, e.g., A.1. Edges among the instances indicate spatial neighboring relationships. Feature A has four instances, B has five instances, C has three instances, and D has four instances in the data set. Fig. 1(b) lists all possible co-locations in $F$, and their *co-location instances* and their corresponding *PI* (the definitions of *co-location instances* and *PI* are provided in Section 3). If $M$ is given as 0.3, we can see that the prevalent co-locations of this data set are {{A, B, C, D}, {A, B, C}, {A, B, D}, {A, C, D}, {B, C, D}, {A, B}, {A, C}, {A, D}, {B, C}, {B, D}, {C, D}} since their PI values are greater than 0.3.

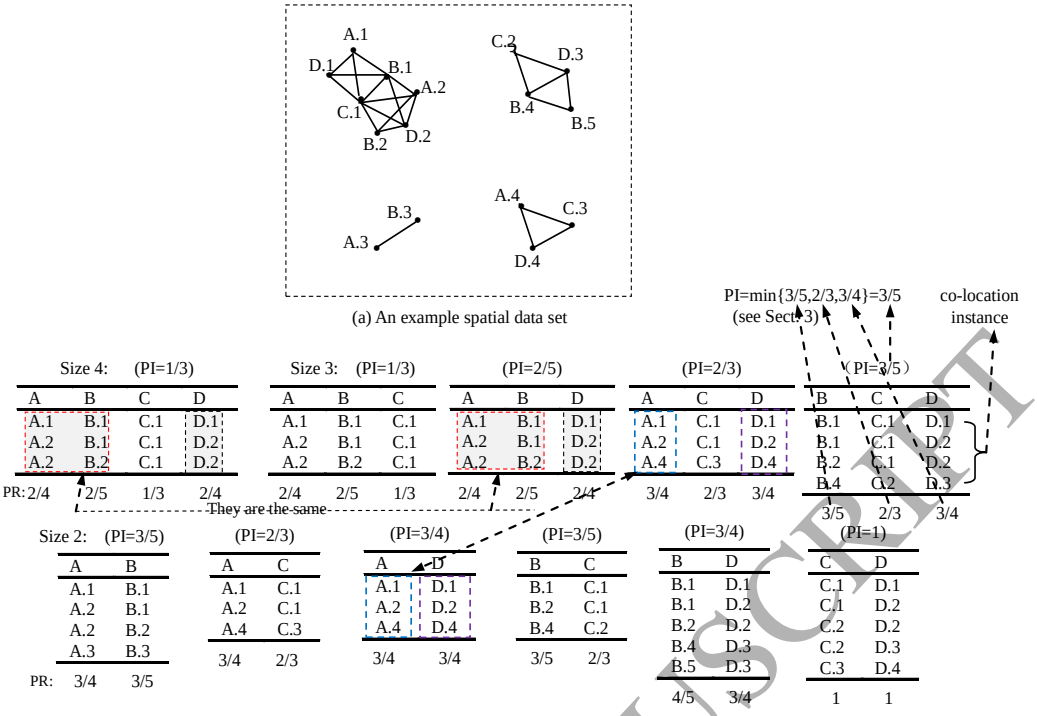



**Fig. 1.** A motivating example

Condensed representations describe small collections of co-locations such that it is possible to infer the original collection of prevalent co-locations. The concept of *maximal co-location patterns* [23,33] is based on a lossy condensed representation, which infers the original collection of interesting co-locations but not their PI values. The introduction of *closed co-location patterns (CCPs)* creates a lossless condensed representation [34], which can infer not only the original collection of prevalent co-locations but also their PI values. A prevalent co-location pattern $c$ is *closed* if there exists no proper prevalent co-location pattern $c'$ such that $c \subset c'$ and PI($c$) = PI($c'$) [34]. This concept is similar to the *closed frequent itemsets* in transactional data sets [16,21,37]. For example, the CCPs of the data set in Fig. 1(a) are {A, B, C, D}, {A, B, D}, {A, C, D}, {B, C, D}, {A, B}, {A, D}, {B, D} and {C, D} because PI({A, B, C, D }) = PI({A, B, C}), PI({A, C, D }) = PI({A, C}) and PI({B, C, D }) = PI({B, C}).

Nevertheless, the above methods can cause serious confusion in mining CCPs as illustrated by the problem in Fig. 1(b). The co-location instance measuring the PI value of pattern {A, B, D} is contained by the co-location instance of its super-pattern

{A, B, C, D} (as shown in the dotted boxes in the co-location instances of {A, B, C, D} and {A, B, D}). The same situation occurs in co-location instances {A, C, D} and {A, D}. This means that the PI values of {A, B, D} and {A, D} can be inferred from their super-pattern {A, B, C, D} and {A, C, D} respectively. However, PI({A, B, D})≠PI({A, B, C, D}) and PI({A, D}) ≠ PI({A, C, D}), so {A, B, C, D}, {A, B, D}, {A, C, D}, and {A, D} are all in the set of CCPs.

The above confusion is essentially caused by the direct projection of the concept of *closed frequent itemsets* in transactional data to CCP mining, where the CCP mining utilizes lossless condensed representation of the prevalent co-location patterns. However, spatial feature interactions are different from the feature relations in transactional data, and spatial co-location pattern mining is different from frequent itemset mining. CCP involves not only a *participation index* to measure the prevalence of co-locations but also a *participation ratio* (which will be defined in Section 3) to measure the prevalence of spatial features in co-locations, which differs from the support/confidence measures of frequent itemset mining.

To address the intrinsic characteristics of spatial feature interactions and CCP mining in spatial co-locations, we first define the concept of a *super participation index* SPI($c|c'$), which is the participation index of the co-location pattern $c$ in its super-pattern $c'$. The concept of *SPI-closed* is then introduced to more effectively and losslessly condense the collections of prevalent co-location patterns. We show that the *SPI-covered* relationship (see Definition 7) is a pseudo partial order in the prevalent co-location set. A theoretical analysis is provided which shows that SPI-closed reduces the number of closed co-locations by up to 50% compared to existing methods when the number of spatial features is sufficiently large. Lastly, an efficient *SPI-closed Miner* is proposed to mine SPI-closed co-locations, and three prunning strategies are developed to make the SPI-closed Miner efficient.

Our experimental evaluation shows that SPI-closed Miner runs much faster than the closed prevalent co-location pattern mining algorithm in [34], which is a very fast closed co-location mining method and also the only one available in the literature.

This is because our method captures richer information in spatial feature interactions and spatial co-locations.

The rest of the paper is organized as follows. Related work is discussed in Section 2. In Section 3, we review definitions related to classic co-location pattern mining, and define the concept and properties of *SPI-closed*. Section 4 presents our *SPI-closed Miner* and Section 5 carries out the qualitative analysis of the miner. The experimental results are presented in Section 6. Section 7 concludes the paper.

## 2. Related work

Spatial co-location pattern mining was first discussed in [18], in which the authors formulated the co-location pattern mining problem and developed a co-location mining algorithm. An extended version of the work in [18] was presented in [8]. The authors in [39] enhanced the co-location pattern in [18] and proposed an approach to find spatial star, clique and generic patterns. Approaches to reduce expensive join operations used for finding co-location instances in [18,8] were proposed in [32,22,29]. The work in [23,33] studied the problem of maximal co-location pattern mining. The problem of co-location pattern mining with spatially uncertain data sets was presented in [24,11,25,15]. The spatial instance distribution information was integrated into prevalence metrics in [17]. The incremental mining and competitive pairs mining of co-location patterns were studied in [12,13]. The concept and mining methods of spatial high utility co-location patterns were presented in [26,30]. Prevalent co-location redundancy reduction problem was studied in [27]. Considering the spatial auto-correlation property in co-location pattern detection approaches, the authors of [3] studied the problem of discovering statistically significant co-location patterns, and [31] studied the problem of co-location pattern mining considering detailed relationships of instances in a continuous space.

Limited work is available on closed co-location pattern mining. By contrast, many algorithms were proposed for finding closed frequent itemsets, such as CLOSET [16], CLOSET+ [21], CHARM [37]. However, to the best of our

knowledge, only the top-*k* closed co-location pattern mining method presented in [34] exists, which identifies closed co-location patterns as focused on in this paper. Although their algorithm is very efficient for mining closed co-locations, it was essentially built on extending the concept of closed frequent itemsets and so leads to a lossy condensed representation of spatial co-locations by the proposed top-*k PI-closed* co-location mining. The algorithms for mining closed frequent itemsets cannot be directly used to mine closed co-location patterns because, in contrast to closed frequent itemset mining in transactional data, spatial objects are embedded in a continuous space without transactional information. Our proposed *SPI-closed* co-location miner captures spatial feature interactions and co-locations by introducing a lossless condensed representation and an efficient discovery process.

## 3. The concept of SPI-closed and its properties

We first review definitions related to classic co-location pattern mining, then define the concept of *super participation index-closed* (*SPI-closed*) and analyze its properties.

### 3.1 Co-location pattern mining

A spatial feature $f_i$ represents a specific aspect of information in a spatial region. For example, the species of a plant in a geographical area is a feature. An occurrence of $f_i$ at a location is called an *instance* of $f_i$. For example, a plant of certain species is an instance. The *spatial neighbor relationship NR* describes the relationships between spatial feature instances. Two spatial feature instances $i_1$ and $i_2$ form an *NR*, denoted as $NR(i_1, i_2)$, if $distance(i_1, i_2) \leq d$, where $d$ is a distance threshold to determine how close the neighbors are.

Given a spatial feature set $F$, a *spatial co-location pattern* $c$ is a subset of the feature set $F$. The number of features in $c$ is called the size of $c$. For example, if $F$= {A, B, C}, {A, B} co-occurs more than a threshold $M$, then it is a prevalent co-location with size 2. The spatial prevalent co-location pattern (PCP) mining

problem can be formulated as follows.

**Definition 1**. (**PCP Mining**) Given a co-location pattern set *CP* and a *quality predicate* (or a *prevalence predicate*) $q$ ($q \rightarrow \{0, 1\}$, where 0 refers to non-prevalent, and 1 indicates prevalent) that measures the quality of *CP*, PCP mining discovers all prevalent co-location patterns $\{c_1, c_2, \ldots c_n\}$, where a co-location pattern $c_i \in CP$, i.e., $q(c_i)=1$.

In practice, to make the predicate $q$ more flexible and applicable for real-life applications, $q$ is usually defined by a quality measure $\partial$: $CP \rightarrow [0, 1]$ in terms of a domain-specific threshold value $M \in [0, 1]$:

$$q(c) = \begin{cases} 1 & if\ \partial(c) \geq M \\ 0 & otherwise \end{cases} \quad [1]$$

The *minimum participation ratio* (*PR*) (called *participation index* (*PI*)) was a frequently used quality measure, as in [8, 22-25, 29-35]. Before defining PR and PI, we define the concepts of *row instance* and *co-location instance*. If there is a set of instances $I=\{i_1, i_2, \ldots, i_m\}$ such that $\{NR(i_j, i_k) \mid 1 \leq j \leq m, 1 \leq k \leq m\}$, then $I$ is called an *NR clique***.** If an *NR* clique $I$ contains all the features in a co-location pattern $c$, but there is not a proper subset of $I$ containing all the features in $c$, then $I$ is a *row instance* of $c$. The set of all row instances of $c$ is called the *co-location instance* of $c$, denoted as $T(c)$.

**Definition 2.** (**Participation ratio**) The *participation ratio* $\mathrm{PR}(c, f_i)$ of feature $f_i$ in a co-location $c$ is the fraction of instances of $f_i$ that occur in $T(c)$, i.e.,

$$\mathrm{PR}(c, f_i) = \frac{Number\ of\ distinct\ instances\ of\ f_i\ in\ T(c)}{Number\ of\ instances\ of\ f_i} \quad [2]$$

**Definition 3.** (**Participation index**) The *participation index* $\mathrm{PI}(c)$ of a co-location pattern $c$ is the minimum participation ratio $\mathrm{PR}(c, f_i)$ among all features $\{f_i\}$ in $c$, *i.e.*,

$$\mathrm{PI}(c) = \min_{f_i \in c}\{\mathrm{PR}(c, f_i)\} \quad [3]$$

A co-location $c$ is considered *prevalent* if $\mathrm{PI}(c) \geq M$, where $M$ is a user-specified

threshold.

For example, for the pattern $c$ = {A, B, C, D} in Fig. 1, the co-location instance $T(c)$={{A.1, B.1, C.1, D.1}, {A.2, B.1, C.1, D.2}, {A.2, B.2, C.1, D.2}}. Feature A has a participation ratio PR($c$, A) of 2/4 since only A.1 and A.2 among the four instances participate in its co-location instance, and PR($c$, B), PR($c$, C) and PR($c$, D) are 2/5, 1/3 and 2/4, respectively. The participation index of $c$ is the minimum of these four participation ratios, i.e. PI($c$) = 1/3.

The PR measures the prevalence strength of a feature in a co-location pattern, and the PI measures the prevalence strength of a co-location pattern. Wherever a feature in a co-location pattern $c$ is observed, all other features in $c$ can be observed in the feature's neighborhood with a probability of PI($c$) $\geq M$. The PI and PR measures satisfy the *anti-monotonicity* property (a *downward closure property*, i.e., PI($c$) $\geq$ PI($c'$) for any $c \subset c'$), which enables level-wise search (like Apriori) [8]. This kind of search method has good performance when a given threshold $M$ is high and the neighborhood relations of spatial data are sparse. However, an Apriori-based co-location discovery algorithm examines all of the $2^k$ subsets of each size $k$ feature set and generates numerous irrelevant patterns.

Lossless condensed representations are the diminished descriptions of the prevalent co-location collections such that it is possible to infer the original collection of prevalent co-locations and their PI values by inference methods. The introduction of *closed co-location patterns* (CCP) creates a lossless condensed representation [34], which can not only infer the original collection of prevalent co-locations but their PI values as well. However, the condensation power of CCP mining methods is quite limited.

### 3.2 A new lossless condensed representation method

By analyzing the properties of the PR and PI measures of spatial co-location patterns, such as their anti-monotonicity, we introduce a new lossless condensed representation method of prevalent co-locations, i.e., *super participation index closed*

(*SPI-closed*) *co-location*, which effectively improves the condensation power of mining CCP. The related definitions and lemmas are as follows.

**Definition 4.** (**The super participation index SPI($c|c'$)**) Let $c$ and $c'$ be two co-locations, and $c \subset c'$. The super participation index SPI($c|c'$) of $c$ in super-pattern $c'$ is defined as the minimum PR($c'$, $f_i$) among all features $\{f_i\}$ in $c$, *i.e.*,

$$\mathrm{SPI}(c|c') = min\{\mathrm{PR}(c', f_i), f_i \in c\} \quad [4]$$

**Example 1**. In the data set of Fig. 1(a), SPI({A, C, D}|{A, B, C, D})=min{PR({A, B, C, D}, A)=2/4, PR({A, B, C, D}, C)=1/3, PR({A, B, C, D}, D)=2/4}=1/3. Similarly, SPI({A, B, D}|{A, B, C, D})=2/5.

**Definition 5. (SPI-closed co-location)** A co-location $c$ is *SPI-closed* if and only if its PI value is greater than the SPI value of $c$ in any of its super-patterns $c'$ which are *SPI-closed*, i.e., if and only if

$$c \subset c' \text{ and } c' \text{ is } SPI\text{-}closed \rightarrow \mathrm{PI}(c) > \mathrm{SPI}(c|c') \quad [5]$$

The *SPI-closed* definition is recursive in the presented form. This is to ensure that an *SPI-closed* co-location set not only can infer the original collection of prevalent co-locations but their PI values as well. Accordingly, the discovery of *SPI-closed* co-locations has to progress from the largest size to size 2.

Recall the traditional concept of *closed* co-location defined in [34]: A co-location $c$ is *closed* if and only if its PI value is greater than the PI value of any of its super-patterns $c'$, i.e., if and only if

$$c \subset c' \rightarrow \mathrm{PI}(c) > \mathrm{PI}(c') \quad [6]$$

In the remainder of this paper, we call such closed co-locations *PI-closed co-locations* to distinguish the classic closed co-locations from our defined closed co-locations.

**Example 2.** Taking the example data set in Fig. 1(a), if $M$=0.3, {A, B, C, D} is an *SPI-closed* co-location. Since PI({A, B, C})=SPI({A, B, C}|{A, B, C, D})=PI({A, B, C, D}) and PI({A, B, D})=SPI({A, B, D}|{A, B, C, D})>PI({A, B, C, D}), {A, B,

C} and {A, B, D} are both *non-SPI-closed* co-locations, but {A, B, D} is *PI-closed*.

**Definition 6**. **(SPI-closed prevalent co-location)** An *SPI-closed* co-location $c$ is an *SPI-closed* prevalent co-location if $c$ is *SPI-closed* and $PI(c) \geq M$, where $M$ is a user-specified threshold.

For simplicity, we use *SPI-closed* co-locations to represent *SPI-closed* prevalent co-location patterns.

**Definition 7**. **(SPI-covered (or PI-covered))** For a co-location $c$, if there is a co-location $c'$ such that $c \subset c'$ and $PI(c) = SPI(c|c')$ ($PI(c) = PI(c')$), we say $c$ is *SPI-covered* (or *PI-covered*) by $c'$.

**Lemma 1.** If $c \subset c'$ and $c$ is *PI-covered* by $c'$, then $c$ must be *SPI-covered* by $c'$.

***Proof.*** If $c'$ PI-covers $c$, then $PI(c) = PI(c')$. According to the anti-monotonicity of PR and PI, $PI(c) \geq SPI(c|c') \geq PI(c')$. Therefore, $PI(c) = SPI(c|c')$ holds, i.e., $c'$ SPI-covers $c$. □

**Lemma 2.** The *SPI-covered* relationship is a pseudo partial order in the prevalent co-location set, such that:

(1) $c$ is *SPI-covered* by $c$. (reflexivity)

(2) if $c$ is *SPI-covered* by $c'$ and $c'$ is *SPI-covered* by $c$, then $c = c'$. (anti-symmetry)

(3) if $c \subset c' \subset c''$, $PI(c) = PI(c')$ and $c'$ is *SPI-covered* by $c''$, then $c$ must be *SPI-covered* by $c''$. (pseudo-transitivity)

***Proof.*** By the concept of *SPI-covered*, it is easy to verify that the first two properties are true. We prove the third statement below.

According to the conditions of the third statement and related definitions, if *$c$ is not SPI-covered by $c''$*, then

$$PI(c) > min\{PR(c'', f_i), f_i \in c\}$$
$$\geq min\{PR(c'', f_i), f_i \in c'\} \qquad (c \subset c')$$

$= min\{\mathrm{PR}(c', f_i), f_i \in c'\}$ (by $c'$ is *SPI-covered* by $c''$ )

$= \mathrm{PI}(c')$ (by the PI definition)

$\Rightarrow$ $\mathrm{PI}(c) > \mathrm{PI}(c')$, contradiction.

Hence, $c$ must be *SPI-covered* by $c''$. □

We note that the *PI-covered* relationship satisfies transitivity, but the *SPI-covered* relationship does not. That is why the condition "$c'$ is *SPI-closed*" is put into Definition 5. Otherwise, the main point of the closure, which is able to deduce the prevalence of deleted patterns by looking at the remaining patterns, will be lost. Furthermore, the process of discovering *SPI-closed* co-locations has to progress from the largest size down to size 2. Our proposed new concept *SPI-closed* outperforms the traditional concept of *PI-closed* defined in [34], i.e., the set of *SPI-closed* co-locations $S_{SPI\text{-}closed}$ is smaller than the set of *PI-closed* co-locations $S_{PI\text{-}closed}$.

**Lemma 3.** If $c \in S_{SPI\text{-}closed}$, then $c \in S_{PI\text{-}closed}$. However, $c \in S_{SPI\text{-}closed}$ might not hold when $c \in S_{PI\text{-}closed}$.

*Proof.* If $c \in S_{SPI\text{-}closed}$, for any of $c$'s super-patterns $c'$ which are in $S_{SPI\text{-}closed}$, we have $\mathrm{PI}(c) > \mathrm{SPI}(c|c')$. According to the anti-monotonicity of PR and PI, $\mathrm{SPI}(c|c') \geq \mathrm{PI}(c')$. Therefore, $\mathrm{PI}(c) > \mathrm{PI}(c')$. For any of $c$'s super-patterns $c''$ which come from $S_{SPI\text{-}closed}$, if $\mathrm{PI}(c) = \mathrm{PI}(c'')$ then there exists $c'''$'s super-pattern $c'''$ which is in $S_{SPI\text{-}closed}$, $c$ is *SPI-covered* by $c'''$ (by Lemma 2.(3)), we thereupon can infer $c \notin S_{SPI\text{-}closed}$ (contradiction). Thus, $\mathrm{PI}(c) > \mathrm{PI}(c'')$. In summary, $c \in S_{PI\text{-}closed}$.

Conversely, we give a counter example: {A, B, D} is a *PI-closed* co-location in the data set of Fig. 1(a), i.e., $\{A, B, D\} \in S_{PI\text{-}closed}$, but it is not *SPI-closed*, i.e., $\{A, B, D\} \notin S_{SPI\text{-}closed}$. □

Accordingly, can we estimate how many fewer co-locations are in $S_{SPI\text{-}closed}$ than in $S_{PI\text{-}closed}$?

For a $k$-size co-location $c$, if $\mathrm{PI}(c) = \min\{\mathrm{PR}(c, f_i), f_i \in c\} = \mathrm{PR}(c, f_s)$, there are $k$-1 ($k$-1)-size co-locations containing feature $f_s$ but only one ($k$-1)-size co-location that

does not contain it. We call co-locations that do not contain the minimum PR feature of their super-patterns *should-be-closed* co-locations since the probability that they are *PI-closed* is generally higher. Other co-locations are called *might-be-closed* co-locations. For the data set in Fig. 1(a) where PI({A, B, C, D}) = PR({A, B, C, D}, C), C is the minimum PR feature of a 4-size co-location {A, B, C, D}. Thus 3-size co-locations {A, B, C}, {A, C, D} and {B, C, D} containing C are *might-be-closed*, and {A, B, D}, which does not contain C**,** is a 3-size *should-be-closed* co-location.

Some of the *should-be-closed* co-locations might be non-SPI-closed. We denote them as *not-SPI-closed*. If the fraction of *should-be-closed* co-locations in *not-SPI-closed* co-locations (denoted as *FR*(*should-be-closed*, *not-SPI-closed*)) is almost equal to *FR*(*should-be-closed* + *might-be-closed*, *not-PI-closed*), we have the following lemma.

Lemma **4.** If $|F| = n$ and *FR*(*should-be-closed*, *not-SPI-closed*) ≈ *FR* (*should-be-closed* + *might-be-closed*, *not-PI-closed*), then $(|S_{PI\text{-}closed}| - |S_{SPI\text{-}closed}|) / |S_{PI\text{-}closed}| \approx (2^{n-1} - n) / (2^n - (n + 1))$.

*Proof.* ① If $|F| = n$, there are $2^n - (n + 1)$ possible co-locations in *F*. For example, if $n$=4, there are 11 possible co-locations (*should-be-closed* + *might-be-closed*) in $F$ = {A, B, C, D}.

② There are at most $2^{n-1}$-$n$ *should-be-closed* co-locations in $2^n - (n + 1)$ possible co-locations. For example, if $F$ = {A, B, C, D}, there is one 3-size *should-be-closed* co-location and three 2-size *should-be-closed* co-locations in 11 possible co-locations (see Fig. 2).

Combining ① with ② under the condition that *FR*(*should-be-closed, not-SPI-closed*)≈*FR*(*should-be-closed+ might-be-closed, not-PI-closed*), we have $(|S_{PI\text{-}closed}| - |S_{SPI\text{-}closed}|) / |S_{PI\text{-}closed}| \approx (2^{n-1} - n) / (2^n - (n + 1))$. □

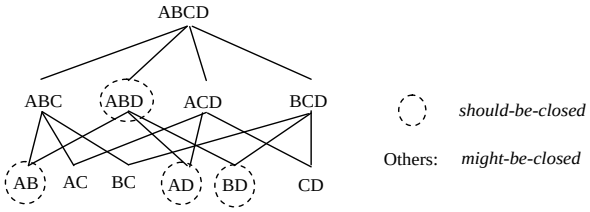



**Fig. 2.** The *Should-be-closed* and *might-be-closed* patterns of the data set in Fig. 1(a)

In general, *FR*(*should-be-closed, not-SPI-closed*) ≤ *FR* (*should-be-closed* + *might-be-closed, not-PI-closed*), hence, $(|S_{PI\text{-}closed}| - |S_{SPI\text{-}closed}|) / |S_{PI\text{-}closed}| \leq (2^{n-1} - n) / (2^n - (n + 1))$ holds.

For example, in Fig. 1(a), $|F| = 4$, accordingly, $(2^{n-1} - n) / (2^n - (n + 1)) = 4 / 11 = 0.36$. If $M = 0.3$, $(|S_{PI\text{-}closed}| - |S_{SPI\text{-}closed}|) / |S_{PI\text{-}closed}| = (8 – 6) / 8 = 0.25$ (see Fig. 1(b)).

We note that $(2^{n-1} - n) / (2^n - (n + 1)) \approx 1/2$ when $n$ is large enough. It means that the introduction of *SPI-closed* can reduce the number of *PI-closed* co-locations by about 50% under the conditions that *FR*(*should-be-closed, not-SPI-closed*) ≈ *FR(should-be-closed + might-be-closed, not-PI-closed*) and the number of spatial features $n$ is large enough.

Distinct from the discovery of frequent itemsets in transactional databases, we can obtain not only PI($c$) for a spatial co-location pattern $c$, but {PR($c$, $f_i$), $f_i \in c$} as well. However, the concept of the traditional *PI-closed* co-locations in [34] only applies to the PI values. The proposed concept of *SPI-closed* co-locations in Definitions 5 and 6 includes the information from both PI and PR, hence the power of the corresponding lossless condensation approach to prevalent co-location collections is effectively strengthened.

## 4. SPI-closed Miner

Based on Lemmas 1, 2 and 3, a direct approach to discovering all SPI-closed co-locations involves the identification of all PI-closed co-locations and then the pruning of non-SPI-closed co-locations. This approach has to compute the PI-closed co-location set first, which is larger than the SPI-closed co-location set. We introduce

SPI-closed Miner, which adopts an FP-growth-like method for directly mining SPI-closed co-locations, and we then develop three pruning strategies to make SPI-closed Miner efficient.

### 4.1 Preprocessing and candidate generation

To generate the smallest possible candidate set of SPI-closed co-locations, the input spatial data set will undergo the following preprocessing: converting the input data to neighborhood transactions, and then extracting features in the converted neighborhood transactions. We explain these stages below.

**1) Converting the input data to neighborhood transactions**

Given a spatial instance $f.i \in S$, the *neighborhood transaction* of $f.i$ is defined as a set that consists of $f.i$ and all other spatial instances having neighborhood relationships with $f.i$, i.e., $NT(f.i) = \{f.i, g.j \in S \mid NR(f.i, g.j) = true \text{ and } f \neq g\}$, where $NR$ is a neighborhood relationship.

For example, the neighborhood transaction of A.1 in Fig. 1(a) is {A.1, B.1, C.1, D.1}. Fig. 3(a) shows the neighborhood transactions of the data in Fig. 1(a). Each instance in the transaction has a neighborhood relationship with the first instance, which is called a *reference* instance.

| Trans. No. | Neighborhood instances | |
|---|---|---|
| 1 | A.1 | B.1,C.1,D.1 |
| 2 | A.2 | B.1,B.2,C.1,D.2 |
| 3 | A.3 | B.3 |
| 4 | A.4 | C.3,D.4 |
| 5 | B.1 | A.1,A.2,C.1,D.1,D.2 |
| 6 | B.2 | A.2,C.1,D.2 |
| 7 | B.3 | A.3 |
| 8 | B.4 | C.2,D.3 |
| 9 | B.5 | D.3 |
| 10 | C.1 | A.1,A.2,B.1,B.2,D.1,D2 |
| 11 | C.2 | B.4,D.3 |
| 12 | C.3 | A.4,D.4 |
| 13 | D.1 | A.1,B.1,C.1 |
| 14 | D.2 | A.2,B.1,B.2,C.1 |
| 15 | D.3 | B.4,B.5,C.2 |
| 16 | D.4 | A.4,C.3 |

**(a) Neighborhood transactions**

extract →

| Trans. No. | Neighborhood features | |
|---|---|---|
| 1 | A | B,C,D |
| 2 | A | B,C,D |
| 3 | A | B |
| 4 | A | C,D |
| 5 | B | A,C,D |
| 6 | B | A,C,D |
| 7 | B | A |
| 8 | B | C,D |
| 9 | B | D |
| 10 | C | A,B,D |
| 11 | C | B,D |
| 12 | C | A,D |
| 13 | D | A,B,C |
| 14 | D | A,B,C |
| 15 | D | B,C |
| 16 | D | A,C |

**(b) Neighborhood transaction features**

**Fig. 3.** Neighborhood transactions and neighborhood transaction features of the data set in Fig. 1(a)

The data structure for storing the neighborhood information was first introduced in [32]. It has several advantages for SPI-closed co-location mining. First, the

neighborhood transactions do not lose any instances, nor any neighborhood relationships of the original data. Second, neighborhood transactions can be easily constructed from the paired neighboring instances in the input data. Third, they can be used to filter the SPI-closed candidate set.

**2) Extracting features in neighborhood transactions**

The lexicographic set of distinct features in the neighborhood transactions is called *neighborhood transaction features*. For example, Fig. 3(b) shows the neighborhood transaction features of neighborhood transactions in Fig. 3(a).

We generate the feature sets for SPI-closed candidates based on these neighborhood transaction features. For the convenience of generating and pruning the SPI-closed candidates, we use a lexicographic prefix-tree structure to store the neighborhood transaction features. This kind of data structure was also used in [34] to generate top-*k* PI-closed co-location candidates. In this paper, we revise the process in [34] employed to generate SPI-closed candidates. Our process for generating and pruning the SPI-closed candidates is described blow.

First, the lexicographic prefix-tree structure is described as follows:

a) It consists of a root labeled as a *reference feature*, and a set of feature neighborhood relationships as the children of the root.

b) Each node consists of three fields: feature-type, count, and node-link, where feature-type denotes a feature this node represents, count registers the number of neighborhood transaction features represented by the portion of the path reaching this node, and node-link links to the next node in the tree carrying the same feature-type.

For example, the lexicographic prefix-tree constructed from the neighborhood transaction features in Fig. 3(b) is shown in Fig. 4(a). We can see that one prefix-tree is built for each reference feature.

Second, all feature sets having a neighborhood relationship with the root node (reference feature) are generated. We call the generated feature sets *star SPI-closed*

*candidates* since all features in a set have neighborhood relationships with their first feature. The output also includes the prevalence information, which indicates the likelihood of its first feature having a neighborhood relationship with all other features in the set. This represents the *upper bound of the participation ratio* (upper PR, or UPR for short). If the UPR of a star SPI-closed candidate is equal to the UPR of any super star SPI-closed candidate in the same reference feature set, we mark it in ***boldface*.** If the UPR of a star SPI-closed candidate is smaller than *M*, we delete it from the star SPI-closed candidates.

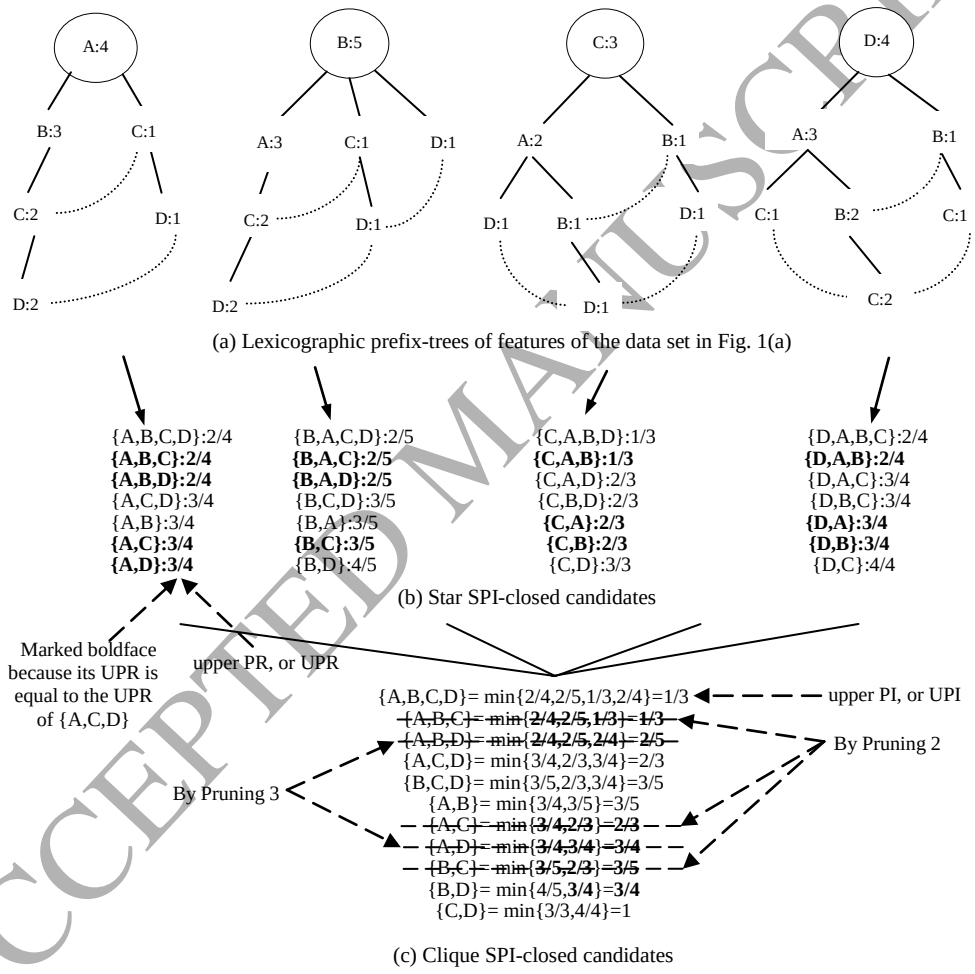



**Fig. 4.** Candidate generation

For example, in the prefix-tree of feature A of Fig. 4(a), we generate all feature sets having a relationship with A and their UPR values: {A, B, C, D}: 2/4, {A, B, C}: 2/4, {A, B, D}: 2/4, {A, C, D}: 3/4, {A, B}: 3/4, {A, C}: 3/4, {A, D}: 3/4. If $M$ = 0.3, no star SPI-closed candidates can be pruned. The feature sets marked in boldface are

**{A, B, C}: 2/4, {A, B, D}: 2/4, {A, C}: 3/4** and **{A, D}: 3/4**. The star SPI-closed candidates generated by prefix-trees in Fig. 4(a) are shown in Fig. 4(b) (assuming $M$ = 0.3).

Third, the star SPI-closed candidates are combined to filter *clique SPI-closed candidates*. This is done as follows. First, $k$ star SPI-closed candidates are combined to a $k$-size clique SPI-closed candidate, and the minimum value of $k$ UPR values forms the *upper participation index* (upper PI, or UPI) of this clique SPI-closed candidate. The super participation index SPI($c|c'$) ($c \subset c'$), calculated based on UPRs, is called *upper SPI*($c|c'$) (or *USPI*($c|c'$)).

Three pruning strategies are incorporated into the above combination process: non-prevalent pruning and two strategies for non-SPI-closed pruning.

***Pruning 1*** (Non-prevalent pruning): If a co-location $c$ is not the star SPI-closed candidate of a certain prefix-tree $f_i$ ($f_i \in c$), $c$ can be pruned.

*Proof*. For a spatial feature $f_i \in c$, if $c$ is not in the set of star SPI-closed candidates of the prefix-tree $f_i$, then UPR($c$, $f_i$) < $M$. We have PI($c$) < UPI($c$) < UPR($c$, $f_i$) < $M$. Hence, $c$ can be pruned. □

For example, if $M$=0.4, co-locations {C, A, B, D} and {C, A, B} in the prefix-tree C are not star SPI-closed candidates. Then {A, B, C, D} and {A, B, C} cannot be combined in the clique SPI-closed candidate generation.

***Pruning 2*** (Non-SPI-closed pruning strategy 1): If the UPI value of a clique SPI-closed candidate $c$ is marked in boldface, and UPI($c$) = UPI($c'$) ($c \subset c'$, $c'$ is a clique SPI-closed candidate), then $c$ can be pruned.

*Proof*. In the process of generating star SPI-closed candidates, a candidate $c$ is marked in boldface when its UPR is equal to that of its certain super star SPI-closed candidate in the same reference feature set. Therefore, if and only if UPI($c$) is marked in boldface, the value of UPI($c$) might be equal to that of its super-patterns. When UPI($c$) = UPI($c'$) ($c \subset c'$, $c'$ is a clique SPI-closed candidate), $c$ is not a clique

SPI-closed candidate, then $c$ can be pruned. □

For example, in Fig. 4(c), UPI({A, B, C}) = UPI({A, B, C, D}), if {A, B, C, D} is a clique SPI-closed candidate, then {A, B, C} can be pruned; the same applies to {A, C} and {B, C}.

***Pruning 3*** (Non-SPI-closed pruning strategy 2): If the UPI value of a clique SPI-closed candidate $c$ is in boldface, and UPI($c$) = USPI($c|c'$) ($c \subset c'$, $c'$ is a clique SPI-closed candidate), then $c$ can be pruned.

*Proof*. First, if UPI($c$) is not in boldface, it is not possible that UPI($c$) = USPI($c|c'$) ($c \subset c'$). Then, according to Definitions 5 and 6, if UPI($c$) = USPI($c|c'$) ($c \subset c'$), $c$ is a non-SPI-closed pattern when $c'$ is an SPI-closed pattern. Therefore, $c$ can be pruned if $c'$ has already been a candidate. □

For example, in Fig. 4(c), UPI({A, B, D}) = USPI({A, B, D}|{A, B, C, D}) = 2/5, and if {A, B, C, D} is a clique SPI-closed candidate, then {A, B, D} can be pruned; the same applies for {A, D}. However, {B, D} cannot be pruned since UPI({B, D})≠USPI({B, D}|{B, C, D}).

As shown in Fig. 4(c), if $M$=0.3, the generated clique SPI-closed candidates and their UPI values are {A, B, C, D}: 1/3, {A, C, D}: 2/3, {B, C, D}: 2/3}, {A, B}: 3/5, {B, D}: 3/4 and {C, D}: 1. We note that all non-SPI-closed co-locations have been pruned in the combination phase for the data set in Fig. 1(a).

In addition, we note that the Pruning 3 strategy contains Pruning 2 strategy, i.e., the co-locations pruned by Pruning 2 strategy will always be pruned by the Pruning 3 strategy. The reasons for keeping Pruning 2 are: ① the computational complexity of Pruning 2 is lower than that of Pruning 3 because we can use a value search strategy in Pruning 2; ② there are generally more co-locations satisfying Pruning 2.

### 4.2 Generating co-location instances and their PI values

Once the clique SPI-closed candidates have been generated, the co-location

instances of each clique SPI-closed candidate and their true PI values need to be computed. The computation process starts from the largest size candidates.

The *candidate co-location instances* of clique SPI-closed candidates are gathered by scanning neighborhood transactions (e.g., Fig. 3(a)). They are not the true co-location instances. True co-location instances can be filtered from the candidate co-location instances by examining a clique relationship between other instances, except for the first instance of the candidate co-location instance. For example, in Fig. 3(a), {A.2, B.2, C.1, D.2} is a true co-location instance of candidate {A, B, C, D}, but {A.2, B.1, C.1, D.2} is not.

For a *k*-size candidate *c*, if PI(*c*) = UPI(*c*) then *c* must be an SPI-closed co-location. Otherwise, we first need to generate all pruned (*k*-1)-size sub-sets of *c*. Next, if PI(*c*) < *M* then *c* can be pruned; otherwise, we need to check whether *c* is an SPI-closed co-location according to Definitions 5 and 6.

Note that the UPI values of size 2 co-locations are their true PI values.

### 4.3 The SPI-closed Miner algorithm

We propose the *SPI-closed Miner* algorithm to enable the above process. The pseudocode below describes its main process.

**Algorithm** SPI-closed Miner

***Input:*** *(1) A feature set, F={$f_1$, $f_2$, …, $f_n$}; (2) A spatial data set, D; (3) A spatial neighborhood distance threshold, d; and (4) A minimum participation index threshold, M.*
***Output:*** *The SPI-closed co-location set Ω.*
***Method:***
**BEGIN**
*// Preprocess and generate star SPI-closed candidates*
1) *NP* =find_neighbor_pairs(*D*,*d*);
2) (*NT*,*ENT*)=gen_neighbor_transactions(*NP*);
3) **for** *i*=1 to *n*
4) $Tree_i$=build_prefix-tree($f_i$,*ENT*);
5) $SNCC_i$=gen_candi_and_cal_upr ($Tree_i$,*M*); *//Generating the star SPI-closed candidates of $Tree_i$ and calculate their UPR*
6) **if** *UPR*(*c*, $f_i$)<*M* *//*c*∈$SNCC_i$*
**then** *c* is pruned from $SNCC_i$;
7) **if** UPR(*c*, $f_i$)=UPR(*c′*, $f_i$) *//*c*, *c′*∈$SNCC_i$ and *c*⊂*c′*
**then** *c* is marked in *boldface*;

*//*Filter *clique SPI-closed* candidates *by combining star SPI-closed* candidates
8) *z*=1;
9) **While** *z*<*n* **do**
10) *l* = largest size of $SNCC_z$

```
11)     while (l>1 and SNCC_z ≠ϕ ) do
12)         for each l-size candidate c in SNCC_z ;
13)             CNCC←combine_and_cal_upi (SNCC_{z+1}, …, SNCC_n);     //CNCC is the set of clique
        SPI-closed candidates
14)             if UPI(c) is boldface and UPI(c)=USPI(c|c'))              //c'∈CNCC and c⊂c'
                    then c is pruned from CNCC;
15)     l=l-1;
16)   z=z+1;

//Calculate true PIs of candidates and obtain the SPI-closed set Ω
17)   Ω←size 2 candidates in CNCC;
18)   l=largest size of CNCC;
19)   while (l>2 and CNCC≠ϕ) do
20)       for each l-size candidate c in CNCC;
21)           SI_c=find_star_instances(c, NT);
22)           CI_c=filter_clique_instances(SI_c, NT);
23)           PI(c)=calculate_true_pi(CI_c);
24)       if PI(c)=UPI(c)
                then move c from CNCC into Ω
25)        else CNCC ← gen_pruned_(l-1)-sub-sets(c, CNCC);
26)             if PI(c)<M or non-SPI-closed(c)                    //per Definitions 5 & 6
                    then prune c from CNCC
27)              else   move c from CNCC into Ω
28)       l=l-1;
29)   Output Ω
END
```

The algorithm *SPI-closed Miner* contains three phases. The first one preprocesses and generates *star SPI-closed* candidates, the second combines star SPI-closed candidates as clique SPI-closed candidates, and the third generates the SPI-closed co-location set Ω by calculating the true PI values of the candidates.

In the first phase, we find all neighboring instance pairs for a given input spatial data set and a neighborhood distance threshold *d*. The instance neighborhood transactions *NT* are generated by grouping the neighboring instances for each instance. The feature neighborhood transactions *ENT* are then obtained from *NT*. Next, a prefix-tree $Tree_i$ of the feature $f_i$ is built based on the neighborhood transaction features of the feature $f_i$ in *ENT*, where $i$=1,2,…,$n$. Lastly, the set of star SPI-closed candidates $SNCC_i$ is generated by using $Tree_i$, and their UPR values are calculated at the same time. For a candidate $c$ in $SNCC_i$, if UPR($c$, $f_i$) < $M$, then $c$ cannot be prevalent, and we can prune $c$ from $SNCC_i$. If UPR($c$, $f_i$) = UPR($c'$, $f_i$) for $c$, $c' \in SNCC_i$ *and* $c \subset c'$, $c$ is marked in boldface.

In the second phase, the set of clique SPI-closed candidates *CNCC* is filtered by combining the star SPI-closed candidates in $SNCC_1$, $SNCC_2$,…,$SNCC_n$. The UPI of each candidate in *CNCC* is computed at the same time. The boldface marks of UPRs

are maintained in the combination process. The combination process starts from the largest size of $SNCC_1$, and ends when no patterns in $SNCC_1$, $SNCC_2$,…, $SNCC_n$ can be combined. If the minimum UPR value (i.e., UPI) is a boldface one for a candidate $c$ in *CNCC*, and UPI($c$) = USPI($c|c'$) (where $c' \in CNCC$ *and* $c \subset c'$), $c$ can be pruned from *CNCC* by pruning strategies Pruning 2 and Pruning 3.

The third phase calculates the true PI values of candidates in *CNCC* and discovers the SPI-closed prevalent co-location set Ω. First, the star co-location instances of a candidate are found by scanning *NT*. The clique co-location instances can then be filtered from the star co-location instances by examining a clique relationship among other instances except for the first instance of the star instance. Next, the true PIs can be calculated based on the clique co-location instances of candidates. For a candidate $c$, if PI($c$) = UPI($c$) then the candidate can be moved from *CNCC* to the SPI-closed co-location set Ω. However, if PI($c$) ≠ UPI($c$), we have to take a number of further steps, as shown in SPI-closed Miner (Steps 25-27). For a $l$-size co-location $c$, if PI($c$) ≠ UPI($c$) then all those $l$-1-size co-locations which were pruned by Pruning strategies 2 or 3 need to be recovered. If PI($c$) < $M$ or PI($c$) = SPI($c|c'$) ($c \subset c'$ and $c'$ is SPI-closed), $c$ can be pruned since it is not prevalent or SPI-closed per Definition 5 and 6, otherwise, $c$ must be a prevalent SPI-closed co-location.

## 5. Qualitative analysis of SPI-closed Miner

Below, we provide a qualitative analysis of the ability of SPI-closed Miner to accurately discover SPI-closed co-locations.

### 5.1 Discover the correct SPI-closed co-location set Ω

SPI-closed Miner can discover the correct SPI-closed co-location set Ω. First, a co-location instance must be a star neighborhood instance and correspond to a neighborhood transaction feature in Miner. Thus, the star SPI-closed candidates and clique SPI-closed candidates can be introduced into SPI-closed Miner, as shown in

Step 6 and Step 14.

Second, in the case of PI($c$) ≠ UPI($c$), we first generate all pruned ($l$-1)-size-candidates of $c$ in *CNCC* in Step 25, following which $c$ is checked and then dealt with based on Definitions 5 and 6 in Steps 26 -27. In addition, the process of the third phase starts from the largest size of *CNCC*.

Third, to avoid duplicative combination, we adopt a backward combination of *SNCC* (the star candidates of feature $z$) at Step 13. Step 9 guarantees the correctness of the combination phase.

### 5.2 The running time of SPI-closed Miner

The running time of SPI-closed Miner is much faster than that of traditional PI-closed co-location mining methods. First, the SPI-closed condition is stronger than the PI-closed condition, accordingly the candidate set generated in SPI-closed Miner must be smaller than that mined by classic PI-closed co-location mining methods.

Second, the majority of the running time in spatial co-location mining is consumed during the generation of co-location instances and in calculating the PI values. Hence, it is preferable to prune non-SPI-closed patterns when generating candidates as far as possible. This is the method adopted in SPI-closed Miner and the top-$k$ closed co-location mining [34]. For the data set in Fig. 1(a), all non-SPI-closed co-locations have been pruned in the combination step.

Third, when there are many star co-location instances that are not true co-location instances, the work of generating and checking their sub-sets is time-consuming. However, a corresponding problem is also evident in PI-closed co-location mining, therefore in this case, the running time of SPI-closed Miner is still faster than that of PI-closed co-location mining methods.

## 6. Experimental evaluation

Various experiments are conducted to verify the effectiveness and efficiency of

the proposed SPI-closed concept and SPI-closed Miner on both synthetic and real data sets. All algorithms are implemented in Visual C++ in a computer with Intel Core i5 3337U @ 1.80GHz, 2GB RAM, and in Microsoft Windows 7.

To the best of our knowledge, the only algorithm to identify closed co-location patterns, as discussed in this paper, is the Top-*k* closed co-location mining algorithm presented in [34]. Accordingly, we create PI-closed Miner based on the Top-*k* CCP mining algorithm presented in [34] as follows. First, top-*k* Miner finds *k* CCPs with the highest prevalence values, so it has no prevalence threshold, while PI-closed Miner finds all CCPs with PI values not less than a given threshold. PI-closed Miner can therefore prune candidate CCPs according to the given threshold. Second, top-*k* Miner traverses the candidate subset tree in a breadth-first manner by raising an internal prevalence threshold to prune candidate co-locations, as it has no pre-determined prevalence threshold, whereas PI-closed Miner uses the depth-first search strategy to discover all CCPs, saving considerable time and space.

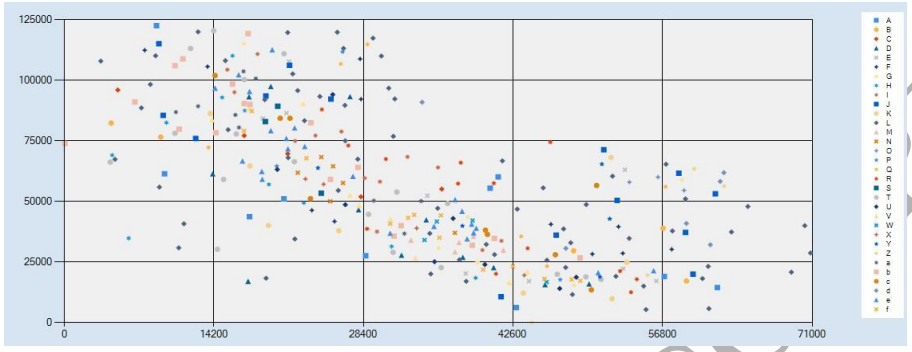
### 6.1 Experiments on real-life data sets

This section examines the performance of the proposed algorithms on three real-life data sets. A summary of the three real data sets is presented in Table 1. The data set Real-1 concerns the rare plant data of the Three Parallel Rivers of Reserved Areas in Yunnan Province, China. It contains 32 features and only 355 instances with a zonal distribution in a 130000m×80000m area as shown in Fig. 5(a). Real-2 is a spatial distribution data set with urban elements, which has more instances than Real-1; the distribution of its instances is even and dense in a 50000m ×80000m area as shown in Fig. 5(b). Real-3 is a vegetation distribution data set of the Three Parallel Rivers of Reserved Areas in Yunnan Province, China. It has the least number of features and the largest number of instances, and its instance distribution is both scattered and clustered in an area of 110000m×160000m as shown in Fig. 5(c).

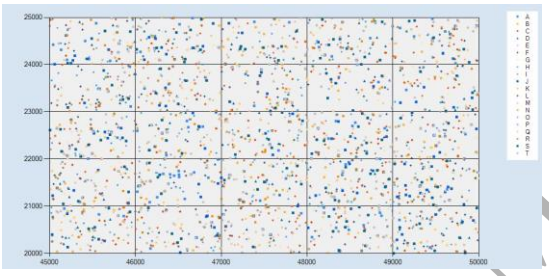
**Table 1** A summary of the three real data sets

| Name | Number of features | Number of instances | (Max, Min) | The distribution area of spatial instances ($m^2$) |
|---|---|---|---|---|
| Real-1 | 32 | 335 | (63, 3) | 130000 × 80000 |
| Real-2 | 20 | 377834 | (60000, 347) | 50000 × 80000 |
| Real-3 | 15 | 501046 | (55646, 8706) | 110000 × 160000 |

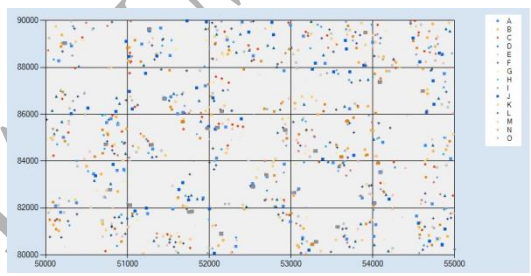
(Max, Min): The maximum and minimum number respectively of the feature's instances in the data sets




**(a)** Spatial distribution of Real-1 data set



**(b)** part of distribution of Real-2



**(c)** part of distribution of Real-3

**Fig. 5.** Spatial distribution of the three real data sets

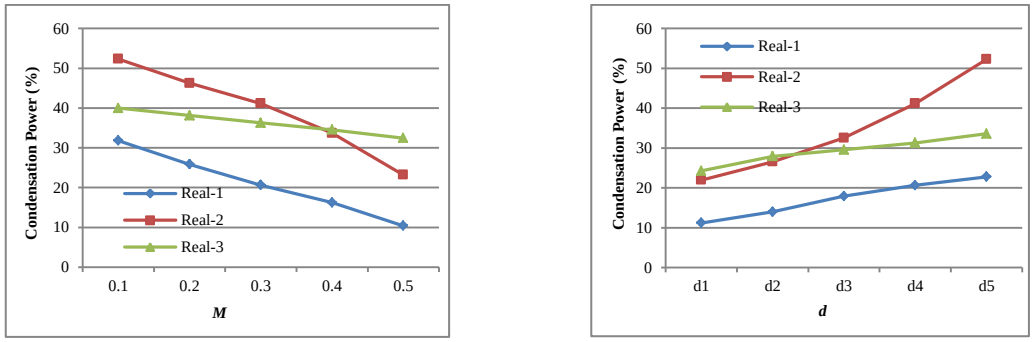
### 1) The effectiveness of SPI-closed Miner

For each real data set, we vary the values of parameters *M* (the minimum participation index threshold) and *d* (the spatial neighbor distance threshold) to verify the condensation power of the SPI-closed co-location mining relative to the result of PI-closed co-location mining using Formula (7).

$$(|S_{PI\text{-}closed}| - |S_{SPI\text{-}closed}|) / |S_{PI\text{-}closed}| \quad [7]$$

where $S_{PI\text{-}closed}$ is the set of PI-closed co-locations and $S_{SPI\text{-}closed}$ is the set of SPI-closed co-locations. The larger the condensation power given by Formula (7), the better is the performance of SPI-closed Miner. The experimental results are shown in Fig. 6(a) - (b). In this experiment, we set the default *d* value of Real-1 as 10000, Real-2 as 4000,

and Real-3 as 10000. The default *M* value is 0.3 for all three real data sets.




(a) By minimum participation index threshold *M*

(b) By neighborhood distance threshold *d*

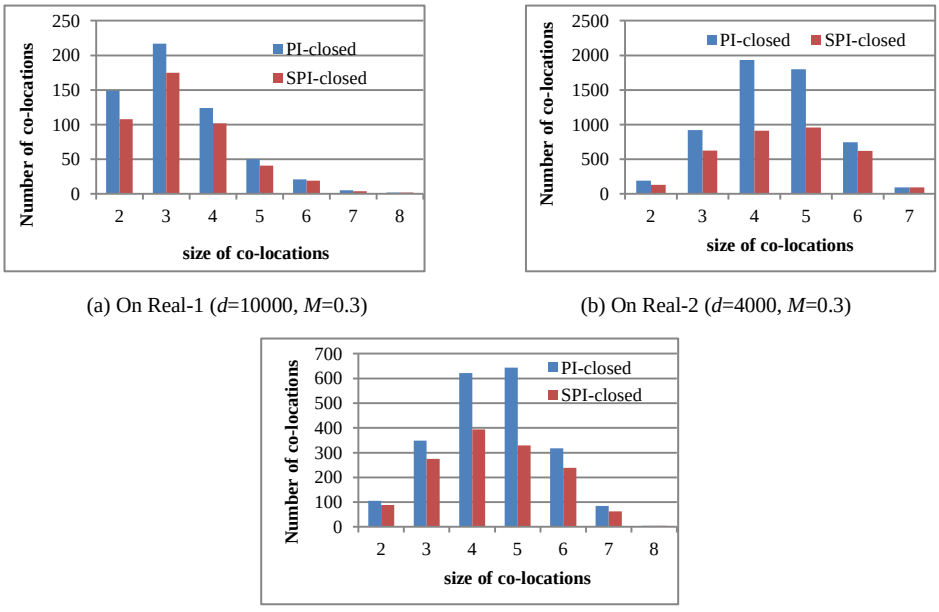
**Fig. 6.** Analysis of the effectiveness of SPI-closed Miner over the three real data sets (In (b), *di*=10000-1000*(4-*i*) for Real-1, *di*=4000-1000*(4-*i*) for Real-2, and *di*=11000-2000*(4-*i*) for Real-3)

We make the following observations from our experiments. First, on all three real data sets, the condensation power is between 10% and 50%, and the mean value is about 30%. The condensation power is highest on the Real-2 data set, and the mean condensation power almost reaches 40%. This is because Real-2 is an even distribution data set. Second, the condensation power increases when *M* becomes smaller or *d* becomes larger. This was anticipated, because there are more prevalent co-locations mined under lower *M* or larger *d*. Third, when *M* or *d* changes, Real-3 faces fewer changes than Real-2. This is because Real-3 is a clustered distribution data set.

Fig. 7(a) - (c) shows the number of mined SPI-closed co-locations compared with the number of PI-closed co-locations by co-location size on the three real data sets using default parameter values. As can be seen, the number of SPI-closed co-locations is less, sometimes much less, than the number of PI-closed co-locations. The largest difference appears in the middle sizes for Real-2 and Real-3, e.g. size 4 and size 5 in Fig. 7(b) - (c), whereas for Real-1 with zonal distribution, there is little difference when size is bigger than 5.

We can find some intuitive insights obtained from the experiments over the real data sets. For example, in the results of Real-2 data set of urban facilities, there are both co-location {Educational institution, bus station, snack bar, small supermarket} and {bus station, snack bar, small supermarket} in the set of PI-closed co-locations,

but only {Educational institution, bus station, snack bar, small supermarket} appears in the set of SPI-closed co-locations. This is because the all row instances of the 3-size {bus station, snack bar, small supermarket} appear in that of 4-size {Educational institution, bus station, snack bar, small supermarket}. This means that the occurrence of {bus station, snack bar, small supermarket} is due to the occurrence of Educational institution in the data set. In other words, the feature "Educational institution" is the key feature in co-location {Educational institution, bus station, snack bar, small supermarket}.




(a) On Real-1 ($d$=10000, $M$=0.3)

(b) On Real-2 ($d$=4000, $M$=0.3)

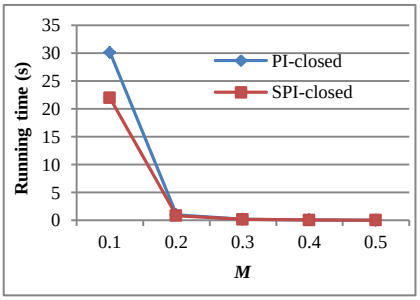
(c) On Real-3 ($d$=10000, $M$=0.3)

**Fig. 7.** Comparison of the mined results w.r.t. co-location size

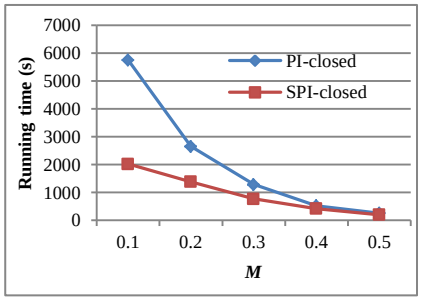
## 2) The efficiency of SPI-closed Miner

The running time of SPI-closed Miner and PI-closed Miner is shown in Fig. 8(a) - (f). Fig. 8 shows that SPI-closed Miner runs much faster than PI-closed Miner when $M$ is small and $d$ is large. We also observe that SPI-closed Miner runs twice as fast as PI-closed Miner in Real-2 when $M$ = 0.1 and $M$ = 0.2 in Fig. 8(b) or $d$ = 5000 in Fig. 8(e), and three times faster in Real-3 when $M$ = 0.1 in Fig. 8(c) or $d$ = 13000 in Fig. 8(f). In addition, SPI-closed Miner is also more space efficient because it avoids the checking of many candidates.

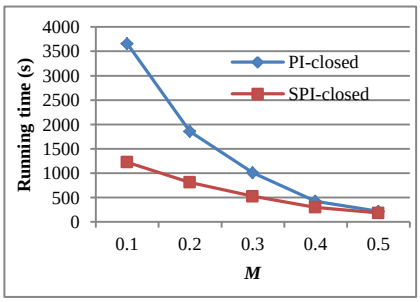
Table 2 compares the number of generated candidates and final results over different sizes by the two algorithms when parameters are set as the default values. For example, the pair value (165/149, 134/108) of size 2 in Table 2 indicates that the number of PI-closed candidates is 165 and the number of PI-closed co-locations is 149, and the related number of SPI-closed is 134 and 108 respectively. As can be seen, the number of SPI-closed Miner candidates is much smaller than the number of PI-closed Miner candidates. Further, we can see that as the size of the candidates grows, the number of SPI-closed Miner-identified candidates is close to the final number in the results. We know that checking a long pattern costs much more time than checking a short one.



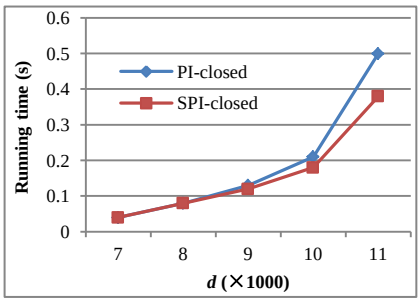

(a) On Real-1 ($d$=10000)



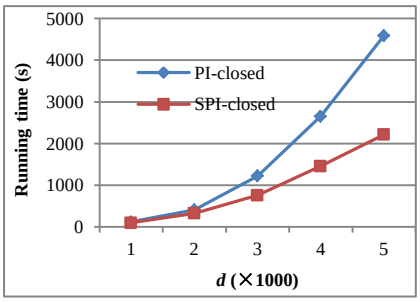

(b) On Real-2 ($d$=4000)



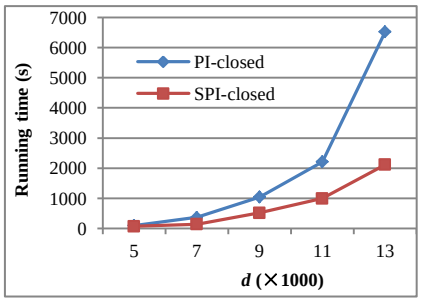

(c) On Real-3 ($d$=10000)




(d) On Real-1 ($M$=0.3)




(e) On Real-2 ($M$=0.3)




(f) On Real-3 ($M$=0.3)

**Fig. 8.** Running time of SPI-closed Miner and PI-closed miner over three real data sets

**Table 2** Comparison of generated candidates and mined results by SPI-closed Miner and PI-closed miner

| Size | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| Real-1 | (165/149, 134/108) | (328/217, 254/175) | (194/124, 135/102) | (148/50, 102/41) | (37/21, 34/19) | (9/5, 8/4) | (7/2, 5/2) | (2/0, 2/0) | (1/0, 1/0) |
| Real-2 | (190/190, 154/132) | (1594/923, 664/626) | (2714/1932, 1108/914) | (1974/1799, 688/956) | (847/744, 401/621) | (107/95, 84/95) | (6/0, 4/0) | (1/0, 1/0) | |
| Real-3 | (105/105, 74/88) | (789/348, 358/275) | (921/621, 547/394) | (1229/643, 448/329) | (384/318, 298/238) | (105/84, 88/62) | (7/4, 3/4) | (1/0, 1/0) | |

1. The two pair values in the entries are PI-closed candidates/PI-closed co-locations, and SPI-closed candidates/SPI-closed co-locations;
2. In these experiments, we set $d$=10000, $M$=0.3 for Real-1, $d$=4000, $M$=0.3 for Real-2 and $d$=10000, $M$=0.3 for Real-3.

### 6.2 Experiments with synthetic data sets

This section examines the scalability of SPI-closed Miner and PI-closed Miner in several scenarios, i.e., different numbers of spatial instances, numbers of spatial features, neighbor distance thresholds, and prevalence thresholds. Synthetic data sets were generated using a spatial data generator similar to that used in [8,32]. Such synthetic data sets allow greater control in studying the effect of corresponding parameters.

The running time of PI-closed Miner exceeds the time limit (20 ks (kiloseconds) > 5 hours) when the number of spatial instances is 600,000, as shown in Fig. 9(a), and when the distance threshold is 10,000, as shown in Fig. 9(c). As shown in Figs. 9(a) - (d), SPI-closed Miner is scalable to large dense data sets, lower $M$, and larger $d$. It performs better than PI-closed Miner in all the experiments. This is because the set of PI-closed co-locations is larger than that of the SPI-closed co-locations and the SPI-closed candidates set is smaller than that of the PI-closed candidates due to the application of the Pruning 3 strategy.

Fig. 9(b) shows interesting cost information: both algorithms cost more time until they reach a peak time cost as the number of features increases, then running time is reduced when the number of features increases further. This is because, as the number of features grows, more time is spent on the prefix-trees of features; when the number of features reaches 40, it drops because the data set is too sparse to have longer prevalent patterns when the total number of spatial instances is fixed. In addition, we can see that the cost difference between the two algorithms in Fig. 9(b) is

the smallest of all the figures. This is because the prefix-tree operation becomes a major factor in SPI-Closed Miner when the number of features grows but the total number of spatial instances is fixed.

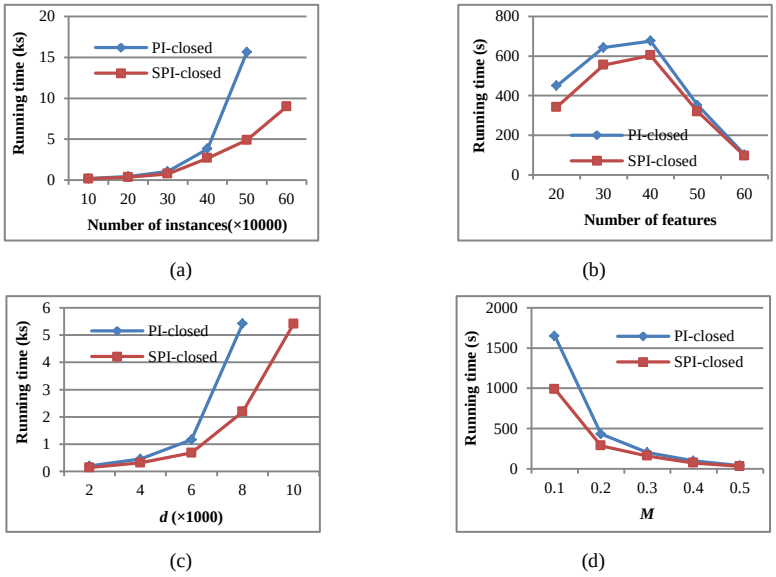



**Fig. 9.** Scalability analysis on synthetic data sets (where (a) $|F|$=20, $d$=1000 and $M$=0.2; (b) $|S|$=200000, $d$=1000 and $M$=0.2; (c) $|F|$=20, $|S|$=200000 and $M$=0; (d) $|F|$=20, $|S|$=200000 and $d$=1000)

We also note that, when the number of spatial instances is 500,000 in Fig. 9(a), SPI-closed Miner runs almost three times faster than PI-closed Miner. At that point, if we compare the number of SPI-closed candidates with the number of PI-closed candidates w.r.t. the co-location size, as shown in Fig. 10, we can see that the number of SPI-closed candidates is far fewer than the number of PI-closed candidates at all sizes, which is why our SPI-closed Miner is so efficient in such situations.

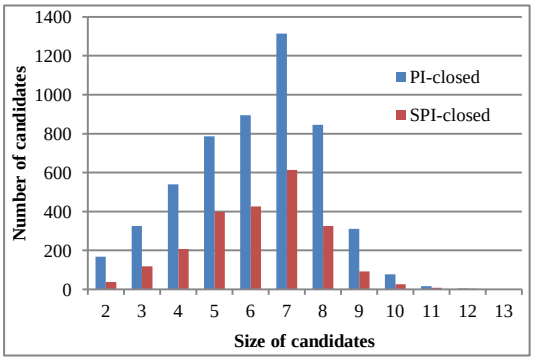



**Fig. 10.** Further analysis on the 500,000 instances in Fig. 9(a)

## 7. Conclusions

In this paper, we present a new lossless condensed representation of prevalent co-location collections and an efficient algorithm *Super Participation Index-closed* (SPI-closed) *Miner*. Both theoretical and experimental analyses show that the proposed SPI-closed concept and its corresponding SPI-closed Miner significantly improve the lossless condensation power of identified spatial co-locations and the efficiency of its execution. In the future, we plan to validate our method with different types of data sets, such as fuzzy data sets and uncertain data sets.

## Acknowledgement

Funding: This work was supported by the National Natural Science Foundation of China (61472346, 61662086, 61762090), the Natural Science Foundation of Yunnan Province (2015FB114, 2016FA026), the Spectrum Sensing and borderlands Security Key Laboratory of Universities in Yunnan (C6165903), the Program for Young and Middle-aged Skeleton Teachers of Yunnan University, and the Project of Innovation Research Team of Yunnan Province.

## References

[1] M. Akbari, F. Samadzadegan, R. Weibel, A generic regional spatio-temporal co-occurrence pattern mining model: a case study for air pollution, J. Geograph. Syst. 17 (2015) 249–274, doi: 10.1007/s10109-015- 0216- 4.

[2] S. An, H. Yang, J. Wang, et al, Mining urban recurrent congestion evolution *patterns* from GPS-equipped vehicle mobility data, Information Sciences, 373 (10) (2016) 515-526.

[3] S. Barua, J. Sander, SSCP: Mining statistically significant co-location patterns, in: Proc. SSTD 2011, pp. 2-20.

[4] L. Cao, Domain driven data mining: Challenges and prospects, IEEE Trans. Knowl. Data Eng. 22 (6) (2010) 755-769.

[5] L. Cao, Coupling learning of complex interactions, Information Processing & Management, 51 (2) (2015) 167–186.

[6] X. Chang, Z. Ma, Y.i Yang, et al, Bi-level semantic representation analysis for multimedia event detection, IEEE Trans. Cybernetics 47(5) (2017) 1180-1197.

[7] X. Chang, Z. Ma, M. Lin, et al, Feature interaction augmented sparse learning for fast kinect motion detection, IEEE Trans. Image Proc. 26( 8) (2017) 3911-3920.

[8] Y. Huang, S. Shekhar, H. Xiong, Discovering colocation patterns from spatial data sets: A general approach, IEEE Trans. Knowl. Data Eng. 16 (12) (2004) 1472-1485.

[9] K. U. Khan, B. Dolgorsuren, T. N. Anh, et al, Faster compression methods for a weighted graph using locality sensitive hashing, Information Sciences, 421 (2017) 237-253.

[10] J. Li, A. Adilmagambetov, M. S. M. Jabbar, et al., On discovering co-location patterns in datasets: A case study of pollutants and child cancers, Geoinformatica, 2016, doi: 10.1007/s10707-016-0254-1.

[11] Z. Liu, Y. Huang, Mining co-locations under uncertainty, in: Proc. SSTD 2013, pp. 429-446.

[12] J. Lu, L. Wang, Q. Xiao, et al, Incremental mining of co-locations from spatial database, in: Proc. FSKD 2015, pp. 648-653.

[13] J. Lu, L. Wang, Y. Fang, et al, Mining competitive pairs hidden in co-location patterns from dynamic spatial databases, in: Proc. PAKDD 2017, pp. 467-480.

[14] Z. Ma, X. Chang, Y. Yang, et al, The many shades of negativity, IEEE Trans. Multimedia, 19(7) (2017) 1558-1568.

[15] Z. Ouyang, L. Wang, P. Wu, Spatial co-location pattern discovery from fuzzy objects. *Int. J. Artif. Intell. Tools*, Vol. 26 (2017) 1750003 (20 pages), doi: 10.1142/S0218213017500038

[16] J. Pei, J. Han, R. Mao, CLOSET: An efficient algorithm for mining frequent closed itemsets, in: Proc. ACM SIGMOD (DMKD Workshop) 2000, pp. 11-20.

[17] C. Sengstock, M. Gertz, T. V. Canh, Spatial interestingness measures for co-location pattern mining, in: Proc. SSTDM (ICDM Workshop) 2012, pp. 821–826.

[18] S. Shekhar, Y. Huang, Co-location rules mining: A summary of results, in: Proc. SSTD 2001.

[19] X. Song, L. Nie, L. Zhang, et al, Interest inference via structure-constrained multi-source multi-task learning, in: Proc. IJCAI 2015, pp. 2371-2377.

[20] F. Verhein, G. Al-naymat, Fast mining of complex spatial co-location patterns using GLIMIT, in: Proc. ICDMW 2007, pp. 679-684.

[21] J. Wang, J. Han, J. Pei, CLOSET+: Searching for the best strategies for mining frequent closed itemsets, in: Proc. ACM SIGKDD 2003, pp. 236-245.

[22] L. Wang, Y. Bao, J. Lu, et al, A new join-less approach for co-location pattern mining, in: Proc. CIT, 2008, pp. 197-202.

[23] L. Wang, L. Zhou, J. Lu, et al, An order-clique-based approach for mining maximal co-locations, Information Sciences 179 (19) (2009) 3370-3382.

[24] L. Wang, P. Wu, H. Chen, Finding probabilistic prevalent colocations in spatially uncertain data sets, IEEE Trans. Knowl. Data Eng. 25 (4) (2013) 790-804.

[25] L. Wang, J. Han, H. Chen, et al, Top-*k* probabilistic prevalent co-location mining in spatially uncertain data sets. Frontiers of Computer Science 10(3) (2016) 488-503.

[26] L. Wang, W. Jiang, H. Chen, Y. Fang, Efficiently mining high utility co-location patterns from spatial data sets with instance-specific utilities, in: Proc. DASFAA 2017, pp. 458-474.

[27] L. Wang, X. Bao, H. Chen, Redundancy reduction for prevalent co-location patterns, IEEE Trans. Knowl. Data Eng. 29 (2017) 1-14, doi: 10.1109/TKDE.2017.2759110.

[28] X. Wang, Y. Zhao, L. Nie, et al, Semantic-based location recommendation with multimodal venue semantics, IEEE Trans. Multimedia 17(3): 409-419 (2015).

[29] X. Xiao, X. Xie, Q. Luo, et al, Density based co-location pattern discovery, in: Proc. SIGSPATIAL, 2008, pp. 11-20.

[30] S. Yang, L. Wang, X. Bao, et al, A framework for mining spatial high utility co-location patterns, in: Proc. FSKD, 2015, pp. 631-637.

[31] X. Yao, L. Chen, L. Peng, et al, A co-location pattern-mining algorithm with a density-weighted distance thresholding consideration, Information Sciences 396 (2017) 144–161.

[32] J. S. Yoo, S. Shekhar, A joinless approach for mining spatial colocation patterns, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1323-1337.

[33] J. S. Yoo, M. Bow, Mining maximal co-located event sets, in: Proc. PAKDD 2011, pp. 351-362.

[34] J. S. Yoo, M. Bow, Mining top-*k* closed co-location patterns, in: Proc. ICSDM 2011, pp. 100-105.

[35] W. Yu, Spatial co-location pattern mining for location-based services in road networks, Expert Systems with Applications, 46 (2016): 324–335.

[36] W. Yu, T. Ai, S. Shao, The visualization and analysis of POI features under network space supported by kernel density estimation, J. Transp. Geogr. 45 (2015) 32–47.

[37] M. J. Zaki, C. J. Hsiao, CHARM: An efficient algorithm for closed itemset mining, in: Proc. SIAM SDM 2002, pp. 457-473.

[38] J. Zhang, L. Nie, X. Wang, et al, Shorter-is-Better: Venue category estimation from micro-video, in: Proc. ACM Multimedia 2016, pp. 1415-1424.

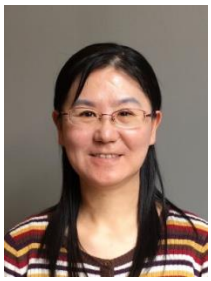
[39] X. Zhang, N. Mamoulis, D. Cheung, et al, Fast mining of spatial co-locations, in: Proc. ACM SIGKDD 2004, pp. 384-393.

[40] L. Zhu, J. Shen, H. Jin, et al, Landmark classification with hierarchical multi-modal exemplar feature, IEEE Trans. Multimedia, 17 (7) (2015) 981-993.

[41] L. Zhu, J. Shen, H. Jin, et al, Content-based visual landmark search via multi-modal hypergraph learning, IEEE Trans. Cybernetics, 45 (12) (2015) 2756-2769.

[42] L. Zhu, J. Shen, X. Liu, et al, Learning compact visual representation with canonical views for robust mobile landmark search, in: Proc. 25th Int. Joint Conf. on Artificial Intelligence (IJCAI'16), New York City, USA, 2016, pp. 3959-3967.

[43] L. Zhu, Z. Huang, X. Liu, et al, Discrete multi-modal hashing with canonical views for robust mobile landmark search, IEEE Trans. Multimedia, 19 (9) (2017) 2066 – 2079.



Lizhen Wang received her M.S. degree in computational mathematics from Yunnan University, China, in 1988 and her PhD in computer science from the University of Huddersfield, UK, in 2008.
She is currently a professor and PhD supervisor at Yunnan University. Her current research interests include spatial data mining, big data analytics, interactive data mining, and their applications.

Xuguang Bao received his M.S. degree in computer application technology from Yunnan University, China, in 2015.
He is currently pursuing a PhD in communication and information systems at Yunnan University. His current research interests include spatial data mining.

Hongmei Chen received her M.S. degree in computational mathematics from Yunnan University, China, in 2001 and her PhD in communication and information systems from Yunnan University, China, in 2012.
She is currently an associate professor at Yunnan University. Her current research interests include data mining.

Longbing Cao received his PhD in pattern recognition and intelligent systems from the Chinese Academy of Science, Beijing, China, and a PhD in computing sciences from the University of Technology Sydney, Australia. Professor Cao is the Founding Director of the Advanced Analytics Institute, University of Technology Sydney, Australia. His current research interests include big data analytics, data mining, machine learning, behavior informatics, complex intelligent systems, agent mining, and their applications.