A Novel Algorithm for Efficiently Mining Spatial Multi-Level Co-Location Patterns

Junyi Li, Lizhen Wang, *Member, IEEE*, Peizhong Yang, and Lihua Zhou

***Abstract*—The spatial co-location pattern is a collection of spatial features in which instances of features prevalently appear in neighboring spatial regions. Due to the heterogeneity of spatial data distribution, the instances of some patterns appear prevalently in the global region (i.e., Global Prevalent Co-location Patterns, GPCPs), while some patterns are not prevalent globally, and their instances are clustered only in some local regions (i.e., Local Prevalent Co-location Patterns, LPCPs). Multi-level co-location pattern mining aims to mine these two types of patterns simultaneously, but existing methods cannot accurately judge the spatial distribution of patterns in a certain region, leading to unsuitable judgment of both GPCPs and LPCPs. To overcome this problem, this paper firstly proposes the relative distribution coefficient to identify the spatial distribution form of patterns, and provides a more refined way for discovering both GPCPs and LPCPs. Secondly, a novel multi-level co-location pattern mining algorithm is proposed by using the relative distribution coefficient as the interest metrics, and some pruning strategies are suggested to improve the mining efficiency. Finally, extensive experiments are conducted on both real and synthetic datasets to verify the effectiveness and efficiency of the proposed method.**

***Index Terms*—Spatial data mining, multi-level co-location pattern, spatial distribution form, relative distribution coefficient.**

I. Introduction

In RECENT years, with the continuous development of spatial information technologies such as remote sensing and satellites, spatial data has exploded. Faced with a large amount of spatial data, how to discover interesting knowledge has become an inevitable problem. Spatial co-location pattern mining (SCPM) is an important branch of spatial data mining, with the aim of mining prevalent co-location patterns from a given spatial dataset. A spatial co-location pattern (or co-location) is a set of spatial features, whose instances often occur closely together in geographical space [1]. SCPM is widely used in many fields, such as ecological protection [2], urban planning [3], transportation [4], etc.

Due to the heterogeneity of spatial data distribution, the instances of some co-locations appear prevalently in the global region, and these co-locations are called the global prevalent co-location patterns (GPCPs). However, the instances of some co-locations are clustered only in some local regions, and those co-locations are named as the local prevalent co-location patterns (LPCPs). More specifically, a GPCP means that there is higher probability that its instances could appear prevalently in the global region, and a LPCP means that there exists at least one local region to make its instances have higher probability of appearing prevalently in the region.

Traditional SCPM methods utilize the participation index (PI) to measure the prevalence of co-locations, but they only work for GPCPs and regrettably miss LPCPs. In practice, the LPCPs are also valuable. For example, in the field of location-based services, LPCPs can provide users with better travel decisions or location selections. Therefore, the local co-location pattern mining (LCPM) is proposed [5], [6], [7], [8], [9], [10], [11], [12]. Existing LCPM methods first divide the global region into different partitions, and then detect LPCPs by utilizing traditional SCPM methods in each partition. Their effectiveness depends on the region division scheme used, where the partition-based scheme may loss neighbor relationships across partitions, while the clustering-based scheme is susceptible to spatial distribution and has higher consumption. Thereafter, the multi-level co-location pattern mining (MLCPM) [13], [14], [15] is proposed to simultaneously mine GPCPs and LPCPs.

It is quite meaningful to mine multi-level co-location patterns, since MLCPM can provide more plentiful information on both global and local levels. For example, in urban planning and construction, the mined GPCPs from the city POI data can provide useful advice with overall urban planning to decision makers, while the LPCPs represent valuable information relative to specific regional planning, since there are different regional characteristics, cultures, and consumers in different cities or even in the same city. In ecological protection, the GPCPs can reveal global symbiotic relationships among multiple species, and the LPCPs can reveal special co-located relationships that cannot be found by traditional SCPM methods, as the co-located relationships between species are often exhibited to be closely related to the local environment where they live.

The existing MLCPM methods treat global non-prevalent co-locations as regional candidates, and then screen the candidates one by one without further metrics for judgment, which leads the

Manuscript received 29 September 2022; revised 10 March 2024; accepted 19 March 2024. Date of publication 25 March 2024; date of current version 7 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62276227, Grant 62306266, Grant 62062066, in part by the Project of Innovative Research Team of Yunnan Province under Grant 2018HC019, and in part by Yunnan Fundamental Research Projects under Grant 202201AS070015, Grant 202401AT070450. Recommended for acceptance by Diane J Cook. *(Corresponding author: Lizhen Wang.)*

Junyi Li, Peizhong Yang, and Lihua Zhou are with the School of Information Science and Engineering, Yunnan University, Kunming 650091, China (e-mail: junyili1028@sina.com; ypz@ynu.edu.cn; lhzhou@ynu.edu.cn).

Lizhen Wang is with the School of Information Science and Engineering and Dianchi College, Yunnan University, Kunming 650091, China (e-mail: lzhwang@ynu.edu.cn).

Digital Object Identifier 10.1109/TKDE.2024.3381178

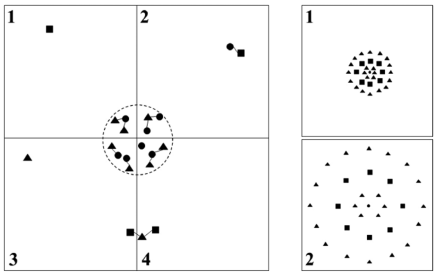


Fig. 1. Examples of the different instance distributions, where each solid symbol (such as ▲, •, ■) represents an instance of its corresponding feature. (a) shows a 2∗2 division, and the co-location {▲,•} has same number of co-location instances in the four regions. (b) shows the similar distribution number of co-location {▲, ■} in the two regions, but instances in the upper one is more clustered than the lower one.

algorithms to be inefficient. Meanwhile, they cannot well detect spatial distribution of co-locations. For example, Fig. 1(a) shows a co-location whose instances' distribution is relatively gather and has less coverage for the study area, but the existing mining methods may identify it as a GPCP. The judgment means that the co-location occurs prevalently across the whole global area by default, which is unsuitable and will mislead users.

Of course, there are also methods to consider the spatial distribution of co-locations [15], [16], by dividing the research region and calculating the difference in the number of co-location instances in different regions. However, they remain with the following problems. First, an unsuitable region division scheme may cause unreasonable judgments about GPCPs or LPCPs (as shown in Fig. 1(a)), and it is not easy to know a suitable scheme in advance. Second, only the number of co-location instances in different sub-regions cannot summarize the distribution states of co-locations. Since GPCPs and LPCPs are relative, judgments of the same co-location might change when the research area is enlarged or reduced. In addition, data distribution of co-location instances should also consider the direction factor, because for the same number of co-location instances, the multidirectional distribution is completely different from the unidirectional distribution.

Therefore, we propose a relative distribution coefficient (RDC) metrics to catch the distribution of a co-location in a certain region, which comprehensively considers quantity and direction factors. The RDC value represents the degree of gather or discrete distribution of a co-location, and we call such judgement as spatial distribution form (SDF), and analyze as many different situations of SDF as possible. Then, we define the GPCPs and LPCPs concepts based on PI and RDC, and discuss the association of global and local prevalence between co-locations and their sub-patterns (or super-patterns). After that, we design a MLCPM-SDF algorithm to mine GPCPs, LPCPs and their prevalent regions efficiently.

The principal contributions of this paper are as follows.

- First, we propose the concept of the relative distribution coefficient of co-locations to measure their spatial distribution form, and based on this we give a formal definition of GPCPs and LPCPs to better reflect their spatial distribution form than existing methods.
- Second, a novel MLCPM-SDF algorithm is proposed to mine GPCPs, LPCPs and their prevalent regions, with some pruning strategies to improve the mining efficiency. The prevalent region for GPCPs is the global research region, while the prevalent regions for LPCPs are generated by a quarter-iterative and regional combination strategy.
- Third, extensive experiments are conducted on both real-world and synthetic spatial datasets by comparing with the existing methods to verify the effectiveness of our proposed metrics and the efficiency of the MLCPM-SDF algorithm.

The remainder of this paper is organized as follows. Section II outlines the basic concepts of SCPM, reviews the related works, and discusses the challenges faced by this work. In Section III, we describe the relevant concepts of our approach and the associated theoretical analysis, as well as propose some pruning strategies. Section IV presents the algorithm framework and the pseudocode, and analyzes the complexity of the algorithm. The experimental results are discussed in Section V. Section VI provides the conclusions and the prospects.

II. Preliminary

A. Basic Concepts

In spatial datasets, different types of spatial objects are represented by **spatial features**, and the appearance of one feature at a particular location in geographic space is called a **spatial instance**. The information of a spatial instance o usually includes the feature type, the instance-id, and its location. The spatial feature set is a collection of different features, denoted as $F = \{f_1, f_2, \ldots, f_n\}$. The spatial instance set is a collection of spatial instances of all features, denoted as $S = \{S_1, S_2, \ldots, S_n\}$, where S_i $(1 \leq i \leq n)$ represents the set of all instances of the spatial features f_i. A spatial co-location pattern is a subset of F, denoted as $C = \{f_1, f_2, \ldots, f_k\}$ $(1 \leq k \leq n)$, $C \subseteq F$, where instances of features in C prevalently appear adjacent in geographic space. The **size** of C is the number of features in C. For example, a specific crocodile in the real world is an instance of the feature crocodile, and the set {crocodile, plover} is a co-location since crocodiles and plovers often live together.

For two instances o and o', if the Euclidean distance between them does not exceed the distance threshold *min_d* specified by users, we call that o and o' have the spatial neighbor relationship NR, denoted as $NR(o, o') \Leftrightarrow (distance(o, o') \leq min\_d)$. For a set of instances $IS = \{o_1, o_2, \ldots, o_k\}$, if IS has the same length as C, and instances of IS covers all features of C and are all neighboring with each other, we say that IS is a **row instance** (RI) of C. The set of all row instances of C is called its **table instance**, denoted as $TI(C)$. The instances in a RI of C constitute a clique relationship under the neighbor relationship NR, and each instance in RI is called a **participating instance** of C. The set of participating instances of C is denoted as $PIC(C) = \{PIF(C, f_1), PIF(C, f_2), \ldots, PIF(C, f_k)\}$, where $PIF(C, f_i) = \{o | o \in RI \wedge RI \in TI(C) \wedge o.t = f_i\}$ is the set of all different participating instances of the feature f_i in C, and $o.t$ is the feature type of the instance o.

To measure the prevalence of co-locations, Shekhar et al. [1] defined the Participation Ratio (PR) and the Participation Index (PI). The **participation ratio** of a feature f_i in C is the ratio of the number of participating instances of f_i to the total number of instances of f_i, denoted as $PR(C, f_i) = |PIF(C, f_i)|/|S_i|$. The **participation index** of a co-location C is the minimum value of the participation ratio of all features in C, calculated as $PI(C) = min_{i=1}^{k}\{PR(C, f_i)\}$.

B. Related Work

Traditional SCPM Methods: SCPM was first proposed by Shekhar et al. [1], who defined the participation index to measure the prevalence of co-locations, i.e., PI, which satisfies the anti-monotonicity, so Apriori-like methods [17], [18] can be used to mine all prevalent co-locations. Huang et al. [19] proposed the join-based algorithm, which generates complete table instances of co-locations by using the join operation. Yoo et al. [20] used a star neighbor model to materialize the spatial neighbor relationship and proposed the joinless algorithm. Wang et al. [21] incorporated the fuzzy theory into the spatial co-location discovery to solve the neighboring degree loss between instances. Wu et al. [22] proposed a novel method to improve the efficiency of co-location mining based on geometric properties of maximal instance cliques. In addition, there are many other mining methods such as the statistics-based method [23], the clique-based method [2], [24], the buffer-based method [25], the method with coupling relation consideration [26], etc.

Methods for mining LPCPs: Existing local co-location pattern mining methods fall into two categories, i.e., the partition-based methods and the clustering-based methods.

(1) The partition-based methods first divide the research area into several local regions, and then mine LPCPs by using traditional methods in these local regions. Their main difference lies in the specified partition scheme used to identify local regions. Celik et al. [5] used a quadtree-based index structure to divide areas, and proposed Zoloc-Miner algorithm to dynamically mining LPCPs. Wang et al. [8] utilized a multi-density clustering algorithm based on maximal cliques to identify prevalent regions of LPCPs. Mohan et al. [7] proposed a neighborhood graph-based method, which divides regions by constructing connected adjacency sub-graphs of co-locations. Qian et al. [10] used k-nearest neighbor graph to partition region. Liu et al. [14] partitioned the region by defining the concept of natural neighborhoods and utilizing the Delaunay triangulation.

The partition-based methods, if improperly partitioned, may lead to the recognition of unreasonable spatial distributions of co-locations, omitting or mis-spliting of regions of LPCPs.

(2) The clustering-based methods use the specified techniques to form clusters of participating instances of a co-location, and represent the clusters as the regions where the co-location lies. Eick et al. [6] viewed the regional co-location mining as a clustering problem, in which an objective function based on the z-score of the continuous variables is utilized to measure the prevalence of co-locations. Deng et al. [13] adopted an adaptive spatial clustering method to generate prevalent sub-regions of LPCPs, where an overlapping method was proposed to improve the computational efficiency, and two constraints were used to check the validity of all sub-regions. Cai et al. [11] applied nonparametric significance test to an adaptive spatial clustering method for mining LPCPs and their prevalent regions.

The clustering-based method also has some steps that need to set thresholds, such as selection of the number of clusters, and the similarity judgment between instances and clusters. Moreover, there is no small computational overhead even for small datasets, since each spatial instance needs to be compared with each cluster during the clustering process, and the clustering speed depends on the spatial distribution of the dataset. Furthermore, the clustering-based method may ignore regions without clustering [12].

Existing MLCPM methods: Existing MLCPM methods usually employ a bottom-up mining framework, i.e., generating candidates from lower size to higher size. In each iteration, they first mine GPCPs, and then identify LPCPs from global non-prevalent co-locations. The method of MLCPM was first proposed by Deng et al. [13], which adopt the same framework of mining from low-size to high-size as traditional methods. In each size, the GPCPs are first mined by using PI measure, and all global non-prevalent co-locations are regarded as candidate LPCPs, then an adaptive spatial clustering method was used to detect LPCPs and their prevalent regions. Liu et al. [14] utilized a heuristic multi-level refinement method based on the proposed concept of natural neighborhood, which used the Delaunay triangulation to generate sub-regions of LPCPs. These methods use the PI as the prevalence measurement, where PI is calculated by considering how many instances in the total instance set participate in the co-location. Such calculation can only reflect the co-located degree of co-locations, but not the distribution state of its participating instances. For a mined GPCP C by traditional methods, if the instances of features are precisely clustered in a local region, the distribution range of its participating instances can only be local, which does not match its named meaning "global". Therefore, Liu et al. [15] proposed a multi-level method based on the uniform coefficient (UC) of the distribution of row instance and several pruning strategies to filter the candidate LPCPs.

Although the existing MLCPM methods can discover GPCPs and LPCPs, most of them cannot make suitable judgment on the spatial distribution of a co-location. There is a MLCPM research considering the distribution of co-location instances based on the proposed UC metrics [15], but it still has the following shortcomings. First, its mining results depend on the dividing scheme used. Second, it only considers the differences in the number of co-location instances within the divided regions and ignores the spatial distribution form of the instances, which may still lead to unsuitable judgment facing the phenomenon in Fig. 1(b). Because the same number of co-location instances is a gather distribution in region 1 and a discrete distribution in region 2, resulting in the same number of instances having completely different coverage of the regions. So, considering only the number of co-location instances in divided regions does not reflect the spatial distribution status of co-locations in the regions.

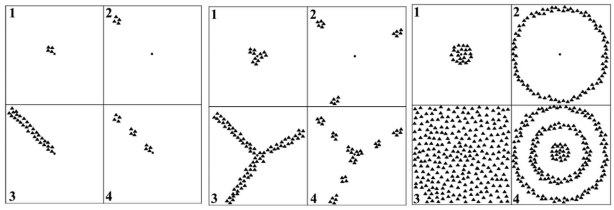


Fig. 2. SDF of instances in different situations. a) Unidirectional situations; b) multi directional situations; and c) circumferential situations.

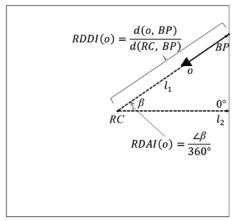


Fig. 3. Illustration of the *RDDI*(o) and *RDAI*(o) of instance o, where the box represents the current research area, the solid point is the area center *RC*, the solid triangle is an instance o, and the dotted arrow line l_2 indicates the 0° direction. The dotted arrow l_1 starting at *RC* passes through o, and intersects the boundary at point *BP*.

C. Challenges

As can be seen from the above discussion, in MLCPM, the problem of effectively distinguishing the global/local distribution of co-locations has still not been well addressed. In this work, we focus on developing a novel metrics to outline the spatial distribution form (SDF) of co-locations in a certain region to reasonably distinguish between GPCPs and LPCPs. To develop such a metrics, the following issues should be considered. (1) SDF is a relative concept, so when considering the SDF of a co-location, the internal spatial location relationship in a co-location and the external relationship of the co-location relative to the region should be considered simultaneously. In this way, when the size of the region changes, the corresponding judgment results can be obtained accordingly. (2) It is promising to develop an efficient MLCPM algorithm by investigating some properties of new metrics. In particular, it hopes to study the influence of new metrics of lower size global/local co-locations on those of higher size co-locations, so as to generate as few candidate global/local co-locations as possible and reduce the computational cost.

III. MLCPM-SDF: A Multi-Level Co-Location Pattern Mining Method Considering the Spatial Distribution Form of Instances

A. Judgments of Spatial Distribution Form

After clarifying the purpose of SDF judgment, we analyze the following different situations as possible, and other complex distribution may be the combination of them. We describe the SDF of instances in a region as discrete or gather, corresponding to the global/local distributed meaning in that region. Considering both the center and boundary of region as references, we outline the situations with same angle of view as near-cluster, far-cluster, multi-cluster and band, and outline situations with different visual angle as unidirectional and multi-directional (as shown in Fig. 2, and only draw three directions or clusters as examples of multi-direction or multi-cluster). Besides, the annular situation (Fig. 2(c)) is the extension of the multi-directional situation, and can be further divided into near-ring cluster, far-ring cluster, multi-ring clusters, and entire bands.

We discriminate the SDF in these situations as follow. The SDF in Fig. 2(b)-3, (b)-4, (c)-3 and (c)-4 are described as **discrete**, since the instances are relatively full in the entire global region. And the rest of the situations are described as **gather**, because the occurring region of instances is relatively not global enough. Besides, the SDF in (b)-3 and (b)-4 will gather more with the direction decreases, and the judgment of such ambiguous phenomenon depends on the measure of SDF and the threshold set, and we designed a quarter-iterative strategy in Algorithm 2 to solve this problem.

B. Method for Judging the SDF of Instances

For judging the SDF of co-location instances, we define the concept of relative distribution coefficient, which can be calculated synchronously during the calculation of PI values of co-locations.

In a certain region, in order to determine the SDF of a given co-location, we first select objects which can represent the region, and then transform the problem into judging the SDF relative to these reference objects. The points in the boundary can serve as references since they delimit the regional range, and the regional center can also be served as a reference object since it can be easily used to the statistics distribution of instances. In Definition 1, we integrate spatial correlations of instances and reference objects by proposing the concept of relative distribution distance. And we choose the participating instances of co-locations for measuring the distribution, as it is not practical to delimit the scope or the center of row instances. The relevant definitions are given below.

Definition 1: **(RDDI & RDDF)** Given a regional center *RC*, a co-location $C = \{f_1, f_2, \ldots, f_k\}$, and a feature f ($f \in C$) with N participating instances (i.e., $|PIF(C, f)| = N$). For any instance o in *PIF*(C, f), let the ray which starts from *RC* and passes through o as l_1, and the intersection point between l_1 and the region boundary as *BP*. The **Relative Distribution Distance of Instance** (RDDI) o is the ratio of the Euclidean distance between o and *BP* to that between *RC* and *BP* (shown in Fig. 3), denoted as *RDDI*(o). The **Relative Distribution Distance of Feature** (RDDF) f in C, denoted as *RDDF*(C, f), is the ratio of $S^2_{rddi}(f)$ to the product of N and the square of $\overline{rddi}_f$, calculated as

$$RDDF(C, f) = \frac{S^2_{rddi}(f)}{N \cdot \overline{rddi}_f^{\,2}} \tag{1}$$

where $S^2_{rddi}(f)$ is the variance of the RDDI values of all instances in *PIF*(C, f), and $\overline{rddi}_f$ is the average value of RDDI.

The RDDF can describe the distribution of C. From a geometrical point of view of the distribution, the RDDI values of all instances will be more similar when $RDDF(C,f)$ is smaller, which indicates that their SDF is more gather. Conversely, the SDF of instances in $PIF(C,f)$ is more discrete when $RDDF(C,f)$ is larger. For the example in Fig. 1(b), the SDF of instances in region 1 is more gather than that in region 2, and their RDDF is indeed smaller than that in region 2. The existing methods cannot effectively distinguish this phenomenon, while the RDDF can. Besides, the calculation of RDDI (shown in Fig. 3) is to ensure that the calculated measurement value is related to current research region.

However, for different SDFs shown in Fig. 2(a) and (b), it cannot be well distinguished based on the RDDF alone. Inspired by the statistics of directional data [27], this paper considers the direction factor for better judging the SDF of instances. We take the rightward direction as reference direction, i.e., the 0° direction, and integrate directional correlations of instances and the reference direction by proposing the concept of relative distribution angle in Definition 2. The relevant definitions are given below.

Definition 2: **(RDAI & RDAF)** Given the same conditions as Definition 1, and let the right ray which starts from RC as l_2, as shown in Fig. 3. For any instance o in $PIF(C,f)$, the **Relative Distribution Angle of Instance** (RDAI) o is the angle passed from l_2 to the position of l_1 in the counterclockwise direction divided by 360°, denoted as $RDAI(o)$. The **Relative Distribution Angle of Feature** (RDAF) f in C, denoted as $RDAF(C,f)$, is the ratio of $S^2_{rddi}(f)$ to the product of N and the square of $\overline{rdai_f}$, calculated as

$$RDAF\left(C,f\right)=\frac{S^2_{rdai}\left(f\right)}{N\cdot\overline{rdai_f}^2} \tag{2}$$

where $S^2_{rddi}(f)$ is the variance of the RDAI values of all instances in $PIF(C,f)$, and $\overline{rdai_f}$ is the average value of RDAI.

Lemma 1: For a co-location C, a feature f ($f\in C$) with N participation instances (i.e., $|PIF(C,f)|=N$), *we have $RDDF(C,f)<1$ and $RDAF(C,f)<1$.*

Proof: Let's discuss $RDDF(C,f)$ firstly. Since

$$S^2_{rddi}(f)=\frac{1}{N}\sum_{i=1}^{N}\left(RDDI(o_i)-\overline{rddi_f}\right)^2$$

$$=\frac{1}{N}\sum_{i=1}^{N}(RDDI(o_i)^2-2RDDI(o_i)\cdot\overline{rddi_f}+\overline{rddi_f}^2)$$

$$=\frac{1}{N}\left(\sum_{i=1}^{N}RDDI(o_i)^2-2\sum_{i=1}^{N}RDDI(o_i)\cdot\overline{rddi_f}+N\overline{rddi_f}^2\right)$$

$$=\frac{1}{N}\left(\sum_{i=1}^{N}RDDI(o_i)^2-N\overline{rddi_f}^2\right)$$

Thus, we have

$$RDDF\left(C,f\right)=\frac{S^2_{rddi}\left(f\right)}{N\cdot\overline{rddi_f}^2}$$

$$=\frac{\frac{1}{N}\left(\sum_{i=1}^{N}RDDI(o_i)^2-N\overline{rddi_f}^2\right)}{N\cdot\overline{rddi_f}^2}$$

$$=\frac{1}{N^2}\left(\frac{\sum_{i=1}^{N}RDDI(o_i)^2}{\frac{1}{N^2}\left(\sum_{i=1}^{N}RDDI(o_i)\right)^2}-N\right)$$

$$\le\frac{1}{N^2}\left(N^2-N\right)=\frac{N-1}{N}<1$$

Similarly, we can also prove $RDAF(C,f)<1$. □

Based on Lemma 1, we know that RDDF and RDAF are both bounded. Next, we will introduce how RDDF and RDAF can be used to describe the SDF of instances in different situations. In Fig. 2, the RDDF values of regions 3 and 4 in Fig. 2(a), (b) and (c) are greater than that of the remaining regions, and the RDAF values for all regions in Fig. 2(a) are corresponding smaller than that in Fig. 2(b) and (c). Therefore, the SDF in Fig. 2(a)-1 and (a)-2 can be well judged as gather, (c)-3 and (c)-4 can be well judged as discrete. And the judgment of SDF in the remaining situations depends on the influence proportion between RDDF and RDAF, (1) the judgment of SDF in Fig. 2(a)-3 and (a)-4 depend more on the RDDF value, and that in Fig. 2(b)-1, (b)-2, (c)-1 and (c)-2 depend more on the RDAF value, but none of them would exceed the threshold set, thus their judgment well hold and is the same as observed in Section III.A; (2) the situations of SDF in Fig. 2(b)-3 and (b)-4 cannot be judged only by RDDF or RDAF, and they should be combined, which is the relative distribution coefficient (RDC) discussed later.

Then, there is a natural question that the RDDF (or RDAF) value of which feature should be selected to describe the SDF of a co-location C. Since participating instances of C are neighboring with each other under the distance threshold min_d, then there is a natural guess that the SDF of participating instances for different features in C are similar, and the RDDF (or RDAF) values of them are close within an acceptable range. Thus, there may be a definition connection between the SDF of features in C, and some special features may have more significant impact. Considering following phenomenon, (1) for the feature with the least number of participating instances, the instances appear to be surrounded by that of other features; (2) if the SDF of the feature with the most gather distribution can be considered as being discrete, the distributions of the remaining features in the co-location must also be discrete. So, we choose the minimum value of RDDF (RDAF) to define the SDF of co-locations.

Definition 3: **(RDDP & RDAP)** For a co-location C, the **Relative Distribution Distance of Pattern** (RDDP) C is the minimum of the relative distribution distance of all features in C, and the **Relative Distribution Angle of Pattern** (RDAP) C is the minimum value of the relative distribution angle of all features in C, calculated as

$$\begin{cases}RDDP\left(C\right)=\min\limits_{f\in C}\left\{RDDF\left(C,f\right)\right\}\\RDAP\left(C\right)=\min\limits_{f\in C}\left\{RDAF\left(C,f\right)\right\}\end{cases} \tag{3}$$

Lemma 2: **(anti-monotone)** *RDDP and RDAP monotonically decrease as the size of co-locations increases.*

Proof: For a co-location C and its any sub-pattern C' ($C' \subset C$), the participating instances of any feature in C must be included in the participating instance set of the same feature in C'. That is, the size of the participating instance set of any feature in C is no more than that in C'. Let's discuss RDDP firstly. By Definition 3, RDDP takes the minimum value of RDDFs, so the problem transforms into the proof that RDDF decreases monotonically as the number of participating instances decreases. The monotonically decreasing property of the coefficient of variation (CV, calculated as $\sqrt{S^2_{rddi}(f)/\overline{rddi_f}}$) is proved by [28]. Because: (1) the square of a function with monotonicity must also have monotonicity; (2) according to Definition 1, the relative distribution distance $RDDI(o)$ of participating instance o ($0 \leq RDDI(o) \leq 1$) can be seen as a random variable; (3) Weibull distribution has characteristics that can present many different types of distributions. Therefore, RDDF decreases monotonically as the number of instances decreases, i.e., $RDDF(C, f) < RDDF(C', f)$ holds. Similarly, the above proof process still holds for RDAF. Thus, both RDDP and RDAP are monotonically decreasing. □

Since both RDDP and RDAP map to a range of 0 to 1 for facilitating the user to set the threshold, we set weights to integrate the impact of distance and direction.

Definition 4: **(RDC)** The **Relative Distribution Coefficient** (RDC) of a co-location C is defined as

$$RDC\,(C) = w_1 RDDP\,(C) + w_2 RDAP\,(C) \tag{4}$$

where w_1 and w_2 are the weight coefficients ($w_1 + w_2 = 1$), which are used to adjust the influence ratio of RDDP and RDAP when judging the SDF of co-location C. A larger RDC value of C means the more discrete SDF of the instances of C. Otherwise, its SDF is more gather. Under a specified threshold *min_rdc*, the SDF of the instances of C is considered discrete when RDC(C) is not lower than *min_rdc*, otherwise the SDF is gather. So the threshold *min_rdc* actually defines the minimum requirement for the discreteness of the instance distribution, and we can say that the instances are distributed discretely if the RDC value reaches *min_rdc*. Furthermore, *min_rdc* is different from the distance threshold *min_d*, the *min_d* is a distance constraint on the neighbor relationship between instances, and since RDC is normalized to the 0-1 interval, the *min_rdc* takes values in the 0-1 interval.

Lemma 3: For a co-location C, (1) if its SDF is discrete, all its sub-patterns have a discrete distribution; (2) if its SDF is gather, then all its super-patterns have a gather distribution.

Proof: For the conclusion (1), given the threshold *min_rdc*, the weights w_1 and w_2, we have $RDC(C) = w_1 RDDP(C) + w_2 RDAP(C) \geq min\_rdc$, because the SDF of C is discrete. For any sub-pattern C' of C, i.e., $C' \subset C$, since $RDDP(C') \geq RDDP(C)$ and $RDAP(C') \geq RDAP(C)$ are hold based on Lemma 2, $RDC(C') = w_1 RDDP(C') + w_2 RDAP(C') \geq w_1 RDDP(C) + w_2 RDAP(C) \geq min\_rdc$. Thus, the distribution of C' is discrete. The conclusion (2) can be proven similarly, and we do not repeat it. □

Pruning Strategy 1: Based on Lemma 3, to find co-locations with discrete distribution, we can filter out such co-locations that one of its sub-patterns has a gather SDF, and select the co-locations that do not have a gather SDF.

C: Discussions on Global and Local Prevalence

Existing MLCPM methods are quite computationally expensive, because (1) the PI needs to be calculated firstly for each candidate co-location; (2) for each candidate LPCP with the PI metrics below the prevalence threshold, the local regions need to be generated and the PI value of the candidate co-location needs to be calculated for each local region. We note that the calculation of PI is time-consuming. In order to speed up the mining process, this section explores the relationships of global or local prevalence between patterns and their super-patterns, and proposes some pruning strategies. First, the participation index metrics on global and local levels are refined as follows:

Definition 5: **(GPR, GPI & LPR, LPI)** For a co-location C, and a feature $f \in C$, the **Global Participation Ratio** (GPR) of f in C is the ratio of the number of different participating instances of f to the total number of instances of f in the global research area, denoted as $GPR(C, f)$. In a local region R, the **Local Participation Ratio** (LPR) of f in C is the ratio of the number of different participating instances of f in R to the total number of instances of f in R, denoted as $LPR(C, f)_R$. Then, we give the concepts of **Global Participation Index** (GPI) and **Local Participation Index** (LPI) in a local region R, calculated as

$$\begin{cases} GPI\,(C) = \min\limits_{f \in C} \{GPR\,(C, f)\} \\ LPI(C)_R = \min\limits_{f \in C} \{LPR(C, f)_R\} \end{cases} \tag{5}$$

This paper considers that the GPCPs should not only satisfy their GPI threshold, but also the SDF should be relatively scattered and covering a certain proportion of regional space. For some GPCPs identified by the existing methods, their SDF are gather and their prevalent regions are local. In this paper, we attribute them to the category of LPCPs. Then, the definitions of GPCPs and LPCPs based on the GPI, LPI and RDC are given as follows.

Definition 6: **(GPCP & LPCP)** For a co-location C, given the prevalence threshold *min_prev* and the RDC threshold *min_rdc*, C is a **Global Prevalent Co-location Pattern** (GPCP), only when its GPI value no less than *min_prev* and its RDC value no less than *min_rdc*. For a co-location C', C' is a **Local Prevalent Co-location Pattern** (LPCP), only when its RDC value is less than *min_rdc* in the global area, and there existing at least one sub-region R with the $LPI(C')_R$ value no less than *min_prev*.

Discussion of the relationship between the RDC(C) and LPCP/GPCP classification of C: (1) a larger RDC value of C means that the wider the distribution of the instances of C, the more globally prevalent C can be. Otherwise, its instance distribution is more gather, and the less likely it is to become globally prevalent. (2) set a threshold *min_rdc*, when RDC(C) is not lower than *min_rdc* and its prevalence threshold is met, C is classified as GPCP, otherwise C will be classified as LPCP (Its prevalent regions will be further identified).

The definitions of GPR/GPI in global and LPR/LPI in a local region R are essentially the same as the traditional definitions of PR/PI, therefore, they also satisfy anti-monotonicity, i.e., given two co-locations C' and C, where $C' \subset C$, we have GPR(C, f)$\leq$GPR(C', f)/LPR(C, f)$_R \leq$LPR(C', f)$_R$ for $f \in C'$, $f \in C$; and GPI(C)$\leq$GPI(C')/LPI(C)$_R \leq$LPI(C')$_R$.

Based on Definition 6 and the anti-monotonicity of GPI/LPI, there are several properties of multi-level co-location mining given as follows. (1) If a pattern C is globally non-prevalent, all its super-patterns $\{C' | C \subset C'\}$ must be globally non-prevalent (because of the anti-monotonicity of GPI). If C is globally prevalent, all its sub-patterns $\{C' | C' \subset C\}$ must be globally prevalent (because of the anti-monotonicity of GPI and Lemma 3). (2) If a pattern C is locally prevalent (i.e., there is at least one local region R with LPI value satisfying the prevalence threshold), all its super-patterns must be globally non-prevalent (because C must be globally non-prevalent according to Definition 6). (3) If a pattern C is both globally and locally non-prevalent, all its super-patterns must be both globally and locally non-prevalent (because of the anti-monotonicity of GPI/LPI).

Pruning Strategy 2: Based on above conclusions, when detecting GPCPs, we can filter out such patterns that one of its sub-patterns is globally non-prevalent or locally prevalent, and select such patterns that all its sub-patterns are globally prevalent. To detect LPCPs, we can filter out such patterns that one of its sub-patterns is locally non-prevalent, and select such patterns that all its sub-patterns are locally prevalent as candidates.

IV. MLCPM-SDF Algorithm

Based on the theoretical study in Section III, a multi-level co-location pattern mining algorithm based on SDF (namely MLCPM-SDF) is proposed in this section. Moreover, we analyze the complexity of MLCPM-SDF, and discuss the correctness and completeness of MLCPM-SDF in mining multi-level co-location patterns.

A. General Framework

Since both GPI/LPI and RDC satisfy the anti-monotone property, the MLCPM-SDF algorithm mines GPCPs and LPCPs size-by-size, and detects the prevalent regions of LPCPs. The general framework is shown in Algorithm 1.

Algorithm 1: General framework for MLCPM-SDF.

Input: (a) instance set S; (b) feature set F; (c) neighbor relationship NR; (d) prevalence threshold min_prev; (e) RDC threshold min_rdc; (f) Minimum region size threshold min_Reg
Output: a set of GPCPs and LPCPs with their prevalent regions
Variables: (a) GP_k: set of size-k GPCPs with their prevalent regions; (b) LP_k: set of size-k LPCPs with their prevalent regions; (c) k: co-location size; (d) C_k: set of size-k candidates; (e) G_Reg: the global research area; (f) L_Regs: set of local prevalent regions of a LPCP
Method:
1 Read data and materialize neighbor relationships;
2 Initialize $GP_1 = F$, $LP_1 = \emptyset$, $k = 2$;
3 **While** GP_{k-1} **or** LP_{k-1} not empty **do**
4 C_k = apriori_gen(GP_{k-1}, LP_{k-1});
5 Initialize $GP_k = \emptyset$, $LP_k = \emptyset$;
6 **For each** $C \notin C_k$ **do**
7 generate $PIC(C)$;
8 calculate $GPI(C)$ and $RDC(C)$;
9 **If** $GPI(C) \geq min\_prev$ **and** $RDC(C) \geq min\_rdc$ **then**
10 $GP_k = GP_k \cup \{C, G\_Reg\}$; //$C$ is a GPCP
11 **Else If** $RDC(C) < min\_rdc$ **then**
12 L_Regs = gen_PreRegs(C, G_Reg, $PIC(C)$);
13 $LP_k = LP_k \cup \{C, L\_Regs\}$; //$C$ is a LPCP
14 $k = k+1$;
15 **Return** union (GP_2, …, GP_{k-1}), (LP_2, …, LP_{k-1});

In Algorithm 1, we first input the spatial data, materialize the neighbor relationship and initialize variables (Steps 1-2). Next, Steps 3-14 is the process of performing multi-level co-location pattern mining from the smaller size to the higher size. In each loop, it first generates the candidate co-location set similar to Apriori method (Step 4), and then detects the prevalent co-locations and their prevalent regions from the candidates. For each candidate, MLCPM-SDF first generates its participating instance set using the CPM-Col algorithm [29], and calculates its GPI and RDC (Steps 7-8), then it identifies the candidate as a GPCP, LPCP or non-prevalent co-location according to Definition 6 and Pruning Strategy 2 (Steps 9-13). Besides, if the candidate is identified as a GPCP, its prevalent region is the global research area (Step 10). If it is identified as a LPCP, the algorithm will detect its prevalent regions by the function **gen_PreRegs()** (Step 12). Step 15 outputs the mining results, i.e., the GPCPs, LPCPs and their prevalent regions. Next, we describe the main steps in detail.

(1) Converting the Input Data to a Set of Neighbor Instances (Step 1): Given a certain input spatial dataset and a distance threshold min_d, we find out all neighboring instances pairs by a geometric method similar to [19]. First, we divide the global research area into $\frac{\sqrt{2}min\_d}{2} * \frac{\sqrt{2}min\_d}{2}$ grids, and all instances are assigned to these grids. Then, for finding the neighbor instances of an instance o in a certain grid LG, all instances in LG are neighboring with o and can be added directly to the result set, and the remaining neighbors of o need to be found only in 20 grids adjacent to LG [30].

(2) Generating Candidate GPCPs and LPCPs (Step 4): Based on Pruning Strategies 1 and 2, the candidate generation phase adopts an Apriori-like method, i.e., the size-k candidate co-locations are generated by joining two size-(k-1) prevalent co-locations and checking its all size-(k-1) prevalent co-lcoations. Among them, the size-k candidate GPCPs are generated by size-(k-1) GPCPs, and the size-k candidate LPCPs come from the set of size-(k-1) GPCPs and size-(k-1) LPCPs.

(3) Generating Participating Instances for all Candidates (Step 7): Existing methods need to generate complete table instances by utilizing the join operation, which costs a high computation. Therefore, we adopt a heuristic search method [29], which only needs to search a part of row instances for

Algorithm 2: gen_PreRegs(C, Reg, $PIC(C)$) detects prevalent regions for a co-location C.

Input: C, Reg, $PIC(C)$
Output: $FR(C)$: a set of prevalent regions of C;
1. initialize $FR(C) = \emptyset$, and k is the size of C;
2. **If** $k = 2$ **then**
3. **If** Reg.size $\leq$ min_Reg **then**
4. **If** if_PreReg(C, Reg, $PIC(C)$) **then**
5. $FR(C) = FR(C) \cup \{Reg\}$;
6. **Return** $FR(C)$;
7. calculate $RDC(C)_{Reg}$;
8. **If** $RDC(C)_{Reg} \geq min\_rdc$ **then**
9. **If** if_PreReg(C, Reg, $PIC(C)$) **then**
10. $FR(C) = FR(C) \cup \{Reg\}$;
11. **Return** $FR(C)$;
12. **Else**
13. divide Reg into quarters;
14. generate $PIC(C)$ in each sub-region q in Reg;
15. **For each** $q \notin Reg$ **do**
16. $FR' =$ gen_PreRegs(C, q, $PIC(C)$);
17. $FR(C) = FR(C) \cup FR'$;
18. **Else**
19. initialize $CFR = FR(C'')$; //CFR is candidate prevalent regions and C'' is the first pattern in $Sub\_P(C)$
20. **For each** $C' \notin Sub\_P(C)$ **do**
21. $CFR = CFR \cap FR(C')$;
22. **For each** $q \notin CFR$ **do**
23. **If** if_PreReg(C, q, PIC(C)) **then**
24. $FR(C) = FR(C) \cup \{q\}$;
25. **Return** $FR(C)$;

generating participating instances. Besides, the calculation of GPI and RDC can be performed synchronously during the process of this step.

(4) Detecting GPCPs and LPCPs (Steps 9-15): After obtaining the GPI and the RDC of candidates, we detect GPCPs and LPCPs based on Definition 6 and Pruning Strategy 2. For each candidate, (1) if both its GPI and RDC satisfy the given thresholds, it is identified as a GPCP. (2) If its RDC do not satisfy the RDC threshold, it is identified as a candidate LPCP. Then, we invoke the function **gen_PreRegs()** shown in Algorithm 2 for detecting whether the candidate has prevalent regions. It can be identified as a LPCP only when the function result is not null. (3) Otherwise, the candidate is identified as a non-prevalent co-location and will be filtered.

B. Detection of Prevalent Regions for LPCPs

Algorithm 2 shows the details of the function **gen_PreRegs()** to detect the prevalent regions of LPCPs, where min_Reg is the minimum region size threshold and $Sub\_P(C)$ represents the set of all size-(k-1) sub-patterns of the size-k candidate C.

For each size-2 candidate LPCP, we adopt a quadratic iterative approach to detect its prevalent sub-region. For the current region Reg, we first determine whether it reaches the recursive termination condition, i.e., the Reg size is lower than min_Reg (Step 3). If it is, Steps 4-5 use the function **if_PreGeg()** to determine whether Reg is a prevalent sub-region of the candidate. If not, we calculate its RDC in region Reg (Step 7) and judge it as follows. (1) If the RDC satisfies the threshold min_rdc, which means that the SDF of the candidate in the current region is discrete, it is necessary to judge the LPI value of C in Reg and decide whether adding Reg into the result set $FR(C)$ by the function **if_PreGeg()** (Steps 8-11). (2) If the RDC is lower than min_rdc, the SDF of the candidate in Reg is gather. In order to detect its specific prevalent regions, we divide the current region into quarters, and then call the function **gen_PreRegs()** iteratively for each sub-region until it reaches the recursion termination condition (Steps 13-17). In this way, the proposed approach can gradually approximate the prevalent regions of co-locations in ambiguous situation. Besides, it is not necessary to calculate the RDC in the first iteration, and we display as this only for the logic and cleanliness of the pseudo-code.

Algorithm 3: if_PreGeg(C, Reg, $PIC(C)$) determines whether Reg is a prevalent region of C.

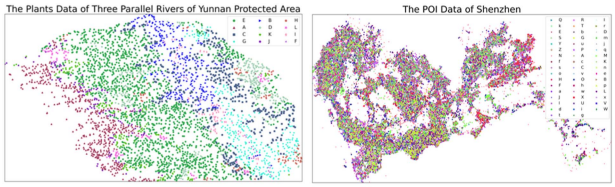
Input: C, Reg, $PIC(C)$
Output: True or False
1. calculate $LPI(C)_{Reg}$;
2. **If** $LPI(C)_{Reg} \geq min\_prev$ **then**
3. **Return** *True*;
4. **Else Return** *False*;

For size-k (k>2) candidate LPCPs, we adopt an Apriori-like approach based on Lemma 3 to generate the prevalent regions of a candidate LPCP, i.e., the prevalent regions of a size-k LPCP are generated from the prevalent regions of all its size-(k-1) sub-patterns. First, we generate the set of candidate prevalent regions CFR by collecting the non-repeating grids covered by the prevalent regions of all its sub-patterns (Steps 19-21). Then, we detect the prevalence of each candidate region, and decide whether one region q can be identified as a prevalent region of C by the function **if_PreReg()** (Steps 22-24). Besides, we need scan once the participating instance set of the candidate co-location for collecting participating instances and calculating the LPI value in each candidate region.

Algorithm 3 shows details of the function **if_PreReg()** to judge whether an input region Reg is a prevalent region of C. Reg is identified as a prevalent region of C when the LPI of C in Reg no lower than the prevalence threshold min_prev, and then the region Reg can be added into the output set $FR(C)$. Otherwise, the function outputs false.

C. Algorithm Analysis

Complexity Analysis: The complexity of the MLCPM-SDF algorithm mainly comes from two aspects. (1) Generating the participating instance set for each candidate co-location using the CPM-Col algorithm [29], i.e., Step 7 in Algorithm 1, and the complexity is $O(k*n_{1*}{n_2}^{k-1})$, where k is the size of the candidate, n_1 is the maximum length of the candidate participating instance set of all candidate co-locations, and n_2 is the maximum length of the instance search space. (2) Detecting the prevalent regions for



(a) the plants dataset (b) the POI dataset

Fig. 4. Spatial distribution of instances in real-world datasets.

each LPCP by the function **gen_PreRegs()**. In the worst case, the function iterates until the termination condition when instances cluster at some local regions. Therefore, the complexity of **gen_PreRegs()** is $O(n_{3*}n_4)$, where n_3 is the maximum length of the participating instance set of all candidate LPCPs, and n_4 represents the maximum recursion depth of the function. Besides, n_1, n_2 and n_3 will decrease as k increases, and k and n_4 is constant in each iteration.

Correctness: In Step 7, the participating instances of all candidate co-locations generated by using the CPM-Col is complete, i.e., there is no participating instance lost, and its correctness has been proved in [29]. Thus, the calculation of GPI for all candidates is correct. For judging the prevalence of a candidate LPCP in each candidate region, we only select participating instances of the co-location in the region, thus the calculation of LPI for all candidate LPCPs within their own prevalent regions are correct. In addition, the MLCPM-SDF algorithm only identifies candidates which satisfy the conditions in Definition 6 as GPCPs or LPCPs. Therefore, the MLCPM-SDF can correctly discover GPCPs and LPCPs.

Completeness: The completeness of the MLCPM-SDF algorithm is guaranteed by following aspects. (1) In Algorithm 1, the neighbor relationship materialized in Step 1 is complete. (2) The generated candidate co-locations are complete, which is guaranteed by the anti-monotonicity of GPI/LPI and Lemma 2. (3) MLCPM-SDF mines GPCPs and LPCPs size-by-size, and the judgment about the prevalence ensures that there is no co-location lost in Steps 9-13. Therefore, MLCPM-SDF can discover all GPCPs and LPCPs.

V. Experimental Evaluation

In this section, we evaluate the effectiveness and the performance of the MLCPM-SDF algorithm.

A. Experimental Setup

In the experimental evaluation, we use both real-world and synthetic spatial datasets.

Real-world Datasets: The first real-world dataset is the plants data of Three Parallel Rivers of Yunnan Protected Area, which includes 12 features and 3855 instances, and the number of instances of features ranges from 40 to 1535. The second real-world dataset is the Shenzhen POI data including 622888 instances with 50 features, and the number of instances of features ranges from 106 to 39746. Fig. 4 shows the distribution of the two real-world datasets.

Synthetic Datasets: All synthetic datasets are generated by using a spatial data generator similar to [20]. First, the global research area is set to $D \times D$ size and divided into regular grids with grid length gl, the size of feature set is F and the total number of instances is N. Then, we generate P core co-locations with average size S, whose features are randomly selected from the feature set. For each core co-location, we generate I row instances (RIs) on average, and the instances of each RI are randomly selected from the instance set of the corresponding feature. For each RI, we randomly choose a grid for locating it and all instances of RI are randomly distributed in the chosen grid. The parameter *clumpy* is used to control the data density, and we locate *clumpy* RIs into a grid when it is selected to locate a RI. Besides, the RI number of all core co-locations follows a Poisson distribution with the mean value I, and the length of all core co-locations follow a Poisson distribution with the mean value S. Table I shows the main parameters setup for generating the synthetic datasets used in each experiment. If there is no description, the parameters take values as follow: $S = 5$, $P = 20$, $I = 200$, $F = 20$, $N = 50000$, $D = 10$ km, $gl = 10$ m, $clumpy = 5$.

Comparison Algorithms: The comparison algorithms used in the experiments are all the existing advanced multi-level co-location mining algorithms, including the multi-level algorithm in [13] (denoted as ML), the multi-level algorithm based on natural neighborhoods (ML-Miner [14]), and the multi-level algorithm based on uniform coefficient (MLCPM-UC [15]).

Experiment Platform: All algorithms are implemented in JAVA, and all experiments are conducted on a computer with a Microsoft Windows 10 operating system, Intel Core i7-8700K CPU @3.70 GHz, and 16 GB main memory.

B. Effectiveness of MLCPM-SDF

1) On the Plants Dataset: First, we conduct experiments on the plants dataset to show the effectiveness of the proposed MLCPM-SDF. The distance threshold *min_d* is set to 1000 m, which ensures that almost every instance has at least one neighbor and the size of the constructed neighboring instance set in all testing algorithms are similar. The remaining parameters are set as follows: *min_prev* $= 0.3$, *min_rdc* $= 0.4$, *min_Reg* $=\sqrt{2}$ *min_d*, and $w_1 = w_2 = 0.5$.

Table II demonstrates the details of main co-locations and lists the corresponding metrics calculated in each algorithm. There are differences in judgments of mined co-locations by algorithms, and we select some representatives to analyze the mining results as follows.

(1) All algorithms identify the co-location {D, E} as a GPCP. From the spatial distribution shown in Fig. 5(a), the instances of D and E occur frequently adjacent (the GPI value meets the *min_prev*), and the prevalent regions have a degree of coverage (the RDC value meets the *min_rdc*).

(2) For the co-location {A, J}, as shown in Fig. 5(b), the instance distributions of A and J are gather (not scattered), and compared with the whole research area, the scattered degree (i.e., the RDC value) does not reach the *min_rdc*, so it is judged more reasonably a LPCP. However, all existing methods identify

TABLE I
EXPERIMENTAL PARAMETERS IN EACH EXPERIMENT ON SYNTHETIC DATASETS

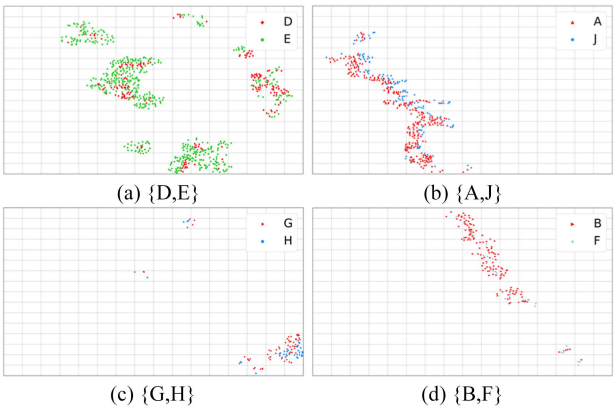
Symbols	Meaning	Experiment on Fig. 7				Experiment on Fig. 8			
		(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
S	Average size of co-locations[a]	3				5			
P	Number of co-locations[a]	10				20			
I	Average number of RIs per co-location[a]	100				200			
F	Number of spatial features	*	6			20			
N	Number of spatial instances	10K	*	10K	50K				
D	Spatial area size (D×D)	10km				*	10km		
clumpy	Number of RIs in a same neighbor area	1		*	1	5			

[a] initial core co-location, * variable values

TABLE II
CO-LOCATIONS IN THE PLANTS DATASET BY ALGORITHMS

Co-locations	ML		MLMiner		MLCPM-UC			MLCPM-SDF		
	GPI	Judge	GPI	Judge	GPI	UC	Judge	GPI	RDC	Judge
{D, E}	0.3655	**GPCP**	0.3491	**GPCP**	0.3491	0.8348	**GPCP**	0.3655	0.4139	**GPCP**
{A, J}	0.6472	GPCP	0.6013	GPCP	0.6013	0.8693	GPCP	0.6472	0.1382	**LPCP**
{D, H}	0.4015	GPCP	0.4461	GPCP	0.4461	0.7963	GPCP	0.4015	0.2173	**LPCP**
{B, F}	0.4716	GPCP	0.6454	GPCP	0.6454	0.6263	**LPCP**	0.4716	0.1362	**LPCP**
{G, H}	0.2027	**LPCP**	0.3089	GPCP	0.3089	0.3783	**LPCP**	0.2027	0.1401	**LPCP**
{H, L}	0.1707	**LPCP**	0.1026	**LPCP**	0.1026	0.5219	**LPCP**	0.1707	0.1847	**LPCP**

* D: Mixed Forest (269); E: Temperate Conifer Forest (1535); A: Ice or Snow Vegetation (479); J: Alpine Screes Sparse Meadow (127); H: Cold Temperate Mountain Hard Lea (82); B: Dry/Hot Shrub (282); F: River (40); G: Warm Temperate Savanna Bushes (301); L: Alpine Meadow (78); (number): the total number of instances per feature.



(a) {D,E} (b) {A,J} (c) {G,H} (d) {B,F}

Fig. 5. SDF of participating instances of co-locations in the plants datasets. Each grid with the side length *min_Reg*≈1,414 m.

it as a GPCP, while our MLCPM-SDF identifies it as a LPCP. The reason may be as follows. Both ML and ML-Miner make judgments by using only the GPI metrics, but ignore the SDF of co-location instances. MLCPM-UC considers the distribution of co-location instances by the proposed UC metrics, but UC focuses on the differences in the number of co-location instances within the sub-regions and cannot summarize the distribution status of the region. There is an example that both the GPCP {D, E} and the LPCP {A, J} have higher UC value, and MLCPM-UC cannot make suitable judgment on such co-locations, even if changing the corresponding UC threshold. MLCPM-SDF uses the RDC metrics to judge the SDF of co-locations, and the RDC can identify the gather property of such SDF by considering the influence of both distance and direction factors.

(3) for co-location {G, H}, only ML-Miner identifies it as a GPCP due to the calculated higher GPI value. The main reason lies in the difference of the neighboring instances set generated by natural neighborhoods in ML-Miner and by methods utilizing the distance threshold. For the co-location {B, F}, only MLCPM-UC and MLCPM-SDF identify it as a LPCP with considering the distribution of instances, and it does have a gather SDF as shown in Fig. 5(d).

Summarizing these results, MLCPM-SDF can address some issues of the existing methods to identify and distinguish GPCPs and LPCPs.

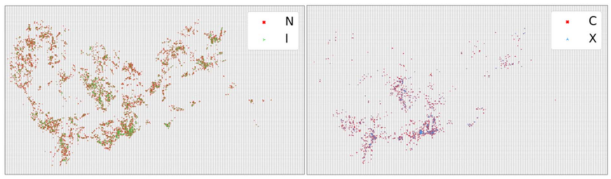
2) On the POI Dataset: Then, we conduct experiments on the POI dataset. Since all existing methods have large time consumption (i.e., exceed 100,000s) when the feature number exceeds 10, we only demonstrate and analyze the mined results by our approach. The distance threshold *min_d* is set to 100 m, and *min_prev* = 0.4, *min_rdc* = 0.3, *min_Reg* = 500 m. Table III demonstrates the details of top-5 co-locations (sorted by their RDC values), and we display the spatial distribution of two representative co-locations in Fig. 6(a) and (b).

For the co-location {N, 1}, we select it for showing whether the mined GPCPs are global enough in terms of their spatial distribution. For the co-location {C, X}, it is identified as a LPCP although its GPI (0.4826) satisfies the prevalence threshold,

TABLE III
TOP-5 CO-LOCATIONS IN THE POI DATASET

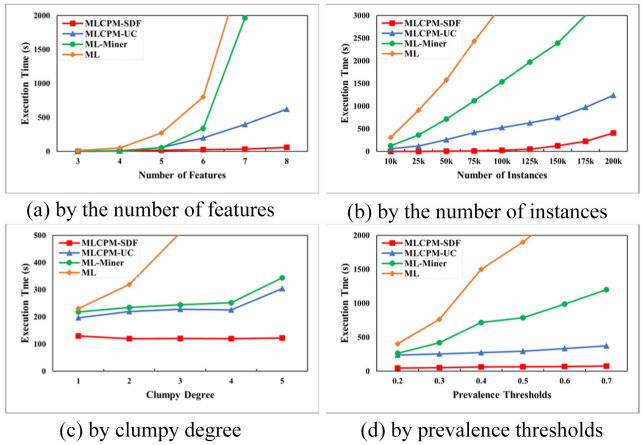
GPCPs			LPCPs							
			Ascending by RDC				Descending by RDC			
Co-locations	GPI	RDC	Co-locations	GPI	RDC	Reg	Co-locations	GPI	RDC	Reg
{N, l}	0.4682	0.3320	{C, V}	0.0001	0.0001	1	{P, d}	0.3329	0.2999	6
{N, f}	0.4665	0.3125	{V, x}	0.0022	0.0001	1	{P, h}	0.2359	0.2998	9
{f, l}	0.7603	0.3091	{V, r}	0.0004	0.0002	1	{H, P}	0.3559	0.2997	2
{Z, s}	0.5788	0.3073	{V, m}	0.0013	0.0006	1	{D, d}	0.1741	0.2996	2
{N, Z}	0.4201	0.3062	{V, a}	0.0017	0.0058	2	{N, s}	0.4149	0.2979	10

* N: Traffic place (23845); f: Comprehensive market (23565); l: Supermarket (17278); Z: Home building materials market (24196); s: Auto Sales (22905); P: Teahouse (11609); d: Appliances and electronics market (14813); h: Casual restaurant (7098); H: Hotel (20428); D: Factory (4645); C: Bank (8846); V: Service area (267); x: Cinema (446); r: Soda fountain store (4960); m: Stationery store (4323); a: Clothing store (19666); (number): the total number of instances per feature.



(a) {N,1} (b) {C,X}

Fig. 6. SDF of participating instances of co-locations in the POI datasets. Each grid with the side length *min_Reg* = 500 m.



(a) by the number of features (b) by the number of instances

(c) by clumpy degree (d) by prevalence thresholds

Fig. 7. Scalability test on common parameters.

because its RDC (0.1963) is lower than the threshold *min_rdc*. The two co-locations have similar GPI values, and the difference is their SDF. From Fig. 6, we can see that the RDC values better reflect the difference in the degree of coverage of the prevalent regions of co-locations.

For the co-location {C, V}, both its GPI and RDC are far from the corresponding thresholds and close to 0, but it presents prevalence within a sub-region. Such co-locations may be instructive either for protecting rare features (features with few instances) or for guiding the development selection of new business sites.

Therefore, the results mined by MLCPM-SDF can reveal more information, and it also enriches the SDF information of co-locations.

C. Performance of MLCPM-SDF

In this part, we evaluate the performance of the proposed MLCPM-SDF algorithm on synthetic datasets. If there is no description, the parameters are set as follows: *min_prev* = 0.4, *min_rdc* = 0.45, *min_d* = 100 m, *min_Reg* = $\sqrt{2}$ *min_d*, and $w_1 = w_2 = 0.5$.

1) Efficiency Comparison: First, we examine the performance of MLCPM-SDF by comparing it with the existing methods (i.e., ML, ML-Miner and MLCPM-UC), and Fig. 7 displays the results. In this experiment, we test the effects of common parameters, which include: (1) parameters for generating synthetic datasets (*the number of features F*, *the number of instances N* and *clumpy*), and their default values are listed in Table I; (2) parameters for verifying the efficiency of algorithms (*prevalence thresholds*). For each value of all parameters, we conduct examinations and summarize the results by randomly generating 10 datasets.

Effect of the number of features: If the number of instances *N* is determined, the number of features *F* affects the complexity of neighbor relationships in the dataset. When *F* increases, more candidates will be generated and be required for calculating their prevalence. Fig. 7(a) shows the execution time of all algorithms, and MLCPM-SDF has smaller time consumption. We analyze the reason as follows. (1) Both ML and ML-Miner take all the global non-prevalent co-locations as candidate LPCPs, while MLCPM-SDF and MLCPM-UC utilize pruning strategies for filtering candidate LPCPs. (2) Both ML-Miner and MLCPM-UC utilize a multi-direction optimization method to mine prevalent regions for each candidate LPCP, which generates prevalent regions by Delaunay triangulation and Convex Hull, and further detects the potential regions by Voronoi diagram. While MLCPM-SDF utilizes the parameter *min_Reg* to control the recursive depth of the algorithm, even if its value is as small as set to *min_d*, which has smaller time consumption.

Effect of the number of instances: Under a certain spatial research area, the number of instances *N* affects the data density. When it increases, the number of instances of features may increase, and more neighbor relationship of instances will

generate. We test the effect of different distance thresholds (i.e., 50 m–500 m) for each value of N, and finally select the results with moderate threshold (not the best results) for MLCPM-SDF to make comparison, since some comparison algorithms have no distance threshold. Fig. 7(b) shows the execution time of algorithms, where MLCPM-SDF and MLCPM-UC have relatively stable time consumption than ML-Miner, and ML is exploding. The main reason may be as follows. (1) All existing methods need to generate all RIs for each candidate co-location, while MLCPM-SDF only search part of RIs to generate participating instances of candidates. (2) for ML-Miner and MLCPM-UC, the time consumption of the multi-direction optimization step is exponentially related to the number of participating instances, while that is approximately constant relation for MLCPM-SDF.

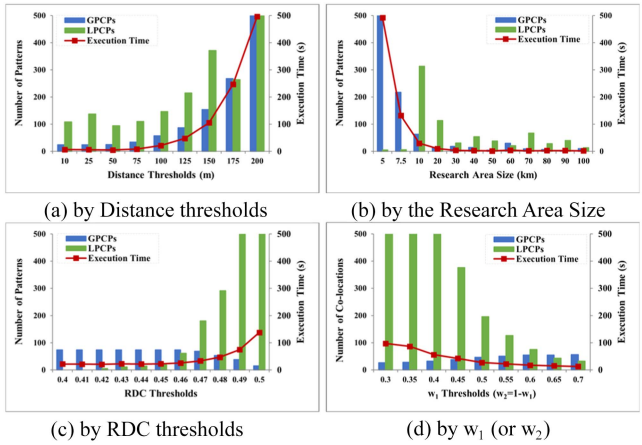
Effect of clumpy degree: The parameter *clumpy* degree controls the number of RIs in the same neighborhood area. With *clumpy* increases, the data density increases and the RI number explosively increases in the same grid. Fig. 7(c) shows the execution time of algorithms, where existing methods have relatively large time consumption, but MLCPM-SDF still stable when *clumpy* increases. The main reason may be that existing methods need to generate all RIs for each candidate, while MLCPM-SDF generates all participating instances of candidates, and the size of the participating instance set remains the same with *clumpy* changing.

Effect of prevalence thresholds: The prevalence threshold *min_prev* controls the number of mined GPCPs and LPCPs. With *min_prev* increases, the number of candidate LPCPs increases since more candidate co-locations become globally non-prevalent. As shown in Fig. 7(d), the execution time of both ML and ML-Miner increase faster when *min_prev* increases, since they do not filter the increasing candidate LPCPs. MLCPM-SDF and MLCPM-UC are stable, because they use some pruning strategies. In fact, for MLCPM-SDF and MLCPM-UC, the number of candidate LPCPs remains constant because the number of co-locations with gather distributions remains constant when only *min_prev* increases, thus these two algorithms are stable.

Summarizing these results, MLCPM-SDF has better performance than the existing methods.

2) The Analysis of the Intrinsic Mechanisms: Next, we examine the intrinsic mechanisms of MLCPM-SDF, i.e., the effect of the specific parameters on it, and Fig. 8 displays the results. The specific parameters contain *the distance threshold*, *the minimum region size*, *the global research area size*, *the RDC threshold* and w_1 (or w_2).

Effect of distance thresholds: The distance threshold *min_d* mainly affects the result for generating the neighbor instance set. The larger *min_d* is, the more neighbor relationship between instances, thus more instances will form the clique relationship, which means an increase in the number of RIs and co-locations. Fig. 8(a) shows the execution time and the number of mined co-locations under different distance thresholds. With the distance threshold increases, both the execution time and the number of mined GPCPs and LPCPs increase, because more participating instances will be generated for each candidate and will be detected by MLCPM-SDF. However, the number of detected LPCPs drops sharply when *min_d* reaches 175 m, the reason



(a) by Distance thresholds (b) by the Research Area Size

(c) by RDC thresholds (d) by w_1 (or w_2)

Fig. 8. Effect of the specific parameters.

may be that these reduced co-locations are identified as GPCPs, which is related to the spatial distribution of the generated dataset.

Effect of min_Reg thresholds: The minimum region size *min_Reg* is used to control the recursive depth in Algorithm 2, and more candidate prevalent regions are generated and are detected for judging the prevalence of each LPCP in the region when *min_Reg* decreases. There is little change in both the execution time and the number of mined co-locations, and only an increase in the number of prevalent regions of some LPCPs, thus we will not display the results in Fig. 8.

Effect of the research area size: When other parameters are determined, the global research area size can affect the spatial distribution of the overall data. When the size decreases, the density of the dataset increases and the more row instances may be generated. Thus, the number of candidates and the execution time will increase, which can be validated by the results shown in Fig. 8(b). Among the results, there are little mined co-locations when the research area size exceeds 20 km, because the density is so low and thus only few neighbor relationships between instances were formed. When the parameter is less than 10 km, both the number of mined GPCPs and the execution time increase, and the reason may be as follows. First, the higher density causes more neighbor relationships formed and more RIs generated, thus it takes more time to detect the participating instances. Second, the smaller area size causes the more discrete distribution of instances, and the increase in the number of participating instances causes an increase in PR and PI values, thus more candidates are eligible to become GPCPs.

Effect of RDC thresholds: The RDC threshold *min_rdc* mainly affects the judgment of GPCPs or LPCPs. With *min_rdc* increases, the spatial distributions of more candidates are identified as gather. We also show the number of mined co-locations and the execution time in Fig. 8(c). When *min_rdc* increases, the number of mined GPCPs decreases, and the number of mined LPCPs and the execution time increase. The reason contains two aspects. First, MLCPM-SDF identifies only prevalent co-locations with discrete distribution (satisfying *min_rdc*) as GPCPs. Second, more co-locations that do not satisfy *min_rdc* are included as candidate LPCPs, thus the time consumption

increases. Meanwhile, we noticed a large increase in the number of LPCPs after at the RDC threshold greater than 0.47. This is because the increase in the number of lower-size LPCPs results the appearance and increase of higher-size LPCPs, and a combinatorial explosion appears for this dataset.

Effect of w_1 (or w_2)*:* The w_1 (or w_2, denoted in Definition 4) mainly affects the influence of distance and direction factors, which affects the mining results of judging the SDF of candidates. As shown in Fig. 8(d), when w_1 increases, the number of mined GPCPs increases and the LPCPs number decreases. The reason may be as follows. For co-locations with discrete distribution, the discreteness in direction may have a greater impact on the discreteness of their distribution within the certain region, then the calculated RDC value increases, and more candidates will be identified as GPCPs. Similarly, for co-locations with gather distribution, the discreteness in distance may affect more, thus the number of LPCPs decreases. The execution time decreases since there are fewer candidate LPCPs needed to be visited for detecting their prevalent regions.

VI. Conclusion and Future Work

In this paper, we propose a relative distribution coefficient (RDC) metrics to catch the spatial distribution form (SDF) of co-locations, which comprehensively considers distance and direction factors. Based on RDC, we give a formal definition of GPCPs and LPCPs to better reflect their distribution than existing methods. And we present a multi-level co-location pattern mining algorithm called MLCPM-SDF to detect GPCPs, LPCPs and their prevalent regions simultaneously, where the prevalent regions of GPCPs are just the global research area, and the prevalent regions of LPCPs are identified by a quadrant iteration and region combination strategy. We perform extensive experiments on both real-world and synthetic datasets to verify the superiority of the proposed MLCPM-SDF algorithm.

In future work, we plan to uncover the fuzzy properties of determining global and local concepts and explore efficient multi-level co-location mining methods based on fuzzy metrics. Moreover, parallelizing multi-level co-location mining to process massive spatial data is promising.

Reference

[1] S. Shekhar and Y. Huang, "Discovering spatial co-location patterns: A summary of results," in *Proc. 7th Int. Symp. Spatio-Temporal Databases*, 2001, pp. 236–256.

[2] X. Bao and L. Wang, "A clique-based approach for co-location pattern mining," *Inf. Sci.*, vol. 490, pp. 244–264, 2019.

[3] X. Yao et al., "A co-location pattern-mining algorithm with a density-weighted distance thresholding consideration," *Inf. Sci.*, vol. 396, pp. 144–161, 2017.

[4] X. Yao et al., "Efficiently mining maximal co-locations in a spatial continuous field under directed road networks," *Inf. Sci.*, vol. 542, pp. 357–379, 2021.

[5] M. Celik, J. M. Kang, and S. Shekhar, "Zonal co-location pattern discovery with dynamic parameters," in *Proc. 7th IEEE Int. Conf. Data Mining*, 2007, pp. 433–438.

[6] C. F. Eick et al., "Finding regional co-location patterns for sets of continuous variables in spatial datasets," in *Proc. 16th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2008, pp. 1–10.

[7] P. Mohan et al., "A neighborhood graph based approach to regional co-location pattern discovery: A summary of results," in *Proc. 19th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2011, pp. 122–132.

[8] D. Wang et al., "An approach based on maximal cliques and multi-density clustering for regional co-location pattern mining," *Expert Syst. Appl.*, vol. 248, 2024, Art. no. 123414.

[9] D. Wang et al., "RCPM_CFI: A regional core pattern mining method based on core feature influence," *Inf. Sci.*, vol. 658, 2024, Art. no. 119895.

[10] F. Qian et al., "Mining regional co-location patterns with kNNG," *J. Intell. Inf. Syst.*, vol. 42, no. 3, pp. 485–505, 2014.

[11] J. Cai et al., "Adaptive detection of statistically significant regional spatial co-location patterns," *Comput. Environ. Urban Syst.*, vol. 68, pp. 53–63, 2018.

[12] Y. Li and S. Shekhar, "Local co-location pattern detection: A summary of results," in *Proc. 10th Int. Conf. Geographic Inf. Sci.*, 2018, pp. 1–15.

[13] M. Deng et al., "Multi-level method for discovery of regional co-location patterns," *Int. J. Geographical Inf. Sci.*, vol. 31, no. 9, pp. 1846–1870, 2017.

[14] Q. Liu et al., "An adaptive detection of multilevel co-location patterns based on natural neighborhoods," *Int. J. Geographical Inf. Sci.*, vol. 35, no. 3, pp. 556–581, 2021.

[15] X. Liu, L. Wang, and L. Zhou, "MLCPM-UC: A multi-level co-location pattern mining algorithm based on uniform coefficient of pattern instance distribution," *Comput. Sci.*, vol. 48, no. 11, pp. 208–218, 2021.

[16] J. Zhao et al., "Mining co-location patterns with spatial distribution characteristics," in *Proc. 5th Int. Conf. Comput. Inf. Telecommun. Syst.*, 2016, pp. 1–5.

[17] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. 20th Int. Conf. Very Large Data Bases*, 1994, pp. 487–499.

[18] J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," in *Proc. 21th Int. Conf. Very Large Data Bases*, 1995, pp. 420–431.

[19] Y. Huang, S. Shekhar, and H. Xiong, "Discovering colocation patterns from spatial data sets: A general approach," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 12, pp. 1472–1485, Dec. 2004.

[20] J. S. Yoo and S. Shekhar, "A joinless approach for mining spatial colocation patterns," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1323–1337, Oct. 2006.

[21] X. Wang, L. Lei, L. Wang, P. Yang, and H. Chen, "Spatial colocation pattern discovery incorporating fuzzy theory," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 6, pp. 2055–2072, Jun. 2022.

[22] P. Wu, L. Wang, and M. Zou, "A maximal ordered ego-clique based approach for prevalent co-location pattern mining," *Inf. Sci.*, vol. 608, pp. 630–654, 2022.

[23] S. Barua and J. Sander, "Mining statistically significant co-location and segregation patterns," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1185–1199, May 2014.

[24] V. Tran et al., "MCHT: A maximal clique and hash table-based maximal prevalent co-location pattern mining algorithm," *Expert Syst. Appl.*, vol. 175, 2021, Art. no. 114830.

[25] Y. Ge, Z. Yao, and H. Li, "Computing co-location patterns in spatial data with extended objects: A scalable buffer-based approach," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 401–414, Feb. 2021.

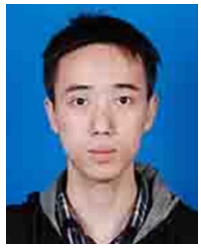
[26] P. Yang, L. Wang, X. Wang, and L. Zhou, "SCPM-CR: A novel method for spatial co-location pattern mining with coupling relation consideration," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5979–5992, Dec. 2022.

[27] K. V. Mardia, "Statistics of directional data," *J. Roy. Stat. Soc.*, vol. 37, no. 3, pp. 349–393, 1975.

[28] F. Wang, "Theorem proving of the coefficient of variation of weibull distribution," *J. Nantong Univ.*, vol. 16, no. 04, pp. 12–14, 2000.

[29] P. Yang et al., "A spatial co-location pattern mining approach based on column calculation," *Scientia Sinica Informationis*, vol. 52, no. 06, pp. 1053–1068, 2022.

[30] J. Li et al., "Fast mining prevalent co-location patterns over dense spatial datasets," in *Proc. 4th Int. Conf. Spatial Data Intell.*, 2023, pp. 179–191.



Junyi Li is currently working toward the postgraduate degree with the School of Information Science and Engineering, Yunnan University. His main research interest is spatial data mining, recommendation systems.

Lizhen Wang (Member, IEEE) received the BS and MSc degrees in computational mathematics from Yunnan University, in 1983 and 1988, respectively, and the PhD degree in computer science from the University of Hundersfield, U.K., in 2008. She is a professor, Ph.D supervisor with the School of Computer Science and Engineering, Yunnan University. She presided more than six projects of the National Natural Science Foundation of China. Her research interests include spatial data mining, interactive data mining, Big Data analytics, and their applications.

Peizhong Yang received the BS and PhD degrees in computer science from Yunnan University, in 2016 and 2021, respectively. He is currently a postdoctoral researcher with the School of Information Science and Engineering, Yunnan University. His research interests include spatial data mining and parallel computing.

Lihua Zhou received the BS and MSc degrees in electronics and information system from Yunnan University, in 1989 and 1992, respectively, and the PhD degree in communication and information system from Yunnan University, in 2010. She is a professor, Ph.D supervisor with the Department of Computer Science and Engineering, Yunnan University. Her main research interests include data mining, machine learning and social network analysis.