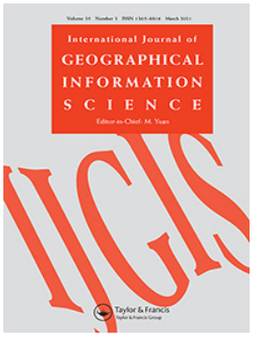

International Journal of Geographical Information Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

An adaptive detection of multilevel co-location patterns based on natural neighborhoods

Qiliang Liu, Wenkai Liu, Min Deng, Jiannan Cai & Yaolin Liu

To cite this article: Qiliang Liu, Wenkai Liu, Min Deng, Jiannan Cai & Yaolin Liu (2021) An adaptive detection of multilevel co-location patterns based on natural neighborhoods, International Journal of Geographical Information Science, 35:3, 556-581, DOI: 10.1080/13658816.2020.1775235

To link to this article: https://doi.org/10.1080/13658816.2020.1775235

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
2021, VOL. 35, NO. 3, 556–581
https://doi.org/10.1080/13658816.2020.1775235

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

An adaptive detection of multilevel co-location patterns based on natural neighborhoods

Qiliang Liu[a,b], Wenkai Liu[b], Min Deng[b], Jiannan Cai[b] and Yaolin Liu[c]

[a]Key Laboratory of Metallogenic Prediction of Nonferrous Metals and Geological Environment Monitoring, Ministry of Education, Central South University, Changsha, Hunan, P.R. China; [b]Department of Geo-informatics, Central South University, Changsha, Hunan, P.R. China; [c]School of Resource and Environmental Science, Wuhan University, Wuhan, P.R. China

ABSTRACT

Multilevel co-location patterns embedded in spatial datasets are difficult to discern due to the complexity of neighboring relationships among spatial features. The neighboring relationships are used to determine whether instances of different spatial features are located in close geographic proximity. When spatial features are distributed unevenly, the neighboring relationships among spatial features cannot be constructed appropriately. Correspondingly, the instances of co-location patterns cannot be generated correctly, and the prevalence of multilevel co-location patterns cannot be measured accurately. To overcome this challenge, this study develops a method to adaptively detect multilevel co-location patterns based on natural neighborhoods. First, locally adaptive neighboring relationships for instances of different spatial features, called 'natural neighborhoods', are defined by considering the formation mechanism of co-location patterns and the local-distribution characteristics of spatial features. Using the natural neighborhoods, we propose a multilevel refining method to identify all global and local co-location patterns algorithmically. We compare the proposed method against three state-of-the-art methods using both simulated and real-life datasets. The comparison shows that the proposed method can discover multilevel co-location patterns from unevenly distributed spatial features more completely and accurately with less *a priori* knowledge for the construction of the natural neighborhoods.

ARTICLE HISTORY
Received 16 December 2019
Accepted 24 May 2020

KEYWORDS
Co-location pattern; natural neighborhood; multilevel patterns; adaptive detection

1. Introduction

Co-location pattern mining is one of the main components of spatial data mining (Guo and Mennis 2009). Given (1) a set of k spatial features, $\boldsymbol{F} = \{f_1, f_2, \ldots, f_k\}$, and their instances $\boldsymbol{I} = \{\boldsymbol{I}(f_1), \boldsymbol{I}(f_2), \ldots, \boldsymbol{I}(f_k)\}$, where each instance of f_i in $\boldsymbol{I}(f_i)$ is a vector < instance ID, feature type, location >, and (2) the neighboring relationships $\boldsymbol{R}$ among different spatial feature instances in $\boldsymbol{I}$, a co-location pattern $\boldsymbol{C}$ is a subset of spatial features ($\boldsymbol{C} \subseteq \boldsymbol{F}$) whose instances form a clique using $\boldsymbol{R}$, and an instance of a co-location pattern is a set of spatial feature instances that includes instances of all

CONTACT Min Deng dengmin@csu.edu.cn
Supplemental data for this article can be accessed here.

features in ***C*** and forms a clique based on ***R*** (Huang *et al*. 2004, Yoo and Shekhar 2006). The problem of co-location pattern detection pairs co-location patterns and their localities such that the co-location patterns are prevalent inside the paired localities (Mohan *et al*. 2011, Li and Shekhar 2018). Owing to spatial heterogeneity, co-location patterns usually exist at multiple levels: some co-location patterns appear globally in the entire study area (global co-location patterns), while others only exist in the local regions of the study area (local co-location patterns) (Celik *et al*. 2007, Ding *et al*. 2011). The discovery of such multilevel co-location patterns will provide new insights on the interaction among the different spatial phenomena, and it has wide applicability in geographic domains. For example, in ecology, discovering the symbiotic relationships among plants of different species, ages, and sizes is crucial for an understanding of the dynamics and specific structures of an ecosystem (Goreaud and Pélissier 2003). In criminology, co-location patterns that are formed by criminal events and socio-economic factors are important for developing effective policing strategies to reduce crime in the local regions (Phillips and Lee 2012). In urban planning, co-location patterns discovered from data on an urban facility (e.g., facility points of interest) allow decision makers to analyze the consistency among the different facilities and optimize urban planning (Yu *et al*. 2017).

Existing methods for the discovery of multilevel co-location patterns usually consist of four steps (Deng *et al*. 2017a, Li and Shekhar 2018): (i) construction of the neighboring relationships ***R*** for instances of different spatial features using a distance threshold, (ii) generation of instances of candidate co-location patterns based on ***R***, (iii) evaluation of the prevalence of global co-location patterns using certain prevalence measures (e.g., participation index), and (iv) identification of local regions where the local co-location patterns are prevalent. Existing studies usually assume that the neighboring relationships are correctly constructed, and more attention has been paid to the following:

(i) To reduce the computational cost of identifying the instances of candidate co-location patterns, some instance filtering and pruning methods were developed, e.g., the partial-join approach (Yoo and Shekhar 2004), joinless approach (Yoo and Shekhar 2006), density-based approach (Xiao *et al*. 2008), order-clique-based approach (Wang *et al*. 2009), and sparse-graph and condensed tree-based approach (Yao *et al*. 2016). The parallel computing technique has also been used to improve the computational performance of instance-connecting operations (Yang *et al*. 2019, Yoo *et al*. 2019).

(ii) To evaluate the prevalence of co-location patterns accurately and objectively, the distance decay effects were considered to define the prevalence measure of co-location patterns (Yao *et al*. 2017, Yu *et al*. 2017), and statistical tests were conducted to estimate their significance (Barua and Sander 2014, Deng *et al*. 2017b, Cai *et al*. 2019). Recently, a visualization method was developed to evaluate the prevalence of size-2 co-location patterns based on human color perception (Zhou *et al*. 2019).

(iii) To detect local co-location patterns, data-unaware space-partitioning heuristics (e.g., Quadtree) were first developed (Celik *et al*. 2007). However, they neglect the spatial distribution of data and might break up potential local co-location patterns. Consequently, clustering-based methods were designed to identify local co-

location patterns by grouping spatial objects or co-location instances (Ding *et al*. 2011, Mohan *et al*. 2011, Wang *et al*. 2013, Deng *et al*. 2017a, Cai *et al*. 2018). Li and Shekhar (2018) found that clustering-based methods may neglect some true local co-location patterns without objects or co-location instance concentrations and suggested enumeration of minimum orthogonal bounding rectangles with different sizes to identify all possible local co-location patterns in the rectangular regions.

(iv) When the densities of the spatial features are even, some existing multilevel co-location mining methods may perform well. However, in practice, spatial features are usually distributed unevenly. In this situation, a unique distance threshold is incapable of constructing appropriate neighboring relationships in the regions with different densities. Correspondingly, the instances of co-location patterns cannot be generated correctly, and the prevalence of multilevel co-location patterns cannot be measured accurately. As a result, some multilevel co-location patterns are very likely to be omitted and/or misjudged. For example, the densities of different spatial features are varied in Figure 1. {*A, B*} is a global co-location pattern, {*A, B, C*} and {*A, B, D*} are local co-location patterns existing in Regions I and II, respectively. Inappropriate neighboring relationships among the instances of different spatial features may seriously affect the mining results in two aspects:

(i) When a small distance threshold is used, many instances of prevalent co-location patterns may be ignored in sparse regions (e.g., Region I). Therefore, the multilevel co-location patterns will be underestimated.

(ii) When a large distance threshold is used, irrelevant instances of candidate co-location patterns may be erroneously constructed in dense regions (e.g., Region II). Consequently, the multilevel co-location patterns will be overestimated.

Based on the above analysis, we can find that the definition of appropriate neighboring relationships for instances of different spatial features is indeed a bottleneck problem in

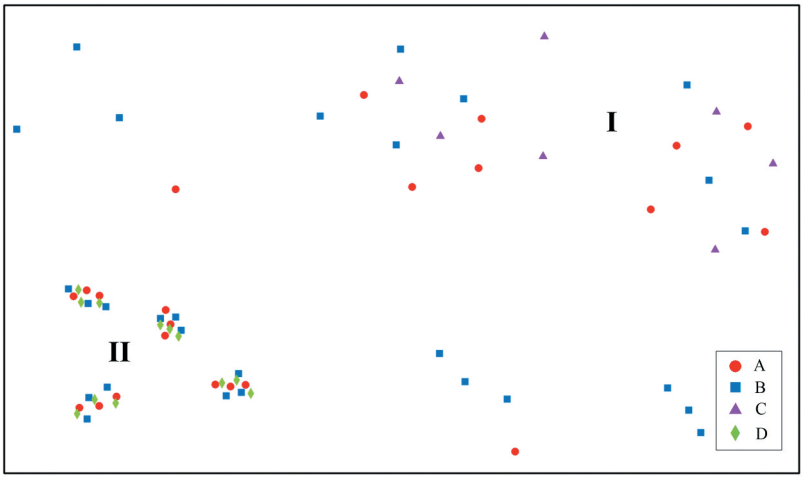

Figure 1. Simulated dataset with unevenly distributed spatial features.

multilevel co-location mining. Scholars have carried out some exploratory work in adaptively constructing neighboring relationships for instances of different spatial features. Yoo and Bow (2012) developed the nearest-neighbor-based and spatial autocorrelation-based approaches to estimate appropriate distance thresholds; however, they only generated a globally fixed distance threshold. To define neighboring relationships using locally adaptive distance thresholds, some scholars adopted a *k*-nearest neighbor graph instead of a fixed distance threshold (Wan and Zhou 2008, Qian *et al*. 2014). Although the neighboring relationships constructed using the *k*-nearest neighbor graph are more adaptive than that constructed using a fixed distance threshold, the determination of an appropriate parameter *k* is difficult (İnkaya *et al*. 2015, Yao *et al*. 2018). To avoid setting the parameter *k*, the Delaunay triangulation network and Voronoi diagram were also used to construct the neighboring relationships based on the topological relationship among the instances of spatial features (Bembenikr and Rybiński 2009, Sundaram and Thnagavelu 2015, Yao *et al*. 2018). However, these Delaunay- or Voronoi-based approaches only identify the first-order neighbors for each spatial feature instance; therefore, some instances of candidate co-location patterns formed by high-order neighbors may be missed (Bembenikr and Rybiński 2009). Moreover, the Delaunay- or Voronoi-based neighboring relationships are usually inappropriate for outliers and spatial feature instances on the border of high-density regions (Estivill-Castro and Lee 2002). Although the multilevel constrained Delaunay triangulation (Shi *et al*. 2016) can identify outliers, the neighboring relationships among instances of spatial features are very likely to be wrongly destroyed.

In summary, the existing methods for constructing neighboring relationships are mainly dependent on user-specified parameters (e.g., the distance threshold or number of nearest neighbors) or the spatial relationship (e.g., the topological relationship). Defining appropriate neighboring relationships for unevenly distributed instances of different spatial features remains a major challenge. Thus, there is a high probability that multilevel co-location patterns are being omitted and/or misjudged. To address this issue, an adaptive method for detecting multilevel co-location patterns based on natural neighborhoods is proposed in this study. The contribution of this work includes three aspects:

(i) We define the natural neighborhoods that can adaptively construct neighboring relationships among unevenly distributed spatial features with less *a priori* knowledge.
(ii) We develop a multilevel refining method based on the natural neighborhoods to automatically discover all the global and local co-location patterns from unevenly distributed spatial features.
(iii) The experimental evaluations show that the multilevel co-location patterns discovered by the proposed method are almost complete and that they have better accuracy than those discovered by the three state-of-the-art methods.

The rest of this paper is organized as follows: Section 2 introduces a new strategy for discovering multilevel co-location patterns based on the natural neighborhoods. Section 3 describes the method of constructing the natural neighborhoods, and Section 4 introduces a multilevel refining method for discovering both global and local co-

location patterns. Section 5 describes the implementation of the proposed method. Section 6 presents the experimental evaluations using both simulated and real-world datasets. Section 7 offers conclusions and outlines future work.

2. Adaptive discovery of multilevel co-location patterns: a new strategy based on natural neighborhoods

To discover multilevel co-location patterns, it is vital to adaptively construct the neighboring relationships for instances of different spatial features. In the study, we argue that the construction of neighboring relationships should be guided by the formation mechanism of co-location patterns.

Co-location patterns can be interpreted by 'induced spatial autocorrelations,' where the spatial autocorrelation of each spatial feature is 'induced' by a spatially autocorrelated underlying variable (Fortin and Dale 2005, Cai *et al*. 2019). Hence, it is meaningless to report a co-location pattern with a high prevalence measure value due to the randomly distributed spatial features (Barua and Sander 2014). Therefore, before constructing the neighboring relationships, randomly distributed spatial features should first be identified and removed. In fact, if two instances of different spatial features are spatial neighbors, the local autocorrelation structures around these two instances would be closely interacting with each other. Based on this idea, to adaptively construct the neighboring relationships for instances of different spatial features, we need to first estimate the range around each spatial feature instance, wherein the local spatial autocorrelation is obvious.

To estimate the range of local spatial autocorrelation around each spatial feature instance, two principles are formulated in this study, i.e., geographic proximity and compactness of connectivity. For the principle of geographic proximity, statistics derived from randomly permutated data are used to estimate the upper bound of the range where the local autocorrelation is obvious. For the principle of compactness of connectivity, the density breakpoint detection method and shared neighbor test are proposed to identify the compact neighbors for each spatial feature instance.

For two instances of different spatial features, if the ranges of local spatial autocorrelation of these two instances are highly overlapped; then, these two instances will be identified as spatial neighbors. In this study, we name this type of spatial neighborhood for instances of different spatial features as 'natural neighborhood,' because it is defined based on the inherent characteristics of the co-location patterns and spatial data, i.e., the formation mechanism of co-location patterns and local distribution of spatial features.

After constructing the natural neighborhood for each spatial feature instance, the instances of candidate multilevel co-location patterns can be generated accurately. The prevalence of multilevel co-location patterns can be measured by certain prevalence measures (e.g., participation index or its variants). A multilevel refining method is further developed to detect all the global and local co-location patterns. Given a prevalence threshold, the global co-location patterns are first identified. Then, non-prevalent co-location patterns at the global level are identified as candidate local co-location patterns. To identify all the local co-location patterns in unknown and arbitrarily shaped subregions, the proximity relationship of the instances of each candidate local co-location pattern is modeled using the Delaunay triangulation network. To reduce the error in the

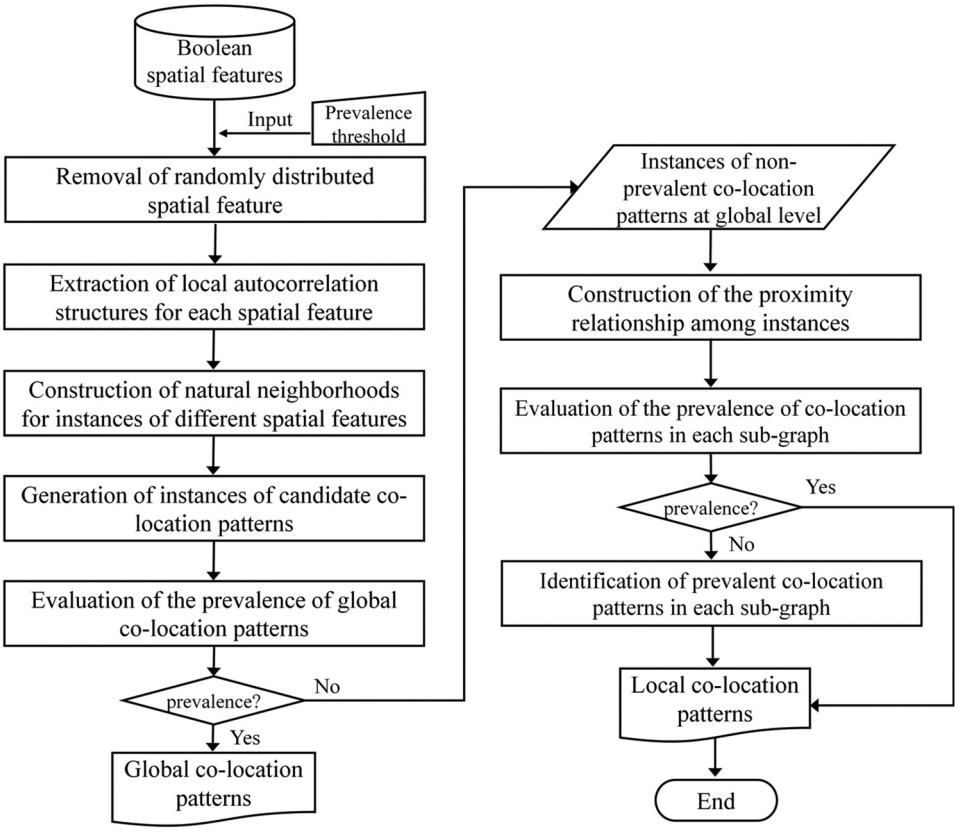

Figure 2. Framework of the natural neighborhoods–based multilevel co-location mining method.

Delaunay triangulation network-based proximity relationship, an edge removal criterion is used to eliminate the abnormally long edges from the Delaunay triangulation network (Estivill-Castro and Lee 2002). The Delaunay triangulation network will be segmented into one or more sub-graphs. The boundary of each sub-graph is delineated using the α-shape method (Edelsbrunner *et al*. 1983), and the prevalence of each candidate local co-location pattern is evaluated in each sub-graph. If a candidate local co-location pattern is prevalent in a sub-graph; then, it will be identified as a local co-location pattern. Otherwise, a multi-direction optimization method is proposed to check whether there are local co-location patterns in this sub-graph.

In Figure 2, the framework of the natural neighborhoods–based multilevel co-location pattern mining method is shown. In the following section, the construction of the natural neighborhood will be presented.

3. Natural neighborhoods: adaptive construction of neighboring relationships among instances of different feature types

Based on the framework shown in Figure 2, to construct the natural neighborhoods, we must first identify and remove the randomly distributed spatial features. In this study, we

use the nearest neighbor index to determine the complete spatial randomness of a spatial feature (Clark and Evans 1954). Then, the local autocorrelation structures of each spatial feature are extracted based on two principles, namely, geographic proximity and compactness of connectivity.

3.1 Robust statistic derived from the permutated dataset for estimating the upper bound of the range of local spatial autocorrelation

It is known that spatial autocorrelation does not exist when the instances of certain spatial features are randomly distributed. Based on this idea, the same number of instances of a spatial feature is randomly permutated in the same study area, and a Delaunay triangulation network is constructed for this randomly distributed dataset. The edges in the Delaunay triangulation network can be used to estimate the upper bound of local autocorrelation. In this study, a robust statistic is constructed to identify the upper bound (D_{UB}):

$$D_{UB} = Median(E) + MAD(E) \tag{1}$$

$$MAD(E) = Median(|e_i - Median(E)|) \tag{2}$$

where $\boldsymbol{E}$ is the set of edges in the Delaunay triangulation network, $Median(\boldsymbol{E})$ is the median of the lengths of edges in $\boldsymbol{E}$, $MAD(\boldsymbol{E})$ is the median absolute deviation of the lengths of edges in $\boldsymbol{E}$, and e_i is the length of the ith edge in $\boldsymbol{E}$.

Considering spatial feature A in Figure 1 as an example, the Delaunay triangulation network constructed for the randomly distributed dataset is illustrated in Figure 3(a), and the estimated upper bound (D_{UB}) is shown in Figure 3(b). For an instance A_i of the spatial feature A, the candidate local autocorrelated neighbors $\boldsymbol{CAN}(A_i)$ of A_i are defined as follows: $\boldsymbol{CAN}(A_i) = \{A_j \square d(A_i, A_j) \leq D_{UB}\}$, where, $d(A_i, A_j)$ is the distance between A_i and A_j. The upper bound only roughly estimates the range of local spatial autocorrelation around each spatial feature instance. For example, in Figure 3(b), there are obvious gaps between instances of feature A in the bottom left. Therefore, we need to further identify compact autocorrelated neighbors for each instance based on the principle of compactness of connectivity.

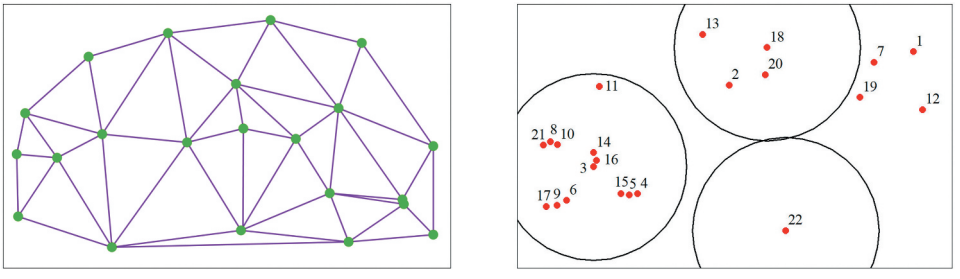

Figure 3. Estimation of upper bound of local autocorrelation: (a) Delaunay triangulation network constructed for randomly permuted data; (b) the estimated upper bound.

3.2 Density breakpoint detection and shared neighbor test for determining local autocorrelated neighbors

To introduce the density breakpoint detection method, we first define the density between two instances A_i and A_j of feature type A as the number of instances of feature A located in the circle passing through A_i and A_j with diameter $d(A_i, A_j)$. The instances of feature A located in that circle can be identified as $\boldsymbol{I}_{ij} = \{A_k \square d(A_k, O_{ij}) \leq d(A_i, A_j)/2\}$ where O_{ij} is the midpoint of line segment A_iA_j. For an instance A_i, all other instances in $\boldsymbol{CAN}(A_i)$ are listed in non-decreasing order of their distance to A_i denoted as $\boldsymbol{CAN}_{order}(A_i)$. Taking A_3 and A_{18} in Figure 3(b) as examples, $\boldsymbol{CAN}_{order}(A_3) = \{A_{16}, A_{14}, A_{15}, A_{10}, A_6, A_5, A_8, A_4, A_9, A_{21}, A_{17}, A_{11}\}$ and $\boldsymbol{CAN}_{order}(A_{18}) = \{A_{20}, A_2, A_{13}\}$. If the distance from two or more instances in $\boldsymbol{CAN}(A_i)$ to A_i is equal; then, these instances will be listed in a non-decreasing order according to the density between these instances and A_i. When the density between A_i and each instance in $\boldsymbol{CAN}_{order}(A_i)$ is calculated one by one, the density is not expected to decrease for the true local autocorrelated neighbors (İnkaya *et al*. 2015). If the density between A_i and an instance A_j decreases, it may indicate that a region with a different density begins. Then, A_j will be identified as a density breakpoint and the instances before A_j will be identified as the density-connected neighbors of A_i and denoted by $\boldsymbol{DCN}(A_i)$.

In Figure 3(b), the densities between A_3 and instances in $\boldsymbol{CAN}_{order}(A_3)$ are 0, 1, 0, 1, 0, 1, 2, 2, 1, 3, 2, and 2, and the densities between A_{18} and instances in $\boldsymbol{CAN}_{order}(A_{18})$ are 0, 1, and 0. The density change point in $\boldsymbol{CAN}_{order}(A_3)$ is A_{15}, and the density change point in $\boldsymbol{CAN}_{order}(A_{18})$ is A_{13}. Therefore, $\boldsymbol{DCN}(A_3) = \{A_{16}, A_{14}\}$ and $\boldsymbol{DCN}(A_{18}) = \{A_{20}, A_2\}$. For A_3, the change point A_{15} indeed indicates the beginning of a different density region (depicted in Figure 4(a)). However, for A_{18}, A_{13} is identified as a change point only because A_{13} is located on the border of the region. To avoid the false discovery of a change point such as A_{13}, we further check whether the density-connected neighbors of A_i and the change point A_j in $\boldsymbol{DCN}(Ai)$ share the same neighbor. If $\boldsymbol{DCN}(A_i) \cap \boldsymbol{DCN}(A_j) \neq \phi$, A_j will be added in $\boldsymbol{DCN}(A_i)$, and we need to further check whether the next change point is a true change point or a border point. For example, $\boldsymbol{DCN}(A_{15}) = \{A_5, A_4\}$, $\boldsymbol{DCN}(A_{13}) = \{A_{18}, A_2, A_{20}\}$, $\boldsymbol{DCN}(A_3) \cap \boldsymbol{DCN}(A_{15}) = \phi$, and $\boldsymbol{DCN}(A_{18}) \cap \boldsymbol{DCN}(A_{13}) \neq \phi$; therefore, A_{15} is a true change point and A_{13} is a border point. Finally, $\boldsymbol{DCN}(A_3) = \{A_{16}, A_{14}\}$ and $\boldsymbol{DCN}(A_{18}) = \{A_{20}, A_2, A_{13}\}$.

After identifying the density-connected neighbors for each spatial feature instance, a shared neighbor test is used to refine the density-connected neighbors to obtain local autocorrelated neighbors. The local autocorrelated neighbors $\boldsymbol{LAN}(A_i)$ of an instance A_i are

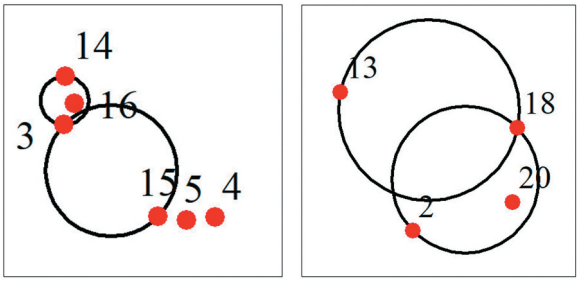

Figure 4. Illustration of different types of density break points: (a) A true density break point A_{15}; (b) A false density break point A_{13}.

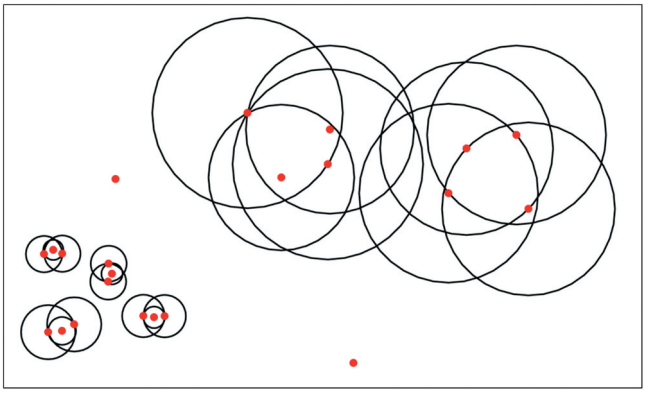

Figure 5. The range of local spatial autocorrelation for each instance of spatial feature *A*.

denoted as $\boldsymbol{LAN}(A_i) = \{A_j \square A_j \in \boldsymbol{DCN}(A_i)$ and $A_i \in \boldsymbol{DCN}(A_j)$, $\boldsymbol{DCN}(A_i) \cap \boldsymbol{DCN}(A_j) \neq \phi\}$. For a spatial feature instance A_i, the range of local spatial autocorrelation ($R(A_i)$) is defined as a circle centered at A_i with radius $d(A_i, A_k)$, where A_k is the farthest neighbor in $\boldsymbol{LAN}(A_i)$. Figure 5 presents the range of local spatial autocorrelation for each instance of spatial feature *A*.

3.3 Construction of the natural neighborhoods

According to the strategy of constructing the natural neighborhoods introduced in Section 2, we can infer that if the spatial autocorrelation of a spatial feature *A* is truly 'induced' by a spatially autocorrelated feature *B*, the ranges of the local spatial autocorrelation of most of the instances of *A* and *B* should highly overlap. Based on this idea, for two instances of different spatial features A_i and B_j, if A_i is located in $R(B_j)$ and B_j *is* located in $R(A_i)$; then, A_i and B_j can be identified as natural neighbors. Therefore, the natural neighborhood of an instance of feature *A* can be defined as

$$NN(A_i) = \{I_j \big| I_j located in R(A_i) and A_i located in R(I_j), I \neq A\} \tag{3}$$

The instances of candidate co-location patterns can be constructed based on the natural neighborhoods. In this study, the co-location pattern refers to the 'clique pattern' (Wang *et al*. 2005). For an instance of a candidate co-location pattern, every two instances of different spatial features should be natural neighbors. The joinless approach (Yoo and Shekhar 2006) was used to construct the instances of all possible co-location patterns (size-2 to size-*k*), where *k* is the number of spatial features. In Figure 6, the neighboring relationships among instances of four features constructed by the natural neighborhoods are shown. One can observe that the instances of candidate co-location patterns can be generated accurately in regions with different densities. Moreover, it can also be seen that although two instances of features *A* and *B* (in red circle) are close to each other, they do not form an instance of {*A, B*}. This is because the instance of *A* is an outlier; therefore, there is no local autocorrelation structure around this instance. Indeed, the natural neighborhoods defined in this study can also avoid an incorrect construction of instances of

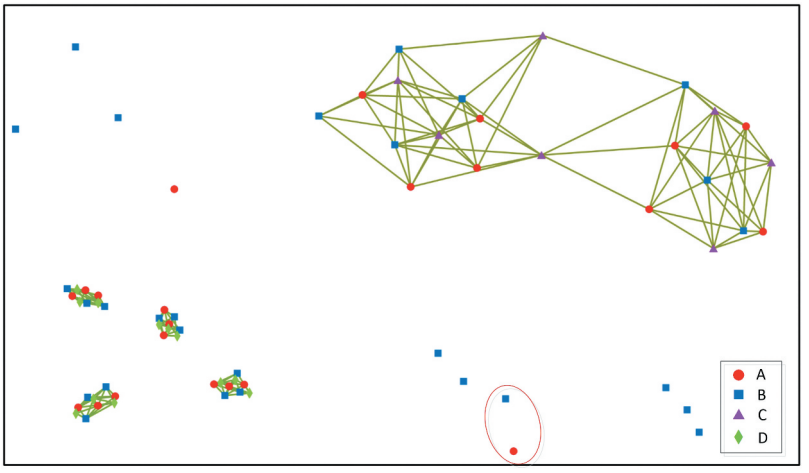

Figure 6. Instances of candidate co-location patterns generated based on natural neighborhoods.

candidate co-location patterns due to the randomly distributed spatial features in local regions.

4. Multilevel refining method for detecting global and local co-location patterns based on natural neighborhood

After the instances of all candidate co-location patterns are constructed accurately based on the natural neighborhoods, a multilevel refining method is further developed to detect both global and local co-location patterns.

4.1 Discovery of global co-location patterns

First, we check whether each candidate co-location pattern is a global co-location pattern. In this study, the participation index (*PI*) (Huang *et al*. 2004) is used as the prevalence measure:

$$PI(C_i) = min_{f_j \in C_i}\{Pr(C_i, f_i)\} \tag{4}$$

$$Pr(C_i, f_j) = |I(C_i, f_j)|/|I(f_j)| \tag{5}$$

where $|\boldsymbol{I}(\boldsymbol{C}_i, f_j)|$ is the number of unique instances of feature f_j in instances of $\boldsymbol{C}_i$, and $|\boldsymbol{I}(f_j)|$ is the number of instances of feature f_j in the entire study area or a sub-region. Given a threshold T, if $PI(\boldsymbol{C}_i) \geq T$; then, $\boldsymbol{C}_i$ will be identified as a global co-location pattern; otherwise, $\boldsymbol{C}_i$ will be classified as a candidate local co-location pattern. Taking the simulated dataset in Figure 1 as an example, if T is set as 0.5; then, {*A, B*} (PI = 0.68) will be identified as a global co-location pattern, whereas {*A, B, C*}(PI = 0.25) and {*A, B, D*} (PI = 0.42) will be categorized as candidate local co-location patterns.

The global co-location patterns also can be identified by using a significance test. Interested readers can refer to the supplement document for details. The significance test

can avoid the discovery of false co-location patterns formed by randomly distributed spatial features (Barua and Sander 2014). In this study, these false co-location patterns will not be generated because randomly distributed spatial features were removed before mining the multi-level co-location patterns. Moreover, the Monte Carlo method for calculating the *p*-values of co-location patterns is indeed time consuming and it is not practical for large datasets. Therefore, we use the prevalence threshold *T* to identify the multi-level co-location patterns in this study.

4.2 Discovery of local co-location patterns

After identifying global co-location patterns, we further check whether each candidate local co-location pattern is a true local co-location pattern. To completely identify the localities of a candidate co-location pattern C_i without a brute-force search, a trimmed Delaunay triangulation network is first used to roughly construct the potential localities of C_i. The Delaunay triangulation network is constructed for all the instances of the features in C_i. A percentile estimation formula developed for skewed distributions (Zhou *et al*. 2011) is used to remove some extremely long edges from the Delaunay triangulation network:

$$P_m = \frac{m - 0.54\,exp(-0.87\lambda) + 0.05}{N - 0.35\,exp(-0.55\lambda) + 0.85} \tag{6}$$

where *N* is the number of edges connecting two instances of a candidate local co-location pattern. The edges are ranked in an increasing order of length from the smallest $m = 1$ to the largest $m = N$, where *m* represents the rank of each edge and λ is the skewness index calculated from the edge length data after a Box–Cox transformation using the maximum likelihood method. In this study, we use the 90th percentile ($m = 90$) to remove the extremely long edges. To delineate each locality of C_i, the α-shape method is used to construct the boundary of all the instances of C_i connected by the trimmed Delaunay triangulation network. The α-shape method is widely used for delineating the shape of a finite set of points in the plane. The only parameter of the α-shape method is the radius of the circles that are used to detect edge points. In this study, the radius increases from 0 with an interval of 5 meters until the polygonal shape for all the instances of the features in C_i is disconnected.

In each locality, *PI* is used to evaluate the prevalence of C_i. If $PI(C_i) \geq T$; then, C_i will be identified as a local co-location pattern. In Figure 7(a), the instances of {*A, B, C*} and {*A, B, D*} connected by the trimmed Delaunay triangulation network are illustrated (dashed lines represent the edges connecting two instances of a candidate local co-location pattern). In Figure 7(b), two local co-location patterns {*A, B, C*} ($PI = 1$) and {*A, B, D*} ($PI = 1$) can be discovered. One can see that the trimmed Delaunay triangulation network can detect localities of candidate local co-location patterns (e.g., {*A, B, C*} in Figure 7) without co-location instance concentrations.

Then, if C_i is not prevalent in a locality connected by the trimmed Delaunay triangulation network (for example, in Figure 8(a), *PI* ({*Calam a., Glyce s.*}) = 0.41), we finally develop a multi-direction optimization method to check whether there are potential local co-location patterns in this locality. A Voronoi diagram is constructed by regarding instances of each feature in C_i as kernels (see the example shown in Figure 8(b)). The Voronoi cells of

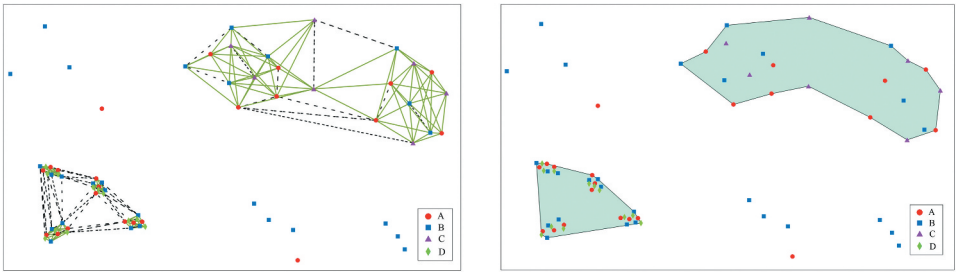

Figure 7. Illustration of discovery of local-colocation patterns: (a) The instances of candidate co-location patterns connected by the trimmed Delaunay triangulation network; (b) Localities of {*A, B, C*} and {*A, B, D*}.

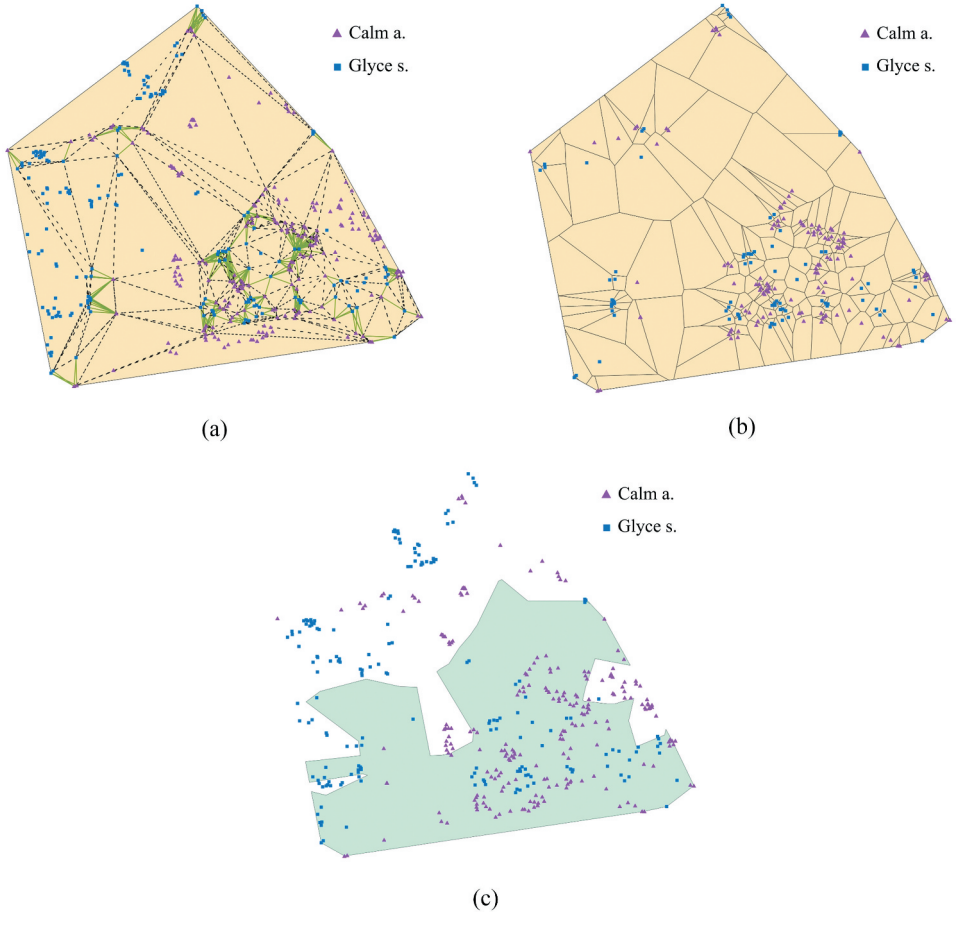

Figure 8. Illustration of the multi-direction optimization method by using two wetland plants: *Calamagrostis angustifolia* (*Calam a.*) and *Glyceria spiculosa* (*Glyce s.*): (a) A sub-graph where {*Calam a., Glyce s.*} is not prevalent; (b) The Voronoi diagram constructed for instances of *Calam a.* and *Glyce s.*; (c) Local co-location pattern {*Calam a., Glyce s.*}.

each instance of C_i are identified as a seed of the locality of a potential local co-location pattern. In each seed, the $PI(C_i)$ is calculated. Two seeds are identified as neighboring seeds if there is the same cell in these two seeds or there are neighboring cells in these two seeds. The multi-direction optimization method starts by selecting an unvisited seed in which the $PI(C_i)$ is larger than the third quartile of the PI values of all the seeds. This seed is regarded as the locality of a potential local co-location pattern. Next, the locality will be combined with a neighboring seed with the highest $PI(C_i)$ at each time to form a new locality, until all the neighboring seeds are combined or the $PI(C_i)$ in a newly built locality is smaller than the threshold T. After each seed has been visited, several overlapping localities may be obtained. Among all the localities where C_i is prevalent, the locality with the maximum area will be identified as the locality of the C_i, and the other localities that are overlapped with this locality are deleted. Figure 8(c) shows the final discovered local co-location pattern {*Calam a., Glyce s.*} (PI = 0.72) from the sub-graph in Figure 8(a).

5. The implementation of the multilevel co-location pattern mining method

The only parameter of the multilevel co-location pattern mining method (***MLMiner***) is the prevalence threshold T. The pseudo-code of the proposed method is described as follows:

Algorithm: MLMiner (Input: *Data, T*)
RemoveRandomFeature (*Data);*
Natural neighborhoods = ConstructNaturalNeighborhoods (*Data);*
CandidatePatterns = GenerateCandidatePatterns (*Data, Natural neighborhoods*);
for each *Pattern* in *CandidatePatterns* **do**
if *Pattern.PI* ≥ *T*
Pattern.Level = Global;
else
Localities = ExtractLocality (*Pattern, T*);
if NotEmpty (*Localities*)
Pattern.Level = Local;
Pattern. Localities = Localities;
end if
end if
end for
end algorithm

Function *RemoveRandomFeature*() removes the randomly distributed spatial features using the nearest neighbor index. Function *ConstructNaturalNeighborhoods*() constructs the natural neighborhood for each spatial feature instance. Function *GenerateCandidatePatterns*() generates the instances of candidate co-location patterns using the joinless approach. Function *ExtractLocality*() checks whether the global non-prevalent co-location patterns are prevalent in the local regions. Next, we first give the pseudo-code of function *ConstructNaturalNeighborhoods*():

function *ConstructNaturalNeighborhoods* (*Data*)
for each *SpatialFeature* in *Data* **do**
Ranges=ConstructRange (*SpatialFeature*);
end for
for each *SpatialFeature* in *Data* **do**
Neighbors =ChangePointDetection (*SpatialFeature, Ranges*);
end for

Natural neighborhoods = *OverlapCheck* (*Data, Neighbors*);
return *Natural neighborhoods*;

In the function *ConstructNaturalNeighborhoods*(), the function *ConstructRange*() calculates the upper bound of the range of the local spatial autocorrelation for each spatial feature instance. Function *ChangePointDetection*() further calculates the range of the local spatial autocorrelation for each spatial feature instance. The function *OverlapCheck*() uses an overlap approach to construct the natural neighborhoods. We further give the pseudocode of function *ExtractLocality* ():

function *ExtractLocality* (*Pattern, T*)
SubGraphs =TrimmedDelaunay (*Pattern);*
RoughLocalities=AlphaShape (*SubGraphs*)
Localities = Empty ();
for each *Locality* in *RoughLocalities* **do**
if *Pattern.PI*≥ *T* in *Locality*
Localities.Append (*Locality*);
else
RefinedLocalities= *MultiDirectionOptimization* (*Locality, T*);
if *NotEmpty* (*RefinedLocalities*)
Localities.Append(*RefinedLocalities*);
end if
end if
end for
return *Localities*;

In function *ExtractLocality*(), the function *TrimmedDelaunay*() connects the instances of the candidate local co-location pattern using the trimmed Delaunay triangulation network. Function *AlphaShape*() constructs the boundaries for the localities of the candidate local co-location patterns. If a candidate co-location pattern is not prevalent in a locality, the function *MultiDirectionOptimization*() will finally check whether there are potential local co-location patterns in this locality:

function *MultiDirectionOptimization* (*Locality, T*)
RefinedLocalities = Empty ();
[*Proximity, Seeds*]= *Voronoi* (*Locality.Instances);*
PISeed=CalculatePI (*Seeds);*
Seeds=ThirdQuartile (*Seeds, PISeed*);
for each *seed* in *Seeds* **do**
RefinedLocality = ExpandOptimization (*Seed, Proximity, T);*
RefinedLocalities.Append (*RefinedLocality*);
end for
RemoveOverlap (*RefinedLocalities*);
return *RefinedLocalities*;

In function *MultiDirectionOptimization*(), the function *Voronoi*() connects the instances of the candidate local co-location pattern in a locality to form the seeds of the locality of that pattern. Function *Voronoi*() also calculates the *PI* value of that pattern in each seed. Function *ExpandOptimization*() constructs the localities of the candidate local co-location pattern from each seed whose *PI* value is larger than that of the third quartile of the *PI* values of all the seeds. For all the localities where the candidate local co-location pattern is

prevalent, the function *RemoveOverlap*() removes the localities overlapped with the locality with the maximum area.

The analysis of the time and space complexities of the proposed method is described here. In this study, we use the *KD*-tree to search the *k*-nearest neighbors. To remove the randomly distributed spatial features, the time complexity does not exceed $O(NlogN)$ and the space complexity is $O(N)$, where N is the number of instances of all the spatial features. The time and space complexity for constructing the natural neighborhood is governed by the *k*-nearest neighbor search using the *KD*-tree; therefore, the time complexity is approximately $O(NlogN)$ (Cormen *et al*. 2009), and the space complexity is $O(NlogN)$. When the joinless approach is used to generate the instances of candidate co-location patterns, the time complexity is approximately $O(NlogN)$ and the space complexity is $O(max\{|C_i|\cdot|\boldsymbol{I}(C_i)|\})$, where $|C_i|$ is the size of the co-location pattern C_i and $|\boldsymbol{I}(C_i)|$ is the number of instances of C_i. The time complexity and space complexity for identifying global co-location patterns can be neglected. When we roughly detect local co-location patterns using the trimmed Delaunay triangulation network, the time complexity is approximately $O(NlogN)$, and the space complexity does not exceed $O(N)$. When we further use the multi-direction optimization method to identify local co-location patterns in the localities identified by the trimmed Delaunay triangulation network (the number of instances of these local co-location patterns is K), the time complexity does not exceed $O(K^2)$, and the space complexity is $O(KlogK)$. Overall, the total time complexity of the proposed method is approximately $O(NlogN+K^2)$ and the total space is $O(max\{|C_i|\cdot|\boldsymbol{I}(C_i)|\}+N+KlogK)$. One can see that the multi-direction optimization operation is the most time-consuming part of the proposed method. However, the value of K is usually small in practice; therefore, the proposed method can be applied to large datasets.

6. Experimental evaluation

6.1 Experiments on simulated datasets

In this section, the performance of the proposed method is evaluated using simulated datasets. The proposed method is compared with three state-of-the-art methods: neighborhood graph-based method (denoted by NG) (Mohan *et al*. 2011), *k*-nearest neighbor graph-based method (denoted by KNNG) (Qian *et al*. 2014), and multilevel method (denoted by ML) (Deng *et al*. 2017a). All methods were implemented in Python on a workstation with a 2.20 GHz CPU and 64 GB memory running the Microsoft Windows 10 operating system.

6.1.1 Parameter setting and evaluation measure

To avoid discovering useless small localities of local co-location patterns, for a local co-location pattern $\boldsymbol{C}_k$, if there is a feature $f_i \in \boldsymbol{C}_k$ that satisfies the condition $I(C_k, f_j)/I(f_j) \leq 0.02$, where $I(f_j)$ is the number of instances of feature f_j in the entire study area; then, $\boldsymbol{C}_k$ will be removed from the mining results. For the proposed method, the prevalence threshold is set as 0.5. The parameters of the three state-of-the-art methods were set based on the suggestions given in the original studies:

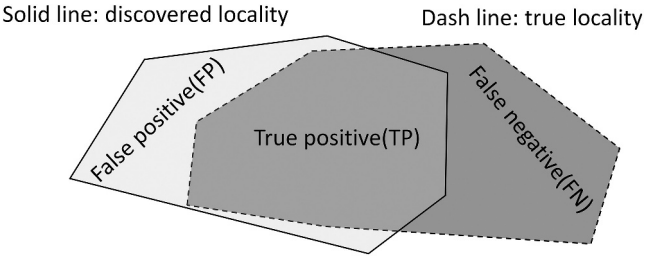

Figure 9. Illustration of the measures for assessing the quality of discovered localities.

- For the NG method, the regional prevalence threshold is set as 0.01 as suggested in the original paper. The distance threshold is estimated using the *L*-function developed by Yoo and Bow (2012).
- For the KNNG method, the prevalence threshold is set as 0.5, and the distance variation threshold is set as 0.6 based on the experimental analysis performed in the original study.
- For the ML method, the prevalence threshold is also set as 0.5, and the distance threshold is also estimated using the *L*-function.

The measure developed by Guo and Wang (2011) was used to evaluate the localities of the multilevel co-location patterns discovered by the four methods. In Figure 9, the dashed-line polygon represents a predefined locality of a co-location pattern, and the solid line polygon denotes the locality discovered by a co-location mining method. For a dataset with n predefined localities of co-location patterns, the precision and recall are defined as follows:

$$Precision = \frac{\sum_{i=1}^{n} TP/(TP_i + FP_i)}{n} \tag{7}$$

$$Recall = \frac{\sum_{i=1}^{n} TP/(TP_i + FN_i)}{n} \tag{8}$$

6.1.2 Generation of simulated data

To quantitatively evaluate the performance of the four methods, a series of simulated datasets were generated in the following five steps:

Step 1: A 200 × 200 study area is divided into four 100 × 100 sub-regions. In each sub-region, n_{parent} parent points are randomly located in each sub-region (Figure 10(a)).

Step 2: For each sub-region, in the buffer of radius r_{parent} centered at each parent point, n_{child} child points are generated around that parent point. Then, all parent points are removed (Figure 10(b)).

Step 3: In the buffer of r_{child} centered at each child point, $n_{grandchild}$ grandchild points of $n_{feature}$ different feature types are randomly generated around that child point. Then, all the child points are removed (Figure 10(c)).

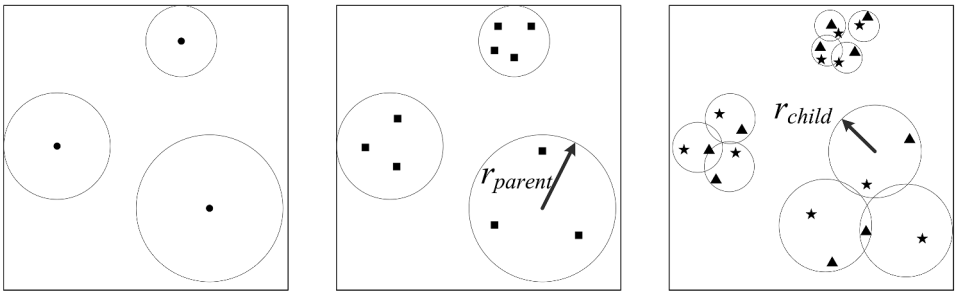

Figure 10. Generation of co-location patterns in a sub-region: (a) Generation of parent points; (b) Generation of child points; (c) Generation of grandchild points.

Step 4: By controlling the feature types of grandchild points, global co-location patterns exist in all the four sub-regions; however, local co-location patterns only exist in two sub-regions.

Step 5: n_{noise} noise points are randomly added to the entire study area (the feature types of noise are also randomly selected).

In the simulated data generator, r_{parent} and n_{child} control the densities of instances of spatial features. r_{child} controls the interaction distance among instances of different features. $n_{feature}$ controls the length of co-location patterns. In sub-regions where local co-location patterns exist, n_{parent} controls the number of localities of local co-location patterns. In this study, we first fix the values of n_{parent} and n_{child} in different sub-regions: Region I ($n_{parent} = 2$, $n_{child} = 40$), Region II ($n_{parent} = 3$, $n_{child} = 40$), Region III ($n_{parent} = 2$, $n_{child} = 40$), and Region IV ($n_{parent} = 2$, $n_{child} = 40$). Then, datasets with different densities of spatial feature instances, types of spatial features, and amount of noise were generated by controlling the parameters r_{parent}, $n_{feature}$, and n_{noise}.

6.1.3 Experimental results

6.1.3.1. Effect of densities of spatial features. To test the effect of spatial feature density on the discovery of multilevel co-location patterns, we first fix $n_{noise} = 50$ and $n_{feature} = 3$ in Regions I and III, and $n_{feature} = 2$ in Regions II and IV. Then, we randomly generate the value of r_{parent} and r_{child} in the study area ($r_{parent} \in [15, 25]$ in Region I, $r_{parent} \in [20, 30]$ in Region II, $r_{parent} \in [25, 35]$ in Region III, and $r_{parent} \in [15, 30]$ in Region IV; $r_{child} \in [3, 4.5]$). Finally, 10 datasets with spatial features of different densities can be obtained. In Figure 11, the number, precision, and recall

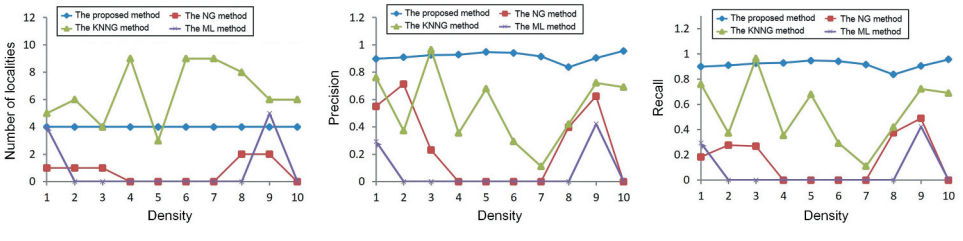

Figure 11. The effect of spatial feature density on mining multilevel co-location patterns: (a) The number of discovered localities; (b) Precision; (c) Recall.

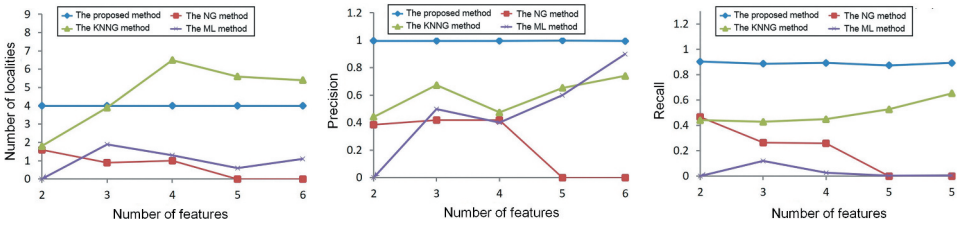

Figure 12. The effect of the number of spatial feature types on mining multilevel co-location patterns: (a) The number of discovered localities; (b) Precision; (c) Recall.

values of the detected localities of co-location patterns are shown. One can observe that multilevel co-location patterns can be discovered by the proposed method correctly and completely (the values of precision and recall usually exceed 0.9). The KNNG method usually incorrectly identifies the localities of co-location patterns, whereas the NG and ML method usually neglect some predefined co-location patterns. The reason may lie in two aspects: (i) the neighboring relationships constructed using the *k*-nearest neighbor graph overestimate the instances of candidate co-location patterns, and (ii) the neighboring relationships constructed using the global fixed distance underestimate the instances of candidate co-location patterns.

6.1.3.2. Effect of number of spatial feature types. To test the effect of the number of spatial feature types on mining multilevel co-location patterns, we first fix $n_{noise} = 50$ and generate different values of r_{parent} and r_{child} in different regions (in Region I, $r_{parent} = 20$, $r_{child} = 3$; in Region II, $r_{parent} = 25$, $r_{child} = 3.5$; in Region III, $r_{parent} = 30$, $r_{child} = 4$; in Region IV, $r_{parent} = 35$, $r_{child} = 4.5$). In Figure 12, one can observe that the proposed method remained effective with an increasing number of feature types. Because the densities of the spatial features are uneven, some co-location patterns are usually overestimated or underestimated by the other three methods.

6.1.3.3. Effect of amount of noise. To test the effect of amount of noise on the mining results, we generate different values of r_{parent}, r_{child}, and $n_{feature}$ in different regions (in Region I, $r_{parent} = 20$, $r_{child} = 3$, $n_{feature} = 3$; in Region II, $r_{parent} = 25$, $r_{child} = 3.5$, $n_{feature} = 2$; in Region III, $r_{parent} = 30$, $r_{child} = 4$, $n_{feature} = 3$; and in Region IV, $r_{parent} = 35$, $r_{child} = 4.5$, $n_{feature} = 2$). In Figure 13, it can be observed that the performance of the proposed method is not

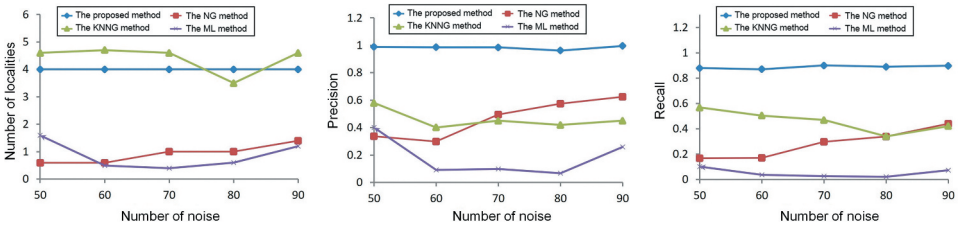

Figure 13. The effect of amount of noise on mining multi-level co-location patterns: (a) The number of discovered localities; (b) Precision; (c) Recall.

sensitive to the amount of noise. Although the other three methods are also not seriously affected by the amount of noise, they cannot detect predefined co-location patterns completely and accurately.

6.1.3.4. Efficiency and scalability of the proposed and three comparison methods.

We generate ten datasets to evaluate the efficiency and scalability of each of the four methods. The number of spatial feature instances in these datasets ranges from 3 K to 50 K. In each dataset, there are one global co-location pattern and one local co-location pattern with five localities formed by three unevenly distributed spatial features. The running time of the three methods on the datasets of different sizes is displayed in Figure 14. The KNNG method is very inefficient (it takes about eight hours to process the dataset with 15 K feature instances); therefore, we do not show the running time of the KNNG. One can see that the proposed method is more efficient than the NG and ML method and it is less sensitive to the size of datasets. By contract, the execution time of the ML method dramatically increases as the size of datasets increases. The running time of different parts of the proposed method is shown in Figure 14(b). It can be found that the multi-direction optimization approach is practical for large datasets.

6.2 Case study: discovery of multilevel symbiotic relationships among wetland plants

The proposed method was further applied for detecting multilevel symbiotic relationships among wetland plants in the Honghe wetland located in Northeast China in the year 2012. In wetland ecology, the growing environment, community composition, and species abundance usually vary in different locations; therefore, the interspecies relationships among different plants are significantly complex and diverse (Zimmer *et al*. 2003, Keddy 2010). The discovery of multilevel symbiosis among wetland plants is vital for understanding the community structure, protecting the ecosystem diversity, and maintaining the ecological balance (Boucher *et al*. 1982, Hubalek 1982). Figure 15 displays the spatial distribution of five plants in the Honghe wetland: *Carex lasiocarpa* (2027 instances), *Carex pseudocuraica* (15,012 instances), *Glyceria spiculosa* (7892 instances), *Calamagrostis angustifolia* (6539 instances), and *Salix brachypoda* (16,246 instances).

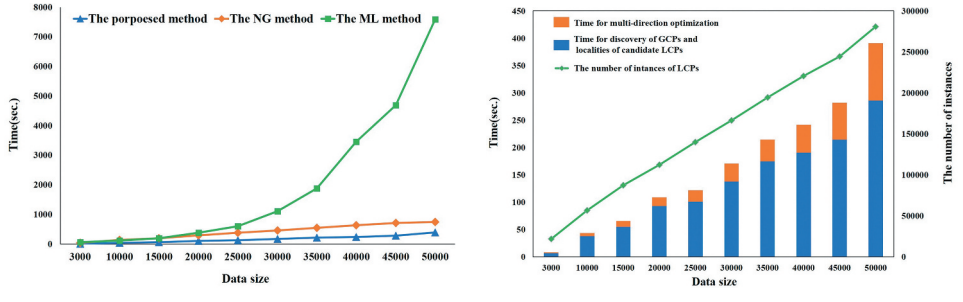

Figure 14. The efficiency and scalability of the three methods with the size of data (GCPs: global co-location patterns; LCPs: local co-location patterns): (a) Running time of the three methods on the datasets of different sizes; (b) Running time of different parts of the proposed method.

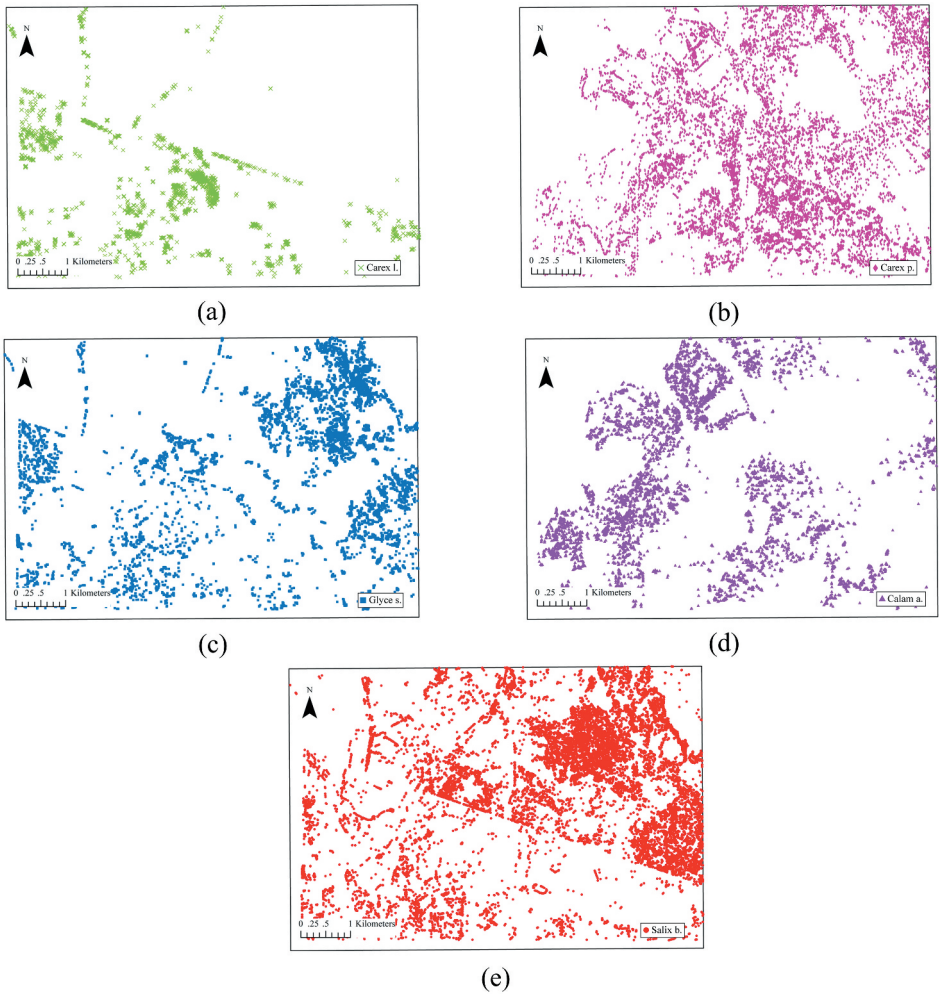

Figure 15. Spatial distributions of wetland plants: (a) *Carex lasiocarpa* (*Carex l.*); (b) *Carex pseudocuraica* (*Carex p.*); (c) *Glyceria spiculosa* (*Glyce s.*); (d) *Calamagrostis angustifolia* (*Calam a.*); (e) *Salix brachypoda* (*Salix b.*).

The NG, KNNG, and ML methods were also applied to the wetland plant dataset. The parameters of the three methods were set based on the suggestions given in the original studies (refer to Section 6.1.1). For the KNNG method, we found that when k increases to 3000, the method still does not terminate. Indeed, the discovered co-location patterns were highly overestimated and meaningless. Therefore, we do not show the results obtained by the KNNG method. In Table 1, the multilevel co-location patterns discovered by the NG, ML, and the proposed method are listed. One can see that the multilevel co-location patterns discovered by the proposed method are generally different from those discovered by the NG and ML method. The proposed method discovers more local co-location patterns than the other two methods. We further evaluate all the detected multilevel co-location patterns based on the rules given by Barua and Sander (2014) and the cross-K function (Ripley 1976). First, for each co-location pattern, we identify all the correct localities discovered by the

Table 1. The multilevel co-location patterns discovered by the NG, ML, and the proposed method.

Multi-level co-location patterns	The Proposed Method					The ML method					The NG method		
	Level	PI_{global}	PI_{min}	PI_{max}	$N_{locality}$	Level	PI_{global}	PI_{min}	PI_{max}	$N_{locality}$	RPI_{min}	RPI_{max}	$N_{locality}$
{Calam a., Carex l.}	Local	0.16	0.72	0.72	1	-	-	-	-	-	0.07	0.07	1
{Calam a., Carex p.}	Local	0.48	0.77	0.96	5	Global	0.77	-	-	1	0.24	0.29	2
{Calam a., Glyce s.}	Local	0.10	0.50	0.50	1	-	-	-	-	-	-	-	-
{Calam a., Salix b.}	Local	0.22	0.66	0.81	4	Local	0.41	0.87	0.97	3	0.07	0.24	2
{Carex l., Carex p.}	Local	0.16	0.72	0.93	2	Local	0.28	0.72	1.00	5	0.14	0.14	1
{Carex l., Glyce s.}	Local	0.26	0.81	0.92	3	Local	0.34	0.90	0.90	1	0.10	0.10	1
{Carex l., Salix b.}	Local	0.19	0.72	0.93	2	-	-	-	-	-	-	-	-
{Carex p., Glyce s.}	Local	0.22	0.56	0.78	3	Global	0.62	-	-	1	0.20	0.27	2
{Carex p., Salix b.}	Local	0.39	0.53	0.53	1	Global	0.79	-	-	1	0.69	0.69	1
{Glyce s., Salix b.}	Global	0.66	-	-	1	Global	0.90	-	-	1	0.08	0.55	3
{Calam a., Carex p., Salix b.}	Local	0.16	0.69	0.72	3	Local	0.41	0.77	0.77	1	0.13	0.26	2
{Carex l., Glyce s., Salix b.}	Local	0.14	0.84	0.84	1	-	-	-	-	-	0.07	0.07	1
{Carex p., Glyce s., Salix b.}	Local	0.13	0.52	0.68	2	-	-	-	-	-	0.12	0.12	3
{Carex l., Carex p., Glyce s.}	-	-	-	-	-	-	-	-	-	-	0.09	0.15	1

I_{global}: participation index calculated at global level; PI_{min}: minimal participation index calculated at local level; PI_{max}: maximal participation index calculated at local level; $N_{locality}$: number of localities of a co-location pattern; RPI_{min}: minimal regional participation index; RPI_{max}: maximal regional participation index.

different methods. Because we do not know the ground truth of the localities of co-location patterns (the measures introduced in Section 6.1.1 cannot be used), we assume that for two localities L_i and L_j, if $(L_i \cap L_j)/L_i > 0.5$ and $(L_i \cap L_j)/L_j > 0.5$; then L_i and L_j can be regarded as the same locality, and the precision and recall defined for supervised classification (Tan *et al.* 2006) are used to quantitatively evaluate the mining results. Table 2 summarizes the assessments of precision and recall of co-location patterns discovered by the three methods. One can observe that many local co-location patterns are missed by the ML method, while many local-co-location patterns are incorrectly identified by the NG method.

To further compare the performance of the three methods, some localities of local co-location patterns discovered by the three methods are displayed in Figure 16. In Figure 16 (a), we can find that the ML method only discovers one locality of pattern {*Carex l., Glyce s.*} in a dense region; however, the localities of {*Carex l., Glyce s.*} in sparse regions are missed. The NG method only identifies a large locality of pattern {*Carex l., Glyce s.*}, and two smaller localities of {*Carex l., Glyce s.*} are missed. In Figure 16(b), we can find that the spatial distributions of *Carex l.* and *Carex p.* are not overlapped; however, the ML method incorrectly identifies {*Carex l., Carex p.*} as a local co-location pattern because the distance threshold for constructing the neighboring relationships is extremely large for this area. In Figure 16(c), we can also observe that {*Carex l., Carex p.*} is incorrectly identified as a local

Table 2. Evaluation of the multilevel co-location patterns discovered by the NG, ML, and the proposed method.

The Proposed Method			The ML method				The NG method	Multi-level co-location patterns
Level	Precision	Recall	Level		Precision	Recall	Precision	Recall
Local	1	1	-	{Calam a., Carex l.}	-	0	1	1
Local	1	1	Global	{Calam a., Carex p.}	0	0	1	0.4
Local	1	1	-	{Calam a., Glyce s.}	-	0	-	0
Local	1	1	Local	{Calam a., Salix b.}	1	0.75	1	0.5
Local	1	1	Local	{Carex l., Carex p.}	0.2	0.5	1	0.5
Local	1	1	Local	{Carex l., Glyce s.}	1	0.33	1	0.33
Local	1	1	-	{Carex l., Salix b.}	-	0	-	0
Local	1	1	Global	{Carex p., Glyce s.}	0	0	0.5	0.67
Local	1	1	Global	{Carex p., Salix b.}	0	0	1	1
Global	1	1	Global	{Glyce s., Salix b.}	1	1	1	1
Local	1	1	Local	{Calam a., Carex p., Salix b.}	1	0.33	0.5	0.5
Local	1	1	-	{Carex l., Glyce s., Salix b.}	-	0	1	1
Local	1	1	-	{Carex p., Glyce s., Salix b.}	-	0	0.67	1
-	-	-	-	{Carex l., Carex p., Glyce s.}	-	-	0	0

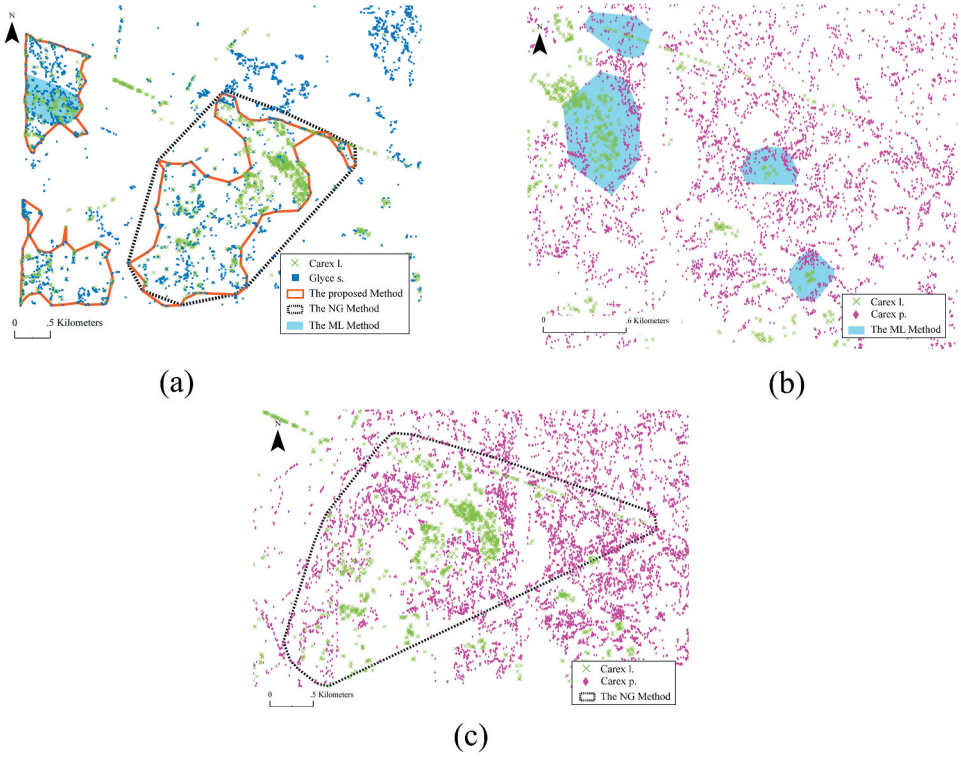

Figure 16. A comparison of the co-location patterns discovered by the NG, ML and the proposed method: (a) Localities of {*Carex l., Glyce s.*} neglected by NG and ML; (b) Localities of {*Carex l., Carex p.*} wrongly identified by the ML method; (c) Localities of {*Carex l., Carex p.*}wrongly identified by NG.

co-location pattern using the NG method because the distance threshold is extremely large for this region. One can infer that if the co-location patterns discovered by the ML and NG methods are used to investigate the spatial interactions among different plants, they are very likely to mislead the researchers.

The proposed method was also applied to the crime and urban facility datasets collected in Portland city, Oregon, USA (http://www.civicapps.org/datasets). Interested readers can refer to the supplement document for details.

7. Conclusions

This study developed an adaptive method based on natural neighborhoods for discovering multilevel co-location patterns. The natural neighborhoods were defined based on the formation mechanism of co-location patterns. The local distribution characteristics of spatial features are used to adaptively construct the neighboring relationships among unevenly distributed spatial features, which is a major challenge in the discovery of multilevel co-location patterns. The natural neighborhoods are locally adaptive and requires less *a priori* knowledge such as user-specified distance threshold and number of nearest neighbors. Therefore, the instances of candidate co-location patterns can be generated accurately. With the help of the natural neighborhoods, a multilevel refining method is proposed to automatically discover all the global and local co-location patterns from unevenly distributed spatial features. The multilevel refining method is an effective heuristic for checking all the possible local co-location patterns in arbitrarily shaped localities without the time-consuming enumeration of all possible localities. When compared with the three state-of-the-art methods, experiments on simulated and real-life datasets show that the proposed method can discover multilevel co-location patterns from unevenly distributed spatial features completely and accurately. The multilevel co-location patterns discovered by the proposed method are more reliable for understanding the spatial interactions among different spatial phenomena.

Future work will mainly focus on two directions. First, the proposed method still requires a user-specified prevalence threshold to identify significant local co-location patterns. By using the natural neighborhood, existing significance tests (Barua and Sander 2014, Deng *et al*. 2017b) can be extended to detect global co-location patterns (details can be found in the supplement document). However, we found that these tests cannot be used to identify local co-location patterns. We need to further develop an efficient significance test for the detection of multilevel co-location patterns based on the natural neighborhood. Second, this study does not consider the temporal dimension of the spatial features; therefore, the natural neighborhood should be further extended to adaptively construct the neighboring relationships for spatiotemporal data.

Acknowledgments

The authors gratefully acknowledge the comments from the editor and the reviewers. This study was funded through support from the National Key Research and Development Foundation of China (No. 2017YFB0503601), National Science Foundation of China (NSFC) (No. 41730105 and 41971353), Natural Science Foundation of Hunan Province (No.2020JJ40669) and Innovation-Driven Project of Central South University (No. 2018CX015).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the National Key Research and Development Foundation of China [2017YFB0503601]; National Natural Science Foundation of China (NSFC) [41971353, 41730105]; Natural Science Foundation of Hunan Province[2020JJ40669]; The Innovation-Driven Project of Central South University [2018CX015].

Notes on contributors

Qiliang Liu received the Ph.D. degree in geographical information science from The Hong Kong Polytechnic University in 2015. He is currently an associate professor at Central South University, Hunan, China. His research interests focus on multi-scale spatio-temporal data mining and spatio-temporal statistics. He has published more than 30 peer-reviewed journal articles in these areas.

Wenkai Liu is currently a Ph.D. candidate at Central South University and his research interests focus on spatio-temporal clustering and association rule mining.

Min Deng is currently a professor at Central South University and the associate dean of School of Geosciences and info-physics. His research interests are map generalization, spatio-temporal data analysis and mining.

Jiannan Cai received the Ph.D. degree in geographical information science from Central South University in 2019. His research interests focus on spatio-temporal association rule mining.

Yaolin Liu is currently a Professor at Wuhan University. His research interests include Geographic Information Science, geographic data mining, and spatial analysis and decision making.

ORCID

Jiannan Cai http://orcid.org/0000-0003-4752-0153

Data and codes availability statement

The data and codes that support the findings of this study are available in 'figshare.com' with the identifier at the permanent link: http://doi.org/10.6084/m9.figshare.12061701

References

Barua, S. and Sander, J., 2014. Mining statistically significant co-location and segregation patterns. *IEEE Transactions on Knowledge and Data Engineering*, 26 (5), 1185–1199. doi:10.1109/TKDE.2013.88.

Bembenikr, R. and Rybiński, H., 2009. FARICS: a method of mining spatial association rules and collocations using clustering and Delaunay diagrams. *Journal of Intelligent Information Systems*, 33 (1), 41–64. doi:10.1007/s10844-008-0076-1.

Boucher, D.H., James, S., and Keeler, K.H., 1982. The ecology of mutualism. *Annual Review of Ecology and Systematics*, 13 (1), 315–347. doi:10.1146/annurev.es.13.110182.001531.

Cai, J.N., *et al*., 2018. Adaptive detection of statistically significant regional spatial co-location patterns. *Computers, Environment and Urban Systems*, 68, 53–63. doi:10.1016/j.compenvurbsys.2017.10.003.

Cai, J.N., *et al*., 2019. Nonparametric significance test for discovery of network-constrained spatial co-location patterns. *Geographical Analysis*, 51 (1), 3–22. doi:10.1111/gean.12155.

Celik, M., Kang, J.M., and Shekhar, S., 2007. Zonal co-location pattern discovery with dynamic parameters. In: *Proceedings of the 7th IEEE International Conference on Data Mining*, 28–31, October, Omaha, NE, IEEE.

Clark, P. and Evans, F., 1954. Distance to nearest neighbor as a measure of spatial relationships in populations. *Ecology*, 35 (4), 445–453. doi:10.2307/1931034.

Cormen, T.H., *et al*., 2009. *Introduction to algorithms*. 3rd edn. London: The MIT Press.

Deng, M., *et al*., 2017a. Multi-level method for discovery of regional co-location patterns. *International Journal of Geographical Information Science*, 31 (9), 1846–1870. doi:10.1080/13658816.2017.1334890.

Deng, M., *et al*., 2017b. Multi-scale approach to mining significant spatial co-location patterns. *Transactions in GIS*, 21 (5), 1023–1039. doi:10.1111/tgis.12261.

Ding, W., *et al*., 2011. A framework for regional association rule mining and scoping in spatial datasets. *Geoinformatica*, 15 (1), 1–28. doi:10.1007/s10707-010-0111-6.

Edelsbrunner, H., Kirkpatrick, D., and Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29 (4), 551–559. doi:10.1109/TIT.1983.1056714.

Estivill-Castro, V. and Lee, I., 2002. Multi-level clustering and its visualization for exploratory spatial analysis. *GeoInformatica*, 6 (2), 123–152. doi:10.1023/A:1015279009755.

Fortin, M.J. and Dale, M.R., 2005. *Spatial analysis: a guide for ecologists*. Cambridge: Cambridge University Press.

Goreaud, F. and Pélissier, R., 2003. Avoiding misinterpretation of biotic interactions with the intertype K_{12}-function: population independence vs. random labelling hypotheses. *Journal of Vegetation Science*, 14 (5), 681–692.

Guo, D. and Mennis, J., 2009. Spatial data mining and geographic knowledge discovery—An introduction. *Computers, Environment and Urban Systems*, 34 (2), 175. doi:10.1016/j.compenvurbsys.2009.11.001.

Guo, D.S. and Wang, H., 2011. Automatic region building for spatial analysis. *Transactions in GIS*, 15 (s1), 29–45. doi:10.1111/j.1467-9671.2011.01269.x.

Huang, Y., Shekhar, S., and Xiong, H., 2004. Discovering colocation patterns from spatial data sets: a general approach. *IEEE Transactions on Knowledge and Data Engineering*, 16 (12), 1472–1485. doi:10.1109/TKDE.2004.90.

Hubalek, Z., 1982. Coefficients of association and similarity, based on binary (presence-absence) data: an evaluation. *Biological Reviews*, 57 (4), 669–689. doi:10.1111/j.1469-185X.1982.tb00376.x.

İnkaya, T., Kayalıgil, S., and Özdemirel, N., 2015. An adaptive neighbourhood construction algorithm based on density and connectivity. *Pattern Recognition Letters*, 52, 17–24. doi:10.1016/j.patrec.2014.09.007

Keddy, P.A., 2010. *Wetland ecology: principles and conservation*. UK: Cambridge University Press.

Li, Y. and Shekhar, S., 2018. Local co-location pattern detection: a summary of results. In: *Proceedings of the 10th International Conference on Geographic Information Science* (GIScience 2018), Article No. 10. Melbourne, Australia, 1–15.

Mohan, P., *et al*., 2011. A neighborhood graph based approach to regional co-location pattern discovery: a summary of results. In: *Proceedings of the* 19*th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, November 1-4, Chicago, IL. New York: ACM, 122–132.

Phillips, P. and Lee, I., 2012. Mining co-distribution patterns for large crime datasets. *Expert Systems with Applications*, 39 (14), 11556–11563. doi:10.1016/j.eswa.2012.03.071.

Qian, F., *et al*., 2014. Mining regional co-location patterns with kNNG. *Journal of Intelligent Information Systems*, 42 (3), 485–505. doi:10.1007/s10844-013-0280-5.

Ripley, B.D., 1976. The second-order analysis of stationary point processes. *Journal of Applied Probability*, 13 (2), 255–266. doi:10.2307/3212829.

Shi, Y., *et al*., 2016. Adaptive detection of spatial point event outliers using multilevel constrained Delaunay triangulation. *Computers, Environment and Urban Systems*, 59, 164–183. doi:10.1016/j.compenvurbsys.2016.06.001.

Sundaram, V.M. and Thnagavelu, A., 2015. A Delaunay diagram-based min–max CP-tree algorithm for spatial data analysis. *Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery*, 5 (3), 142–154.

Tan, P.N., *et al*., 2006. *Introduction to data mining*. London: Pearson Education, Inc.

Wan, Y. and Zhou, J., 2008. KNFCOM-T: a k-nearest features-based co-location pattern mining algorithm for large spatial data sets by using T-trees. *International Journal of Business Intelligence and Data Mining*, 3 (4), 375–389. doi:10.1504/IJBIDM.2008.022735.

Wang, J.M., Hsu, W., and Lee, M.L., 2005. A framework for mining topological patterns in spatio-temporal databases. In: *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, October 31-November 5, Bermen, Germany. ACM: 429–436.

Wang, L., *et al*., 2009. An order-clique-based approach for mining maximal co-locations. *Information Sciences*, 179 (19), 3370–3382. doi:10.1016/j.ins.2009.05.023.

Wang, S., *et al*., 2013. Regional co-locations of arbitrary shapes. *In*: M.A. Nascimento, *et al*., ed.. *Advances in spatial and temporal databases. SSTD 2013. Lecture notes in computer science*. Vol. 8098. Berlin, Heidelberg: Springer, 19–37.

Xiao, X.Y., *et al*., 2008, Density based co-location pattern discovery. In *Proceedings of the16th ACMSIGSPATIAL International Conference on Advances in Geographic Information Systems*, November 5-7, Irvine, CA. ACM, 102–114.

Yang, P.Z., Wang, L.Z., and Wang, X.X., 2019. A MapReduce approach for spatial co-location pattern mining via ordered-clique-growth. *Distributed and Parallel Databases*. doi:10.1007/s10619-019-07278-7.

Yao, X.J., *et al*., 2016. A fast space-saving algorithm for maximal co-location pattern mining. *Expert Systems with Applications*, 63, 310–323. doi:10.1016/j.eswa.2016.07.007.

Yao, X.J., *et al*., 2017. A co-location pattern-mining algorithm with a density-weighted distance thresholding consideration. *Information Sciences*, 396, 144–161. doi:10.1016/j.ins.2017.02.040.

Yao, X.J., *et al*., 2018. A spatial co-location mining algorithm that includes adaptive proximity improvements and distant instance references. *International Journal of Geographical Information Science*, 32 (5), 980–1005. doi:10.1080/13658816.2018.1431839.

Yoo, J.S., *et al*., 2019. Parallel co-location mining with MapReduce and NoSQL systems. *Knowledge and Information Systems*. doi:10.1007/s10115-019-01381-y.

Yoo, J.S. and Bow, M., 2012. Mining spatial colocation patterns: a different framework. *Data Mining and Knowledge Discovery*, 24 (1), 159–194. doi:10.1007/s10618-011-0223-0.

Yoo, J.S. and Shekhar, S., 2004. A partial-join approach for mining co-location patterns. In: *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems*, November 12-13, Washington, DC. ACM, 241–249.

Yoo, J.S. and Shekhar, S., 2006. A joinless approach for mining spatial colocation patterns. *IEEE Transactions on Knowledge and Data Engineering*, 18 (10), 1323–1327. doi:10.1109/TKDE.2006.150.

Yu, W.H., *et al*., 2017. Spatial co-location pattern mining of facility points-of-interest improved by network neighborhood and distance decay effects. *International Journal of Geographical Information Science*, 31 (2), 280–296. doi:10.1080/13658816.2016.1194423.

Zhou, M., *et al*., 2019. A visualization approach for discovering colocation patterns. *International Journal of Geographical Information Science*, 33 (3), 567–592. doi:10.1080/13658816.2018.1550784.

Zhou, Y., *et al*., 2011. Development of percentile estimation formula for skewed distribution. *Acta Physica Sinica*, 60 (8), 089201.

Zimmer, K.D., Hanson, M.A., and Butler, M.G., 2003. Interspecies relationships, community structure, and factors influencing abundance of submerged macrophytes in prairie wetlands. *Wetlands*, 23 (4), 717–728. doi:10.1672/0277-5212(2003)023[0717:IRCSAF]2.0.CO;2.