Mathematical Biosciences and Engineering

MBE, 18(6): 8223–8244.
DOI: 10.3934/mbe.2021408
Received: 13 April 2021
Accepted: 13 August 2021
Published: 18 September 2021

http://www.aimspress.com/journal/MBE

Research article

Spatial co-location pattern mining based on the improved density peak clustering and the fuzzy neighbor relationship

Meijiao Wang*, Yu chen, Yunyun Wu and Libo He

School of Information and Network Security, Yunnan Police College, Kunming 650223, China

***Correspondence:** Email: wangmj0871@163.com; Tel: +86087165198660; Fax: +86087165183096.

Abstract: Spatial co-location pattern mining discovers the subsets of spatial features frequently observed together in nearby geographic space. To reduce time and space consumption in checking the clique relationship of row instances of the traditional co-location pattern mining methods, the existing work adopted density peak clustering to materialize the neighbor relationship between instances instead of judging the neighbor relationship by a specific distance threshold. This approach had two drawbacks: first, there was no consideration in the fuzziness of the distance between the center and other instances when calculating the local density; second, forcing an instance to be divided into each cluster resulted in a lack of accuracy in fuzzy participation index calculations. To solve the above problems, three improvement strategies are proposed for the density peak clustering in the co-location pattern mining in this paper. Then a new prevalence measurement of co-location pattern is put forward. Next, we design the spatial co-location pattern mining algorithm based on the improved density peak clustering and the fuzzy neighbor relationship. Many experiments are executed on the synthetic and real datasets. The experimental results show that, compared to the existing method, the proposed algorithm is more effective, and can significantly save the time and space complexity in the phase of generating prevalent co-location patterns.

Keywords: spatial data mining; spatial co-location pattern; density peak clustering; fuzzy neighbor relationship; cluster fuzzy participation index

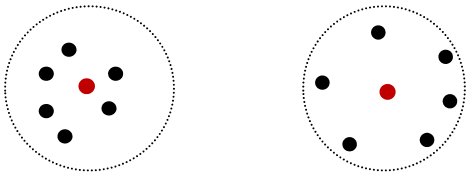
1. Introduction

As an important branch of spatial data mining, spatial co-location pattern mining has been a research hot recently. A prevalent spatial co-location pattern represents a subset of spatial features

whose instances are frequently located together in a spatial neighborhood. There are many application domains such as Earth science, public health, biology, transportation, etc. [1].

The traditional co-location pattern mining framework materializes the neighbor relationship between spatial instances according to a certain distance threshold set by the users, which is very time-consuming to repeatedly check the clique relationship of row instances, and also very space-consuming to store row instances. To address this problem, Fang et al. [2] materialized the spatial neighbor relationship between instances by clustering technology, and based on the Density Peak Clustering, they proposed the algorithm for Mining Prevalent Co-locations (the DPC-MPC algorithm). The DPC-MPC algorithm employed the basic idea of density peak clustering to cluster the instances. After finding the cluster center through the decision graph, the remaining instances were fuzzy clustered for overlapping distribution according to the membership value. The sum of the membership values of an instance to all clusters was 1. After clustering, the neighbor relationship between instances was simplified in the cluster. Namely, a λ-cluster (λ cut set of a cluster) was regarded as a cluster, and the instances in the cluster meet the λ-proximity. Finally, in the prevalent co-location filtering phase, the membership of an instance to the cluster in the table instance was used to calculate the prevalence measurement of the co-location. However, the DPC-MPC algorithm had the following two shortcomings:

First, when calculating the local density of an instance in the process of density peak clustering, only the number of data points in the circle with the instance as the center and the cutoff distance as the radius was counted, ignoring the fuzziness of neighbor relationship between the center instance and other instances in the circle. For example, Figure 1 shows two circles with the same radius. The red dot is the center of the circle, and both circles have a density of 6. However, it can be observed that the points around the center of (a) are more compact than those around the center of (b). That is to say, the local density of the center of the circle in (a) should be larger than that in (b). However, the DPC-MPC algorithm treated the two cases as the same, which affected the accuracy of local density and relative distance in density peak clustering, and further affected the accuracy of the clustering results.



(a) (b)

Figure 1. Local density.

Secondly, because the sum of the membership of an instance to all clusters was constrained by 1, when λ takes as 0, each λ-cluster is the whole data set, so that the participation index of all features was the value 1, it is unreasonable to take λ as 0; when λ is greater than 0, the contribution of an instance to the participation ratio of its feature must be less than 1. That is to say, if an instance is not in a λ-cluster, it will always take away part of its contribution to the participation ratio. The larger the λ is, the greater the contribution to the participation ratio to be taken away, causing the maximum value of the participation ratio is on longer 1, but a certain value in the interval < 0, 1 >. So the participation ratio calculated by the membership value lacks of accuracy. The reason is that the overlapping division of instances usually occurs at the intersection of two or more clusters, but the

DPC-MPC algorithm strictly divided instances into each cluster according to the membership value, and the membership values were finally used to calculate the participation ratio.

To solve the above two shortcomings, we need to adopt a technology that can not only distinguish the difference of local density between instances, but also correctly measure the similarity between an instance and its cluster center. A fuzzy neighbor method was proposed to improve the calculation of the local density of density peak clustering [3]. It defined a fuzzy proximity function to measure the contribution of a data point to the local density of the center. The fuzzy proximity function could measure the similarity between the data point and the center within the cutoff distance, but couldn't measure the similarity between any two points on the spatial data set. Therefore, it cannot solve the second shortcoming. Kernel density estimation is a function used to estimate the unknown density, which is widely used in many types of research, including density peak clustering. It can distinguish the differences of local density, and also can measure the similarity between two instances. However, the fuzzy neighborhood relationship, which can also solve the above two shortcomings, is more effective than the kernel density estimation method [4]. Therefore, in this paper, the co-location pattern mining based on the improved density peak clustering and the fuzzy neighbor relationship is studied.

The following is the main contributions of this paper:

1) To make the clustering results better applied to the co-location pattern mining, three improvements are adopted to the classical density peak clustering algorithm: Define the local density and relative distance based on the fuzzy neighbor relationship; Propose an automatic generation strategy of cluster center instances; Propose a new overlapping allocation strategy of instances;

2) A new measure of co-location pattern prevalence, cluster fuzzy participation index, is given in the framework of co-location pattern mining based on clustering;

3) A co-location pattern mining algorithm based on the improved density peak clustering and the fuzzy neighbor relationship is designed;

4) Compared with the existing algorithms, the effectiveness of the proposed algorithm is verified, and the efficiency and the memory consumption in the stage of co-location pattern generation are evaluated.

Section 2 describes the related works. The preliminaries are given in Section 3. Some related definitions are provided in Section 4. Section 5 presents the algorithm. Section 6 performs the experiments to verify the proposed algorithms. Finally, the conclusion is presented in Section 7.

2. Related work

The concept of co-location pattern was first proposed by Shekhar et al. [5]. Unlike association rule mining in transaction databases, which uses support degree to measure the frequency of transactions, spatial co-location pattern mining employs the participation index to measure the prevalence of co-location patterns. A join-based method which was an Apriori-like algorithm for co-location pattern mining was proposed by Huang [1]. For reducing the join operations between table instances of the join-based algorithm, the join-less and the partial join algorithms were presented in [6,7]. The prefix-tree structure was adopted in the CPI-tree algorithm [8] and the iCPI-tree algorithm [9] to prune the candidate co-locations and reduce the memory consumption for storing table instances. For discovering co-location patterns on uncertain data sets, lots of research had been done in [10,11]. And the work in [12] aimed to find co-location patterns from the interval data. For more efficiently and losslessly compressing the prevalent co-location patterns than the closed co-locations strategy, Wang et al. [13] put forward the concept of the SPI-closed co-location.

The work in [14] studied the strategy of reducing the redundancy of prevalent co-locations according to the distribution information of the instances. The research on finding maximal co-location patterns was conducted in [15–17]. Parallel co-location mining based on map-reduce was studied in [18,19] for massive spatial data. The fuzzy set theory had been integrated with the co-location pattern mining [2,20,21].

The density peak clustering (DPC) algorithm was originally proposed in 2014 [22]. Over the past six years, various methods have been adopted to extend the classic DPC algorithm. The paper [23] proposed a robust clustering by computing the local density relative to the K-nearest neighbors independent of the cutoff distance and using two new point assignment strategies to assign the remaining points to the most probable clusters. An improved density peaks clustering algorithm with fast finding cluster centers was proposed for the large-scale data set [24]. Liu et al. [25] put forward the shared-nearest-neighbor-based clustering by fast searching and finding of density peaks (SNN-DPC). The constraint-based clustering by fast search and find of density peaks (CCFDP) method was studied in [26]. [27] presented a new semi-supervised density peaks clustering algorithm (SSDPC) which used constraint projection.

As far as we know, no work adopted the fuzzy neighbor relationship in the co-location pattern mining based on the density peak clustering framework, which will be addressed in this paper.

3. Preliminaries

3.1. Density peak clustering

Density peak clustering (DPC) [22] is to find high-density areas separated by low-density areas, which is based on the following two hypotheses: 1) The density of a cluster center is large enough, and the density of any data point around it is not more than which of itself; 2) The distance between a cluster center and a higher density cluster center is also large enough.

The DPC algorithm defines two important parameters: local density and relative distance. It determines the cluster centers and assigns other instances to the clusters according to these two parameters. The definitions of the two parameters are given below.

Definition 1. (Local Density) [22]**.** Given a data set $S = \{s_1, s_2, \ldots, s_n\}$, a cutoff distance d_c, the local density ρ_i of data point s_i ($s_i \in S$) is defined as the number of data points that are closer than d_c to point s_i, namely,

$$\rho_i = \sum_{s_j \in S, j \neq i} \chi(dist(s_i, s_j) - d_c)$$
$$\chi(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases} \quad (1)$$

where, $dist\ (s_i, s_j)$, $i \neq j$, is the distance between s_i and s_j, and d_c is a parameter set by the user. The literature [22] proved by experiments that, when the data set is large enough, the value of d_c has little influence on the final clustering result, and provided a method of selecting an appropriate distance: First, arrange the distances between all data points in ascending order (the distance matrix between all data points is obtained by preprocessing and is the input of the algorithm), then select the distance corresponding to the percentage (0.5~5% is recommended)) as the cutoff distance.

Definition 2. (Relative Distance) [22]**.** The relative distance δ_i of the data point s_i is defined as the minimum distance between the point s_i and any other point with a higher local density. For the point

with the highest local density s_i, δ_i is defined as the maximum distance between s_i and the other data points, formalized as:

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j > \rho_i} (dist(s_i, s_j)), & if \exists j \quad s.t. \quad \rho_j > \rho_i \\ \max\limits_{j} (dist(s_i, s_j)), & otherwise \end{cases} \tag{2}$$

The DPC algorithm first calculates binary pairs (ρ_i , δ_i) for each data point. All binary pairs (ρ_i , δ_i) are plotted into a decision diagram with ρ_i as the horizontal axis and δ_i as the vertical axis. Then select the points with larger ρ_i and δ_i manually, that is, the points close to the upper right corner of the decision graph and obviously different from other data points as the clustering center. Finally, classify the remaining data points into the cluster of the nearest sample points with higher local density according to the order of local density from large to small.

3.2. Fuzzy neighbor relationship between spatial instances

Let there be a spatial instance data sets *S*, fuzzy neighbor relationship (FNR) [4] is used to measure the proximity level between instances in *S*. Taking the Euclidean distances *ED* between the instances in *S* as the domain, where $ED \to [0,\infty)$, FNR is a fuzzy subset on *ED*. Namely, there is a mapping for FNR: $ED \to [0,1]$, $d \to \mu(d)$, where $d \in ED$ represents the Euclidean distance between any two specific instances in *S*, μ is the proximity function of FNR, and $\mu(d)$ is the membership value of Euclidean distance *d*, which refers to the probability of d belonging to FNR.

4. Related definitions

4.1. Density peak clustering based on FNR

This section defines the local density and the relative distance of an instance based on fuzzy neighbor relationship (FNR).

Definition 3. (Local density based on FNR). Given the set of spatial instances *S*, the fuzzy neighbor relationship FNR on *S*, the local density ρ_i^{FNR} of s_i ($s_i \in S$) based on FNR is defined as the sum of the membership value between the center point and the data points that are closer than d_c to point s_i. The formal expression is as follows:

$$\rho_i^{FNR} = \sum_{s_j \in S, j \neq i} \mu(dist(s_i, s_j)), \quad dist(s_i, s_j) \le d_c \tag{3}$$

In the DPC algorithm, the contribution (equal to the membership value) of a data point to the density of the center is 0 or 1. The definition of ρ_i^{FNR} considers the difference in the distance between instance within the cutoff distance and the center. In this case, the contribution to the local density is a certain value on the interval [0,1], which is more accurate than that in the DPC algorithm.

Figure 2 shows an example spatial instance data set and the coordinates of each instance in the data sets. The membership function of FNR on the data sets is shown in Eq (4).

$$\mu(d)=\begin{cases}1 & d<1\\ -\frac{1}{4}(d-1)+1 & 1\le d\le 5\\ 0 & d>5\end{cases} \tag{4}$$

Suppose that the cutoff distance $d_c = 3$. Table 1 lists the local density and the local density based on FNR of some instances. It can be observed that a local density value was shared by multiple instances shared. In contrast the local density based on FNR distinguished the local density of each instance, which was helpful to select the cluster center instance accurately.

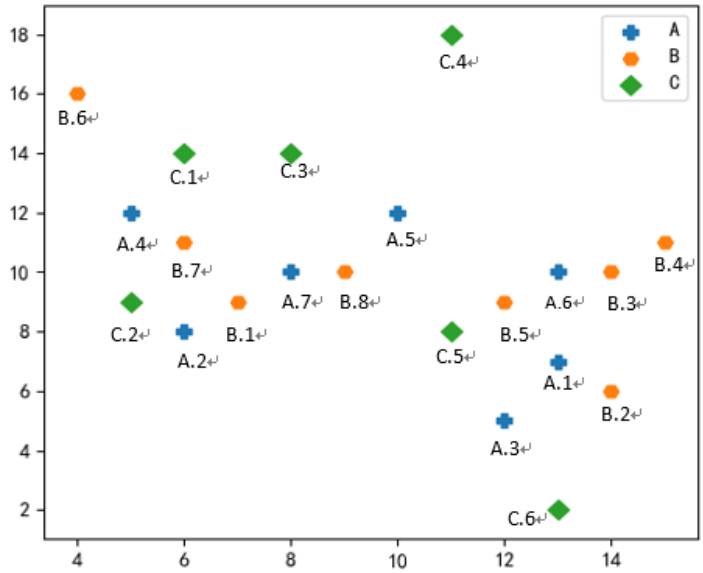


Figure 2. An example spatial instance data sets.

Table 1. The local densities of some instances.

Instances	A.1	A.2	A.3	A.4	A.5	A.6	A.7	B.1	B.2	B.3
The local density based on FNR	3.47	2.84	1.38	2.09	1.78	3.63	3.67	3.93	1.59	2.59
The local density	5	4	2	3	3	5	5	5	2	4

Definition 4. (Relative distance based on FNR). Given the set of spatial instances S, the fuzzy neighbor relationship FNR on S, the relative distance of $s_i\,(s_i \in S)$ based on FNR δ_i^{FNR} is defined as the minimum distance between s_i and the other instances with higher local density. For the instance s_i with the highest local density, δ_i^{FNR} is defined as the largest distance between s_i and the other instances, formalized as:

$$\delta_i^{FNR}=\begin{cases}\min\limits_{j:\rho_j^{FNR}>\rho_i^{FNR}}\left(dist\left(s_i,s_j\right)\right), & if\ \exists j \quad s.t. \quad \rho_j^{FNR}>\rho_i^{FNR}\\ \max\limits_{j}\left(dist\left(s_i,s_j\right)\right), & otherwise\end{cases} \tag{5}$$

The relative distance based on FNR is similar to that in the DPC algorithm. The critical difference between the two is that the relative distance based on FNR is calculated by the local density based on FNR, so it is more precise than that in the DPC algorithm.

4.2. Automatically generate cluster center instances

In the classic DPC algorithm, the decision graph is constructed by local density and the relative distance to select the points with larger local density and relative distance as cluster centers. However, this method is carried out manually with a certain subjectivity. To avoid the errors caused by human intervention in cluster center selection, a strategy based on the statistical characteristics of local density and relative distance was proposed in [28] to generate cluster centers automatically. It selected the instances with local density greater than the mean of local density and relative distance greater than twice the relative distance standard deviation as cluster centers. Unlike the classic DPC algorithm, in the clustering for co-location pattern mining, an instance may belong to multiple different clusters. That is to say, it is allowed to overlap between clusters. Therefore, in this paper, we select the instances with the local density greater than the mean value of the local density and the relative distance greater than the relative distance standard deviation as the cluster center instead.

Definition 5. (Cluster center instance). Given the spatial instance set S, let $E(\rho)$ be the mean value of the local density based on FNR of S, let $\sigma(\delta)$ be the standard deviation of the relative distance based on FNR, if the local density ρ_i^{FNR} and relative distance δ_i^{FNR} of the instance s_i satisfy:

$$\begin{cases} \rho_i^{FNR} > E(\rho) \\ \delta_i^{FNR} > \sigma(\delta) \end{cases} \tag{6}$$

Then s_i is a cluster center instance.

Table 2 lists the selected cluster centers by the decision graph and the cluster center automatic generation strategy respectively. The validity of the result will be evaluated in subsequent experiments in this paper.

Table 2. The selected cluster center instances by two different selection methods.

Selection strategy	The cluster center instances
Automatic generation strategy	A.1, A.6, B.7, C.1
Decision graph	A.6, B.7

4.3. Overlapping allocation of instances

According to the basic idea of density peak clustering, after determining the cluster centers, it is necessary to divide the remaining instances into clusters with higher density. In this way, an instance must and can only be allocated to a specific cluster. Because clustering results in this paper are applied to co-location pattern mining, as mentioned above, instances may belong to multiple row instances, which will happen not only in a single cluster but also among multiple clusters.

To realize overlapping division of instances, the definition of a cluster is as following.

Definition 6. (Cluster). Given a set of spatial instances S, a set of m cluster centers $v = \{v_1, v_2 \ldots, v_m\}$, a set of m clusters $V = \{V_1, V_2 \ldots, V_m\}$, a user-defined proximity threshold β ($0 \le \beta \le 1$), The cluster

V_i ($1 \le i \le m$) is defined as the set of instances whose proximity to the cluster center v_i is no less than β. Namely,

$$V_i = \{(s_j, \mu(dist(v_i, s_j)) \mid s_j \in S, \mu(dist(v_i, s_j)) \ge \beta\} \tag{7}$$

where, the membership value $\mu(dist(v_i, s_j))$ is also referred to the membership of s_j to the cluster V_i. For the cluster center v_i, the membership to the cluster V_i is assigned the value 1.

According to the membership threshold β given by the user, instances can be divided into clusters in one step. After the division, an instance may not be divided into any cluster, or may be divided into multiple clusters. For the example datasets in Figure 2, set β = 0.1. Table 3 lists the clustering results after overlapping allocation of the instances to the cluster centers A.1, A.6, B.7 and C.1. The first instance in each cluster is the cluster center instance. Each cluster also stores the proximity of the instances to the cluster center. For example, the proximity of B.2 to A.1 is 0,898. The underlined instances are overlappingly allocated.

Table 3. Clustering results by overlapping allocation (β = 0.1).

Cluster ID	Instances of clusters
Cluster-1	{<A.1, 1.0>, <A.3, 0.9039>, <B.2, 0.898>, <u><B.3, 0.46></u>, <u><B.5, 0.69></u>, <u><B.4, 0.133></u>, <u><C.5, 0.69></u>}
Cluster-2	{ <A.6, 1.0>, <B.3, 1.0>, <B.4, 0.69>, <B.5, 0.898>, <C.5, 0.542>, <u><A.5, 0.35></u>, <u><B.8, 0.25></u>, <u>< B.2, 0.22 ></u>}
Cluster-3	{<B.7, 1.0>, <A.2, 0.5>, <A.4, 0.898>, <A.5, 0.22>, <A.7, 0.69>, <B.1, 0.69>, <B.8, 0.46>, <C.2, 0.69>, <u><C.3, 0.35></u>}
Cluster-4	{<C.1, 1.0>, <B.6, 0.54>, <C.3, 0.75>, <u><A.5, 0.132></u>, <u><A.4, 0.69></u>}

4.4. Prevalence measurement and properties

According to the clustering results, there are the following new definitions.

Definition 7. (β-proximity). Given a set of spatial instances S, a set of m clusters on S V=$\{V_1, V_2,\ldots,V_m\}$, a cluster $V_t \in V(1 \le t \le m)$, for any two instances s_i, s_j in S, if $s_i \in V_t$ and $s_j \in V_t$, then s_i and s_j satisfy β-proximity.

For example, in Cluster-1, A.1, A.3, B.2, B.3, B.5, B.4 and C.5 all satisfy β-proximity pairwise.

From the above definition, we can see that both instances in a cluster satisfy the β-proximity relationship. The α-proximity [4] is obtained by calculating the proximity between two instances. The β-proximity requires that the two instances are in the same cluster. Then a cluster is also a clique. Therefore, we can obtain the cluster row instances and cluster table instances which are defined below.

Definition 8. (Cluster row instance and cluster table instance). Given a set of spatial instances S, a set of m clusters on S V=$\{V_1, V_2,\ldots,V_m\}$, for a co-location c, if the subset I of the cluster $V_t (V_t \in V)$ satisfies: 1) I contains all instances of the features in c; 2) There is no subset of I containing the instances of all features in c, then I is called a cluster row instance of c. The set of all cluster row instances of c is called the cluster table instance of c, denoted as VT (c).

For example, in Cluser-1, for the co-location c = {A,B,C}, {A.1, B.2, C.5} is a cluster row instance of c; the cluster table instance of c, $VT(c)$ = {{A.1, B.2, C.5},{A.1, B.3, C.5}, {A.1, B.4, C.5}, {A.1, B.5, C.5}, {A.3, B.2, C.5}, {A.3, B.3, C.5},{A.3, B.4, C.5},{A.3, B.5, C.5}}.

Unlike the fuzzy row instance and fuzzy table instance in the classic co-location pattern, the cluster row instance and cluster table instance do not need to be generated and stored.

Definition 9. (Cluster Fuzzy Participation Ratio (VFPR) and Cluster Fuzzy Participation Index (VFPI)). Given a set of spatial instances S, a set of m clusters on S $V = \{V_1, V_2,\ldots,V_m\}$, a size-$k$ candidate co-location $c = \{o_1, o_2,\ldots, o_k\}$, for the feature $o_u \in c$, the cluster fuzzy participation ratio VFPR of o_u is defined as the ratio of the sum of the membership of an instances that do not recur in the cluster table instances of c to the cluster in which they are located and the total number of the instances of o_u .

$$VFPR(c,o_u)=\frac{\sum_{i=1}^{m}\max_{s_j\in\pi_{o_u}(VT(c))}(\mu(dist(v_i,s_j)))}{N(o_u)} \tag{8}$$

where, $\pi_{o_u}(VT(c)$ is the projection of the feature o_u on the table instance VT (c); $N(o_u)$ is the total number of the instances of o_u.

In the above formula, for an instance that is overlapping allocation, only the maximum membership of the cluster it belongs to is counted in the summation expression.

The cluster fuzzy participation index of the co-location c is defined as the minimum cluster fuzzy participation ratio of all features in c, namely,

$$VFPI(c)=\min_{u=1}^{k}\{VFPR(c,o_u)\} \tag{9}$$

The following is the analysis of the range of cluster fuzzy participation ratio (VFPR) and cluster fuzzy participation index (VFPI). Because the range of the membership of an instance to a cluster is [0,1], when an instances of feature o_u of c all appear in the cluster table instance, and each instance has a membership value of 1 to the cluster it belongs, the cluster fuzzy participation ratio takes the maximum value of 1; when the membership value of an instance to its cluster takes a value in the interval (0,1), the cluster fuzzy participation ratio increases with the increase of the membership value, When all instances of the feature o_u do not appear in the cluster table instance, the cluster fuzzy participation ratio of o_u takes the minimum value of 0. Therefore, the value range of the cluster fuzzy participation ratio is also [0,1]. According to the definition, the range of the cluster fuzzy participation index is also [0,1].

Given a minimum cluster fuzzy participation threshold *min_vfprev*, if $VFPI(c)\geq \min\_vfprev$, then the co-location c is prevalent.

For example, in Figure 2, for the co-location c={A,B}, the fuzzy participation ratio of A is

$$VFPR(c,\mathrm{A})=\frac{\sum_{i=1}^{4}\max_{s_j\in\pi_{\mathrm{A}}(VT(c))}(\mu(dist(v_i,s_j)))}{N(A)}=0.7321,$$

and of B is

$$VFPR(c,\mathrm{B})=\frac{\sum_{i=1}^{4}\max_{s_j\in\pi_{\mathrm{B}}(VT(c))}(\mu(dist(v_i,s_j)))}{N(B)}=0.7722,$$

then the fuzzy participation index of c is $VFPI(c)$= 0.7321. If min_*vfprev* = 0.5, then c is prevalent.

The cluster fuzzy participation ratio and cluster fuzzy participation index has anti-monotonicity and downward closure property.

Lemma 1. (Anti-monotonicity). The cluster fuzzy participation ratio (VFPR) and the cluster fuzzy participation index (VFPI) are anti-monotone with increasing of the size of the co-location pattern.

Proof. For a co-location pattern c, if an instance of the feature in c appears in a cluster row instance of a super-set of c, then the instance must also appear in the cluster row instance of c, not vice versa. Since the calculation of the cluster fuzzy participation ratio considers the maximum membership value of an instance to the cluster it belongs. So, the cluster fuzzy participation rate VFPR is monotonically non-increasing. According to the definition of the cluster fuzzy participation index (VFPI), VFPI is also monotonically non-increasing.

Lemma 2. (Downward closure properties). In the co-location pattern mining based on improved density peak clustering and fuzzy neighbor relationship, if a co-location c is prevalent, all its subsets are prevalent too; if c is not prevalent, all its supersets are also not prevalent.

Proof. Available from Lemma 1.

5. Algorithms

The co-location pattern mining algorithm based on improved density peak clustering and fuzzy neighbor relationship (CPM-IDPCFNR algorithm) is described as follows:

Algorithm 1. the CPM- IDPCFNR algorithm

Input:
O: set of spatial features, S: spatial data set, d_c: cutoff distance, μ: proximity function of FNR, β: membership threshold

Variables:
FNR: fuzzy neighbor relationship, k: size of co-location, v: set of cluster center, V: set of cluster, *Den*: set of local density based on FNR, *RD*: set of relative distances based on FNR, C_k: size-k co-location, P_k: set of size-k prevalent co-locations, *min_vfprev*: the minimum fuzzy participation index threshold

Output:
co-location pattern set P with $VFPI \geqslant min\_vfprev$

Steps:
1) FNR = get_fuzzy_neighbor_relationship (S, μ);
2) Den = calculate_local_density (FNR, d_c);
3) RD = get_peaks (Den, S);
4) v = select_cluster_center (Den, RD);
5) V_i = assign_to_cluster (S, FNR, β, v);
6) $k = 1$; $P_1 = O$;
7) while(not empty P_k-1) do
8) C_k = generate_candidate_co-locations (P_k-1);
9) P_k = select_prevalent_co-locations (C_k, V, *min_vfprev*);
10) $P = P \cup P_k$;
11) $k = k + 1$;
12) end while

The CPM-IDPCFNR algorithm mainly includes the following three steps:

1) Calculate the fuzzy neighbor relationship of the spatial data set (line 1). According to the proximity function of the fuzzy neighbor relationship, the grid division technique is adopted to

calculate the fuzzy neighbor relationship of the spatial data set;

2) Cluster the spatial data set (lines 2–5). The local density and relative distance of the instances based on FNR are calculated. Then the cluster center instances are automatically generated according to the local density mean and relative distance variance. Finally, the remaining instances are assigned to the clusters overlappingly.

3) Filter prevalent co-location patterns (lines 7–12). Generate size-k candidate co-location patterns from size-k-1 prevalent co-location patterns, calculate the cluster fuzzy participation index of each candidate co-location, and select the cluster fuzzy participation index no less than the minimum cluster fuzzy participation threshold.

6. Experiments

This section conducts an experimental evaluation of Algorithm 1 (CPM-IDPCFNR algorithm) on real data sets and synthetic data sets. The primary purpose is: 1) Evaluate the three adopted strategies by comparing the densities of instances, the clusters selected and overlapping allocation of instances CPM-IDPCFNR and DPC-MPC [2] , and evaluate the effectiveness of the mining results of CPM-IDPCFNR by analyzing and comparing the mining results of CPM-IDPCFNR, CPFNR [4] and DPC-MPC. The prevalent measures of the three algorithms are cluster fuzzy participation index VFPI, fuzzy participation index FPI and fuzzy participation index DFPI (named in this paper) respectively. 2) Evaluate the performance of CPM-IDPCFNR in generating prevalent co-locations by analyzing and comparing the number of prevalent co-locations generated, running time and memory consumption.

All the algorithms in the experiment are implemented in Java language and run on a Windows 7 operating system PC with an Intel Core i7-6700 processor, a main frequency of 3.4 GHz, and a memory of 8 GB.

6.1. Data set

Two real data sets are used in the experiments in this section. The real data set Real-1 is a rare plant data set in the “Three Rivers in Parallel” region of Yunnan Province. It contains 31 features and a total of 336 instances. The vegetation distribution data set of Gaoligong Mountains, which contains 25 features and a total of 13,350 spatial instances. The synthetic datasets in the experiment are generated using the data generator similar to the literature [1]. All data sets are normalized to a 2000 × 2000 space. The proximity function of FNR of all spatial data sets used in the experiment is defined as follows:

$$\mu(d)=\begin{cases}1 & d\le a\\ -\dfrac{(d-a)^2}{(b-a)^2}+1 & a<d\le b\\ 0 & d>b\end{cases} \tag{10}$$

where, d is the Euclidean distance between any two instances in the spatial data set, a and b are the parameters of proximity function.

6.2. Evaluation of the effectiveness of the proposed algorithm

The related parameter settings in the experiments are shown in Table 4.

Table 4. Experimental parameter information on real data sets.

Parameters	Meaning	CPM-IDPCFNR		CPFNR		DPC-MPC	
		Real-1	Real-2	Real-1	Real-2	Real-1	Real-2
a	proximity function parameter	20	20	20	20	-	-
b	proximity function parameter	150	100	150	100	-	-
$\alpha/\beta/\lambda$	proximity function	0.1(β)	0.1(β)	0.01(α)	0.01(α)	0.2(λ)	0.2(λ)
d_c	cutoff distance	54	32	54	32	54	32
$min\_(v)\,fprev$	minimum(cluster) fuzzy participation index	0.3					

6.2.1. Evaluation of the three strategies adopted

In this section, to evaluate the three strategies adopted in our algorithm, we analyze some intermediate results of the CPM-IDPCFNR algorithm and the DPC-MPC algorithm on real-1. Tables 5 and 6 list the top 10 local densities based on FNR (ρ_i^{FNR}) and their instance indexes(i) in the CPM-IDPCFNR algorithm as well as the the top 10 local densities(ρ_i) and the instance indexes(i) of the DPC-MPC algorithm. We can observe that ρ_i^{FNR} can tell the difference in density while ρ_i cannot do (several instances share the same ρ_i). We employ the Non-Repeat Ratio (NRR) as the effectiveness indicator of the density value, $NRR = \frac{n_{non}}{n_{total}} * 100\%$, where n_{non} is the number of no repeated non-zero density values, n_{total} is the number of non-zero density values. We can get that $NRR(\rho_i^{FNR}) = 99.8\%$, and $NRR(\rho_i) = 0$. Therefore, thanks to the FNR strategy, ρ_i^{FNR} is more effective than $\rho^{FNR.}$. Moreover, because the relative distances of instances are obtained according to the local density, the relative distance based on FNR δ_i^{FNR} of CPM-IDPCFNR is more effective than the relative distance δ_i of DPC-MPC.

Table 5. Top 10 ρ_i^{FNR} and their corresponding instance indexes of CPM-IDPCFNR on Real-1.

index(i)	128	155	11	32	30	56	315	12	153	48
ρ_i^{FNR}	6.829	6.752	5.877	5.866	5.835	5.807	5.744	4.996	4.904	4.885

Table 6. Top 10 ρ_i and their corresponding instance indexes of DPC-MPC on Real-1.

index(i)	128	155	11	30	32	56	315	8	12	33
ρ_i	7	7	6	6	6	6	6	5	5	5

Table 7. Comparison of the selected cluster center instances by different selection strategies.

Algorithms	Selection Strategy	Indexes of the cluster centers
CPM-IDPCFNR	automatically generation	128, 32, 56, 8, 319, 14, 162, 24, 104, 99, 40, 191, 211, 116, 138, 13, 258
DPC-MPC	decision graph	128, 56, 315

Table 7 compares the clusters generated by different selection strategies of the two algorithms. We can find that CPM-IDPCFNR gets more clusters than DPC-MPC with only three clusters, and the former case is more consistent with the characteristics of spatial instances gathering in a

relatively small range. Table 8 compares the times of each instance overlapping allocation by the two algorithms. In the DPC-MPC algorithm, each instance is assigned to all clusters, which has been proved in Section I that the mining results lack of accuracy. In the CPM-IDPCFNR algorithm, each instance is allocated to no more than 3 clusters, which is consistent with the basic characteristics of the spatial proximity of instances compared to that of DPC-MPC.

Table 8. Comparison of the overlapping allocation times of instance.

Algorithms	Overlapping allocation times	Number of instances
CPM-IDPCFNR	3	2
	2	39
	1	191
	0	87
DPC-MPC	3 (equal to the number of clusters)	333

Table 9. Size-2 prevalent co-location patterns on Real-1.

Co-locations	CPFNR		CPM-IDPCFNR		DPC-MPC	
	FPI	Rank	VFPI	Rank	DFPI	Rank
{Fritillaria delavayi, ordyceps}	0.5871	1	0.5129	1	0.5741	106
{Fritillaria delavayi, Megacarpaea delavayi Franch}	0.5702	2	0.4139	9	0.6807	16
{Long bract fir, Fritillaria delavayi}	0.5605	3	0.4399	5	0.6807	16
{taxus yunnanensis, cephalotaxus lanceolata}	0.5335	4	0.3705	16	0.4578	301
{Long bract fir, Saussurea gossypiphora}	0.5078	5	0.5062	2	0.7572	4
{Yunnan fish wood, taxus yunnanensis}	0.4874	6	0.3502	21	0.4578	301
{Long bract fir,Trillium tschonoskii}	0.4749	7	0.4273	32	0.4064	407
{taxus yunnanensis, Yunnan fish wood}	0.4727	8	0.4204	7	0.4931	191
{Long bract fir, Megacarpaea delavayi Franch}	0.4611	9	0.4194	10	0.7572	4
{Long bract fir, ordyceps}	0.4601	10	0.3564	6	0.5910	79

Table 10. Size-3 prevalent co-location patterns on Real-1.

Co-location	CPFNR		CPM-IDPCFNR		DPC-MPC	
	FPI	Rank	VFPI	Rank	DFPI	Rank
{Long bract fir, Fritillaria delavayi, Megacarpaea delavayi Franch}	0.396	1	0.4194	1	0.6807	20
{taxus yunnanensis, Yunnan fish wood, cephalotaxus lanceolata}	0.3729	2	0.3502	4	0.4578	2300
{Fritillaria delavayi, Megacarpaea delavayi Franch, Cordyceps}	0.3719	3	0.3297	6	0.5741	455
{Long bract fir, Fritillaria delavayi, Cordyceps}	0.3605	4	0.3297	6	0.5741	455
{Long bract fir, Cordyceps,Saussurea gossypiphora}	0.3424	5	0.3911	63	0.4064	4064
{Saussurea gossypiphora, Fritillaria delavayi, Megacarpaea delavayi Franch}	0.3289	6	0.3126	16	0.6807	20
{Long bract fir, Saussurea gossypiphora, Fritillaria delavayi}	0.3191	7	0.2997	14	0.6807	20
{Long bract fir, Megacarpaea delavayi Franch, Cordyceps}	0.3174	8	0.3297	6	0.5741	455
{Long bract fir, Saussurea gossypiphora, Megacarpaea delavayi Franch}	0.3099	9	0.3126	14	0.7572	2
{magnolia sieboldii, Fritillaria delavayi, Cordyceps}	0.2859	10	0.3804	3	0.5741	455

6.2.2. Evaluation of the validity of the mining results

Tables 9 and 10 list the mining results of the size-2 and size-3 co-locations of the three algorithms on Real-1. The mining results of the size-2 and size-3 co-locations of the three algorithms on Real-2 are shown in Tables 11 and 12 respectively. It can be observed that 70% of the top 10 of the size-2 and size-3 co-locations mined by the CPM-IDPCFNR algorithm are in that of the CPFNR algorithm, which is more accurate than the classic Join-less algorithm [4]. But no more than 20% of the top 10 of the size-2 co-locations of the DPC-MPC algorithm are in that of the CPFNR algorithm, and for the size-3 co-locations, less than 10%.

Table 11. Mining results of size-2 co-locations on Real-2.

Co-locations	CPFNR		CPM-IDPCFNR		DPC-MPC	
	FPI	Rank	VFPI	Rank	DPI	Rank
{broad-leaf forest, Pinus yunnanensis}	0.9393	1	0.8752	1	0.7907	99
{Alnus cremastogyne Burk, broad-leaf forest}	0.8586	2	0.8319	3	0.7907	99
{Alnus cremastogyne Burk, Pinus yunnanensis}	0.8384	3	0.839	2	0.8223	80
{Abies fabri , tsuga chinensis}	0.8228	4	0.6709	4	0.8825	47
{broad-leaf forest, tsuga chinensis}	0.7449	5	0.6639	5	0.7907	99
{Fargesia spathacea Franch, Abies fabri }	0.7138	6	0.4076	25	0.8825	47
{tsuga chinensis, Miscellaneous fill}	0.6755	7	0.5996	10	0.8863	38
{tsuga chinensis, Pinus yunnanensis}	0.6637	8	0.6128	9	0.8223	80
{Fargesia spathacea Franch, Miscellaneous fill}	0.6443	9	0.3021	34	0.9188	21
{Fargesia spathacea Franch, tsuga chinensis}	0.6056	10	0.4023	19	0.8863	38

Table 12. Mining results of size-3 co-locations on Real-2.

Co-locations	CPFNR		CPM-IDPCFNR		DPC-MPC	
	FPI	Rank	VFPI	Rank	DPI	Rank
{Alnus cremastogyne Burk, broad-leaf forest, Pinus yunnanensis}	0.8159	1	0.8251	1	0.7907	374
{broad-leaf forest, tsuga chinensis, Pinus yunnanensis}	0.5847	2	0.5912	2	0.7907	374
{Fargesia spathacea Franch, Abies fabri , tsuga chinensis}	0.572	3	0.3902	17	0.8825	121
{Abies fabri, broad-leaf forest, tsuga chinensis}	0.5132	4	0.5226	4	0.7907	374
{Abies fabri , tsuga chinensis, Miscellaneous fill}	0.473	5	0.4579	7	0.8825	121
{Alnus cremastogyne Burk, tsuga chinensis, Pinus yunnanensis}	0.4159	6	0.4458	10	0.8223	287
{Fargesia spathacea Franch, tsuga chinensis, Miscellaneous fill}	0.4099	7	0.2976	25	0.8863	85
{Alnus cremastogyne Burk, broad-leaf forest, tsuga chinensis}	0.4018	8	0.4476	9	0.7907	374
{Quercus, Alnus cremastogyne Burk, broad-leaf forest}	0.3689	9	0.5331	3	0.7907	374
{tsuga chinensis, Pinus yunnanensis, Miscellaneous fill}	0.3665	10	0.4533	8	0.8223	287

Tables 13 and 14 list the co-locations in the top 10 of size-2 and size-3 co-location patterns of CPM-IDPCFNR that do not appear in the top 10 of the size-2 and size-3 co-locations of the mining results of the CPFNR algorithm. Figures 3 and 4 are the distribution diagrams of these co-locations respectively. It can be observed that the instances of these co-locations frequently appear together, and the size-3 co-locations on Real-1 all contain the rare feature “Berneuxia thibetica”, the size-2

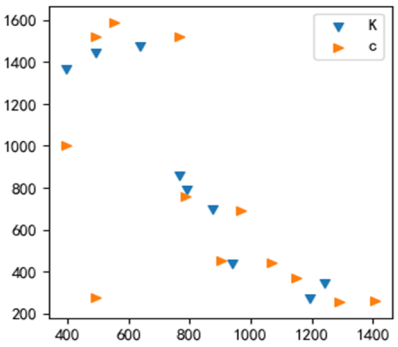
co-locations {Castanea mollissima, walnut} and {Quercus, Alnus cremastogyne Burk} on Real-2 are also composed of rare features.

Table 13. Partial top-10 size-2 and size-3 co-locations of the mining results of CPM-IDPCFNR on Real-1.

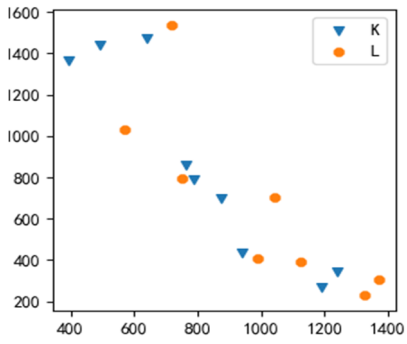
Size -2 co-locations	VFPI	Rank	Size-3 co-location	VFPI	Rank
{magnolia sieboldii, Cordyceps}	0.4781	3	{Long bract fir, Berneuxia thibetica,Saussurea ssipiphora}	0.3812	2
{Long bract fir, Fritillaria delavayi}	0.4487	4	{Long bract fir, Berneuxia thibetica, Fritillaria delavayi}	0.3491	5
{taxus yunnanensis, Fritillaria delavayi}	0.4195	8	{Berneuxia thibetica, Saussurea ssipiphora,Fritillaria delavayi}	0.3282	10

Table 14. Partial top-10 size-2 and size-3 co-locations of the mining results of CPM-IDPCFNR on Real-2.

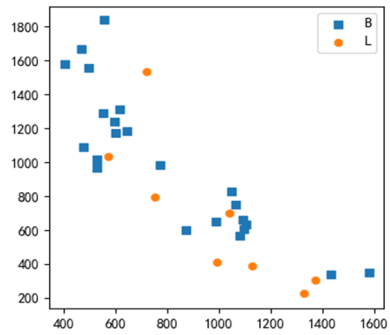
Size-2 co-location	VFPI	Rank	Size-3 co-location	VFPI	Rank
{Castanea mollissima, walnut}	0.6632	6	{Quercus, broad-leaf forest, Pinus yunnanensis}	0.5075	5
{Quercus, Alnus cremastogyne Burk}	0.6567	7	{Quercus, Alnus cremastogyne Burk, Pinus yunnanens}	0.5049	6
{pinus densata, Birch}	0.6414	8			



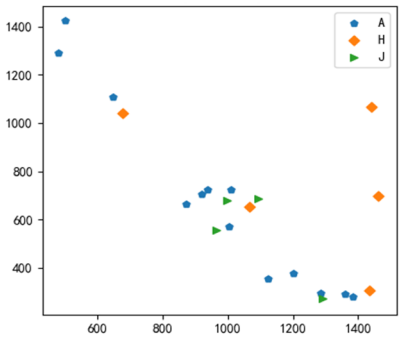
(a){magnolia sieboldi(K), Cordyceps(c)}



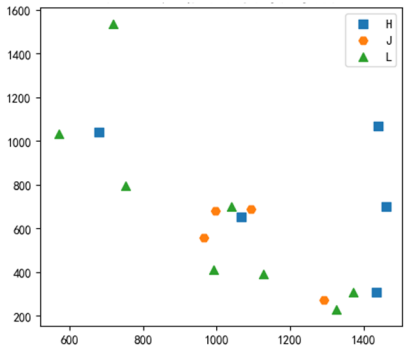
(b){Cordyceps(K), Fritillaria delavayi(L)}



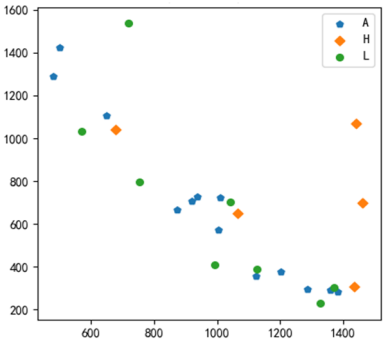
(c){taxus yunnanensis(B), Fritillaria delavayi(L)}



(d) {Long bract fir(A), berneuxia thibetica decne(H), Saussurea gossypiphora(J)}

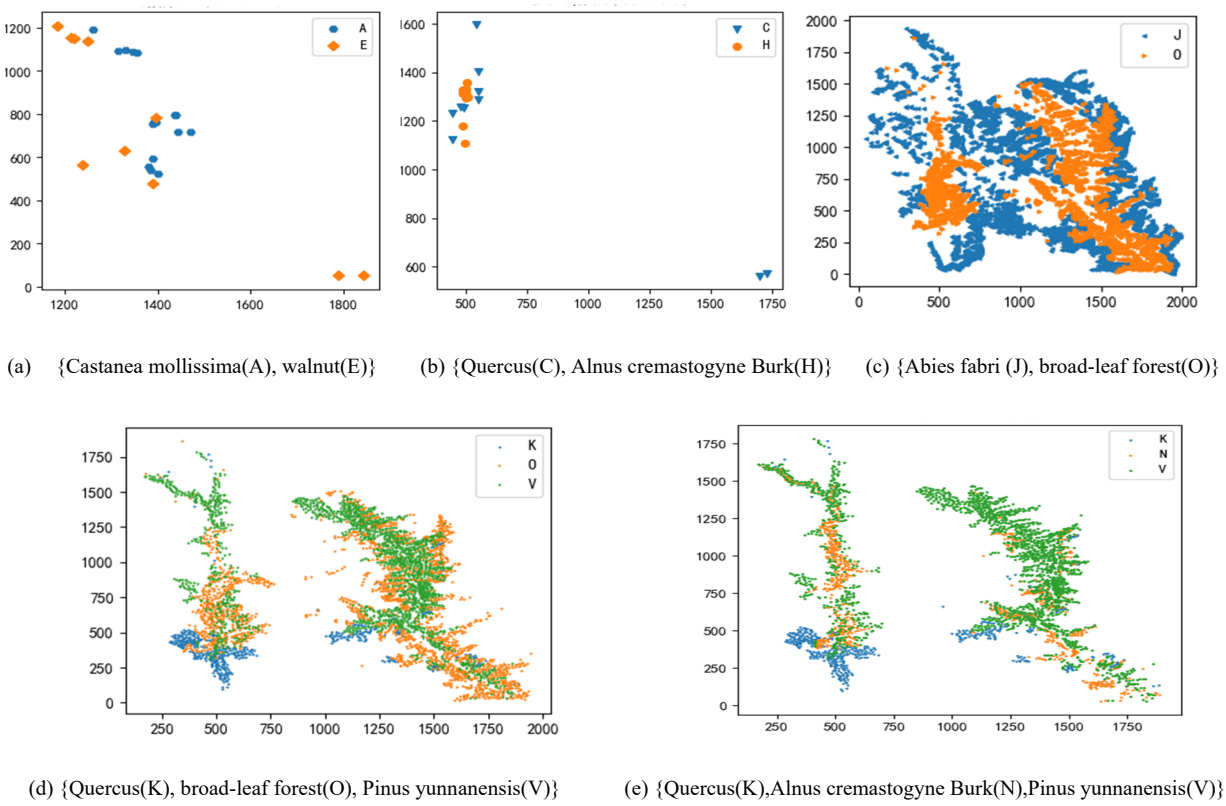


(e){berneuxia thibetica decne(H), Saussurea gossypiphora(J), Fritillaria delavayi(L)}



(f) {Long bract fir(A), davidia involucrata(H), Cordyceps(L)}

Figure 3. The distribution diagram of instance of partial co-locations on Real-1.



(a) {Castanea mollissima(A), walnut(E)} (b) {Quercus(C), Alnus cremastogyne Burk(H)} (c) {Abies fabri (J), broad-leaf forest(O)}

(d) {Quercus(K), broad-leaf forest(O), Pinus yunnanensis(V)} (e) {Quercus(K),Alnus cremastogyne Burk(N),Pinus yunnanensis(V)}

Figure 4. The distribution diagrams of instances of partial co-locations on Real-2.

It can be seen from the above that the mining results of the CPM-IDPCFNR algorithm is more effective than the DPC-MPC algorithm and truly reflect the actual distribution of instances, and the co-locations with rare features are more likely to be mined.

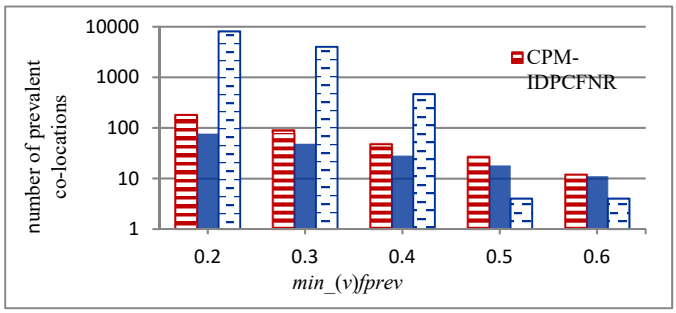
6.3. Performance evaluation of generating prevalent co-location patterns

This section analyzes and compares the number of prevalent co-location patterns generated by the CPM-IDPCFNR algorithm, DPC-MPC algorithm, and CPFNR algorithm on Real-2 and the synthetic data set, as well as the running time and memory consumption during the phase of generating prevalent patterns to evaluate the performance of the CPM-IDPCFNR algorithm.

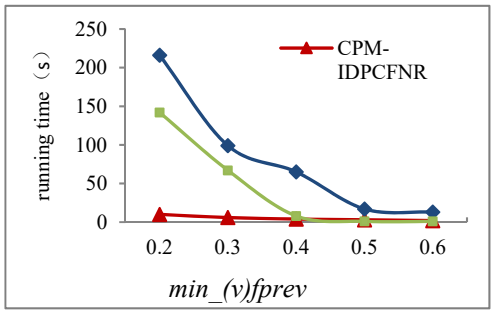
6.3.1. Performance evaluation on real data sets

The effect of the minimum (cluster) fuzzy participation index threshold. Figure 5 shows the number of prevalent co-locations, running time, and memory consumption generated by the three algorithms when the fuzzy participation index threshold takes different values on Real-2(the other parameter settings are shown in Table 4). It can be observed that the CPM-IDPCFNR algorithm produced more prevalent co-locations than the CPFNR algorithm, but the former consumes much less time than the latter. When the fuzzy participation index threshold is no less than 0.5, the DPC-MPC algorithm generates more prevalent co-locations than the CPFNR algorithm, but the former consumes significantly less time than the latter too. From the perspective of memory consumption, the memory cost of the CPM-IDPCFNR algorithm and the DPC-MPC algorithm is

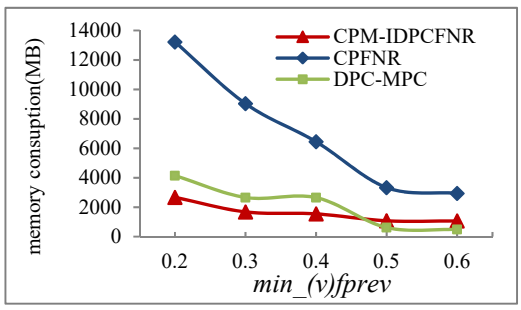
much lower than that of the CPFNR algorithm, and the smaller the (cluster) fuzzy participation index threshold, the greater the difference in memory cost. Because the CPFNR algorithm must check the clique relationship of star instances when generating fuzzy row instances of candidate patterns, which is very time-consuming. The smaller the fuzzy participation index threshold, the more candidate co-locations generated, the more time the clique relationship test process consumes, then the more memory to store fuzzy row instances. The other two algorithms use clustering technology to materialize the spatial proximity relationship between the instances. The proximity relationship is simplified within the cluster. It does not require time and space to generate and store the cluster row instances of the candidate co-locations, saving a lot of time and space. It can also be observed that as the fuzzy participation index threshold decreases, the number of prevalent co-locations generated by the DPC-MPC algorithm grows rapidly. When the fuzzy participation index threshold is no less than 0.5, the number of prevalent co-locations generated by the DPC-MPC algorithm is smaller than that of the other two. But when the fuzzy participation index threshold is less than 0.5, the number of prevalent co-locations it generates is much higher than that of the other two, resulting in a greatly increased time and memory consumption than that of the CPM-IDPCFNR algorithm. Because when the DPC-MPC algorithm assigns instances to clusters overlappingly, fuzzy clustering technology is used to force an instance to be assigned to each cluster with a certain membership, which leads to a much greater participation index of the co-locations than that of the other two, easily leading to the super co-location and its subsets have the same participation index. Therefore, when the fuzzy participation index threshold becomes smaller, the generation of high size co-locations will increase rapidly, and the time consumed will also increase sharply. However, the memory consumption increases relatively slowly, which is far lower than that of the CPFNR algorithm.



(a) Number of prevalent co-locations



(b) Running time

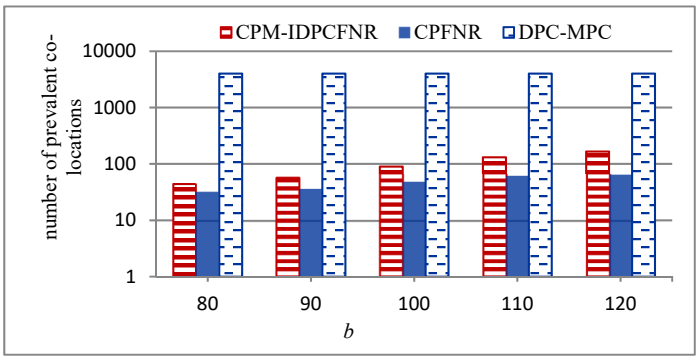


(c) Memory consumption

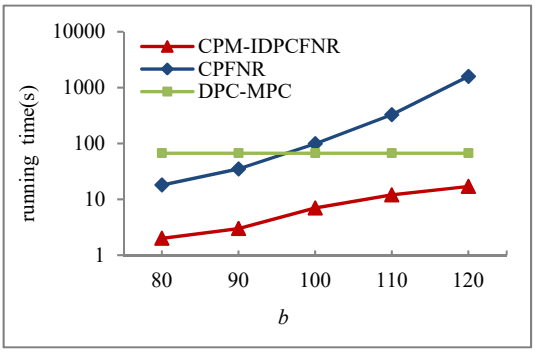
Figure 5. The effect of the minimum (cluster) fuzzy participation index threshold.

As can be seen from the above description, since the mining of co-location patterns based on clustering saves time and memory, the CPM-IDPCFNR algorithm shows good performance when generating prevalent co-locations.

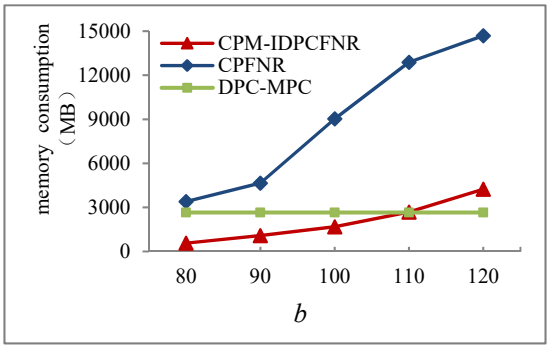
The effect of proximity function. The values of the experimental parameters are shown in Table 4. Figure 6 shows the number of prevalent co-locations, running time and memory consumption generated by the three algorithms when the proximity function parameter b (distance threshold) on Real-2 takes different values. Since there is no proximity function in the DPC-MPC algorithm, the number of generated prevalent co-locations, running time and memory consumption remain unchanged, while the DPC-MPC algorithm produced more prevalent co-locations than the other two algorithms. Since the time and memory consumed by the CPFNR algorithm increase with the increase of the parameter b. When b is less than 100, the running time of the DPC-MPC algorithm is higher than that of the CPFNR algorithm. When b is greater than 100, the former consumes less time and memory than the latter. The time and memory consumed by the CPM-IDPCFNR algorithm also increases slowly as b increases, and it is always significantly less than that of the CPFNR algorithm. The larger the b, the greater difference between the time and memory consumption of the two. But the number of prevalent co-locations generated by the CPM-IDPCFNR algorithm is always higher than that of the CPFNR algorithm during the whole process of b change. The above description once again illustrates that the clustering-based co-location pattern mining framework saves time and space when generating prevalent co-location patterns.



(a) Number of prevalent co-locations



(b) Running time



(c) Memory consumption

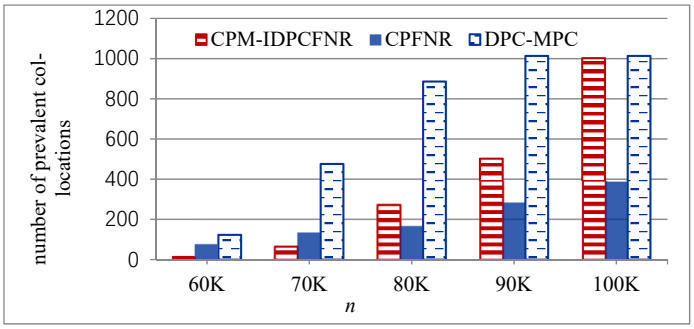
Figure 6. The effect of proximity function.

6.3.2. Performance evaluation on synthetic data sets

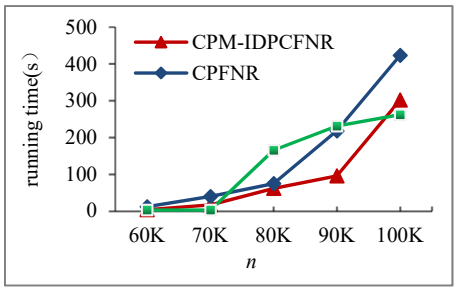
The experimental parameter information on the synthetic data set is shown in Table 15.

Table 15. The experimental parameter information on the synthetic data set.

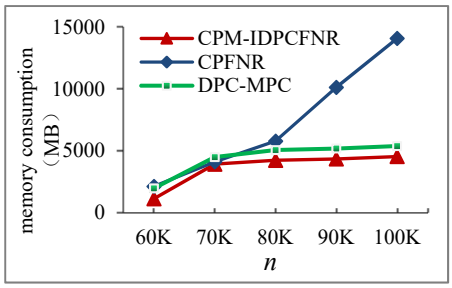
Parameters	Means	Default value		
		CPM-IDPCFNR	CPFNR	DPC-MPC
a	parameter of proximity function	5	5	—
b	parameter of proximity function	20	20	—
$\alpha/\beta/\lambda$	proximity(membership) threshold	0. 1(β)	0.01(α)	0.4(λ)
d_c	cutoff distance		15	
N	number of feature		10	
$min\_(v)fprev$	minimum (cluster) fuzzy participation threshold		0.4	



(a) Number of prevalent co-locations



(b) Running time



(c) Memory consumption

Figure 7. The effect of the number of instances.

The effect of the number of instances. Figure 7 shows the number of prevalent co-locations, running time, and memory consumption of the three algorithms when the number of instances is 60K, 70K, 80K, 90K and 100K (the other parameters are shown in Table 15). As shown in Figure 7(a), the number of prevalent co-locations generated by the three algorithms increases as the number of instances increases. The DPC-MPC algorithm has the fastest growth rate, while the CPFNR algorithm has the slowest. The number of prevalent co-locations generated by the DPC-MPC algorithm is always the largest, while that of the CPFNR algorithm is always the smallest. The

CPM-IDPCFNR algorithm generates more prevalent co-locations than the CPFNR algorithm, but consumes less time and memory. When the number of instances is less than 80K, the DPC-MPC algorithm consumes less time than the other two because it generates fewer clusters than the CPM-IDPCFNR algorithm in the clustering process. When the number of instances is 80K and 90K respectively, the number of prevalent co-locations generated by the DPC-MPC algorithm is higher than that of the other two; thus it takes more time than the other two. However, when the number of instances is 100K, the DPC-MPC algorithm and the CPM-IDPCFNR algorithm produce the same number of co-locations, and the former consumes less time but more memory. When the number of instances is 70K, the DPC-MPC algorithm consumes more memory than the CPFNR algorithm (the former generates more prevalent co-locations). In other cases, the former consumes less memory than the latter. When the number of instances is not less than 70K, the memory consumption of the two cluster-based co-locations mining algorithms grows extremely slowly, showing that the cluster-based co-location mining framework has good stability in memory consumption.

The effect of membership threshold. Figure 8 shows the time consumption of generating prevalent co-location patterns of the CPM-IDPCFNR algorithm on three synthetic datasets (60K, 80K, 100k) as the membership threshold β varies (the other parameters are shown in Table 15). It can be seen that with the increase of β, the running time on each dataset decreases. Because the larger the β, the smaller the cluster size, the less prevalent co-locations generated, and thus the less time consumed.

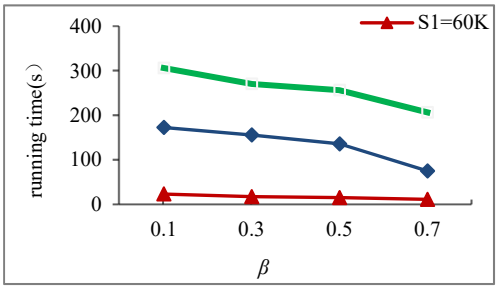


Figure 8. The effect of membership threshold.

7. Conclusions

This paper explores a co-location pattern mining framework based on clustering and studies the co-location pattern mining algorithm based on a combination of improved density peak clustering and fuzzy proximity relationships. Three improvement strategies are adopted for the classic density peak clustering, and a new prevalence measure of the co-location, cluster fuzzy participation index, is proposed. A co-location pattern mining algorithm based on the improved density peak clustering and the fuzzy neighbor relationships is given. Experiments show that the proposed algorithm is effective and greatly saves time and space during the generation of prevalent co-location patterns. For the future plan, we will employ fuzzy trigonometric function or other fuzzy methods for the co-location mining, and also mean to extend the co-location pattern mining based on the clustering framework to the other types of spatial data, e.g., uncertain data and spatial-temporal data.

Acknowledgments

This work is supported in part by the key research program of Science and Technology Department of Yunnan Province (No.2019BC003), in part by grants (No.2021J0695, No.2020J0484) from the Education Department of Yunnan Province Foundation and in part by grants (No.19A021, No.19A010) from Scientific research project of Yunnan Police College.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Y. Huang, S. Shekhar, H. Xiong, Discovering colocation patterns from spatial data sets: a general approach, *IEEE Educ. Act. Dep.*, **16** (2004), 1472–1485.
2. Y. Fang, L. Wang, T. Hu, Spatial co-location pattern mining based on density peaks clustering and fuzzy theory, in *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, Springer, Cham, (2018), 298–305.
3. M. Du, S. Ding, X. Yu, A robust density peaks clustering algorithm using fuzzy neighborhood, *Int. J. Mach. Learn. Cybern.*, **12** (2017), 1–10.
4. M. Wang, L. Wang, L. Zhao, Spatial co-location pattern mining based on fuzzy neighbor relationship, *J. Inf. Sci. Eng.*, **35** (2019).
5. S. Shekhar,Y. Huang, Discovering spatial co-location patterns: A summary of results, in *International symposium on spatial and temporal databases*, Heidelberg, Springer, (2001), 236–256.
6. J. S. Yoo, S. Shekhar, M. Celik, A join-less approach for colocation pattern mining: a summary of results, in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, IEEE, (2005), 813–816.
7. J. S. Yoo, S. Shekhar, J. Smith, J. P. Kumquat, A partial join approach for mining co-location patterns, in *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, ACM Press, (2004), 241–249.
8. L. Wang, Y. Bao, J. Lu, J. Yip, A new join-less approach for co-location pattern mining, in *2008 8th IEEE International Conference on Computer and Information Technology*, IEEE, (2008), 197–202.
9. L. Wang, Y. Bao, Z. Lu, Efficient discovery of spatial co-location patterns using the iCPI-tree, *Open Inf. Syst. J.*, **3** (2009), 69–80.
10. L. Wang, P. Wu, H. Chen, Finding probabilistic prevalent co-locations in spatially uncertain data sets, *IEEE Trans. Knowl. Data Eng.*, **25** (2013), 790–804.
11. L. Wang, P. Guan, H. Chen, L. Zhao, Mining co-locations from spatially uncertain data with probability intervals, in *International Conference on Web-Age Information Management*, Springer, (2013), 301–314.
12. L. Wang, H. Chen, L Zhao, L. Zhou, Efficiently mining co-location rules on interval data, in *International Conference on Advanced Data Mining and Applications*, Springer, Berlin, (2010), 477–488.
13. L. Wang, X. Bao, L. Zhou, Redundancy reduction for prevalent co-location patterns, *IEEE Trans. Knowl. Data Eng.*, **30** (2008), 142–155.

14. L. Wang, X. Bao, H. Chen, Effective lossless condensed representation and discovery of spatial co-location patterns, *Inf. Sci.*, **436** (2018), 197–213.
15. L. Wang, L. Zhou, J. Lu, J. Yip, An order-clique-based approach for mining maximal co-locations, *Inf. Sci.*, **179** (2009), 3370–3382.
16. X. Yao, L. Peng, L Yang, T. Chi, A fast space-saving algorithm for maximal co-location pattern mining, *Expert Syst. Appl.*, **63** (2016), 310–323.
17. J. S. Yoo, M. Bow, Mining maximal co-located event sets, in *Pacific-Asia conference on knowledge discovery and data mining*, (2011), 351–362.
18. J. S. Yoo, D. Boulware, D. Kimmey, A parallel spatial co-location mining algorithm based on MapReduce, in *2014 IEEE International Congress on Big Data*, (2014), 25–31.
19. P. Yang, L. Wang, X. Wang, A parallel spatial co-location pattern mining approach based on ordered clique growth, in *International Conference on Database Systems for Advanced Applications*, (2018), 734–742
20. Z. Ouyang, L. Wang, P. Wu, Spatial co-location pattern discovery from fuzzy objects, *Int. J. Artif. Intell. Tools*, **26** (2017), 1750003.
21. P. Wu, L. Wang, Y. Zhou, Discovering co-location from spatial data sets with fuzzy attributes, *J. Front. Comput. Technol.*, **7** (2013), 348–358.
22. A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, *Science*, **344** (2014), 1492–1496.
23. J. Xie, H. Gao, W. Xie, X. Liu, P. W. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors, *Inf. Sci. Int. J.*, **350** (2016), 19–40.
24. X. Xu, S. Ding, Z. Shi, An improved density peaks clustering algorithm with fast finding cluster centers, *Knowl.-Based Syst.*, **158** (2018), 65–74.
25. R. Liu, H. Wang, X. Yu, Shared-nearest-neighbor-based clustering by fast search and find of densitypeaks, *Inf. Sci.*, **450** (2018), 200–226.
26. R. Liu, W. Huang, Z. Fei, K. Wang, J. Liang, Constraint-based clustering by fast search and find of density peaks, *Neurocomputing*, **330** (2019), 223–237.
27. S. Yan, H. Wang, T. Li, J. Chu, J. Guo, Semi-supervised density peaks clustering based on constraint projection, *Int. J. Comput. Intell. Syst.*, **14** (2020),140–147.
28. R. Mehmood, R. Bie, H. Dawood, H. Ahmad, Fuzzy clustering by fast search and find of density peaks, in *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)*, (2015), 258–261.

AIMS Press