MPCD: An Algorithm for Discovering Multilevel Prevalent Co-location Patterns from Heterogeneous Distribution of Spatial Datasets

Vanha Tran(✉), Thiloan Bui, and Hoangan Le

FPT University, Hanoi 155514, Vietnam
{hatv14,loanbt7}@fe.edu.vn, anlhhe163613@fpt.edu.vn

Abstract. Discovering spatial prevalent co-location patterns (SPCPs) is an important area of research in spatial data analysis, aiming to identify hidden spatial relationships between objects (spatial instances), e.g., their simultaneous occurrences in space. The significance of discovering SPCPs is evident in various fields such as medicine, biology, urban planning, and many others. However, many challenges are currently faced in the process of discovering SPCPs, including dealing with heterogeneous distribution spatial datasets and the computational cost required for multilevel SPCP discovery, which often encounters inefficient time complexity. To address these challenges, in this paper, we propose a new method applied density-wise clustering. After splitting the dataset into clusters based on both density levels and distribution, which we called 'density-wise clusters', we adopt the Bron-Kerbosch algorithm and utilize a MC-hash structure to enhance the SPCP mining process. Our method is examined and compared with other two efficient methods on both simulated and real-world datasets. The results show that our proposed method not only effectively addresses the mentioned difficulties but also yields noteworthy results in discovering SPCPs in spatial data.

Keywords: Multilevel co-location patterns · Heterogeneous distribution · Density-wise clusters · MC-hash structure

1 Introduction

With the relentless advancement of technology, a vast amount of spatial data is collected daily. Among these, spatial prevalent co-location pattern (SPCP) mining is considered a crucial component since it can expose. SPCPs demonstrate sets of objects co-occurring in space with significant frequency [8]. Therefore, exploring and studying SPCPs have the potential to provide insights for the development and application in various fields such as urban planning [18], public health [11], business [20], transportation [21], agricultural science [13], etc.

However, determining SPCPs faces many challenges due to the uneven distribution of real-world data. Figure 1 is an example of data points in space with

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14814, pp. 101–119, 2024.
https://doi.org/10.1007/978-3-031-64608-9_7

different distributions and density variations. Traditional SPCP mining methods often rely on a single threshold to measure the association (e.g., neighbor relationship) between objects (spatial instances) in space. This can lead to inaccuracies in output, e.g., if the threshold is too low, sparse events may be easily underestimated, or conversely, if the threshold is set too high, less significant events may be wrongly evaluated. Other existing methods consider the relationships between instances by partitioning data into regions or clusters with different densities [15,19]. While this strategy demonstrates high efficiency with relatively accurate results, computational costs increase significantly, slowing down the mining process due to the need to leverage abundant data volumes at the current time.

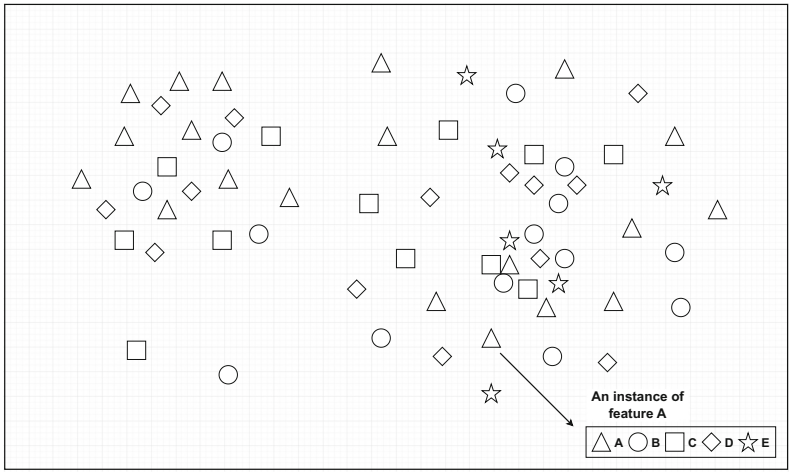


Fig. 1. Example of heterogeneous distribution data.

To address these limitations, we propose a method to improve the accuracy and speed of mining SPCPs in heterogeneous distribution spatial data. Our method consists of three stages, as follows:

1. We partition the input dataset into different subsets based on density levels of data objects. Subsequently, on each subset, the fuzzyDBSCAN [9] algorithm is employed to separate the points into 'density-wise clusters'.
2. Within each density-wise cluster, all maximal cliques are determined using the robust Bron-Kerbosch algorithm [10]. From there, candidate co-location patterns are identified. In this stage, a compressed maximal clique hash (MC-hash for short), which is developed to store the neighbor relationships between instances compactly, is used for reducing computational cost.
3. Finally, across all regions, combining SPCPs in these regions are examined to determine whether they are prevalent enough to become global patterns based on the global index (GI). If not, they are claimed as local patterns.

The remaining portion of the paper is organized as follows. Section 2 reviews relevant studies on this topic. Then, we detail the specific steps of the proposed method in Sect. 3. Section 4 presents experiments and analysis. Finally, the conclusion is led in Sect. 5.

2 Related Work

An important branch of spatial data mining is spatial co-location mining, which focuses on discovering co-occurring patterns of spatial objects within the same geographic area. Shekar et al. [16] laid the foundation for the field of SPCP mining and introduced algorithms for finding SPCPs based on set intersection operations. However, this algorithm has several drawbacks such as low efficiency, insufficient accuracy, and limited scalability. After that, there are many efficient algorithms developed such as joinless [23], overlapping maximal-based [17], instance driver schema [1], and so on.

In SPCP mining, the challenge lies in effectively handling large data and uneven distribution densities. This is because certain patterns may only emerge in specific regions (without clear hierarchical partitioning). However, in the context of mining multiple levels, this significantly impacts the accuracy and efficiency of the task. The existing algorithms for mining multilevel SPCPs currently can roughly divided into three categories, i.e., rule-based [12,15], clustering-based [3,13,19], and network constrained-based [6,14] algorithms. In which, employing clustering across the entire space is proved more efficiently to address the drawback in mining multilevel SPCPs in heterogeneous distribution data [19].

DBSCAN [5] is a clustering algorithm that uses density to determine clusters and noise points in data space. It does not require prior knowledge of the number or shape of clusters. This makes DBSCAN an effective tool in many real-world applications such as image analysis, signal processing, and spatial data mining. Compared to DBSCAN, Fuzzy DBSCAN [9] has several outstanding advantages, e.g., parameter selection is simple, it only requires a single parameter, i.e., the degree of fuzziness, instead of two parameters ϵ and $MinPts$ in DBSCAN. Fuzzy DBSCAN also has better ability to handle clusters with different densities and shapes, enhancing the effectiveness and flexibility of the clustering algorithm.

In paper [2], L-density is used to measure the density of data points in space. L-density, or density level, is a new concept that distinguishes between clusters with different densities and identifies clusters with varying density in data space. The L-density calculation method in k-DBSCAN provides a new approach to improving the performance and results of clustering algorithms, especially when data exhibits density variations.

Eppstein et al. presented the Bron-Kerbosch algorithm [4] as a recursive algorithm to enumerate all maximal cliques in a graph. This algorithm works by narrowing down potential candidate sets and recursively calling to search for maximum cliques. It is quite simple, easy to understand, highly efficient for sparse graphs, and significantly improves performance compared to previous methods, making the analysis of complex networks more feasible.

Multi-level SPCP mining allows the detection of co-locations at different scales, from detailed and specific patterns to general and common patterns, or vice versa. This helps create a comprehensive view of the structure of geographical space and the close relationships between objects or locations. In this work, we design an efficient algorithm that employs the above technique, i.e., Fuzzy DBSCAN, maximal cliques, and SPCP mining to discover multilevel SPCPs.

3 Proposed Method

To efficiently explore multilevel SPCPs, the entire spatial dataset needs to be partitioned into subsets for processing based on their differing appearances. Data points with close relationships, holding the potential for co-location patterns, often exhibit similarity in density and distribution. This is the driving force behind the formation of density-wise clusters.

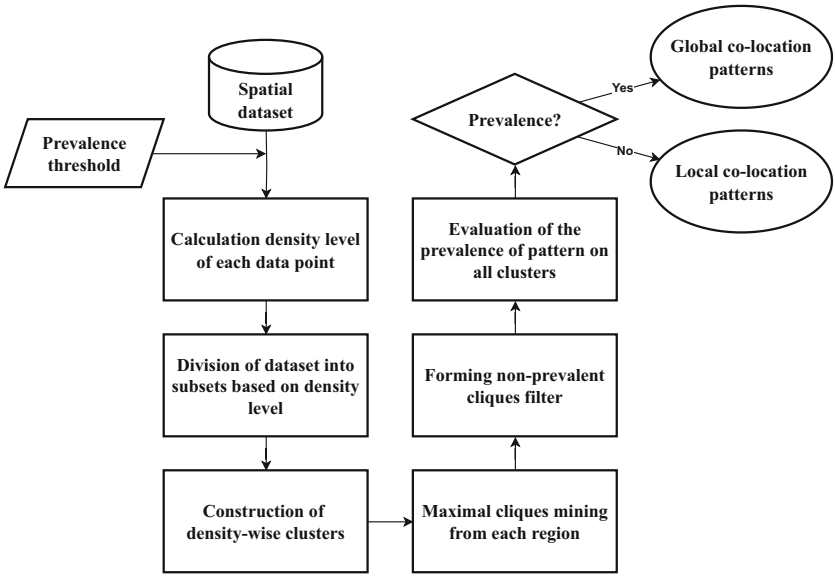
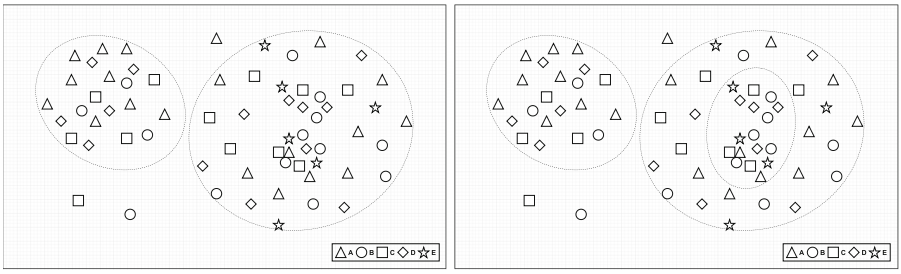


Fig. 2. The proposed framework for discovering multilevel SPCPs form heterogeneous distribution of spatial datasets.

Figure 2 shows the pipeline of the proposed method for discovering multilevel SPCPs from heterogeneous distribution of spatial dataset. Once density-wise clusters are formed across the entire dataset, SPCPS are explored. Starting with the search for maximal cliques, we optimize computation costs and operational time by leveraging the Apriori property and a compressed MC-hash. After SPCPs are found across all density-wise clusters, the global index (GI) is used to classify patterns into global and local categories. The following sections will provide specific instructions for this method.

3.1 Construction of Density-Wise Clusters

Heterogeneous spatial data always exhibit regions with varying densities and distributions. To achieve high efficiency and accuracy, the entire dataset will be partitioned into smaller groups based on the density and distribution of the data points. Partitioning the data based on only one of the two factors, density or distribution, is insufficient, as illustrated in the Fig. 3. Therefore, it is necessary to process the data based on both factors to form sets of data points in space that are more closely linked, which we will refer to as density-wise clusters.



(a) Data clustered based on distribution **(b)** Data clustered based on density

Fig. 3. Illustration of spatial data clustered by different factors.

Firstly, the dataset is partitioned into subsets based on density value using the k-level partitioning method. We borrow the definition from KDBSCAN [2]:

Definition 1 (l-Density). *The l-density of a point P, denoted as l-density(P), is defined as follows:*

$$l - density(P) = \frac{1}{l} \sum_{i=1}^{l} d_i \tag{1}$$

Definition 2 (Density Level). *Density level of a point P, denoted by $density - level(P)$ is an integer number, labeled by k-means algorithm. For two points P and Q, if their density levels are the same, then the $l - density$ of P and Q are approximately similar. Note that, density-level is only a categorical parameter.*

After applying the k-level partitioning method, we obtain subsets of data with the same density level, or subsets containing points with the same density. On each subset, we further proceed to partition the data points based on distribution using the Fuzzy DBSCAN [9] algorithm. The result we obtain is density-wise clusters containing points that are similar in density and distribution, demonstrating close coherence with each other. An example of density-wise clusters is shown in Fig. 4.

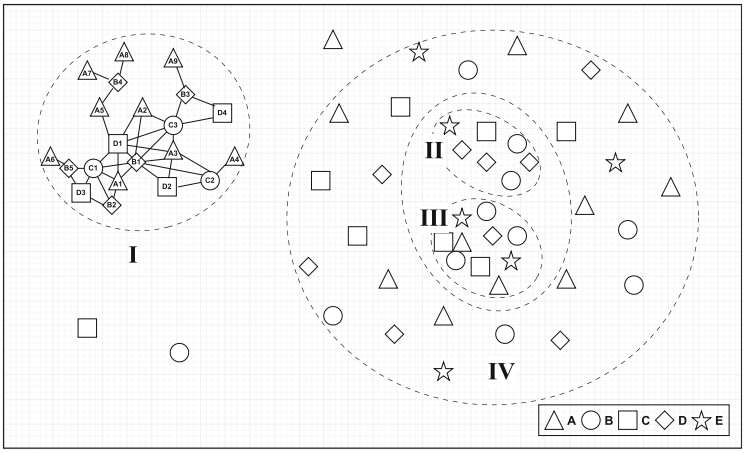


Fig. 4. Density-wise clusters forming based on both density and distribution.

3.2 Multi-level SPCP Mining from Clusters

After obtaining the density-wise clusters, the next step is to conduct mining on each of these clusters. Firstly, we will utilize the robust Bron Kerbosch algorithm [10] to find maximal cliques in each cluster.

Algorithm 1. Bron-Kerbosch-Pivot

P: A data set vertices of a cluster
R: A maximal clique
X: A set of vertices excluded from the clique
1: **if** $P \cup X = \emptyset$ **then**
2: report R as a maximal clique and store in MC-hash
3: **end if**
4: choose a pivot $u \in P \cup X$ {choose u to maximize $|P \cap \Gamma(u)|$}
5: **for** each vertex $v \in P \setminus \Gamma(u)$ **do**
6: BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
7: $P \leftarrow P \setminus \{v\}$
8: $X \leftarrow X \cup \{v\}$
9: **end for**

Algorithm 1 operates by traversing through potential vertex subsets and checking if they can be extended into maximal vertex subsets. This process is carried out recursively, with the selection of pivot vertices and removal of visited vertices aiding in improving the algorithm's efficiency. The final result is all maximal vertex subsets in the cluster. When applying algorithm 1 to cluster 5 in Fig. 4, discovered the maximal cliques show in the left side of Fig. 5.

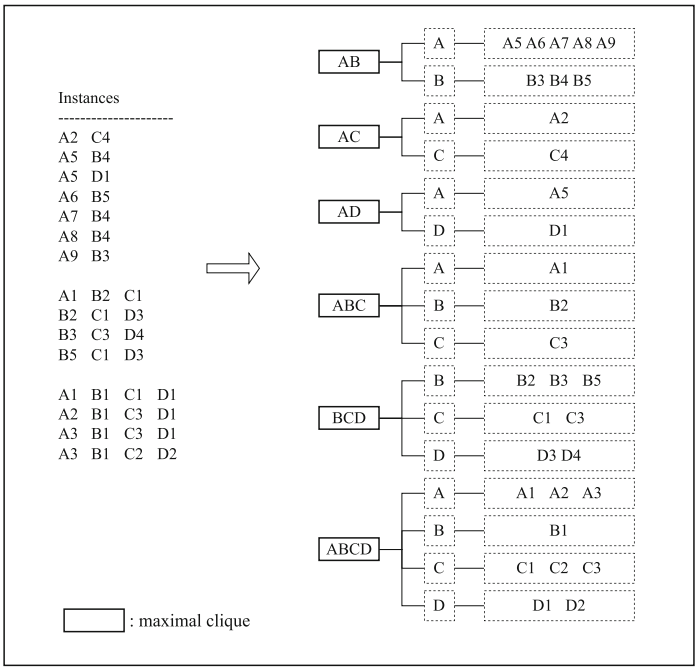


Fig. 5. MC-hash structure with the maximal cliques in cluster 5 of Fig. 4.

Each maximal cliques is saved into the MC-hash structure as shown in the right side of Fig. 5, with the key being the co-location pattern and the corresponding value being the list of neighbor objects. From the maximal cliques, patterns are mined based on one of the fundamental theories of data mining, namely the Apriori property, i.e., all subsets of a frequent item set must be frequent. If an item set is infrequent, all its super sets will be infrequent. Starting with patterns of size = 2, we use the participation index (PI) (that is shown in Eq. 2) as a measure of popularity:

$$PI(C_i) = min_{f_j \in C_i}\{Pr(C_i, f_i)\} \tag{2}$$

$$Pr(C_i, f_j) = \frac{|I(C_i, f_j)|}{|I(f_j)|} \tag{3}$$

where $|I(C_i, f_j)|$ is the number of unique instances of feature f_j in instances of Ci, and $|I(f_j)|$ is the number of instances of feature f_j in the entire study area or a sub-region. Given a threshold T, if $PI(C_i) \geq T$; then, C_i will be identified as a prevalent pattern; otherwise, C_i will be classified as a non-prevalent pattern.

Algorithm 2 checks the cliques in the MC-hash, χ. First, the prevalent patterns of size 2 are preserved, while the non-prevalent patterns of size 2 are used as a standard to eliminate non-prevalent patterns of larger sizes, helping to by

Algorithm 2. Discovering SPCP algorithm

```
Sk = ∅, L1 = ∅, L2 = ∅
for each clique zi in χ do
    if size of zi = k then
        check zi in Sk with conditional and add value χ of zi to Sk
    else if size of zi > k then
        subcliques = generate_combinations(zi)
        for each clique wi in subcliques do
            check wi in Sk with conditional and add all instances of wi to Sk
        end for
    end if
end for
for each co-location pattern cp in Sk do
    if cp not in non_prev_2 then
        Pi = calculate_pi(cp)
        if Pi ≥ θ then L1 add cp
        else L2 add cp
        end if
    end if
end for
return L1, L2
```

pass redundant computations. The set S will be used to contain non-prevalent size 2 patterns. Then, patterns of larger sizes, starting from 3, 4, 5, and so forth, will be mined. For each pattern, if the pattern contains any pattern from set S, then that pattern will be non-prevalent. If the pattern does not contain any pattern from set S, we will calculate the PI value of that pattern. If the PI value is greater than the threshold, the pattern will be considered prevalent. This entire process is further accelerated by utilizing a compressed maximal clique hash structure, i.e., MC-hash.

The search for prevalent patterns is carried out sequentially across all clusters. Once all prevalent patterns have been identified across the entire dataset, they are classified into global and local patterns. We propose a Global index (GI) calculated as the ratio of the number of clusters in which the pattern is prevalent to the total number of clusters:

$$GI(P) = \frac{\text{number of density - wise clusters that pattern P is prevalent}}{\text{total number of density - wise clusters}} \tag{4}$$

All patterns with a GI greater than the predefined threshold will be classified as global patterns. Correspondingly, the remaining patterns will be considered as local patterns.

The MPCD algorithm, i.e., Algorithm 3, initially, the algorithm construction of density-wise clusters by KD-tree structure, K-means and fuzzy DBSCAN to cluster data points into groups based on density. Subsequently, the algorithm multi-level pattern mining from clusters by mining maximal cliques and utilizes

Algorithm 3. The proposed algorithm, MPCD

Input: ρ: A data set points, θ: Prevalence threshold, α: Global threshold
Output: Γ: List of global co-location patterns, Λ: List of local co-location patterns

$\Gamma = \emptyset$, $\Lambda = \emptyset$, $S = \emptyset$
root = build_kdtree(ρ)
$k_neigh = 100$
Find_k_nearest_neighbors for each point in ρ with root
η = find_Ldens(ρ, root)
Δ = KMeans(η) ▷ Hash density level of all point
$\Theta = \emptyset$
for each data in Δ **do**
    Θ = fuzzyDBSCAN(Δ_i) ▷ Hash of all cluster after using fuzzy K-DBSCAN algorithm
end for
for each cluster Θ_i in Θ **do**
    $prev_2 = \emptyset$, $non_prev_2 = \emptyset$
    χ = BronKerboschWithPivot(Θ_i) ▷ Using MC-hash structure
    $prev_2, non_prev_2 \leftarrow$ mining_cliques_lv2(χ, θ)
    $S \leftarrow S \cup prev_2$
    max_len_co-location = max(χ)
    for $N = 3$ to max_len_co-location **do**
        $S \leftarrow S \cup$ mining_cliques_lvk($\chi, N, non_prev_2, \theta$) ▷ List prevalence co-location pattern
    end for
end for
for each prevalence co-location pattern ρ_i in S **do**
    if $\rho_i \geq \alpha$ **then** $\Gamma \leftarrow \Gamma \cup \rho_i$
    else$\Lambda \leftarrow \Lambda \cup \rho_i$
    end if
end for
return Γ, Λ

a data structure called MC-hash to store neighbor object subgroups of previously generated clusters. Finally, the algorithm classifies the discovered patterns into two types: global and local, based on a provided prevalence threshold.

4 Experiments and Analysis

In this section, comprehensive experiments were conducted to evaluate the effectiveness and efficiency of the proposed method. We carefully selected the following algorithms for comparison: For the KNNG method, the distance variation threshold is set at 0.6, as determined from experimental analysis conducted in the original study. For the Joinless method, the distance threshold is also determined using the L-function, as proposed by Yoo and Bow (2012) [22]. For our method, the parameter k-neighbors is set to 100 based on experimental results. All algorithms were coded in C++ and run on a computer with the configuration

of Chip Apple M1, Memory 16 GB, MacOS Ventura 13.0. All algorithms were run on both artificial and real datasets for comparison.

4.1 Experiments on Synthetic Datasets

This process generates simulated spatial datasets for evaluating three methods of spatial data handling. A 200 × 200 research area is divided into four 100 × 100 subareas and named regions I, II, III, and IV. Random parent points are generated in each subarea, controlled by the parameter $n-parent$. Child points are created around each parent point within a buffer zone defined by $r-parent$. These child points are then replaced by grandchildren points, each associated with a randomly chosen feature type from a set of $n-feature$ types. Noise points are added to increase data diversity, with parameters such as feature point density, interaction range, number of features, and noise level controlled by $r-parent$, $n-feature$, and $n-noise$. This process enables the generation of diverse datasets for evaluating spatial data processing methods.

4.1.1 Parameter Setting and Evaluation Measure

To efficiently explore co-location patterns within a heterogeneous distribution of spatial datasets, experiments have established a PI of 0.2 to avoid discovering insignificant small localities of local co-location patterns. The parameters for the three comparison methods were configured according to recommendations provided in the original studies:

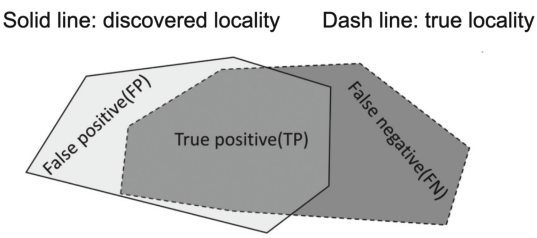


Fig. 6. Illustration of the measures for assessing the quality of discovered localities.

The measure developed by Guo and Wang (2011) [7] was employed to assess the accuracy of the multilevel co-location patterns discovered by the three methods. In Fig. 6, the dashed polygon represents a predefined area of the co-location pattern, while the solid polygon represents the area detected by the co-location pattern mining method. For each dataset, precision and recall can be calculated as follows:

$$Precision = \frac{\sum_{i=1}^{n} TP/(TP_i + FP_i)}{n} \tag{5}$$

$$Recall = 1 - \frac{\sum_{i=1}^{n} TP/(TP_i + FN_i)}{n} \tag{6}$$

Precision is calculated as the average ratio of true positive (TP) regions (the intersection of patterns generated by the specified instances and features on the ground truth dataset) over the total number of true positive and false positive (FP) regions (the sum of these regions represents what the methods explore, where FP is the complement region of TP). Additionally, the measurement method provides a Recall index with similar TP and false negative (FN) regions. A Precision score approaching 1 indicates that what the methods explore is more accurate. However, we also need to consider the Recall parameter to determine whether the algorithm is truly accurate. In order to make it easier to observe here, we use the recall formula in Eq. (6) that is using 1 to subtract the original recall value. Thus, in Eq. (6), as a higher Recall score indicates that many co-location patterns are still unexplored or misidentified.

4.1.2 Effect of Number Instance of Spatial Features

The data is generated with the following parameters: noise = 50, and the number of instances gradually increases: 5,000, 10,000, 15,000, 20,000, 25,000. Five features are generated as follows: [A, B, C] within region [I, III], [B, C, D, E] within region [II, IV]. Ten features are generated as follows: [C, D, E, F, G] within region [I, III], [A, B, C, D, H, I, J] within region [II, IV]. Fifteen features are generated as follows: [A, B, C, D, E, F, G, H, I, J] within region [I, III], [A, B, C, D, E, K, L, M, O] within region [II, IV]. The generated data set is illustrated in Fig. 7.

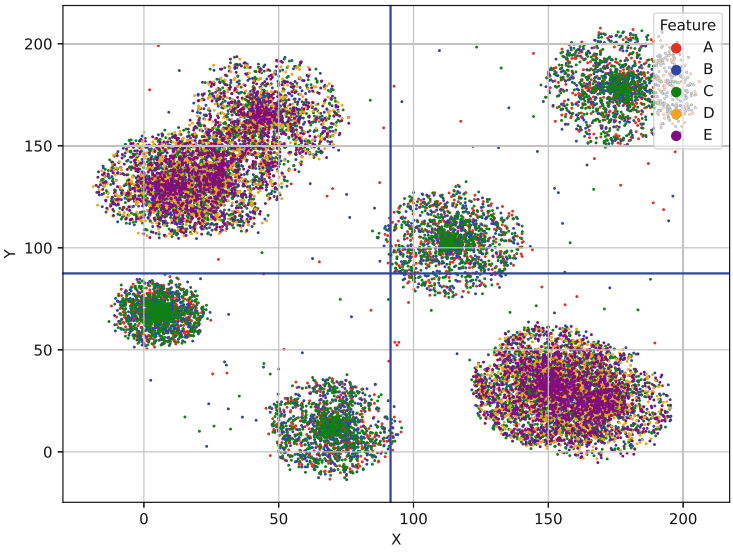
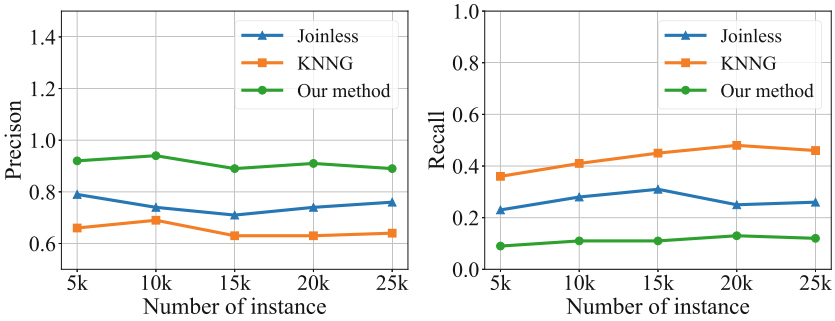
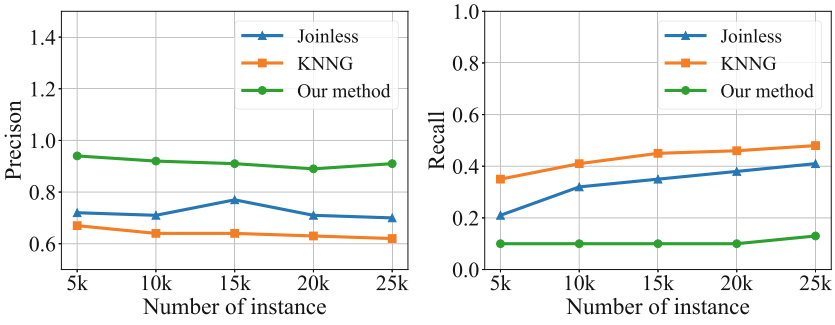


Fig. 7. The generated data set.

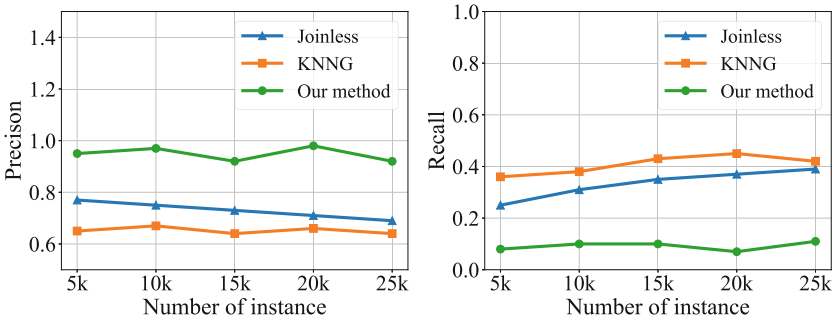
When conducting experiments simultaneously increasing the number of features and instances and providing indices regarding precision and neglect in exploring multilevel co-locations in Fig. 8, it can be observed that the proposed method exhibits a higher precision (0.93) compared to the Joinless method (0.76) and the KNNG method (0.65). Combining this with recall, the proposed method (0.12) outperforms the Joinless method (0.35) and the KNNG method (0.42). This demonstrates that the proposed method is more effective than the other two



(a) Precision and recall on datasets with 5 features



(b) Precision and recall on datasets with 10 features



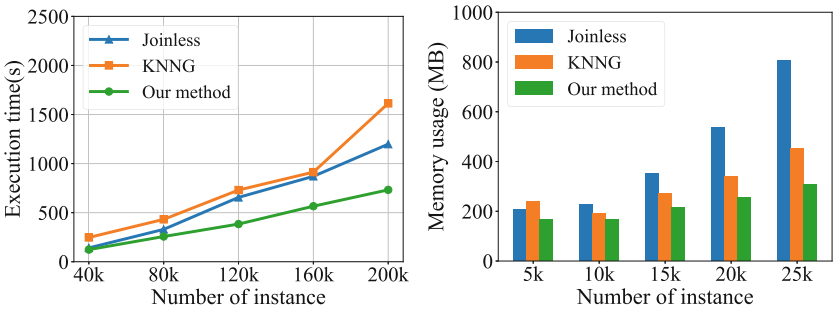
(c) Precision and recall on datasets with 15 features

Fig. 8. The effect of the number of instances on mining multilevel SPCPs.

methods. The main reason is that the Joinless method overlooks many sparse data regions that still hold value, while the KNNG method evaluates various candidate locations differently.

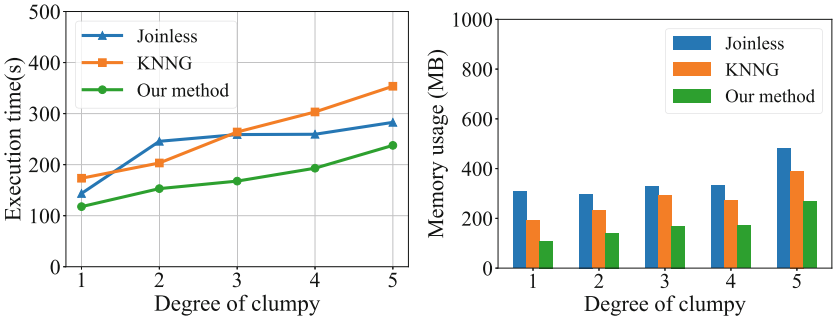
4.1.3 Scalability of Density and Instance

In this experiment, we tested the scalability of density and instance on a self-generated dataset with large feature and instance parameters to demonstrate the ability to expand the density of neighboring regions, describing the number of cases with the same location generated within the same neighboring region, controlled by the concentration level. We utilized 20 features, increased the number of instances gradually by 40.000, with clumpy values ranging from 1 to 5.



(a) The execution time. **(b)** The memory consumption.

Fig. 9. The execution time and memory consumption of three algorithms on different numbers of instances (prevalence threshold is 0.2 for all cases).



(a) The execution time. **(b)** The memory consumption.

Fig. 10. The scalability of the three algorithms on different densities of neighborhood areas.

From Fig. 9, we can observe that the scalability of the KNNG and Joinless methods is relatively low due to the large range and dense density, resulting in a high number of instances within the space around the large distance threshold.

Performance is further compared based on the concentration level of the dataset in Fig. 10. We can see that as the clumpy increases, the runtime and RAM usage of the other two methods are higher compared to the proposed method. This is because the distance threshold parameters of the Joinless and KNNG methods need to be increased to accommodate sparse data regions, significantly affecting their ability to process densely populated areas. On the other hand, the proposed method demonstrates flexibility in handling heterogeneous distributions of spatial datasets.

4.2 Experiments on Real Datasets

This experiment utilizes real-world Point Of Interest (POI) data from various geographical regions to evaluate both the effectiveness and scalability of identified multilevel co-location patterns. Four distinct datasets are employed for this purpose.

The Shenzhen POI dataset consists of 100,000 data points and is used to assess the efficacy of the three methods under investigation. California POI Alpha dataset encompasses 97,000 data points and 26 different feature types, with instances ranging from a maximum of 13,481 for feature A to a minimum of 835 for feature W. Shanghai POI dataset, comprising 120,000 data points and 15 feature types, exhibits varying instance counts across its features, with feature A having the highest instance count at 25,476 and feature O the lowest at 240 instances. The Beijing POI dataset contains 241,000 data points and includes 30 feature types, with feature A registering the highest instance count at 35,453 instances and feature D the lowest at 2,423 instances. These datasets are instrumental in examining the scalability of instance counts.

4.2.1 Effect of the Discovered Multilevel SPCPs

The experiment was conducted on the Shenzhen POI dataset with the following parameter settings: a prevalence threshold of 0.2 to determine co-location pattern levels. In Table 1, the experiment listed important co-location patterns and their levels for each method. For the proposed method, PI is calculated as GI according to the formula mentioned. For the other two methods, PI is calculated using the original formula. To determine the local levels in the Joinless, we also consider co-locations below the prevalence threshold as local patterns. For the KNNG, we determine global patterns by averaging the PI of each co-location pattern across the data regions.

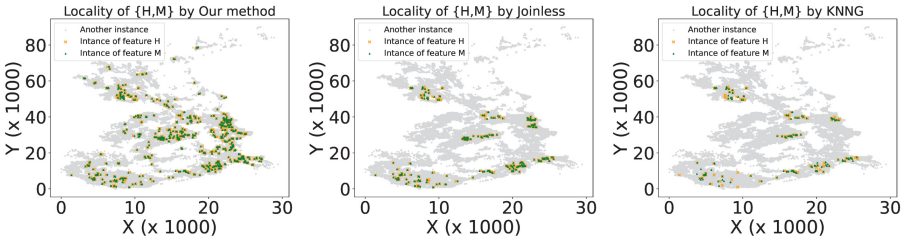
Let's consider co-location pattern {C,D,G}. With our method, we found it with a PI value of 0.43 and identified it as a global pattern. This is correct because the children of {C,D,G}, namely $PI(\{C,D\})= 0.76$, $PI(\{C,G\}) = 0.74$, and $PI(\{D,G\}) = 0.72$, are all global patterns. However, both Joinless and KNNG methods also found co-location patterns that are children of {C,D,G}

Table 1. The multilevel SPCPs discovered by the KNNG, Joinless, and MPCD

Multi-level co- location patterns	The Proposed Method		Joinless		KNNG	
	Level	PI	Level	PI	Level	PI
{A,C}	Global	0.76	Global	0.69	Global	0.58
{A,D}	Global	0.76	Global	0.64	Global	0.49
{B,K}	Local	0.13	Local	0.17	–	–
{C,D,G}	Global	0.43	Local	0.18	Local	0.18
{C,D}	Global	0.76	Global	0.64	Global	0.26
{C,G}	Global	0.74	Global	0.64	Global	0.23
{D,G}	Global	0.72	Local	0.17	Local	0.14
{D,P}	Local	0.12	Local	0.17	–	–
{F,J}	Local	0.15	–	–	Local	0.06
{F,P}	Local	0.13	–	–	–	–
{G,R}	Local	0.17	–	–	Local	0.06
{H,M}	Local	0.15	Local	0.13	Local	0.03
{N,P}	Local	0.17	–	–	Local	0.03

and identified them as global patterns, whereas {C,D,G} is only at the local pattern level. This can be explained by the fact that the other two methods missed many patterns during the exploration process, especially in regions with heterogeneous spatial data. This is evident in patterns such as {B,K}, {D,P}, {F,J}, {F,G}, {N,P}, {G,R}, and many others not listed in this experiment.

Through Table 1, the multilevel co-location patterns detected by Joinless, KNNG, and the proposed method are listed. It can be observed that the multilevel co-location patterns detected by the proposed method generally differ from those detected by the other methods.



(a) The proposed method **(b)** Joinless **(c)** KNNG

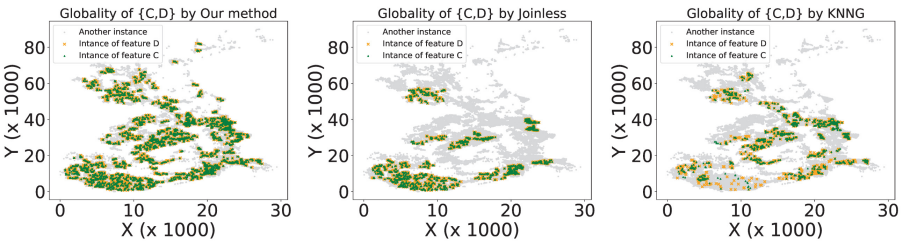
Fig. 11. Comparison of the locality co-location pattern {H,M} detected by Joinless, KNNG, and the proposed method.

The proposed method identifies more global and local co-location patterns compared to the other two methods. We further evaluate all the detected

multilevel co-location patterns based on listing the top co-location patterns with the highest PI at the global and local levels. Subsequently, we compute the PI value for each co-location pattern across the methods.

This experiment has selected two co-location patterns and illustrates the instances of the patterns discovered by the three compared methods. Through visualization, we can better understand the data presented in Table 1.

To further compare the performance of the three methods, some instances of the local co-location patterns detected by all three methods are displayed in Fig. 11. We can observe that the Joinless method, depicted in Fig. 11(b), only identifies locality patterns H,M within densely populated areas with a small exploration range, thus missing out on {H,M} patterns in sparsely populated regions. On the other hand, Fig. 11(c) illustrates that the KNNG method fails to detect {H,M} locality patterns at the outer edges of large regions, and some regions misidentify {H,M} locality patterns.



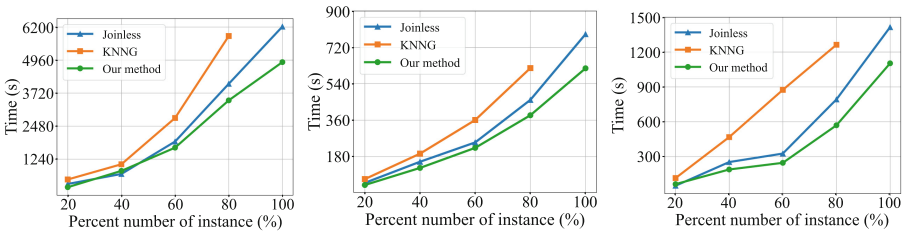
(a) The proposed method **(b)** Joinless **(c)** KNNG

Fig. 12. A comparison of the locality co-location pattern {C,D} detected by Joinless, KNNG, and the proposed method is presented.

Through Fig. 12(a), it is evident that the proposed method effectively explores global patterns {C,D} across various regions and density levels. However, Fig. 12(b) demonstrates that the Joinless method only searches within densely populated data regions, thus missing global patterns {C,D} in sparsely populated areas due to the inflexible distance threshold parameter. On the other hand, Fig. 12(c) illustrates that the KNNG method predominantly delineates global patterns {C,D} within regions of similar and discrete densities, overlooking many global patterns in sparsely populated areas.

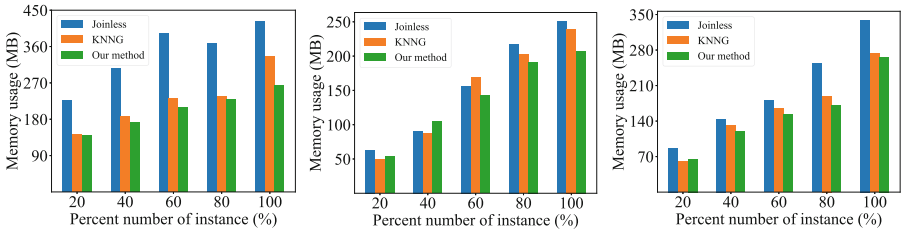
4.2.2 Scalability

To verify the performance and scalability when varying the dataset sizes, this experiment divides three large datasets into smaller datasets with sizes corresponding to 20, 40, 60, and 80 percent of the original data points randomly. The experiment is conducted with PI of 0.2 across all three comparison methods. For the Joinless method, the experiment sets the distance threshold to 200 for the Beijing and Shanghai datasets, and to 800 for the California dataset. For the



(a) Beijing POI (b) California (c) Shanghai POI

Fig. 13. The execution time of the compared algorithms.



(a) Beijing POI (b) California (c) Shanghai POI

Fig. 14. The memory usage of the compared algorithms.

KNNG method, the experiment sets epsilon to 0.6 and alpha to 0.001. Regarding the method proposed in this paper, the global threshold is set to 0.5.

As the number of data points increases shown in Fig. 13, the runtime of our method is lower compared to the other two methods. This is because our method efficiently partitions the data based on heterogeneous density, resulting in faster and more effective processing. On the other hand, the presence of multiple feature types in the data slows down the other two methods.

All parameters in this experiment are kept constant to record the memory usage during the execution of the three methods. The results are depicted in Fig. 14. The heterogeneous distribution of the data makes it challenging to control the number of neighbors within the distance threshold range, as it can vary significantly. Moreover, the diversity in the number of features results in the other two methods consuming more memory space, while our method partitions the data into smaller regions and employs prevalence co-location pattern exploration.

5 Conclusion

In this study, we propose a novel method for exploring spatial co-location patterns. Initially, the data is clustered based on both density level and distribution, forming density-wise clusters. Subsequently, candidate co-location patterns

are generated using the powerful Bron-Kerbosch algorithm and employing the MC-hash structure. Finally, the Global Index is utilized to classify patterns into Global and Local categories. Experiments conducted on both simulated and real-world datasets have demonstrated the effectiveness of this method. We compare the proposed method with two other well-known approaches.

References

1. Bao, X., Wang, L.: A clique-based approach for co-location pattern mining. Inf. Sci. **490**, 244–264 (2019)
2. Debnath, M., Tripathi, P.K., Elmasri, R.: K-dbscan: identifying spatial clusters with differing density levels. In: DMIA, pp. 51–60. IEEE (2015)
3. Deng, M., Cai, J., Liu, Q., He, Z., Tang, J.: Multi-level method for discovery of regional co-location patterns. Int. J. Geogr. Inf. Sci. **31**(9), 1846–1870 (2017)
4. Eppstein, D., Löffler, M., Strash, D.: Listing all maximal cliques in large sparse real-world graphs. J. Exp. Algorithm. **18**, 3 (2013)
5. Ester, M., Kriegel, H.P., Sander, J.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
6. Ghosh, S., Gupta, J., Sharma, A., An, S., Shekhar, S.: Towards geographically robust statistically significant regional colocation pattern detection. In: ACM SIGSPATIAL, pp. 11–20 (2022)
7. Guo, D., Wang, H.: Automatic region building for spatial analysis. Trans. GIS **15**, 29–45 (2011)
8. Huang, Y., Shekhar, S., Xiong, H.: Discovering colocation patterns from spatial data sets: a general approach. IEEE Trans. Knowl. Data Eng. **16**(12), 1472–1485 (2004)
9. Ienco, D., Bordogna, G.: Fuzzy extensions of the dbscan clustering algorithm. Soft. Comput. **22**(5), 1719–1730 (2018)
10. Johnston, H.: Cliques of a graph-variations on the Bron-Kerbosch algorithm. Int. J. Comput. Inf. Sci. **5**(3), 209–238 (1976)
11. Li, J., Adilmagambetov, A., Mohomed Jabbar, M.S.: On discovering co-location patterns in datasets: a case study of pollutants and child cancers. GeoInformatica **20**, 651–692 (2016)
12. Li, Y., Shekhar, S.: Local co-location pattern detection: a summary of results. In: 10th International Conference on Geographic Information Science (GIScience 2018). Schloss-Dagstuhl-Leibniz Zentrum für Informatik (2018)
13. Liu, Q., Liu, W., Deng, M.: An adaptive detection of multilevel co-location patterns based on natural neighborhoods. Int. J. Geogr. Inf. Sci. **35**(3), 556–581 (2021)
14. Liu, W., Liu, Q., Deng, M.: Discovery of statistically significant regional co-location patterns on urban road networks. Int. J. Geogr. Inf. Sci. **36**(4), 749–772 (2022)
15. Qian, F., Chiew, K., He, Q., Huang, H.: Mining regional co-location patterns with kNNG. J. Intell. Inf. Syst. **42**, 485–505 (2014)
16. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: a summary of results. In: Jensen, C.S., Schneider, M., Seeger, B., Tsotras, V.J. (eds.) SSTD 2001. LNCS, vol. 2121, pp. 236–256. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-47724-1_13
17. Tran, V., Wang, L., Zhou, L.: Mining spatial co-location patterns based on overlap maximal clique partitioning. In: MDM, pp. 467–472 (2019)

18. Tran, V., Wang, L., Zhou, L.: A spatial co-location pattern mining framework insensitive to prevalence thresholds based on overlapping cliques. Distrib. Parall. Datab. **41**(4), 511–548 (2023)
19. Wang, D., Wang, L., Wang, X., Tran, V.: An approach based on maximal cliques and multi-density clustering for regional co-location pattern mining. Exp. Syst. Appl. **248**, 123414 (2024)
20. Wang, L., Tran, V., Do, T.: A clique-querying mining framework for discovering high utility co-location patterns without generating candidates. ACM Trans. Knowl. Discov. Data **18**(1), 1–42 (2023)
21. Yao, X., Jiang, X., Wang, D., Yang, L., Peng, L., Chi, T.: Efficiently mining maximal co-locations in a spatial continuous field under directed road networks. Inf. Sci. **542**, 357–379 (2021)
22. Yoo, J.S., Bow, M.: Mining spatial colocation patterns: a different framework. Data Min. Knowl. Disc. **24**, 159–194 (2012)
23. Yoo, J.S., Shekhar, S.: A joinless approach for mining spatial colocation patterns. IEEE Trans. Knowl. Data Eng. **18**(10), 1323–1337 (2006)