DOI: 10.15849/IJASCA.241130.12

Int. J. Advance Soft Compu. Appl, Vol. 16, No. 3, November 2024
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Effective Maximal Co-location Pattern Mining- A Hybrid Approach for Spatial and Spatio-temporal Data

Swati Meshram[1], Kishor P. Wagh[2]

[1]Research Scholar, Dept of Computer Science and Engineering, Government College of Engineering, Amravati, Maharashtra, India.
E-mail :swati.meshram@computersc.sndt.ac.in
[2]Dept of Information Technology, Government College of Engineering, Amravati, Maharashtra, India.
E-mail : kishorpwagh2000@gmail.com

Abstract

Spatial co-location patterns refer to a set of distinct spatial features often found in proximity over a study region. Spatial co-location pattern mining is a process of discovering co-location patterns in global and local regions. However, this relationship of co-occurrence is not uniformly observed. That is, few patterns are discovered at global regions but are not found at local regions and vice versa. Such pattern discovery is based on a single prevalent threshold value in various previous research works. Moreover, this single prevalent threshold would not be suitable to detect maximal patterns globally and locally. Alternatively, it would either miss certain patterns globally or locally due to non-uniform distribution of data instances. To discover the spatial co-location patterns, this paper presents a prevalent region mining algorithm to mine spatial co-location at global and local regions based on distribution of data. Additionally, the effectiveness of this algorithm is proven by comparing with various other state of art algorithms. The algorithm is implemented and evaluated on synthetic and real dataset.

Keywords: *Co-location, grid structure, Haversine distance, neighborhood.*

1. Introduction

In the digital age, with the proliferation of smartphones, Internet of Things (IoT) devices and global positioning system and numerous applications running on them has led to generation of large-scale Spatio-temporal data. This tremendous data offers new opportunities and challenges for analysis and making intelligent decisions.

Received 5 August 2024; Accepted 5 November 2024

Spatial and Spatio-temporal data has a particular interesting type of pattern named as Co-location [1]. For instance, {Residential, Super market, Bank} is a pattern where co-existence among the facilities is commonly observed to leverage the interdependence relationship for their advantage. The colocation pattern discovery is defined as identification of objects that exist together globally as entities or are commonly found together in a specific local geographical area. Exploration of interdependence relationship among the location-based objects puts forth socio-economic domain insights. Spatial co-location mining has been applied in the domains such as urban resource planning and retail [2], public health [3], crime analysis [4] and road network analysis [5], etc.

Traditional co-location pattern mining [6] methods had applied joins [7-9] and association rules like Apriori [10] and FPgrowth [11] along to obtain co-location patterns and utilized prevalence measures for the mining process. These methods required excessive storage and large computational capability to generate co-location patterns. With increase in the feature set of data resulted into redundant patterns generation and increased search space. Some methods adopted clique based search to determine co-location patterns which is a time consuming process as it is a NP-hard problem. They give optimal result but are found unsuitable due to higher complexity of the method. Moreover, co-location patterns are found to be prevalent based on threshold. Measures based on single prevalence threshold could lead to omission of certain rare but important patterns. Such pattern's significance is higher in local region but lower at the global level. Existing methods treat these patterns equally whose significance is computed merely based on a single distance threshold value and without considering data distribution across the regions. Data distribution is the spread of the data in the region. The distribution indicates the reach of the data which may be clustered or spread, which needs to be handled differently at global and local level without losing the neighborhood relationship between the objects.

We propose prevalent regions to capture the neighborhood relationship based on data distribution in the region. The method is important as its weighted measure of prevalence is not merely based on counting of instances but the closeness distance between the instances captures the weight age of the region. To make the computation efficient and minimal we use the concept of grid structure along with the density on standard real point of interest spatial data and spatio-temporal boids dataset.

Our contribution in this research paper is outlined as follows:

- We propose a prevalent region mining algorithmic framework for discovering spatio-temporal co-location mining to identify patterns for varied data distribution based on neighborhood.
- We adopt a grid structure to explore the co-location patterns.
- We propose a weighted method to efficiently filter the co-location patterns and estimate the importance of the co-location rules.
- The weighted prevalent region co-location mining method is implemented and evaluated for correctness over synthetic and real spatial dataset.

We list the structure of our paper as; section 2 discusses the relevant literature. Section 3 covers the basic framework model proposed in methodology. Section 4 discusses the output and results with its interpretation in discussion. Finally, section 5 gives the conclusion of the work carried out.

2. Related Work

Shashi Shekhar and Huang et.al [6] introduced co-location mining and proposed a measure of usefulness or interestingness of the patterns using a metric named as participation index (PI). It is the minimum ratio of every feature participating in the candidate pattern. And a candidate co-location pattern becomes prevalent if its participation index is higher than the prevalent threshold. Then join-based [7], partial join[8] and join-less[9] methods to discover spatial co-location patterns were conceptualize. Although the traditional methods were simple techniques of computing local and global density whose computations were merely based on counting the instances similar to Apriori [10] and Fp-growth [11] algorithms etc. These algorithms adopt association rule mining methods to filter the instances after joining. The Clustering [12] approach for co-location was adopted, which clusters the regions of high density to form co-location patterns. That results in over counting or undercounting of data instances and generation of extensive patterns. These methods relied on time consuming tests for patterns based on counting and ignore the importance of the regions. Depth-first traversal and CPI-tree structure were the alternatives techniques to the traditional join-based methods [13]. Celik et.al [14] adopted a quad tree based structure for partitioning the study area and gave a Zoloc-miner algorithm. In [15] a k-nearest neighbor method for hierarchical partitioning of the spatial space has been adopted.

Further the work of discovering co-location pattern were carried out to be based on determining clique-based neighborhood instances [16-18]. The instances which could not fit in clique-based neighborhood were discovered by star neighborhood instances to make the search easier and less complex. Selection of co-location patterns required to determine the prevalence of such candidate co-location patterns for each participating feature. These methods of partitioning, clique based and star neighborhood-based techniques in large data sets led to high memory consumption, high computational complexity, and back tracking.

V. Trans et.al [19] had applied Delaunay triangulation scheme for generation of neighbors participating in the CP. This method was a distance threshold free method, which is an efficient technique with a shortcoming of excessive triangulation generation of edges and filtering. Another research work by Li.et.al. [3], adopted grid based transact ionization along with buffer overlay method to discover co-location patterns. The other approach of co-location pattern mining based on clustering [20-21], which groups neighbors closer to selected centroids formed clustering colocation patterns in spatial domain.

Spatial and Spatio-temporal instances are found in the Earth's continuous space. Due to the absence of discrete closed boundaries in space, existing techniques takes additional time in searching co-located object instances. They require to store large number of co-location instances in memory before pruning. Measures based on single prevalence threshold could lead to omission of certain rare but important patterns. Existing methods treat these patterns equally whose significance is computed merely based on a single distance threshold value and without considering data distribution across the regions. Data distribution is an emphasis on how the data is located in the region. The distribution indicates the reach of the data which may be clustered or spread, which needs to be handled differently at global and local level. We need to detect prevalent regions [28], the regions of interest with data distribution which is higher locally and is condensed or spread locally but may not be realized or identified with a single global distance threshold to limit the search space. With increase in volume and complexity of location data, there is a need of simpler and effective approach for co-location pattern analysis in broader socio-economic landscape. Table 1 presents the review of recent works in the field of co-location pattern mining of spatial data.

3. Methodology

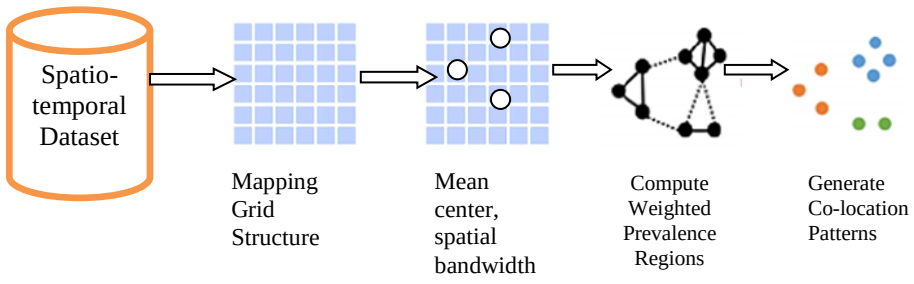

Figure 1: The proposed framework for spatial co-location pattern mining.

The outline of the proposed framework is given in the Fig.1**.** First step is to input the spatio- temporal data, map it to grid structure. Select the high density regions. Find the mean centers and spatial bandwidth of the regions. Compute the weights of the regions and generate prevalent regions with higher weight threshold. Prune the lesser prevalent regions and transform the remaining prevalent regions as co-location patterns. The framework is hybrid as it utilizes grid structure and density based strategy.

A spatial-temporal dataset comprises of location records representing spatial coordinates of the instances(I_m) or events, time of the event along with its non-spatial attributes called as a feature or facility. A collection of distinct features observed in the study area contribute to form a feature set $F = \{f_1, f_2, ..., f_n\}$. A Spatio-temporal co-location pattern is a set of features that are distinct and are in the neighborhood

relationship given as CP = $\{f_1, f_2, ..., f_k\}$ where $k \leq n$, represents the size and $f \in F$. Any instance I_r could be a neighbor of an instance I_m, if the spatial distance between the given instances is less than the distance threshold and time difference between the

Table 1. Review of recent co-location pattern mining approaches.

Reference	Approaches	Method	Limitations and Remarks
[17]	Clique based co-location	Metric - Critical Distance computation, Neighborhood - Star and Clique	Higher complexity. Clique generation- NP hard problem
[19]	Delaunay Triangulation(DT) based Co-location	DT based neighborhood	Requires more storage space as for the same three vertices, three times DT triangle are stored. Complexity is higher.
[22]	Sliding window	FP-Growth generation, Area neighborhood relationship. Suitable for extended objects, buffer overlap, cell operations. Metric – Minimum participation index	Not parallel method.
[23]	Fuzzy co-location	Fuzzy membership, k-nearest neighbors, cliques. Metric – Relative distance and Fuzzy prevalence index.	High complexity
[25]	Regional partition based	Partition distance threshold, core based nearest affiliation neighborhood. Metric – core participation index	Efficiency dependents on core features selection.
[26]	Branch – depth extension	Loss-utility based pruning	User defined threshold, parallelism for concurrent execution required.
[27]	Multi density and Maximal Clique based co-location	Local Partitioning strategy	Effectiveness on region partitioning
[29]	Weighted directed graph of network	Metric -Network based prevalence index.	High complexity.
[30]	Clique based co-location	IDS and NDS tree, Euclidean distance clique. Metric- Neighborhood distance threshold	Large storage space required.
[31]	Multi-level co-location	Metric Global PI, Local PI, Prevalent regions. Apriori generation of co-location.	High complexity

two events is less than given temporal threshold. Spatio-temporal neighborhood relationship is given as,

$$NR(\mathrm{I_m}) = \left\{ \forall I_r \mid spatial\_dist(I_m, I_r) < distance\_threshold \Lambda temporal\_dist(I_m, I_r) < temporal\_threshold \right\} \quad (1)$$

The spatial distance in the above equation is Haversine distance and given by the formula:

$$distance = 2R.\arcsin\sqrt{\sin^2(lat_2 - lat_1) + \cos(lat_2).\cos(lat_1) + \sin^2(lon_2 - lon_1)} \quad (2)$$

where (lat_1, lon_1), (lat_2,lon_2) refer to the spatial coordinates of the two points locations under comparison. 'R' refers to the Earth mean radius = 6371 km and distance is the Haversine distance.

Then the participation ratio for f_i in CP gives the number of instances of f_i participating in CP to the total number of instances of feature type f_i. The measure participation index PI of a co-location pattern CP is the minimum participation ratio among every feature participating in CP, given as PI(CP) = argmin$\forall f_i \in$ CP {PR (CP, fi)}. This measure participation index represents the interestingness of the pattern CP and is called as prevalence measure.

We propose four novel algorithms to extract prevalent regions on spatial and spatio temporal dataset for co-location pattern mining.

3.1 Grid Formation and Instance Mapping:

As there is absence of closed boundaries, search space is continuous and to minimize the comparison based on distance, we place a logical grid which limits the distance for comparison. A logical grid formed by the intersection of latitudes and longitudes. This network of squares forms a spatial or geographic grid and individual square is known as a grid cell. The grid cells density is defined as a measure of the total number of data points in it. Given a set of instances $I_{m,}$in the grid space 'G'with grid cells as {G_1,G_2…, G_i}. Computation of grid cells density given by:

$$Density(G_i) = \forall_{\mathrm{Im} \in Gi} count(I_m) \quad (3)$$

Algorithm 1: Grid formation and Grid cell density

Step 1: Obtain minimum and maximum Cartesian coordinates to form spatial grid.

$$\begin{aligned} Min\_X &\leftarrow \min(\forall I_i.X) \\ Min\_Y &\leftarrow \min(\forall I_i.Y) \\ Max\_X &\leftarrow \max(\forall I_i.X) \\ Max\_Y &\leftarrow \max(\forall I_i.Y) \end{aligned} \quad (4)$$

Step 2: Create necessary hash map and arrays for the grid to obtain the density of grid cells.

$$Grid = createhashmap(\lambda, \lambda)$$
$$Grid\_Density = [][] \quad (5)$$

Step3: Assign data instances to grid and compute density of grid cells

$$\begin{aligned} &For\ each\ instance\ \mathrm{I}_m\ \ \mathrm{I}\ do \\ &\quad i\ =\ (I_m.x\ -\ \mathrm{Min\_X})\ /\ \lambda \\ &\quad j\ =\ (I_m.y\ -\ \mathrm{Min\_Y})\ /\ \lambda \\ &\quad Grid[i][j].append(I_m) \\ &\quad GridCell\_Density[i][j]\ +=\ 1 \end{aligned} \quad (6)$$

Step 4: Compare grid cells density with min_density threshold and mark true for higher density and false to drop the cell.

$$\begin{aligned} &For\ each\ i\ in\ \lambda\ do \\ &\quad For\ each\ j\ in\ \lambda\ do \\ &\quad Grid\_Cell[i][j] = false \\ &\quad If\ \ GridCell\_Density[i][j])\ >\ min\_density \\ &\quad\quad Grid\_Cell[i][j] = true \end{aligned} \quad (7)$$

Step 5: Return Grid, GridCell_Density,Grid_Cell

3.2 Prevalent Regions:

Prevalent regions are defined as the regions of interest locally and/or globally due occurrence of instances which form neighborhood and the spread is limited by spatial bandwidth. The computation of weight is based on difference of distance space and time among mean center and instances with respect to the spatial and temporal bandwidth. Here, the difference between the mean center temporal interval or period and instances I_m temporal interval is given as

$$temporal\_dist(\mathrm{I_m}.T, I_r.T) = \begin{Bmatrix} 1 & if\ \mathrm{I_m}.T - I_r.T = 0 \\ 0 & otherwise \end{Bmatrix} \quad (8)$$

Those grid cells whose weighted average is higher than the weight threshold are formed as prevalent regions as the distribution of instances is concentrated with minimum distance.

*Algorithm 2***:** Generate prevalent regions and candidate co-location patterns.

Step 1: Computation for grid cells greater than threshold density and determine the mean center of grid cell and spatial bandwidth 'h_s'.

$$\begin{aligned} &For\ each\ interval\ T\ of\ timeset\ do: \\ &\quad For\ each\ Grid\_Cell\ of\ Grid\ do: \\ &\quad\quad If\ the\ GridCell\_Density(G_i)\ <min\_density\ then \\ &\quad\quad\quad continue \\ &\quad\quad \mu\_centre_{Gi} = \frac{1}{Density_{Gi}} \sum_{m=1}^{Density_{Gi}} I_m, \forall\ I_m \in G_i\ \wedge NR(\mu\_centre_{Gi}) = I_m \\ &\quad\quad Radius_{Gi} = avg(spatial\_distance(\mu\_centre_{Gi}, Im)) \\ &\quad\quad h_s = Radius_{Gi} \end{aligned} \quad (9)$$

Step 2: Compute the Weighted average prevalence index of region to find prevalence of region.

$$W_{\mu\_centre_{Gi}} = \sum_{\forall I_m \in G_i} \exp\left(-\frac{\mathrm{P}\,\mu\_centre_{Gi_S} - I_{m_S}\,\mathrm{P}^2}{h_s^2 * (n-1)}\right) * exp\left(-\frac{\mathrm{P}\,\mu\_centre_{Gi_T} - I_{m_T}\,\mathrm{P}^2}{h_T^2}\right) \quad (10)$$

Step 3: *If* $W_{\mu\_centre_{Gi}} < W_{TH}$ *then continue*;

If $W_{\mu\_centre_{Gi}} > W_{TH}$ *then*

$$PRegion \leftarrow PRegion \cup G_i \quad (11)$$

$$Centres \leftarrow Centres \cup \mu\_centre_{Gi}$$

Step 4: Generate Candidate Co-location patterns for prevalent regions.

For each $I_m \in G_i$ *do*

If $(|\mu\_centre_{Gi} - I_m| < Radius_{Gi})$ *then*

If $(I_m.f$ *not in* $CCP(G_i))$ *then*

$$CCP(G_i) \leftarrow CCP(G_i) \cup I_m.f \quad (12)$$

$$Sort\ (CCP\ (G_i))$$

Step 5: Return *CCP*

Now, generation of candidate patterns is performed for the prevalent regions. To make the patterns compact by removal of redundant features and mapping to the similar class of patterns using hash search is required for efficient storage and retrieval. We aim to reduce the size of patterns, for repeated feature types.

Algorithm 3: Generate spatial co-location patterns by mining prevalent regions.

Step 1:

$k = 1$

While CCP_k *is not empty do*

Remove redundant neighbour features.

Step 2: *generate* $hash(CCP_k)$

Step 3:

if $hash(CCP_k)$ *not in hashmap then*

$hashmap(key = hash(CCP_k))$

$hashmap(value = G_i, T)$

else

append $hashmap(value = G_i, T)$

$k++;$

Step 4:

for each key in hashmap do

$HC(hash(CCP_k)).append(count(hash(CCP_k).value))$

Step 5:

Display CCP_k

Step 6: *Stop*

Algorithm 4 constructs a tree, for sequence of temporal intervals.Generate temporal sequence from hashmap(CCP_K) key and value list from all temporal intervals in hash map. For every node in the tree recursively call the node and visit, display the nodes and keep on visiting its child nodes to obtain the temporal sequence of features.

Algorithm 4:

Step 1:

$k = 1$

While CCP_k *is not empty do*

$Sequence = hashmap(\ CCP_K\ )$

Step 2: $CurrentNode = root$

Step 3:

For every element in Sequence

If the element not in CurrentNode.Children then

$Append\left(CurrentNode.Children\left(element, count = 1\right)\right)$

Else

$Update\left(CurrentNode.Children\left(element, count + 1\right)\right)$

Step 4:

$//\ Generate\ a\ Subsequence\ temporal\ tree\ for\ each\ CCP_k$

$if\ \ Node\left(T_s.count > temporal\_threshold\right):$

$Output\ temporal\ sequence\left[CCP_k\right]\ =\ display\_pattern\left(Node, level, count\right)$

$display\_pattern\left(child.node, level + 1, count\right)$

Step 5: *Stop*

4. Results and Discussion

4.1 Dataset

Table 2: Dataset Details

Dataset	Type	Records	Features
Synthetic	Spatial Synthetic	1000	Education, Financial Institution, Insurance, Healthcare, Hotel, Retail, Residential
ShenzhenPOI	Spatial Real Standard	40000	A to M
POI	Spatial Real standard	40000	Populated places, monuments, educational institutions, airports, glaciers etc
Boids	spatio-temporal Standard	200 boids x 2500 records	200 boids

4.2 Experimental Setup

The proposed algorithm is implemented in python 3.11.7 in Anaconda navigator v2.4.0 Jupyter notebook 7.0.8 on Windows 10 operating system.

The temporal bandwidth h_T is set to 1 temporal unit (e.g. one day or interval).It was run on all the listed datasets given in table 2.

Table 3: Result of proposed algorithm based on weighted prevalence, POI dataset.

Grid Cell/ Region ID	Average Distance	Weight	Neighbors	Pattern
4597941	2.76878	0.33448	2	A, A
4524668	4.51984	0.15611	3	C,C
4621772	4.66085	0.04489	2	B,E
4736431	3.09666	0.80010	11	C,D,E,F,G
4689900	3.28451	0.85127	17	C,D,E,F,H,J,R
4643261	2.23211	0.95610	28	B,C,D,E,F,G,I,L
4596650	2.72356	0.91037	20	A,D,E,F,G,H,K,M,N
4625395	3.17191	0.58888	5	B,E,G,I,J
4514346	1.91838	0.90019	9	E,G,H,L,Q
4707799	2.67034	0.62164	4	D,E,H
5041662	3.10190	0.79950	11	A,D,E,G,H,I,J,P
4654068	3.75380	0.63473	8	D,E,F,G,H,N
5034594	0.72640	0.92739	2	C,F
4714990	3.95465	0.8023	18	B,C,D,E,G,H,J,K
4991601	4.37005	0.27994	4	A,C,D
4539408	3.16567	0.51268	4	C,E,L
4804320	2.56182	0.88751	14	B,C,D,E,F,I,L,N
4693495	3.26695	0.91421	30	C,D,E,F,G,H,I
4403223	2.21778	0.85328	8	A,D,F,H,I

Table 3 presents the result on POI dataset based on the proposed weighted prevalence method. It includes various patterns from size – 2 to more size of neighbors often found. We observe that the method is able to generate patterns of bigger size. So we call the method as maximal, yet it is able to eliminate redundancy by sorting the features and dropping the repeated features. Fig 2. shows the comparison of proposed method with tradition methods as Apriori and Clustering. These methods generate more patterns due to over counting and redundancy while the proposed weighted method eliminated the unnecessary generation of candidate patterns. Reducing the computation and pruning process.

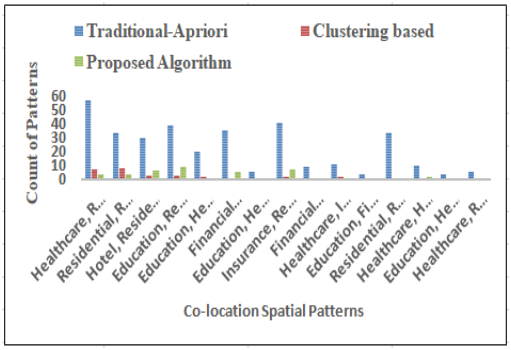

Figure 2: Comparison of traditional Apriori, clustering based and proposed weighted prevalence region method on spatial data on synthetic dataset.

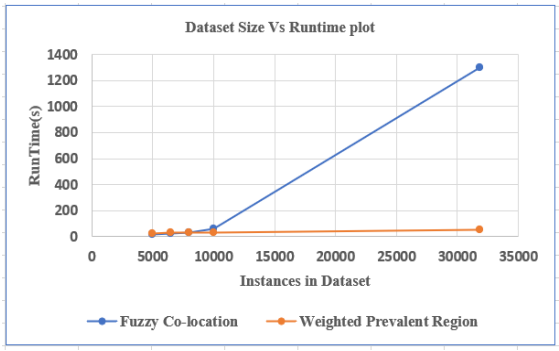

Figure 3: Comparison of number of Instances of Shenzhen POI dataset on runtime based on fuzzy co-location[23] and proposed weighted prevalent region algorithm.

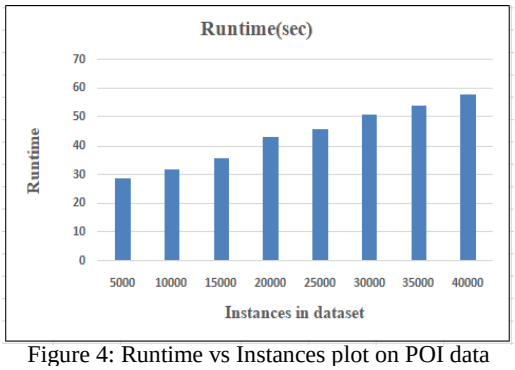

Figure 4: Runtime vs Instances plot on POI data

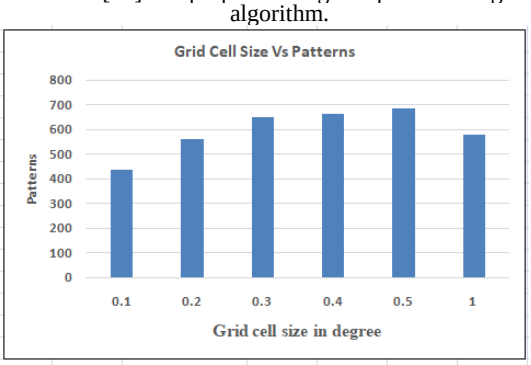

Figure 5: Gridcellsize vs patterns ploton POI data

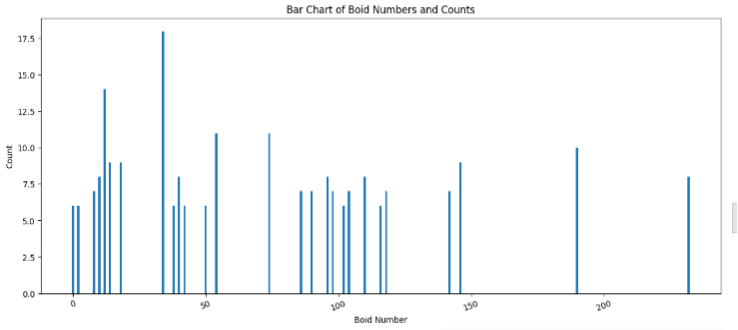

Figure 6: Spatio-temporal maximal co-location on the boids dataset

4.3 Evaluation Measure

The evaluation measure used for testing the algorithms are runtime of the algorithms and prevalence index.The prevalence index and weight of prevalent regions. Prevalence Index for a pattern C is given as

$$Pi(C) = min_{f_i \in C}\{pr(f_i, C)\} \qquad (14)$$

Where Pr is the participation ratio of the pattern C involving feature f_i.

Table 4: Comparison of proposed algorithm weight measure with traditional PI measure.

Top Patterns	Weight of Prevalence Region	Prevalence Index
{A,C,D,E,F,G,H,I}	0.92	0.348591549
{A,C,F,G,H,K,L}	0.67	0.235915493
{B,C,D,G,I}	0.59	0.207746479
{C,D,G,H,I}	0.35	0.123239437
{B,D,G}	0.24	0.084507042

Table 4 clearly indicates that the prevalence index measure is based purely on counting of the occurrences of features proximity, whereas the weighted prevalence measure is based on regional presence of the feature types and distance between the neighbors. So this measure would not miss any rare patterns occurring globally and locally. We the patterns that have negative weight values are pruned.

Effect of dataset size on runtime: As figure 3 shows, as the data instances are increased to the fuzzy co-location algorithm [23], it takes more runtime which is exponential time due to fuzzy maximal grid clique generation for each instance. With less number of data instances for co-location, less number of maximal fuzzy grid cliques is constructed. As dataset size is increased, the runtime increases exponentially. This is due to the reason that fuzzy algorithm involves k-nearest neighbors, and fuzzy grid-cliques computation which majorly rises exponentially with increase in dataset size. However, the weighted prevalence region runtime increases linearly. It takes very less time than the fuzzy co-location algorithm because the proposed algorithm does not compute cliques and Delaunay triangle, which require more processing time and extra storage for duplicate triangles and cliques. But we incorporate the grid cell partitioning into the prevalent regions and weight computation of prevalent region. Figure 4 shows the increase in number of instances, the runtime increases moderately and linearly.

Effect of Grid-Cell Size: The grid cell size 'λ' used for experiment were 0.1, 0.2, 0.3, 0.4, 0.5,1 degree. The bandwidth was set to 2 km for the experiment. We observe that with grid cell size as 0.1 degree, types of patterns identified were 434. With increase in grid cell size as 0.2 and further to 0.3, 0.4 degrees, the types of patterns increased to 560 and 648. But the number of weighted prevalent regions for the same patterns were reduced as can be seen for pattern type 'A,A'. This relates to the fact as more instances could fit in a grid cell leading to change in patterns or are expanded. If we further increase and make the grid cell size 1 degree we find that the types of patterns are reduced to 580, this is due to the fact that the search circumference increases leading to more features in a pattern, consequently reducing redundant patterns. Figure 5, shows the impact of the grid cell size on number of patterns found in the given data. Figure 6, shows the spatio-temporal boids that are being often co-located with other boids. The grid size limits the maximum distance to be considered for comparison for boids to be co-located often in time.

Effect of bandwidth h_s: The bandwidth is the indicator of spread of data instances. By limiting the spread bandwidth which is smaller than the grid size, helps in providing the concentrated or high-density regions. And the clusters formed do not take all the instances of the grid and the shape of grid. Rather will be formed from the cluster centre and constrained on the bandwidth for weight computation. In the experiment we initialized the bandwidth as the distance threshold with 2km. By setting it to 1km was computing weights that were large negative values even for the average distance 3.73 km, and 0.0005 weights for the average distance 2.75 km and when the bandwidth was 3km resulted to weight of 0.65 for 2.70km and 0.46 weights for 3.59km. By setting the bandwidth to 2km, prevalence weight of the region was computed as 0.26 for average distance 3.06km, 0.35 for 2.70km, 0.55 for 2.03 km. and 0.88 for 0.91km. This setting gave better result.

Effect of Distance and Weight threshold: The distance threshold was initialized to the 3 km. When it is assigned a small value, it results into very less patterns generated when higher than 3km results in inclusion of more patterns. When the weight threshold value is small in the range of 0.01 to 0.1, it is found that the run time of the algorithm is reduced. As we increase these parameters the algorithm takes more time as more data instances would be required to be stored in the hash map leading to overall increase in algorithm run time.

Relationship of weight and average distance: Large region weight and large average distance indicates the distribution of data is spread across the grid cell i.e region. If the weight of the grid cell or region is small and average distance is large, describes the distribution of data is spread over larger area and the proximity between the neighboring instances is they are less likely to become neighbors. If the weight within the grid cell is high and average distance is small represents that the data instances are clustered. Weight is representation of density and average distance is representation of spread of data, is clustered or scattered.

5. Conclusion and Future Works

We have proposed a spatio-temporal grid structure based weighted prevalent region to mine co-location patterns. The dropping of grid cells technique with less density regions helps in reduced computation. Prevalent regions formed are the regions with higher data distribution with minimal distance leading to higher weights. Weights are a measure of importance of the co-location along with its size. It is demonstrated that grid size is important in number and size of patterns that would be generated. If the grid size is too small, it leads to small size of patterns become formed. If the grid size is too large would lead to less number of bigger patterns. The reason is wider reach of instances in the grid cell and inclusion of maximal types along with redundancy would lessen the number of distinct types of co-location patterns generated. The bandwidth which limits the spread of the instances and regions provides high density regions. And the clusters formed do not take all the instances of the grid and the

shape of grid. Rather will be formed from the cluster centre and constrained on the bandwidth for weight computation. The experimental evaluation with the dataset-birds, Shenzhen POI, synthetic have shown our algorithm is relatively effective with parameter settings. Our algorithm could be adjusted with other parameter settings to obtain more number of co-location patterns.

Our proposed method is effective and is distinct from other methods as it not based on only counting of instances and supporting it by frequency. Nor the computations are based on fuzzy notion neither do we employ nearest neighbour searching for all the instances which takes up major computation time. We utilize the grid structure division of the study area so that long distance computations should not be carried out. We reduce the computations by using the notion of density which signifies whether the region should be further analysed if it is densely populated. The prevalence or importance of the region is provided by the weightage of the region. The weighted prevalence region is truly based on the scope of data spread, radius and density. Further our technique is simple and does not involve large computations reducing the complexity of the algorithm like as is done in case of cliques' search. We use efficient data structure like hashmap and tree for storage of results and faster retrieval. Most other papers distance calculation is based on Euclidean distance formula. Our approach is based on Haversine distance formula which is more practical as it considers the curvature of the Earth in computations.
Further work could be carried out to optimize the proposed algorithm for efficient retrieval and generation of compact patterns. We plan to include threads for parallel execution and enhance scalability. Better design of data structures could be considered for overall improvement of the algorithm and propose an alternative to distance threshold.

References

[1]. Meshram, S., & Wagh, K. P. (2023). Spatial co-location pattern mining—A survey of recent trends. In *Congress on Intelligent Systems* (pp. 265–280). Springer Nature.

[2]. Chen, Y., Chen, X., Liu, Z., & Li, X. (2020). Understanding the spatial organization of urban functions based on co-location patterns mining: A comparative analysis for 25 Chinese cities. *Cities, 97,* 102563. https://doi.org/10.1016/j.cities.2019.102563

[3]. Li, J., Adilmagambetov, A., Mohomed Jabbar, M. S., Zaïane, O. R., Osornio-Vargas, A., & Wine, O. (2016). On discovering co-location patterns in datasets: A case study of pollutants and child cancers. *Geoinformatica*, *20*(4), 651–692.

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

[5]. Yao, X., Jiang, X., Wang, D., Yang, L., Peng, L., & Chi, T. (2021). Efficiently mining maximal co-locations in a spatial continuous field under directed road networks. *Information Sciences, 542*, 357–379. https://doi.org/10.1016/j.ins.2020.06.057

[6]. Shekhar, S., & Huang, Y. (2001). Discovering spatial co-location patterns: A summary of results. In *Advances in Spatial and Temporal Databases* (pp. 236–256). Springer.

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

[8]. Yoo, J. S., Shekhar, S., Smith, J., & Kumquat, J. P. (2004). A partial join approach for mining co-location patterns. In *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems* (pp. 241–249).

[9]. Yoo, J. S., & Shekhar, S. (2006). A joinless approach for mining spatial colocation patterns. *IEEE Transactions on Knowledge and Data Engineering, 18*(10), 1323–1337. https://doi.org/10.1109/TKDE.2006.150

[10]. Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the VLDB* (pp. 487–499).

[11]. Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD* (pp. 1–12).

[12]. Inoue, R., & Shiode, S. (2023). Colocations of spatial clusters among different industries. *Computational Urban Science, 3*, 35. https://doi.org/10.1007/s43762-023-00107-9

[13]. Wang, L., Bao, Y., Lu, J., & Yip, J. (2008). A new join-less approach for co-location pattern mining. In *2008 8th IEEE International Conference on Computer and Information Technology* (pp. 197–202). https://doi.org/10.1109/CIT.2008.4594673

[14]. Celik, M., Kang, J. M., & Shekhar, S. (2007). Zonal co-location pattern discovery with dynamic parameters. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)* (pp. 433–438).

[15]. Qian, F., Chiew, K., He, Q., Huang, H., & Ma, L. (2013). Discovery of regional co-location patterns with k-nearest neighbor graph. In *Advances in Knowledge Discovery and Data Mining* (pp. 174–186). Springer. https://doi.org/10.1007/978-3-642-37453-1_15

[16]. Bao, X., & Wang, L. (2019). A clique-based approach for co-location pattern mining. *Information Sciences, 490*, 244–264. https://doi.org/10.1016/j.ins.2019.03.072

[17]. Baride, S., Saxena, A. S., & Goyal, V. (2023). Efficiently mining colocation patterns for range query. *Big Data Research, 31*, 100369. https://doi.org/10.1016/j.bdr.2023.100369

[18]. Kim, S. K., Kim, Y., & Kim, U. (2011). Maximal cliques generating algorithm for spatial co-location pattern mining. In *Secure and Trust Computing, Data Management and Applications: 8th FIRA International Conference, STA* (pp. 241–250). Springer.

[19]. Tran, V., & Wang, L. (2020). Delaunay triangulation-based spatial colocation pattern mining without distance thresholds. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *13*, 282–304.

[20]. Meshram, S., & Wagh, K. P. (2021). Mining intelligent spatial clustering patterns: A comparative analysis of different approaches. In *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 325–330).

[21]. Liu, Z., Chen, X., Xu, W., Chen, Y., & Li, X. (2021). Detecting industry clusters from the bottom up based on co-location patterns mining: A case study in Dongguan, China. *Environment and Planning B: Urban Analytics and City Science*, *48*(9), 2827–2841.

[22]. Zhang, J., Wang, L., Tran, V., & Zhou, L. (2023). Spatial co-location pattern mining over extended objects based on cell-relation operations. *Expert Systems with Applications*, *213*, 119253. https://doi.org/10.1016/j.eswa.2022.119253

[23]. Zhou, T., Wang, L., Wang, D., & Tran, V. (2024). Fuzzy regional co-location pattern mining based on efficient density peak clustering and maximal fuzzy grid cliques. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS42022134

[24]. Ghosh, S., Gupta, J., Sharma, A., An, S., & Shekhar, S. (2022). Towards geographically robust statistically significant regional colocation pattern detection. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoSpatial Simulation* (pp. 11–20).

[25]. Wang, D., Wang, L., Jiang, X., & Yang, P. (2024). RCPM_CFI: A regional core pattern mining method based on core feature influence. *Information Sciences, 658*, 119895.

[26]. Yang, P., Wang, L., Zhou, L., & Chen, H. (2024). A fast spatial high utility co-location pattern mining approach based on branch-and-depth-extension. *Information Sciences*, *666*, 120407.

[27]. Wang, D., Wang, L., Wang, X., & Tran, V. (2024). An approach based on maximal cliques and multi-density clustering for regional co-location pattern mining. *Expert Systems with Applications, 248*.

[28]. Li, L., Wang, L., Yang, P., & Zhou, L. (2024). A novel algorithm for efficiently mining spatial multi-level co-location patterns. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2024.3381178

[29]. Yao, X., Jiang, X., Wang, D., Yang, L., Peng, L., & Chi, T. (2021). Efficiently mining maximal co-locations in a spatial continuous field under directed road networks. *Information Sciences, 542*, 357–379. https://doi.org/10.1016/j.ins.2020.06.057

[30]. Bao, X., & Wang, L. (2019). A clique-based approach for co-location pattern mining. *Information Sciences, 490*, 244–264. https://doi.org/10.1016/j.ins.2019.03.072

[31]. Li, J., Wang, L., Yang, P., & Zhou, L. (2024). A novel algorithm for efficiently mining spatial multi-level co-location patterns. *IEEE Transactions on Knowledge and Data Engineering*.

[32]. Inman, J. (1835). *Navigation and Nautical Astronomy: For the Use of British Seamen* (3rd ed.). W. Woodward, C. & J. Rivington.

[33]. Abpeiker, S., & Kasmarik, K. (2023). Motion behaviour recognition dataset collected from human perception of collective motion behaviour. *Data in Brief, 47,* 108976. https://doi.org/10.1016/j.dib.2023.108976