Co-location Pattern Mining Under the Spatial Structure Constraint

Rodrigue Govan, Nazha Selmaoui-Folcher, Aristotelis Giannakos, Philippe Fournier-Viger

Co-location pattern mining under the spatial structure constraint *

Rodrigue Govan[1][0000−0002−4087−7056], Nazha Selmaoui-Folcher[1][0000−0003−1667−3819], Aristotelis Giannakos[2], and Philippe Fournier-Viger[3][0000−0002−7680−9899]

[1] Institute of Exact and Applied Sciences, University of New Caledonia, F-98851 Nouméa Cedex-France
{rodrigue.govan, nazha.selmaoui}@unc.nc
[2] EPROAD, Université de Picardie Jules Verne
[3] Big Data Institute, College of Computer Science and Software Engineering, Shenzhen University, China

Abstract. Most methods to find spatial co-location patterns (subsets of object features that are geographically close to one another) employ standard proximity measures (e.g. Euclidean distance). But for some applications, these measures do not work well since the spatial structure is not considered. This article proposes CSS-Miner, a co-location pattern mining approach under the spatial structure constraint. In this case, the street network of a city is used as a constraint. CSS-Miner has been applied to two real datasets with different points of interest.

Keywords: co-location · data mining · spatial data · spatial structure.

1 Introduction

Discovering co-location patterns is a data mining task that aims at extracting knowledge and insights that integrate the spatial dimension to help decision-makers. A co-location (or *co-location pattern*) is a subset of spatial features that are frequently located in the same region. Despite numerous studies [8, 9, 14], most co-location pattern mining methods use standard distance functions (e.g. the Euclidean distance) to assess the proximity of spatial objects. For applications such as demographic analysis via points of interest (POIs), the Euclidean distance is not suitable since a path between two spatial objects can be significantly different from their Euclidean distance. Hence, other distance measures should be used.

In this paper, we propose CSS-Miner (CSS stands for **C**o-location under the **S**patial **S**tructure constraint), a co-location pattern mining approach for identifying interesting co-locations under the constraint of the spatial structure of a city's street network. CSS-Miner first constructs a graph under the spatial structure constraint using a shortest path algorithm, and then extracts maximal

*This work was supported by the ANR Grant SpiRAL ANR-19-CE35-0006-02.

cliques to obtain spatial patterns. For evaluation, the proposed approach was applied on two datasets from the cities of Paris and Chicago, which allowed discovering relevant patterns.

The article is organized as follows. Section 2 reviews relevant work on spatial pattern mining, focusing on the event-based approach. Section 3 describes the proposed CSS-Miner approach to consider the spatial structure constraint. Then, section 4 presents the data used for evaluation and the discovered patterns. Finally, a conclusion is drawn and perspectives are discussed.

2 Related work

Huang et al. [6] described two main approaches for spatial pattern mining: the sequence-based approach and the event-based approach used in this paper.

The event-based approach (or join-less approach) focuses on the location of spatial objects and their proximity. Initially proposed by Shekhar et al. [9], this approach extracts subsets of objects that are spatially close together, and are called co-locations.

In this paper, we propose a method adopting the event-based approach to leverage the spatial dimension of objects and their proximity. To apply the event-based approach under the spatial structure constraint, maximal clique mining is used to extract co-location patterns. Therefore, the next sub-sections 2.1 and 2.2 respectively give an overview of approaches for maximal clique mining and key studies on co-location pattern mining and their interestingness measures.

2.1 Maximal clique mining

(**Complete graph**) Let $G = (V, E)$ be a graph with $V = \{v_1, v_2, \ldots, v_n\}$ the set of vertices and $E \subseteq \{(v_i, v_j) \in V^2 \mid \forall i, j \in \{1, \ldots, n\} \text{ and } i < j\}$ the set of edges (in this setting, all graphs considered are undirected.) If $(v_i, v_j) \in E$, then v_i and v_j are adjacent. A graph is complete if each pair of graph vertices is connected by an edge (adjacent).
(**Clique**) Let $G = (V, E)$ be a graph and $g = (V_g, E_g)$ be a subgraph such that $V_g \subseteq V$ and $E_g \subseteq \{(v_{g,i}, v_{g,j}) \in E \mid v_{g,i} \in V_g \wedge v_{g,j} \in V_g \text{ and } i \neq j\}$. A clique of G is a subgraph $g \subseteq G$ such that g is complete.
(**Maximal clique**) Given $G = (V, E)$ a graph and $g \subset G$ a clique, the clique g is said to be maximal if and only if there exists no clique g' such that $g \subset g' \subseteq G$.

Valiant [13] has shown that mining all maximal cliques is #P-complete. We can particularly mention the algorithm proposed by Tomita et al. [10] for its $O(3^{n/3})$ worst-case complexity in an n-vertex graph which is optimal as a function of n but also Cazals et al. [3] who consider a recursive approach to improve the mining performance.

Maximal clique mining methods are commonly used to mine co-location patterns [1,11]. By defining a graph network where vertices represent spatial objects and edges represent their neighborhood then by applying a maximal clique mining method, we can obtain subsets of objects that are all neighbors to each other.

Therefore, in this paper, we will use the approach proposed by Tomita et al. [10] for its speed given the size of our datasets detailed in the section 4.1.

2.2 Co-location pattern mining and interestingness measures

The event-based approach projects spatialized data with their coordinates and defines the proximity between each spatial object to extract patterns. In this section, we recall the co-location mining framework proposed in Shekhar and Huang [9], Huang et al. [6] and Yoo and Shekhar [14]. Let $\mathcal{F}$ be a set of features and $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ be a database of spatial objects. Each object in $\mathcal{O}$ consists of a tuple `<object_id, location, feature>`, where `feature` $\in \mathcal{F}$. For example, in Fig. 1b, $\mathcal{F} = \{A, B, C\}$, $\mathcal{O} = \{A_1, B_2, \ldots, C_3\}$ with $A_1 =< 1, (x_1, y_1), A >$, $B_2 =< 2, (x_2, y_2), B >$, etc. A co-location $\mathcal{C}$ is a subset of features $\mathcal{F}$ associated to spatial objects $\mathcal{O}$. These co-location patterns represent pattern frequently located in neighbor objects. The neighborhood relationship is defined as a binary relation $\mathcal{R}(o, o')$ between two spatial objects o and o'. $\mathcal{R}$ can be based on a distance threshold between two objects, or based on their intersection. Several studies have been done in this vein [7,14]. Recently some researchers used a proximity measure that is not the Euclidean distance. For example, Yu [15] proposed the shortest path length as proximity measure. However, the author utilized a sequence-based approach with a limited number of neighbors, which can miss out some relevant information.

In the join-less approach, to determine if two objects are spatially close, the user sets a maximum distance threshold d. A graph is then constructed with vertices representing the spatial objects. Two vertices are adjacent if the associated spatial objects' distance falls within a threshold d (i.e., the spatial distance measure between these two vertices is less than d).

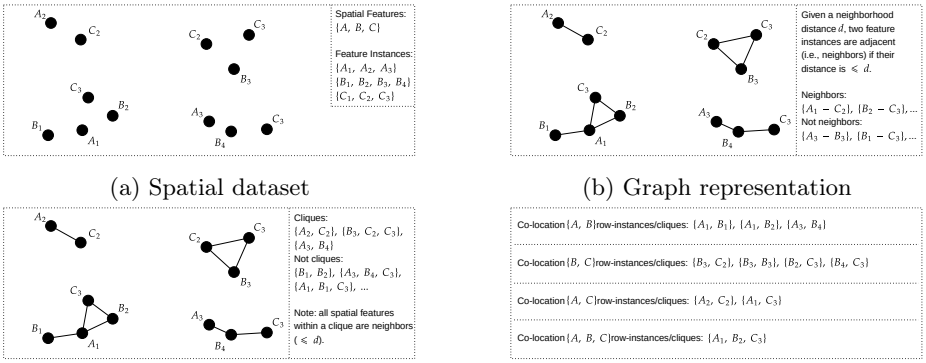
Interestingness measures have been developed to quantify interesting patterns. To measure whether a co-location pattern is interesting or not, the participation index (or prevalence), based on the participation ratio is used.
(**Participation ratio**) Let $\mathcal{C}$ be a co-location pattern. For an instance $f_i \in \mathcal{C}$, the participation ratio is given by:

$$Pr(f_i, \mathcal{C}) = \frac{|\{ \text{ instances of } f_i \text{ participating in } \mathcal{C})\}|}{|\{ \text{ instances of } f_i \ \}|} \tag{1}$$

Given the example of Fig. 1, let $\mathcal{C} = \{A, B\}$ be a co-location candidate and $I_{\mathcal{C}} = \{(A_1, B_1), (A_1, B_2), (A_3, B_4)\}$ be the set of row-instances of $\mathcal{C}$. With A and B, two features having respectively, 3 and 4 instances, we have $Pr(A, \{A, B\}) = \frac{|\{A_1, A_3\}|}{|\{A_1, A_2, A_3\}|} = \frac{2}{3}$ and $Pr(B, \{A, B\}) = \frac{|\{B_1, B_2, B_4\}|}{|\{B_1, B_2, B_3, B_4\}|} = \frac{3}{4}$.
(**Participation index**) Let $\mathcal{C}$ be a co-location candidate, $I_{\mathcal{C}} = \{I_1^{\mathcal{C}}, ..., I_k^{\mathcal{C}}\}$ be the set of row-instances of $\mathcal{C}$ and $\mathcal{F} = \{f_1, \ldots, f_n\}$ be the set of spatial features from the database $\mathcal{O}$. The participation index is defined by:

$$Pi(\mathcal{C}) = \min_{f_i \in \mathcal{C}} Pr(f_i, \mathcal{C}) \tag{2}$$

Using the previous example, we have as participation index:



(a) Spatial dataset (b) Graph representation

(c) Set of cliques (d) Co-location patterns

Fig. 1: Example of co-location patterns based on a set of cliques from a spatial dataset.

$$Pi(\{A,B\}) = \min_{f_i \in \{A,B\}} Pr(f_i, \{A,B\}) = \min(\tfrac{2}{3}, \tfrac{3}{4}) = \tfrac{2}{3}$$

In this paper, the prevalence measure will be used to determine whether co-location patterns in section 4 are relevant or not.

As mentioned before, methods based on the join-less approach mostly used standard distance functions as proximity measure for spatial objects. By using standard distance measures, we may lose the spatial structure. For this reason, we will use the shortest path length as proximity measure.

2.3 Shortest path search

Over the last decades, the shortest path search has been a major problem in graph theory. The speed of search depends entirely on the number of vertices and edges in a graph. One of the first solutions was introduced by Dijkstra [4].

More recently, Varia and Kurasova [12] proposed an accelerated version of Dijkstra's algorithm, by adding two components: a bidirectional search and a parallelized process. To find the shortest path between two vertices v_i and v_j, authors applied Dijkstra's algorithm to find the shortest path from v_i to v_j and from v_j to v_i. Since Dijkstra's algorithm is based on a priority queue, parallel and bidirectional components use two priority queues. With these components, the two paths move forward simultaneously. According to their results, the improved approach is at least twice as fast as the standard algorithm.

To leverage the spatial structure constraint and accelerate the process, the parallel bidirectional Dijkstra's algorithm will be used.

3 Methods

Let consider a set of spatial objects $\mathcal{O}$ with a set of features $\mathcal{F}$. Let G_S be a graph representing the spatial structure as $G_S = (V_S, E_S)$ where V_S a set of vertices representing objects and E_S a set of edges.

3.1 Taking into account the spatial structure constraint

To analyze POIs, the spatial structure constraint is carried out in several steps:

1. For each spatial object $o_i \in \mathcal{O}$, we associate it in the spatial structure G_S with the closest object noted $o_S \in V_S$ (through the Euclidean distance);
2. We apply Dijkstra's algorithm for each object from V_S to the other objects located within a radius d according to the Euclidean distance;
3. If the shortest path length between two objects from V_S is lower than the threshold d, then they are considered as neighbors.

To avoid unnecessary shortest path searches, we only apply the shortest path algorithm between two objects of V_S if these two objects are respectively associated to two objects of $\mathcal{O}$. Here, the Euclidean distance is only used in order to limit the number of shortest path search. Applying a distance radius threshold with the Euclidean distance will prevent computing irrelevant shortest paths. By triangular inequality, a spatial object located outside a distance radius d from another spatial object has a shortest path length greater than or equal to d.

3.2 Graph construction

To extract our spatial patterns (co-locations) which are the maximal cliques, we chose to go on a graph construction $G = (\mathcal{O}, E_{\mathcal{O}})$ (under the spatial structure constraint) where $E_{\mathcal{O}} = \{(o_i, o_j) \mid \exists (o_{S,i}, o_{S,j}) \in E_S, D_{sp}(o_{S,i}, o_{S,j}) \leq d, \forall (i,j) \in [\![1,n]\!]^2, i \neq j\}$ with $o_{S,i}$ representing the object from the spatial structure associated to the spatial object $o_i \in \mathcal{O}$ and D_{sp} representing the distance obtained by Dijkstra's shortest path algorithm if it exists.

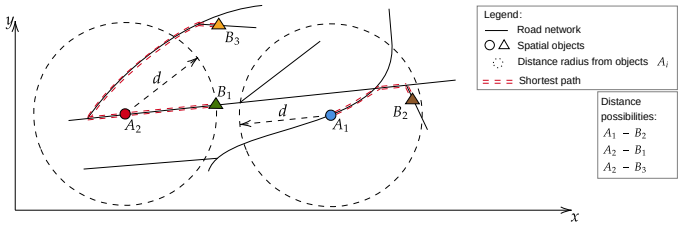


Fig. 2: Three possibilities of distance CSS-Miner can encounter

In the Fig. 2, A_i and B_i are objects from V_S explained in the section 3.1. With d as the distance radius and the shortest path length threshold, we have:

- $d_2(A_2, B_3) > d$ so CSS-Miner will not compute $D_{sp}(A_2, B_3)$;
- $d_2(A_1, B_2) \leq d$ so CSS-Miner will compute D_{sp} and get $D_{sp}(A_1, B_2) > d$ so we will not consider A_1 and B_2 as neighbors;
- $d_2(A_2, B_1) \leq d$ so CSS-Miner will compute D_{sp} and get $D_{sp}(A_2, B_1) < d$ so we will consider A_2 and B_1 as neighbors.

In our approach, CSS-Miner processes two graphs: The first one representing the spatial structure and the second one representing the relationship of our spatial dataset created with the first graph.

4 Experimental Results

We apply CSS-Miner on two real datasets. Both have been created by collecting data from OpenData [4]. The first dataset is located in Paris city with High Schools, Movie theaters, Bicycle stations, Parks and Subway station variables having respectively 239, 85, 996, 722 and 326 spatial objects (2368 objects in total). The second dataset is located in Chicago city with High Schools, Bus station, Rail Lines station, Fast food chains, Bicycle stations and Parks variables having respectively 142, 5606, 124, 877, 1402 and 613 spatial objects (8764 objects in total). For each dataset, the entire process was carried out with a AMD Ryzen 7 3700X 8-core processor with 64GB of RAM. It took respectively, about 2 and 5 hours to run the entire process on Paris and Chicago datasets.

Although we aim to analyze and understand the young population behavior, CSS-Miner is applicable to other demographic analysis, for instance: What are the daily habits of a manager compared to a student? Another POIs analysis can also be useful to develop a decision support tool to help developing the tourism of a city. Finally, the POIs analysis remain a very large subject to study.

4.1 Data Preprocessing

To integrate the spatial structure constraint, it is necessary to get access to that information. In this case, we used the road network as spatial structure. Here, we assume that the path is taken on foot because we wanted to integrate only data from OpenData platforms where the traffic noise is not always available. To get access to the road network of Paris and Chicago, we used OSMnx methods [2]. Once the street network is retrieved, it can be converted into a graph network with roads as edges and road intersections as vertices. At the end, the graph associated to Paris street network has 42,870 vertices and 241,016 edges and the graph associated to Chicago has 184,476 vertices and 1,217,928 edges.

4.2 Results

The Table 1 shows us the possible activities near High Schools in Paris, in particular Parks and Movie theaters. Due to limited page number, the Table 1 only displays few extracted patterns. We note through extracted co-location patterns, the ubiquity of High Schools and Bicycle variables, which also show us that the city of Paris helps young population to get around the city autonomously and practice a physical activity. It would be interesting to apply CSS-Miner to other french cities offering this service in order to confirm this trend.

[4] opendata.paris.fr/, data.iledefrance.fr/, data.cityofchicago.org/

Since CSS-Miner integrates the road network as spatial structure constraint, we compared our co-location patterns with the ones without this constraint i.e., using only the Euclidean distance. The results show us that by taking into account the road network, co-location patterns not always have a prevalence greater than prevalence with the Euclidean distance as proximity measure.

Indeed, the extracted co-location patterns without constraint used a distance threshold equal to 500 (meters), just as CSS-Miner. By triangular inequality, a walking distance between two spatial objects is greater than or equal to their Euclidean distance. Therefore, without constraint, the co-location candidates contain more spatial objects, increasing the probability to have a high number of feature instances per variable, which can reduce their prevalence. This also explains why the {Parks, High Schools, Bicycle} co-location pattern has a decreasing prevalence from 0.89 to 0.56 by adding the Movie theaters variable.

Table 1: Extracted co-location pattern prevalence (Pi from equation 2)

City	Co-location pattern	Pi under constraint	Pi without constraint
Paris	{Parks, High Schools, Bicycle}	0.89	0.89
	{Parks, High Schools, Movie theaters, Bicycle}	**0.56**	0.44
	...		
Chicago	{Bus, Fast food chains, High Schools, Bicycle}	**0.58**	0.5
	{Bus, Fast food chains, High Schools}	**0.33**	0.17
	...		

Moreover, without constraint, the algorithm extracted patterns CSS-Miner did not extract: {High Schools, Subway} and {Parks, High Schools, Movie theaters, Subway} with a prevalence equal to 0.31 and 0.14 respectively without the constraint. It shows that even if the spatial objects are close to one another using the Euclidean distance, their shortest path length do not verify our proximity criterion, so they cannot be considered as close. At the end, CSS-Miner can extract more relevant patterns and filter not so relevant patterns.

The results show that most of High Schools in Chicago have a Fast food chains around it, so young population in Chicago will be more tempted to go eat in a Fast food at lunch or after school. The ubiquity of High Schools and Fast food chains variables can also be a sign of malnutrition in the US, at least in Chicago. To confirm this affirmation, it would be interesting to apply CSS-Miner in other US cities and verify the relevancy on a national scale. It would also be interesting to get a Fast food dataset in Paris (unavailable on the OpenData) to reveal if Fast food chains in Paris target young population as in Chicago.

5 Conclusion and perspectives

In this paper, we introduced CSS-Miner, a co-location pattern mining approach integrating the spatial structure. We described how this constraint has been

defined and taken into account, particularly with a road network and a shortest path search algorithm. To extract co-location patterns, we used the maximal clique mining approach with a restricted search radius and editable depending on the use case. Then, we applied the approach on two real datasets.

The next step of our work will be to integrate knowledge from experts [5], such as urban planners and geographers to verify the relevancy of the extracted patterns. Moreover, CSS-Miner will be applied on larger datasets to estimate the performance. Finally, future work will consider the altitude as spatial structure.

References

1. Bao, X., Wang, L.: A clique-based approach for co-location pattern mining. Information Sciences **490**, 244–264 (2019)
2. Boeing, G.: Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. Computers, Environment and Urban Systems **65**, 126–139 (2017)
3. Cazals, F., Karande, C.: A note on the problem of reporting maximal cliques. Theoretical computer science **407**(1-3), 564–568 (2008)
4. Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)
5. Flouvat, F., Van Soc, J.F.N., Desmier, E., Selmaoui-Folcher, N.: Domain-driven co-location mining: Extraction, visualization and integration in a gis. Geoinformatica **19**, 147–183 (2015)
6. Huang, Y., Shekhar, S., Xiong, H.: Discovering colocation patterns from spatial data sets: a general approach. IEEE Transactions on Knowledge and data engineering **16**(12), 1472–1485 (2004)
7. Kim, S.K., Lee, J.H., Ryu, K.H., Kim, U.: A framework of spatial co-location pattern mining for ubiquitous gis. Multimedia tools and applications **71**(1), 199–218 (2014)
8. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: International Symposium on Spatial Databases. pp. 47–66. Springer (1995)
9. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: A summary of results. In: International symposium on spatial and temporal databases. pp. 236–256. Springer (2001)
10. Tomita, E., Tanaka, A., Takahashi, H.: The worst-case time complexity for generating all maximal cliques and computational experiments. Theoretical Computer Science **363**, 28–42 (2006)
11. Tran, V., Wang, L., Chen, H., Xiao, Q.: Mcht: A maximal clique and hash table-based maximal prevalent co-location pattern mining algorithm. Expert Systems with Applications **175**, 114830 (2021)
12. Vaira, G., Kurasova, O.: Parallel bidirectional dijkstra's shortest path algorithm. Databases and Information Systems VI, Frontiers in Artificial Intelligence and Applications **224**, 422–435 (2011)
13. Valiant, L.: The complexity of enumeration and reliability problems. SIAM Journal on Computing **8**(3), 410—-421 (1979)
14. Yoo, J.S., Shekhar, S.: A joinless approach for mining spatial colocation patterns. IEEE Transactions on Knowledge and Data Engineering **18**(10), 1323–1337 (2006)
15. Yu, W.: Spatial co-location pattern mining for location-based services in road networks. Expert Systems with Applications **46**, 324–335 (2016)