Preprint of:
Govan, R., Selmaoui, N., Giannakos, A., Fournier-Viger, P. (2019). Co-location pattern mining under the spatial structure constraint. Proc. 34th International Conference on Database and Expert Systems Applications (DEXA 2023), Springer, to appear.

Co-location pattern mining under the spatial structure constraint

Rodrigue Govan[1], 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
aristotelis.giannakos@u-picardie.fr
[3] Big Data Institute, College of Computer Science and
Software Engineering, Shenzhen University, China
philfv@szu.edu.cn

Abstract. Spatial co-location pattern is a subset of object features that are geographically close to one another. The majority of existing methods employ standard proximity measures (e.g. Euclidean distance). However, depending on the study area, these standard measures do not work well. The spatial structure has to be considered. This article proposes CSS-Miner, a co-location pattern mining approach under the spatial structure constraint. In this case, we use the street network of a city as a constraint. CSS-Miner has been applied to two datasets from the cities of Paris and Chicago by selecting different POIs.

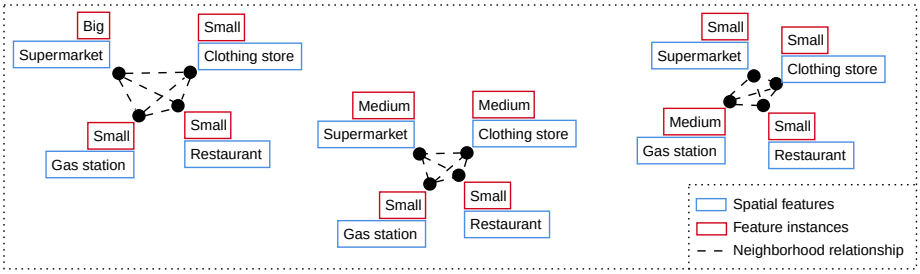
Keywords: co-location · pattern mining · spatial data · spatial structure.

1 Introduction

In the field of data mining, discovering co-location patterns is done to extract knowledge and insights that integrate the spatial dimension and can help decision-makers. A co-location (or *co-location pattern*) is a subset of spatial features that are frequently located in the same region. For example, shopping centers in a city often contain big supermarkets, small restaurants, clothing stores and a gas station. Hence, if we consider shopping centers around a city as a co-location pattern, its spatial features can be described by supermarkets, restaurants, clothing stores and gas stations. A supermarket can be a spatial feature having values such as small, medium and large as feature instances. A graph representation of this example is presented in Fig. 1.

A lot of research has been done on mining co-location patterns [15, 19, 27]. Pattern mining methods for co-location discovery were applied in various fields such as to analyze business density [6], explain anthropic phenomena [1] and to explain soil erosion [18]. However, despite numerous use cases, most of these

Fig. 1: Example of a co-location pattern with its spatial features and feature instances.



methods rely on standard distance functions to assess the proximity of spatial objects such as the Euclidean distance. But in some use cases, it is desirable to choose other functions for measuring distances between spatial objects. For instance, in the case of behavior analysis via points of interest in a city, the Euclidean distance does not make sense since a path between two spatial objects can be significantly different from their Euclidean distance.

Depending on the case study, it can be essential to take into account the spatial structure, since it influences the distribution of spatial objects in the area. If standard distance functions are used such as the Euclidean distance in urban analysis, all information about the city's spatial structure is lost. To keep that information, other distance functions must be used. However, depending on the density of a city, integrating a city's structure can increase the complexity of the analysis, not only in terms of data processing but also in terms of input parameter settings as well.

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. The approach first constructs a graph under the spatial structure constraint using a shortest path algorithm. Then, CSS-Miner extracts maximal 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. described two main approaches for spatial pattern mining: the sequence-based approach and the event-based approach [12].

The sequence-based approach consists in transforming spatial objects into sequential data and then applying a standard frequent itemset pattern mining algorithm. This approach was introduced by Koperski and Han [15].

The event-based approach (also named join-less approach) focuses on the locations of spatial objects and their proximity. Initially proposed by Shekhar et al. [19], this approach extracts subsets of objects that are spatially close together, and are called co-locations. In the same way as for the sequence-based approach, interestingness measures have been defined to keep only the most interesting co-location patterns.

In the literature, it is found that the sequence-based approach is more commonly used than the join-less approach, and all the above methods rely on the Euclidean distance to assess spatial relationships. In this paper, we design a method based on an alternative distance measure to handle a constraint that we call the spatial structure constraint. The proposed method adopts 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 $E \subseteq \{(v_i, v_j) \in V^2 \mid \forall i, j \in \{1, \ldots, n\} \text{ and } i \neq j\}$ the set of edges. If two vertices v_i and v_j are linked i.e., $(v_i, v_j) \in E$, 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 $g \subset g' \subseteq G$.

With the event-based approach, it is possible to extract co-location patterns through maximal clique mining. Valiant [24] has shown that enumerating all maximal cliques is #P-complete. In the same way as standard frequent itemset pattern mining methods, maximal clique extraction checks every combination of vertices from a graph to obtain maximal cliques. We can particularly mention the algorithm proposed by Bron et al. [4] and Tomita et al. [21] for its $O(3^{n/3})$ worst-case complexity in an n-vertex graph which is optimal as a function of n but also Moon et al. [17] and Cazals et al. [5] who consider a recursive call in the algorithm to improve the maximal clique mining performance.

In the literature, maximal clique mining methods are commonly used to mine co-location patterns [2, 16, 22, 26]. 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 Bron et al. [4] then adapted by Tomita et al. [21] 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 idea of the event-based approach is to project spatialized data with their coordinates and to define the proximity between each spatial object in order to extract patterns. In this section, we recall the co-location mining framework proposed in Shekhar and Huang [19], Huang et al. [12] and Yoo and Shekhar [27]. 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 the Fig. 2b, $\mathcal{F} = \{A, B, C\}$, $\mathcal{O} = \{A_1, B_2, \ldots, C_3\}$ 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'. Depending on user requirements and use cases, $\mathcal{R}$ can be based on a distance threshold between two objects, or based on their intersection. Several works have been done in this vein, including Yoo and Shekhar [27], Wang et al. [25] and Kim et al. [14]. Most of these works are usually based on the Euclidean distance to evaluate the proximity between spatial objects. But more recently, some works have been done on co-location pattern mining using a different proximity measure from the Euclidean distance. Yu [28] proposed in his paper the shortest path length as proximity measure. However, the author proposed the method by adding a parameter which is the maximum number of object neighbors. By setting this parameter, it ensures a fast pattern mining time but it also limits the size of co-location patterns which can miss out some information that might turn out to be relevant to users. Then, Yu et al. [29] added a distance-decay function to find the spatial dependence between spatial objects. It consists of weighting the contribution of a co-location pattern in the interest measure.

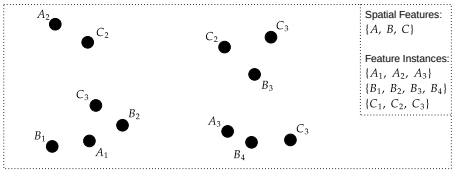
The join-less approach is based on the definition of a neighborhood threshold. To determine if two objects are spatially close, we set a maximum distance threshold d. Once the neighborhood is defined, the graph is constructed with the spatial objects representing the vertices. Two vertices are adjacent if the associated spatial objects' distance falls within the threshold d (i.e., the spatial distance measure between these two vertices is lower than d).

For spatial pattern mining methods, interestingness measures have been developed in order to quantify interesting patterns. To measure whether a co-location pattern is interesting or not, the participation index, based on the participation ratio is used. The participation index is also called the prevalence. We then speak about prevalent spatial pattern.

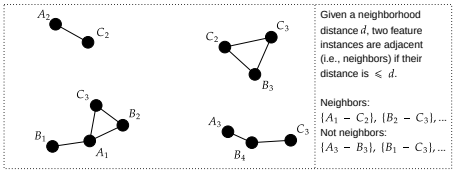
(**Participation ratio**) Let $\mathcal{C}$ be a co-location pattern. For an instance $f_i \in \mathcal{C}$, the participation ratio is given by:

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

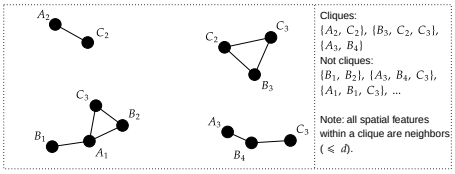
Given the example of the Fig. 2, let $\mathcal{C} = \{A, B\}$ be a co-location candidate and $I_{\mathcal{C}} = \{(A_1, B_1), (A_1, B_2), (A_3, B_4)\}$ be the set of row-instances of $\mathcal{C}$. With A and B, two features having respectively, 3 and 4 instances, we have $Pr(A, \{A, B\}) = \frac{|\{A_1, A_3\}|}{|\{A_1, A_2, A_3\}|} = \frac{2}{3}$ and $Pr(B, \{A, B\}) = \frac{|\{B_1, B_2, B_4\}|}{|\{B_1, B_2, B_3, B_4\}|} = \frac{3}{4}$.



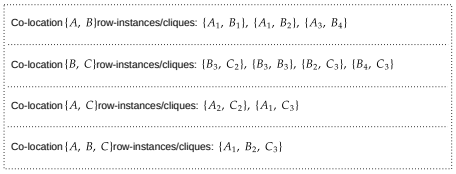
(a) Spatial dataset with spatial features and its features instances



(b) Graph representation given a neighborhood distance threshold d



(c) Set of cliques from graph representation



(d) Co-location patterns from the set of cliques

Fig. 2: Example of co-location patterns based on a set of cliques from a spatial dataset.

(**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:

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

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

The participation index measure has been defined for point data. However, with increased data collection, we now have different types of data (lines, polygons, ...). In their paper, Akbari et al. [1] proposed a participation index variant measure for each type of data. To take into account all types of data, authors proposed to restrict the mining region by applying Voronoi diagram around core elements. In their case, core elements is the spatial feature to analyze described by point data. Once the Voronoi diagram applied, each Voronoi's cell represents

a co-location row-instance. Then, to consider polygon/line data, they weight spatial objects by the proportion of the spatial object located inside the Voronoi region.

In this paper, we aim to integrate a spatial structure constraint to mine co-location patterns. Applying Voronoi diagram would restrict the mining region but it would not necessarily restrict the distance measure. Thus, we will not use the variant measure proposed in [1]. Therefore, we will have to convert our polygon data into point data by taking the polygon's center of gravity (mean of all polygon coordinates).

As mentioned before, join-less approach works mostly used standard distance functions as proximity measure between spatial objects. Depending on the case study, by using standard distance measures such as the Euclidean distance, we may lose the spatial structure. We will, then, 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 the numbers of edges in a graph. One of the first solutions was introduced by Dijkstra [7]. Then, new algorithms were developed to accelerate the shortest path search [9, 13, 20].

More recently, Varia and Kurasova [23] 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, the bidirectional component uses two priority queues. Their method is run by executing one step on each side in a single period. According to authors, the algorithm stops somewhere between v_i and v_j. However, the process is not symmetrical, it depends on the number of edges of all the visited nodes. The problem in the bidirectional component is that two paths are taken with one step at a time. Each path will be shorter, but they are moving each in turn. That is why authors proposed to add a second component: parallel computing. With this component, the two paths can move forward at the same time. By adding these two components, according to their results, the improved approach is at least twice as fast as the standard algorithm, depending on the number of vertices in the graph.

In order to leverage the spatial structure constraint and accelerate our process, the bidirectional and parallel 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

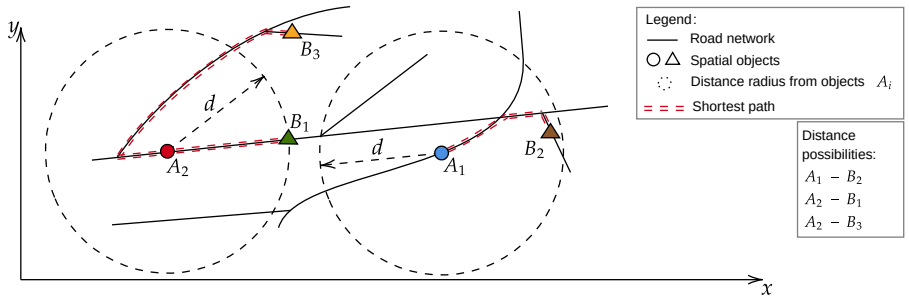
To analyze points of interest in a spatial structure (e.g., an urban area), the length associated to the shortest path taken between two locations (x_i, y_i) and (x_j, y_j) associated to spatial objects o_i and o_j respectively, seems the most adapted. In order to extract co-location patterns, we have sought to include the spatial structure constraint.

The integration of the 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 compute the shortest paths according to Dijkstra's method 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 computations, we only apply the shortest path algorithm between two objects of V_S if these two objects are respectively associated to two objects of $\mathcal{O}$. Even though the Euclidean distance is not used as proximity measure in our approach, we still use it in order to limit the number of shortest path computation. Applying a distance radius threshold with the Euclidean distance will prevent computing irrelevant shortest path. By triangular inequality, a spatial object located outside a distance radius d from another spatial object has a shortest path distance greater than or equal to d.

Fig. 3: Three possibilities of distance CSS-Miner can encounter.



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.

The Fig. 3 shows the three possibilities CSS-Miner can encounter. Here, 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 threshold, we have as Euclidean distance, $d_2(A_2, B_3) > d$, so CSS-Miner will not run the shortest path algorithm and will not consider A_2 and B_3 as neighbors under the spatial structure constraint. On the other hand, we have $d_2(A_1, B_2) \leq d$, so our algorithm will run the shortest path algorithm. However, we have $D_{sp}(A_1, B_2) > d$, so we will not consider A_1 and B_2 as spatially close. Finally, we have in the Fig. 3, the spatial object B_1 located within the distance radius d of A_2. CSS-Miner will compute the shortest path, get $D_{sp}(A_2, B_1) < d$ and consider these two spatial objects as spatially close. Therefore, the associated value in the adjacency matrix will be equal to 1.

At the end, in CSS-Miner, we are processing 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

In this paper, we apply our approach on two real datasets (see data description in Table 1. The first one is created by collecting data from OpenData platforms of Paris[4] and its suburbs[5]. The second dataset is also created by collecting data from the OpenData platform of Chicago[6].

Table 1: Description of datasets.

City	Variable	Attributes	# Modalities	# Spatial Objects
Paris	High Schools	Type of High School	7	239
	Movie theaters	# Seats available (*)	5	85
	Bicycle	Station capacity (*)	8	996
	Parks	Type of Park	9	722
	Subway	Line of the station	16	326
Chicago	High Schools	Type of High School	13	142
	Bus	# Lines on station	12	5,606
	Rail Lines	# Lines on station	6	124
	Fast food chains		1	877
	Bicycle	Station capacity (*)	8	1,402
	Parks	Type of Park	13	613

(*): The data have been discretized by quantile.

[4] https://opendata.paris.fr/
[5] https://data.iledefrance.fr/
[6] https://data.cityofchicago.org/

For each dataset, the entire process was carried out on a computer with a AMD Ryzen 7 3700X 8-core processor, 64GB of RAM and a NVIDIA GeForce RTX 2060 SUPER GPU with 8GB of dedicated RAM. It took respectively, about 2 and 5 hours to run the entire process on Paris and Chicago datasets.

This use case aims to analyze and understand the young population behavior in a big city. This approach is totally generic, since we can apply it to a population analysis according to the socio-professional category, for example: What are the daily habits of a manager compared to a student? Another points of interest analysis can also be useful to develop a decision support tool to help developing the tourism of a city. Finally, the points of interest analysis remain a very large subject to study.

4.1 Data Preprocessing

In order 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. We assume that the path is taken on foot. We made this choice 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 [3]. Authors made OSMnx easy to use, one can retrieve street network from coordinates or just by providing the city name via its Python package. 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.

For the Paris dataset, we collected datasets of Movie theaters, High Schools, Self-service bicycle stations, Green spaces and Subway stations of Paris. To restrict the scope of analysis, we only kept spatial objects of Paris (and not its regions around). We also chose to only process point data. Therefore, the Green spaces variable which is initially polygon data is reduced to point data by taking the centroid (the center of gravity). In addition, since co-location pattern only works with categorical data, we discretized two variables (Movie theaters and Self-service bicycle stations) by quantile. At the end, we have 2,968 spatial objects described by the table 1.

For the Chicago dataset, we collected datasets of Bus stops, Rail lines, Green spaces, High Schools, Self-service bicycle stations and Fast food chains of Chicago. The same process as the Paris dataset has been done on this Chicago dataset. At the end, we have 8,764 spatial objects described by the table 1.

For both datasets, we converted all the coordinates into the projected coordinate system WGS 84 / Pseudo-Mercator (EPSG:3857). This coordinate system allows us to compute distances in meters.

At the pruning step, we set a radius threshold of $500m$ ($d = 500$). Each object will only be compared to objects within this radius. At the graph construction and its associated adjacency matrix, to determine if two objects (two vertices) are contiguous, we thresholded the walking distance to the same radius threshold

($d = 500$). Thus, if the shortest path found between two spatial objects is lower than $500m$, then its associated value in the adjacency matrix is equal to 1. Otherwise, it is equal to 0. In addition, to avoid any loop in the graph (a vertex adjacent to itself), we set every value of the diagonal to 0.

4.2 Paris dataset

Following the data processing and the graph construction, we have run the maximal cliques mining process. Since this paper is about young population behavior analysis, the Table 2 only shows co-location patterns containing the High Schools variable.

Table 2: Extracted Paris co-location pattern prevalence.

Co-location pattern	Prevalence under constraint	Prevalence without constraint
{Green Spaces, High Schools, Self-service bicycle}	0.89	0.89
{High Schools, Self-service bicycle}	0.86	0.86
{Green Spaces, High Schools, Self-service bicycle, Subway station}	0.78	0.89
{High Schools, Movie theaters, Self-service bicycle}	0.71	0.71
{High Schools, Self-service bicycle, Subway station}	0.71	0.71
{High Schools, Movie theaters, Self-service bicycle, Subway station}	0.71	0.71
{Green Spaces, High Schools, Movie theaters, Self-service bicycle}	**0.56**	0.44

The Table 2 shows us the possible activities near High Schools in Paris, in particular parks and movie theaters. We note through these co-location patterns, the ubiquity of High Schools and Self-service bicycle variables, which also shows us that the city of Paris helps young population to get around the city autonomously and at the same time, 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.

Since CSS-Miner approach integrates the road network as spatial structure constraint, the idea is to see if there is any difference compared to co-location patterns without this constraint i.e., using only the Euclidean distance. These results show us that by taking into account the road network, co-location patterns under constraint not always have a prevalence greater than prevalence with the Euclidean distance as proximity measure.

We can explain as follow. 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 instances per variable, which can reduce their prevalence. This also explains why the {Green Spaces, High Schools, Self-service bicycle} co-location pattern has a decreasing prevalence from 0.89 to 0.56 by adding the Movie theaters variable. Indeed, by adding a variable into a co-location pattern, it increases the number of spatial objects contained in the co-location, which can decrease the prevalence.

Finally, without considering the spatial structure constraint i.e., by using the Euclidean distance as proximity measure, the algorithm extracted some patterns CSS-Miner did not extract. These patterns are: {High Schools, Subway station} and {Green Spaces, High Schools, Movie theaters, Subway station} with a prevalence equal to 0.31 and 0.14 respectively without considering the spatial structure constraint. These two patterns have a prevalence equal to 0 under the constraint. It shows that even if the spatial features 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, by taking into account the spatial structure to define the proximity measure, we can extract more relevant patterns and filter not so relevant patterns based on the study area.

4.3 Chicago dataset

As the Paris dataset, the Table 3 only shows co-location patterns containing the High Schools variable from the Chicago dataset.

Table 3: Extracted Chicago co-location pattern prevalence.

Co-location pattern	Prevalence under constraint	Prevalence without constraint
{Bus, Fast food chains, High Schools, Self-service bicycle}	**0.58**	0.5
{Bus, Fast food chains, High Schools, Rail Lines, Self-service bicycle}	0.38	0.38
{Bus, Fast food chains, High Schools}	**0.33**	0.17
{Bus, Fast food chains, High Schools, Rail Lines}	0.3	0.3
{Bus, Fast food chains, High Schools, Parks}	0.17	0.17
{Bus, Fast food chains, High Schools, Rail Lines, Self-service bicycle, Parks}	0.15	0.15

Prevalences from Table 3 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 the young population of Chicago. To confirm this affirmation, it would be interesting to apply CSS-Miner in big cities from the USA and verify if we can extract the same co-location patterns. It would also be interesting to get a Fast Food dataset in Paris to reveal if Fast Food chains in Paris target young population the same way as in Chicago. Unfortunately, the Fast Food dataset in Paris is not available on OpenData platforms. We note from these co-location patterns that High Schools in Chicago have a mean of public transportation nearby which shows us that Chicago is well connected. Just as the Paris dataset, based on prevalence values, we have more relevant co-location patterns under constraint than without for the same reasons mentioned before.

5 Conclusion and perspectives

In this paper, we introduced CSS-Miner, a co-location pattern mining approach under the spatial structure constraint. We described how this constraint have 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. At the end, thanks to the OpenData platforms of Paris and Chicago, we have been able to create two real datasets.

However, during the data processing step, we chose to transform all our spatial objects into points. The next step of our work will be to keep initial type data (points, polygons, lines, ...). In addition, during the shortest path search step, a comparison between spatial objects and the road network is done. In order to optimize the shortest path search, an additional pruning step of the road network might be necessary. There are several works done on pruning, for instance a soft filter pruning for convolutional neural network [11] or an online graph pruning applied on grid maps [10]. So the next step of our work will be to prune the graph associated to the road network in order to accelerate our shortest path search process.

Moreover, CSS-Miner is an explanatory analysis method, so the next step of our work will be to integrate knowledge from experts [8], such as urban planners and geographers, in order to verify the relevancy of the extracted spatial patterns.

Finally, in this paper, we assumed that the path taken is on foot. For the next step, to consider the path taken by a car, we will intend to consider the spatial structure as a directed graph, since all roads taken by a car are not bidirectional. Moreover, in order to extract interesting co-location pattern, it is necessary to integrate the temporal dimension with peak hours impacting the traffic network. However, adding this temporal dimension requires data unavailable on OpenData platforms, therefore the next steps will be to integrate edge direction and to use APIs provided by Google and other traffic management companies.

References

1. Akbari, M., Samadzadegan, F., Weibel, R.: A generic regional spatio-temporal co-occurrence pattern mining model: a case study for air pollution. Journal of

Geographical Systems **17**(3), 249–274 (2015)
2. Bao, X., Wang, L.: A clique-based approach for co-location pattern mining. Information Sciences **490**, 244–264 (2019)
3. Boeing, G.: Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. Computers, Environment and Urban Systems **65**, 126–139 (2017)
4. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Communications of the ACM **16**(9), 575–577 (1973)
5. Cazals, F., Karande, C.: A note on the problem of reporting maximal cliques. Theoretical computer science **407**(1-3), 564–568 (2008)
6. Chiu, J., Khezerlou, A.V., Zhou, X.: Understanding business location choice pattern: A co-location analysis on urban poi data. In: Proceedings of the 2nd INFORMS Workshop on Data Science, Phoenix, AZ, USA. vol. 3 (2018)
7. Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)
8. 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)
9. Fredman, M., Tarjan, R.: Fibonacci heaps and their uses in improved network optimization algorithms. In: 25th Annual Symposium onFoundations of Computer Science, 1984. pp. 338–346 (1984). https://doi.org/10.1109/SFCS.1984.715934
10. Harabor, D., Grastien, A.: Online graph pruning for pathfinding on grid maps. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 25, pp. 1114–1119 (2011)
11. He, Y., Kang, G., Dong, X., Fu, Y., Yang, Y.: Soft filter pruning for accelerating deep convolutional neural networks. arXiv preprint arXiv:1808.06866 (2018)
12. 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)
13. Johnson, D.B.: Efficient algorithms for shortest paths in sparse networks. J. ACM **24**(1), 1–13 (jan 1977). https://doi.org/10.1145/321992.321993, https://doi.org/10.1145/321992.321993
14. 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)
15. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: International Symposium on Spatial Databases. pp. 47–66. Springer (1995)
16. Kwan Kim, S., Kim, Y., Kim, U.: Maximal cliques generating algorithm for spatial co-location pattern mining. In: Secure and Trust Computing, Data Management and Applications: 8th FIRA International Conference, STA 2011, Loutraki, Greece, June 28-30, 2011. Proceedings 8. pp. 241–250. Springer (2011)
17. Moon, J., Moser, L.: On cliques in graphs. Israel J. Math. **3**, 23—-28 (1965)
18. Selmaoui-Folcher, N., Flouvat, F., Gay, D., Rouet, I.: Spatial pattern mining for soil erosion characterization. In: New Technologies for Constructing Complex Agricultural and Environmental Systems, pp. 190–210. IGI Global (2012)
19. 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)

20. Thorup, M.: Undirected single-source shortest paths with positive integer weights in linear time. J. ACM **46**(3), 362–394 (may 1999). https://doi.org/10.1145/316542.316548, https://doi.org/10.1145/316542.316548
21. 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)
22. 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)
23. 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)
24. Valiant, L.: The complexity of enumeration and reliability problems. SIAM Journal on Computing **8**(3), 410—-421 (1979)
25. Wang, L., Bao, Y., Lu, Z.: Efficient discovery of spatial co-location patterns using the icpi-tree. The Open Information Systems Journal **3**(1) (2009)
26. Yao, X., Peng, L., Yang, L., Chi, T.: A fast space-saving algorithm for maximal co-location pattern mining. Expert Systems with Applications **63**, 310–323 (2016)
27. 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)
28. Yu, W.: Spatial co-location pattern mining for location-based services in road networks. Expert Systems with Applications **46**, 324–335 (2016)
29. Yu, W., Ai, T., He, Y., Shao, S.: Spatial co-location pattern mining of facility points-of-interest improved by network neighborhood and distance decay effects. International Journal of Geographical Information Science **31**(2), 280–296 (2017)