DE GRUYTER OPEN

Int. J. Appl. Math. Comput. Sci., 2017, Vol. 27, No. 4, 681–695
DOI: 10.1515/amcs-2017-0047

amcs

METHODS FOR MINING CO–LOCATION PATTERNS WITH EXTENDED SPATIAL OBJECTS

ROBERT BEMBENIK [a,*], WIKTOR JÓŹWICKI [a], GRZEGORZ PROTAZIUK [a]

[a] Institute of Computer Science
Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: {R.Bembenik,G.Protaziuk}@ii.pw.edu.pl, w.jozwicki@gmail.com

The paper discusses various approaches to mining co-location patterns with extended spatial objects. We focus on the properties of transaction-free approaches EXCOM and DEOSP, and discuss the differences between the method using a buffer and that employing clustering and triangulation. These theoretical differences between the two methods are verified experimentally. In the performed tests three different implementations of EXCOM are compared with DEOSP, highlighting the advantages and downsides of both approaches.

Keywords: spatial data mining, co-location patterns, extended objects.

1. Introduction

According to Fayyad *et al*. (1996), data mining is the application of specific algorithms to extracting patterns from data. Spatial data mining refers to patterns that relate to space, and aims to capture knowledge. Li *et al*. (2016) posit that the captured knowledge may refer to spatial or non-spatial properties of the analyzed objects; it is previously unknown, potentially useful, and ultimately understandable. This knowledge can uncover descriptions and predictions of patterns of spatial objects, such as spatial rules, general relationships, summarized features, conceptual classification, and detected exceptions.

Uncovered spatial rules can be divided into co-location rules and spatial association ones. The former concern co-occurrences of objects (Shekhar and Xiong, 2007) and are based on spatial co-location patterns (Xiong *et al*., 2004), i.e., subsets of Boolean spatial features frequently located in close proximity. An instance of a co-location pattern is a set of spatial objects which satisfy feature and neighborhood constraints. A co-location rule is a rule of the form in which one co-location pattern indicates a probable occurrence of another one. It refers to patterns of features that do not intersect. Spatial association rule discovery (Shekhar and Xiong, 2007) is a similar problem, but focuses on finding patterns related to defined reference features. Hence, a spatial association pattern specifies reference features and a set of features which tend to occur in their neighborhood. This does not suggest whether they are neighbors to each other as well, but provides information on frequently occurring features in the proximity of these specified as a reference. In this paper we focus on spatial co-location rules.

The above-mentioned problems of finding patterns in spatial data consider spatial objects mostly to be points representing specific instances of places belonging to categories of places (e.g., restaurants, shops, houses). However, spatial objects not always can be represented in the form of a single point. Examples of such objects are the so-called extended objects, such as roads, railways, rivers, electrical networks, etc. To discover patterns concerning such objects, we need a dedicated representation of these objects for the purpose of use in a knowledge discovery process and specialized data mining algorithms. This paper analyses and compares two such methods of representation, EXCOM and DEOSP, and their impact on results of the data mining process.

EXCOM (Xiong *et al*., 2004) is a buffer-based approach. It relies on the notion of a neighborhood of an extended spatial object and constitutes an expansion of the method proposed by Shekhar and Huang (2001), allowing one to consider extended objects (straight lines, line strings and collections thereof) in the discovered patterns. The method relies on a spatial database system (SDS) and extensive use of the buffer operation (employed

*Corresponding author

in SDSes for proximity analysis and constituting a zone around a map feature measured in units of distance).

DEOSP is another method allowing one to discover co-location patterns for extended objects, although it does not allow operating on such extended objects as areas. It is an extension of FARICS (Bembenik and Rybiński, 2009) by a fairly complex algorithm of determining triangulation for line segments. The method does not require using spatial joins. Relations among instances of spatial features are determined based on the neighborhood model utilizing the Dalaunay diagram structure in which edges and triangles indicate cliques (co-location patterns). Line segments building up linear objects are represented by points, while point objects are treated as degenerated line segments. DEOSP limits searching for frequent spatial patterns to areas with large concentration of objects, assuming that they are sufficiently representative for a problem at hand. In this approach it is necessary to create groups, yet clustering is not realized on complex objects but on points and points on line segments constituting relations among objects.

Example application domains for spatial co-location rules and co-location rules relating to extended spatial objects can be applied in urban computing. Urban computing is a process of acquisition, integration, and analysis of big and heterogeneous data generated by a diversity of sources in urban spaces, such as sensors, devices, vehicles, buildings, and humans, to tackle the major issues that cities face (Zheng *et al*., 2014). The goal of urban computing is to improve peoples lives, city operation systems, and the environment unobtrusively and continuously. The areas of urban computing that can benefit from application of spatial co-location/association rules are urban planning and place recommendation services.

Urban planning concerns optimal localization of service points and buildings in the urban space. In the case of service points, a badly chosen location can cause losses up to thousands or even millions of dollars. A simple example may be a popular café in the downtown of the city which is always crowded and a similar place a few hundred meters down the street which has a lot less visitors. The key to solving the problem of cafés' popularity here is an analysis that allows determination of factors influencing the prosperity (or its lack) of a given service point. External factors here can be the location in a large/small agglomeration, a neighborhood of other service points, the proximity of a communication route, or fashion (another factor is the attractiveness of the place itself). By using the data shared by LBSN services (location based social networks, such as Foursquare) in combination with spatial data mining techniques, it is possible to solve this problem efficiently (Karamshuk *et al*., 2013). Rules discovered in data mining analysis show what types of places are frequently located in close proximity. Additional analysis of LBSN data concerning these places (e.g., the number of check-ins, times when the check-ins occur, characteristics of people that do the check-ins) can help uncover the full picture of the situation behind the popularity of the places.

The knowledge of mutual interrelationships between objects/groups of objects/communication routes seems to be crucial for the problem of place recommendation. Interrelationships in the form of spatial co-location or association rules can be discovered using spatial data mining techniques discussed in this paper. If one knows, for example, that a lot of people (the number of people visiting the places can be inferred based on LBSN check-in data) are interested in cafés located in the neighborhood of cinemas with parking in the vicinity of the main road, then he or she is able to make precise and potentially useful recommendations. Having knowledge collected empirically in this way, one will not recommend places (that may seem good at first sight) not fulfilling all criteria of a place discovered as worth recommending for the given context (e.g., known history of places favoured by the user seeking recommendation).

The rest of the paper is structured in the following way. In Section 2 we discuss related work. In Section 3 methods of finding spatial patterns in datasets with extended objects, i.e., an approach using buffers around objects (EXCOM) and an approach utilizing Delaunay diagrams (DEOSP), are discussed. Section 4 presents experimental comparison of EXCOM and DEOSP. Finally, Section 5 concludes the paper.

2. Related work

Approaches to mining co-location patterns, especially those with extended spatial objects found in the literature, according to Li *et al*. (2014), can be classified into support-confidence co-location mining and statistical test co-location mining.

The first group of algorithms follows the concept of association rule mining introduced by Agrawal and Srikant (1994). In the reference-feature approach proposed by Koperski and Han (1995) neighboring objects are computed in order to materialize transaction sets around the instances of the reference objects. Transactions are built around one object type specified by the user. Then, association rules are created using the *apriori* algorithm. the neighborhood is defined in terms of the user specified distance d. Appice *et al*. (2005) present a spatial data mining system called ARES employing the reference-feature approach using spatial hierarchies that assists in the process of extracting spatial association rules. The system can find rules including multiple object types and extended spatial objects, e.g., road networks, water networks, urban areas, green areas. It employs a client-server architecture. The server side

is responsible for the data mining process allowing the use of hierarchies and applying constraints. GUI is located on the client side and allows one to control the data mining process. Geometrical, directional and topological features necessary for the mining of the rules are extracted from the spatial database by a middle layer that constitutes the coupling between the spatial database and the inductive logic programming system SPADA (Lisi and Malerba, 2004)—the server side of ARES. The advantage of ARES is that it allows the user to specify their criteria to reduce the pattern search space in the rule discovery process. The criteria are expressed as pattern constraints in the following form: *pattern_constraint(AtomList, Min_occur, Max_occur)*. *AtomList* is the list of conjunctive constraints, while *Min_occur* and *Max_occur* specify respectively the minimum and the maximum number of constraints that the pattern has to satisfy. The constraints can concern either the antecedents or consequents of spatial association rules. The constraints defined in ARES do not prevent the generation of candidate rules but the evaluation of their confidence.

Loglisci *et al*. (2010) considers a descriptive data mining approach to discovering relational disjunctive patterns in spatial networks. Such an approach allows considering variants of spatial relationships existing between two objects. The approach utilizes the SPADA system and is composed of three steps: (i) extraction of infrequent conjunctive patterns that can be upgraded to the disjunctive form, (ii) accommodation of background knowledge to exploit the similarity among the spatial relationships in the process of generation of disjunctive patterns, (iii) generation of disjunctive patterns by iterative integration of disjunctive patterns with pair-wise joining. Objects in the spatial network being considered here are divided into target (TO) and non-target (NTO). Properties and relationships of objects are represented by predicates such as key predicates identifying the analyzed objects, predicates defining the values taken by TOs and NTOs, binary predicates relating TO and NTO with other NTOs, is_a predicate associating NTO with symbols in user-defined taxonomy. An example of a spatial association rule enriched with the disjunctive pattern is *district(A), [comes_from(A,C)—external_ends_at(A,C)],is_a(C,road), comes_from(A,B), is_a(B,rail)*, which introduces the disjunctions and states that the road named *C* can be connected to the district named *A* through two possible alternative ways.

Kim *et al*. (2014) propose a framework for co-location pattern mining that uses a transaction-based approach and employs maximal cliques as transaction-type datasets. The proposed framework can be used to mine co-location rules among both point-type as well as extended-type objects. The main building blocks of the framework encompass neighboring graph generation, generation of maximal cliques and application of association analysis methods. The neighboring graph generation phase requires a distance parameter and creates vertexes for spatial objects connecting them with edges wherever neighbor relationships are found. This step is realized by means of a GIS engine. The definition of a transaction here allows spatial objects to be included in more than one spatial co-location transaction.

In the event-centric model proposed by Shekhar and Huang (2001), neighborhoods are used instead of transactions and a participation index (PI) as a measure of prevalence. PI is anti-monotonic and helps prune the search space in the process of searching for prevalent co-location patterns. The *participation index* of a co-location $C = \{f_1, f_2, \ldots, f_k\}$ is defined as $\min_{f_i \in C}\{Pr(C, f_i)\}$, where $Pr(C, f_i)$ is the participation ratio for the feature f_i in the co-location *C*. The value $Pr(C, f_i)$ is a fraction of instances of f_i that participate in any instance of the co-location *C*. A co-location is prevalent if its PI value is greater than some user-specified threshold. A high value of the participation index indicates that spatial features in the co-location pattern appear together with high probability. The ideas presented in this approach that are suitable for mining co-location rules in datasets comprising point objects were further developed by Xiong *et al*. (2004) to include extended objects.

Statistical approaches to mining spatial co-location patterns claim to overcome the problem of deciding on the threshold value of the participation index (used in support-confidence based co-location mining approaches) to filter out the discovered rules in the pruning step of algorithms. Statistical test co-location mining (Barua and Sander, 2011; Adilmagambetov *et al*., 2013; Li *et al*., 2014) uses a statistical test to decide whether an observed co-location is significant.

Barua and Sander (2011) argue that the PI threshold should not be global but decided on based on the distribution and the number of instances of each individual feature involved in a co-location. To achieve this, a prevalence measure is computed to measure the spatial dependency among features in a co-location. Next the null hypothesis of no spatial dependency against a hypothesis of spatial dependency among the spatial features is tested. For each co-location pattern the probability p measuring the prevalence value under a null hypothesis is computed with the computationally expensive randomization test. In the null model, features maintain a similar spatial distribution as in the observed data without any inter-dependency among the features. The pattern is considered statistically significant if for a given level of significance α the prevalence measure value computed from the observed data fulfills $p \leq \alpha$.

The ideas presented by Barua and Sander (2011)

are only applicable to objects represented as points. Adilmagambetov *et al*. (2013) expand the approach so that it can be applied to extended spatial objects. The presented approach is transaction based. The transactions are created using a grid with imposed points. Buffers are built around spatial (point or linear) objects. Grid points may intersect one or several spatial objects and their buffers. A transaction is defined as a set of features corresponding to these objects. The size of the grid has significant influence on the obtained results. As the authors observed, too large a distance between grid points may lead to omission of some regions of space while too short a distance leads to a greater number of transactions and in the effect dramatically extends the amount of required computation. The expected support is used as a measure of prevalence. In the proposed algorithm the number of elements of the rules is limited to three, because statistical significance used to prune insignificant co-location rules does not have the monotonic property.

Li *et al*. (2014) propose an algorithm that overcomes the limitations in the co-location rule size reported by Adilmagambetov *et al*. (2013). It allows discovering co-location rules with one fixed consequent. The idea is based on the property of potential significance that is monotonic in some situations. Because of that, co-location rules are discovered in a fashion similar to *apriori* approaches. The algorithm first arranges the set of antecedent features in ascend order of their frequencies, then similarly as in the *apriori* algorithm, the candidate generation process is performed, and finally the rules are pruned based on their z-score value representing the upper bound for the binominal distribution if it is lower than the assumed minimum threshold value.

The various approaches to mining co-location patterns with extended spatial objects have their advantages, yet all are based on the neighborhood defined in terms of a user-specified distance. Statistical approaches to mining spatial co-location patterns promise to overcome some limitations of support-confidence methods, but they are very computationally expensive. We claim that an event-centric support-confidence approach does not have to be computationally aggravating and returns reasonable results with no distance parameter. To this end, we compare EXCOM and DEOSP.

3. Discovering co-location patterns in datasets with extended spatial objects with EXCOM and DEOSP

In this section we discuss two methods of mining co-location patterns with extended objects: a buffer-based approach EXCOM and DEOSP—an approach utilizing Delaunay diagrams. The purpose of this discussion is identification of key concepts utilized by these methods so as to allow their comparison. Detailed presentation of both approaches is made by Xiong *et al*. (2004) and Bembenik *et al*. (2014).

3.1. EXCOM. EXCOM (Xiong *et al*., 2004) is a buffer-based approach. A buffer is a standard operation available in most spatial database systems and returns a geometry covering all points within a given distance from the input geometry (PostGIS, 2017). The buffer function creates a rounded buffer around a point, line, or polygon (Oracle, 2017). The approach used in EXCOM is based on the notion of a neighborhood of an extended spatial object. Spatial objects are represented by features and their instances. A road can be an example spatial feature. Specific roads scattered across the city are instances of that feature. The resultant co-location patterns take into consideration spatial features.

The EXCOM algorithm consists of two phases: filtering and refinement. In the filtering phase only coarse co-location patterns are discovered, whereas in the refinement phase proper co-location patterns are discovered based on the coarse ones. Such an approach is possible because co-location patterns constitute a subset of coarse co-location patterns. This property is proven by Xiong *et al*. (2004).

The key concepts characteristic for this approach, i.e., the definitions of notions used in the algorithm as well as a description of the algorithm, are given in the consecutive subsections.

3.1.1. Definitions relative to EXCOM.

Definition 1. (*Size-d Euclidean neighborhood—point*) (*Xiong* et al., *2004*) The size-d Euclidean neighborhood of a point location (denoted by $N(p)$) is a circle hid radius d with p as its center.

Definition 2. (*Size-d Euclidean neighborhood—spatial object*) (*Xiong* et al., *2004*) The size-d neighborhood of an extended spatial object (denoted by $N(o)$) (e.g., polygon, linestring) is defined by the buffer operation of size d for the object.

Examples of buffer-based neighborhoods for sample spatial objects are presented in Fig. 1. In the top row spatial objects are presented, whereas neighborhoods associated with these objects are in the bottom row. As can be seen, shapes of neighborhoods are different and depend on those of spatial objects.

Further definitions of notions used in the buffer-based model include the following (Xiong *et al*., 2004):

Definition 3. (*Euclidean neighborhood—feature*) (*Xiong* et al., *2004*) The Euclidean neighborhood $N(f_j)$ of a feature f_j is the union of $N(i_l)$ for every instance i_l of the feature f_j.

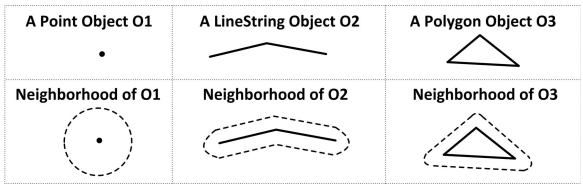


Fig. 1. Examples of buffer-based neighborhoods for points and sample extended objects) (Xiong *et al*., 2004).

Definition 4. (*Euclidean neighborhood—feature set*) (*Xiong* et al., *2004*) The Euclidean neighborhood $N(f_1 f_2 \dots f_k)$ for a feature set $C = \{f_1 f_2 \dots f_k\}$ is the intersection of $N(f_i)$ for every feature f_i in C.

Definition 5. (*Coverage ratio*) (*Xiong* et al., *2004*) The coverage ratio $Pr(f_1 f_2 \dots f_k)$ for a feature set $C = \{f_1 f_2 \dots f_k\}$ is computed according to the following formula: $N(f_1 f_2 \dots f_k)/D$, where $N(f_1 f_2 \dots f_k)$ is the Euclidean neighborhood of the set C and D is the total area of the plane.

The prevalence is represented by the coverage ratio, i.e., if the coverage ratio is greater than a user-specified minimum prevalence threshold, the feature set is a co-location pattern.

Definition 6. (*Conditional probability*) (*Xiong* et al., *2004*) The conditional probability $Pr(C2|C1)$ of a co-location rule $C_1 \rightarrow C_2$ is the probability of finding the neighborhood of C_2 in the neighborhood of C_1. It can be computed as $N(C_1 \cup C_2)/N(C_1)$, using the co-locations C_1 and $C_1 \cup C_2$.

Definition 7. (*Bounding neighborhood—object*) (*Xiong* et al., *2004*) $BN(o)$, the bounding neighborhood of a spatial object o (e.g., point, polygon, line-string), is defined as $MBBR(Buffer(MOBR(o), d))$, where $MOBR$ is the minimum object bounding rectangle, $Buffer$ is the buffer operation with buffer size d, and $MBBR$ is the minimum buffer bounding rectangle.

Definition 8. (*Euclidean bounding neighborhood—feature*) (*Xiong* et al., *2004*) The Euclidean bounding neighborhood $BN(f_j)$ of a spatial feature f_j is the union of $BN(i_l)$ for every instance i_l of the spatial feature f_j.

Definition 9. (*Euclidean bounding neighborhood—feature set*) (*Xiong* et al., *2004*) The Euclidean bounding neighborhood $BN(f_1 f_2 \dots f_k)$ for a feature set $CC = \{f_1, f_2, \dots, f_k\}$ is the intersection of $BN(f_i)$ for every feature f_i in CC.

Definition 10. (*Coarse-level coverage ratio*) (*Xiong* et al., *2004*) The coarse-level coverage ratio $CPr(f_1 f_2 \dots f_k)$ for a set $CC = \{f_1, f_2, \dots, f_k\}$ is $BN(f_1 f_2 \dots f_k)/D$, where $BN(f_1 f_2 \dots f_k)$ is the Euclidean bounding neighborhood of the set CC and D is the total area of the plane.

Definition 11. (*Coarse-level co-location pattern*) (*Xiong* et al., *2004*) A coarse-level co-location pattern is a set of spatial features with a coarse-level coverage ratio greater than a user-specified minimum prevalence threshold.

The coverage ratio is used to determine prevalence of a feature set. If for a given set the coverage ratio is larger than a value declared by a user, the set is a co-location pattern. The conditional probability is the measure of the confidence of a rule. If for a candidate co-location pattern the conditional probability is larger than the user-defined threshold, then such a rule is a co-location rule.

The main challenge in the buffer-based approach is the multitude of spatial join operations using spatial intersections. Due to the high cost of such operations, the structure of the EXCOM algorithm has been based on the filter-and-refine paradigm. Using the buffer and a double operation of determining the minimum bounding box, it aims to reduce the number of spatial joins through initial elimination of spatial sets that cannot participate in co-location patterns.

3.1.2. EXCOM algorithm. As mentioned at the beginning of Section 3.1, the algorithm consists of filter and refine phases.

The first phase (filtering) is based on constructing auxiliary objects in the form of bounding neighborhoods, using the buffer operation and applying the MBR (minimum bounding rectangle) twice. Owing to that, spatial join operations are always performed on rectangles, which simplifies computations. These rectangles are spatially summed up which results in the bounding neighborhood of a spatial object. Spatial features that build up a candidate for a coarse co-location pattern in the filtering phase are intersected spatially to result in a bounding neighborhood of that candidate. To determine whether a feature set is a coarse co-location pattern, it is necessary to compute the coarse-level coverage ratio and compare it with a user-defined threshold.

In the second phase (refinement), a filtered set of coarse co-location patterns from the first phase is used for performing operations of creating regular buffers, spatial joins on non-rectangular objects and candidate coverage ratio computations. Based on the results of these computations, a decision is made whether the candidates are co-location patterns.

Two-phase reduction of the number of candidates is possible if spatial objects are represented appropriately, e.g., using a quad-tree and at the same time geometrical filtering of the space, and due to the fact that the coverage ratio of co-location patterns is monotonically non-increasing with the size of the co-location patterns

increasing. One has to note that geometric filtering is used only to determine coarse co-location patterns of size 2. For more complex patterns, combinatorial searching is used for a set of patterns of a size decremented by one.

3.2. DEOSP. DEOSP (Bembenik *et al*., 2014) is another method allowing one to discover co-location patterns for extended objects (straight lines, line strings and collections thereof), although it does not allow operating on extended objects such as areas. DEOSP is based on structures related to the Delaunay diagram.

3.2.1. Definitions relative to DEOSP. The main notions of DEOSP are presented below as reported by Bembenik *et al*. (2014). Spatial tessellations utilized for mining co-locations including extended spatial objects in this approach are the Voronoi diagram, the Delaunay triangulation, as defined by Okabe *et al*. (2009), the constrained Delaunay triangulation and the conforming Delaunay triangulation, whose definitions are given below.

Definition 12. (*Constrained Delaunay triangulation)* (*Okabe* et al.*, 2009*) For a given planar straight-line graph $G(P_g, L_g)$ representing obstacles and a set Q of points, the constrained Delaunay triangulation is a triangulation spanning $P = P_g \cup Q$ satisfying the condition that the circumcircle of each triangle does not contain in its interior any other vertex which is visible from the vertices of the triangle. Constrained triangulation is generally different from a Delaunay triangulation. But if we add a set S of points on L_g, then the constrained Delaunay triangulation spanning $P \cup S$ may coincide with the ordinary Delaunay triangulation spanning $P \cup S$. Such a special Delaunay triangulation is called a constrained Delaunay triangulation.

Definition 13. (*Conforming Delaunay triangulation)* (*Okabe* et al.*, 2009*) For a given planar straight-line graph $G(P_g, L_g)$ representing obstacles and a set Q of points, we consider a set S of additional points and construct the ordinary Delaunay triangulation $D(P_g \cup Q \cup S)$ spanning $(P_g \cup Q \cup S)$. If all line segments in L_g are the union of the edges of $D(P_g \cup Q \cup S)$, we call $D(P_g \cup Q \cup S)$ the *conforming Delaunay triangulation*.

3.2.2. Mining extended spatial objects: An outline of the method. In order to accomplish the process of mining co-location rules with extended spatial objects, the following assumptions need to be made:

- Line segments are represented as end points without intermediate points on the line segment.
- Additional points are selected in such a way that there exist edges between endpoints of the line segments or the existing edges build a line segment that links them.
- A constrained Delaunay triangulation corresponding to the topology of the triangulation for line segments is utilized.
- Only nearby objects are taken into consideration.

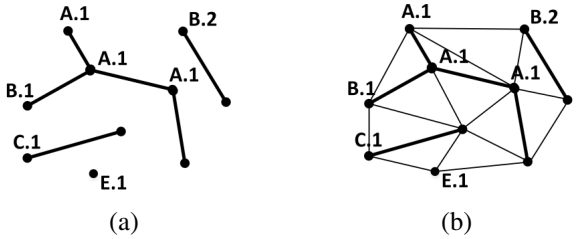
An *apriori*-like operation is employed to generate candidates and co-location rules.

Taking into consideration the aforementioned spatial tessellations as well as the assumptions, the mining process for extended spatial objects can be realized in the following steps: (i) construction of a constrained Delaunay triangulation, (ii) construction of a conforming Delaunay triangulation, (iii) selection of triangles for processing, (iv) removal of relations among objects that do not belong to the same groups, (v) finding co-location instances, (vi) discovering co-location rules in spatial data.

We present these steps in detail after Bembenik *et al*. (2014).

Construction of a constrained Delaunay triangulation. Input data in the form of coordinates of point objects and line segment endings is used to create a Delaunay triangulation between endings of the imposed line segments. In the process of triangulation construction, each node stemming from a point object is labeled with its type and instance. Line segments appearing in line objects are constraints defined by the beginning and ending of the line segment. Endings of the imposed segments have the same type and instance. Only the first segment ending receives a label. Sample input data and a constrained Delaunay triangulation are shown in Fig. 2. The sample data (Fig. 2(a)) contain constraints in the form of line segments labeled A, B and C and their instances (e.g., A.1, B.1). The only point in this dataset is instantiated as E.1. Figure 2(b) shows the constrained Delaunay triangulation for the sample data; thick lines represent constraints.

Construction of a conforming Delaunay triangulation. Based on the triangulation created in the first step of the method, a conforming Delaunay triangulation is created,

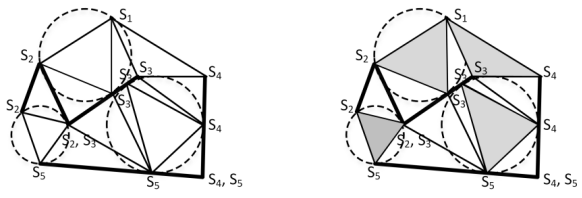


(a) (b)

Fig. 2. Sample input data (a), constrained Delaunay triangulation with labeled vertices (b) (Bembenik *et al*., 2014).

i.e., a triangulation having all edges being Delaunay ones. The process is based on the following property: each constrained triangulation can be transformed to a conforming Delaunay triangulation (Rineau, 2017). In most cases it is necessary to introduce additional points being vertices to the triangulation. They are labeled with the type and instance. This is achieved in such a way that all vertices of the imposed edges receive a label of the type and instance of the first ending of the edge. Because of possible intersections of objects having a common ending point, as shown in Fig. 3(a), for segments S.2 and S.3, each point can have many labels. In this example the common vertex has two labels: S.2, S.3. Adding a new point to the triangulation is yet dependent on the imposed edges of one segment, so adding labels refers only to the label of the edge ending related to this segment, ignoring the remaining edges. It is visible for added points labeled S.3 that do not constitute the ending of this segment. The outcome of this step is a constrained Delaunay triangulation consistent with Delaunay triangulation for points with labels representing object instances, which is shown in Fig. 3(a).

Selection of triangles for processing. The purpose of this step is elimination of triangles that bring in redundant information concerning topology of objects. It is noteworthy that the occurrence of a combination, such as S.2, S.2, S.5, does not indicate the existence of an imposed edge in the triangle. Such is the case, e.g., for the triangle in Fig. 3(b) having such a combination. The triangle can be rejected when there is no other correct combination for it. It then complies with the assumptions of a Voronoi diagram where all generators are different. One should, however, expect that the relation between two objects is included in a different triangle of the diagram. The outcome of this step consists in selected triangles of the constrained Delaunay triangulation for the labeled points representing instances of different objects. Rejected triangles are not considered in further processing.



(a) (b)

Fig. 3. Constrained Delaunay triangulation consistent with the Delaunay triangulation: the thick line denotes imposed edges, the labels denote object instances represented by vertices (a). Selection of triangles with non-redundant information: the triangles kept are marked in gray (b) (Bembenik *et al*., 2014).

Removal of relations among objects that do not belong to the same groups. Removal of relations among objects belonging to various groups follows the method used in the NSCABDT algorithm (Yang and Cui, 2008). For all retained triangles, further referred to as correct, the mean length of edges ($Global\_mean$) and the global standard deviation ($Global\_stddev$) are calculated. To reject or retain an edge between two vertices, it is necessary to compare the length of edges with the value of a discriminating function given by the following formula:

$$F(v) = Global\_mean + Global\_stddev \times \frac{Global\_mean}{Local\_mean(v)}$$

where $Local\_mean(v)$ denotes the mean length of the incident edges of vertex v, computed with the formula

$$Local\_mean(v) = \frac{1}{K}\sum_{k=1}^{K} Len(e_k).$$

The edges for which the relation $Len(e_k) > F(v)$ holds are removed. The outcome of the step are edges connecting objects belonging to the same clusters. They can be regarded as edges of triangles with missing edges.

Finding co-location instances. The purpose of this step is to determine which objects build up cliques. Having information on cliques, it is feasible to search for co-location rules. The cliques build up object clusters described by their correct label combinations. If one or two edges in the triangle were rejected, each retained edge is a two-element clique for the combination of labels of different types and instances. Consequently, for a sample triangle with edges (1, 1) – A.1, (2, 2) – B.1, (3, 3) – A.2, C.2, after rejecting edges (1, 1) – (2, 2) we get instances of a clique for the retained: A.1, C.2 – the correct combination of edges (1, 1) – (3, 3); B.1, A.2 and B.1, C.2 are instances of a clique for the edges (2, 2) – (3, 3).

Discovering co-location rules. The last step of the method is realized similarly as in the *apriori* algorithm. To compute co-location rules, we need to use the notions of a participation ratio, a participation index for a group of types, and the notion of a co-location rule as well as the confidence of a co-location rule. We use those concepts based on their definitions in (Bembenik and Rybiński, 2009).

3.3. Comparison of the existing methods and implementation remarks.

Below, we provide a summary of the properties of the algorithms EXCOM and DEOSP.

1. **Required parameters given by the user** (aside from prevalence and confidence).
   EXCOM: Buffer size.
   DEOSP: No additional parameters required.

2. **Given extent of the object neighborhood**.
   EXCOM: All objects in the range of the buffer.
   DEOSP: Only the nearest neighbors, i.e., objects directly linked with an edge in the triangulation.

3. **Adjustment to neighborhood distribution and protection of distant objects**.
   EXCOM: The method does not adjust to objects irregularly remote, but protects objects fairly and very distant outside of the buffer.
   DEOSP: The method adjusts to objects irregularly remote, but protects only very distant objects from other clusters.

4. **Does the closeness of objects influence the quality of the pattern?**
   EXCOM: Yes, neighboring objects that are closer to each other generate a higher coverage ratio than distant neighboring objects.
   DEOSP: No.

5. **Do objects' shapes influence the quality of the pattern?**
   EXCOM: Yes, linear objects most often influence a higher value of the ratio than point objects similarly distant from each other.
   DEOSP: No.

6. **Object types**.
   EXCOM: All extended objects.
   DEOSP: Only linear and point objects.

The EXCOM algorithm was implemented in Oracle 11g DBMS with spatial extension. To achieve the best results, three versions of the algorithm were implemented and the best-performing ones were selected for comparison of EXCOM and DEOSP. The first version of the algorithm is consistent with the algorithm introduced by its authors. When analysing whether a candidate is a pattern or a coarse pattern, the spatial neighborhoods of objects are intersected with each other in turn (each neighborhood separately). Finally, having a set of neighborhoods when considering candidate features, we compute bounding neighborhoods, avoiding a double counting of the overlapping area at the same time. The second version of the algorithm introduces an alternative approach to computing the area used by the neighborhood of a candidate working on feature layers. In this approach at the beginning a layer representing the neighborhood of a given feature is created by summing up neighborhoods of objects of that feature on a plane. Having neighborhoods of all features separately, one can easily determine neighborhoods of feature sets. It is worth pointing out that in this approach the problem of overlapping neighborhoods of singular objects does not exist anymore.

In the third version of the implementation, spatial objects are processed similarly as in the first one. The only difference lies in prevalence pruning. For calculating the area, a function creating a layer being a union of neighborhoods of each candidate is used. The three implementations of EXCOM were run on the same datasets with the same parameters. For each version the same results were expected. In fact, the actual results were very similar but not identical. Most frequently, differences in the coverage ratios appear in at most 10% of the cases and mostly amount to around 1%. We suspect that this is due to the tolerance factor in Oracle Spatial. The smaller tolerance values the smaller the discrepancies and longer running times. In the efficiency tests the third implementation of EXCOM performed best. It was more than three times faster than the second version. The first version was so slow that it was eliminated in preliminary experiments.

DEOSP was implemented as a C++ program.

4. Experimental evaluation

In the experiments we used the following real datasets from the MetroGis website (`www.datafinder.org`) in shapefile format from the Twin Cities area: main roads, bus garages and largest shopping centers. The database of main roads contains 199 linear objects belonging to 4 road types: county road, interstate, state highway and US highway. Each object is composed of many segments. The bus garages database contains 16 objects of 2 types: metro transit MT garages and regional provider RP garages. Shopping centers database contains 330 objects of 6 types: community center, downtown center, mega center, neighborhood center, regional center and sub-regional center. The dataset is well suited for the comparison of EXCOM and DEOSP as it contains both objects represented as points and as polylines. As a result, it is possible to discover co-locations among points, points and polylines as well as polylines.

We ran three experiments focused on (i) efficiency comparison of EXCOM and DEOSP, (ii) different buffer sizes for EXCOM and the choice of an optimal buffer for further experiments, (iii) comparison of patterns and rule quality for rules generated by both methods with focus on differences on particular examples. All tests were performed on a PC with an Intel Core 2 Duo 2.27 GHz processor, 4 GB of RAM running Windows 7 Professional.

In the coming subsections we present the outcomes of the experiments in detail.

4.1. Efficiency comparison. Datasets used in this comparison concerned locations of bus garages and roads. We ran different implementations of EXCOM for different buffer sizes and compared the running times to DEOSP.

The comparison encompassed the following buffer sizes: 150, 500 and 1000 meters. Results of the conducted tests are collected in Table 1.

The third implementation of EXCOM was most computationally efficient. In the tests it was more than three times faster than version 2. The poor performance of version 1 for a buffer size of 150 m excluded it from further experiments. The same dataset was processed by DEOSP in 0.527 s, which makes it 40 to 60 times faster (depending on the buffer size in EXCOM). Of course, the conducted experiments do not allow direct comparison of efficiency of these two methods due to essential differences in their implementation. However, the following observations seem to be justified:

- the approach using a buffer, despite improvements, is typically much slower than the method not requiring this kind of operations;
- using general SDBMS for performing complex spatial operations seems not to provide comparable efficiency with dedicated implementation of such functionality.

4.2. Choice of an optimal buffer size for EXCOM. The goal of the experiment presented here was to determine the difficulty of choosing the buffer size in EXCOM, but also deciding what buffer size could be used in the next experiment. To realize the goal, two datasets were used, namely, main roads and shopping centers. A visualization of the utilized data is given in Fig. 4.

The basic input parameters, i.e., coverage ratio and confidence threshold, were set to very small values (10^{-10} for both cases) so that both candidates covering very small areas (even below 1 square meter) were reported as interesting patterns, as well as each candidate rule was a co-location rule. The test covered 5 runs of the EXCOM. The buffer size was incremented for each run and switched between 20, 50, 100, 300 and 500 meters.

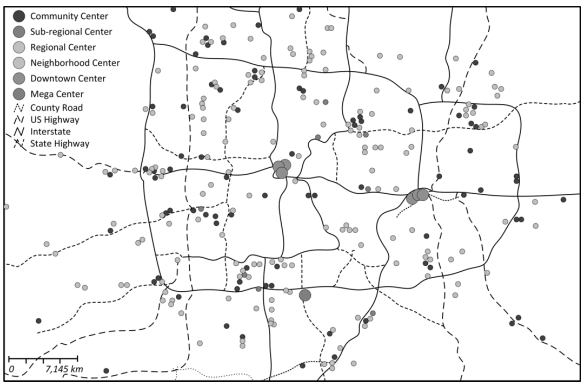


Fig. 4. Visualization of data consisting of roads and shopping centers.

Figure 5 presents visualization of all buffers for a selected area. In each run statistics concerning the run, and output sets (candidates, patterns, candidate and co-location rules) were gathered. A summary of these statistics is contained in Table 2.

The problem that we encountered during the tests concerned difficulties with aggregating union for a large number of objects by the Oracle Spatial system, which forced us to reduce the buffer size to 500 m. The diagram in Fig. 6 shows that the number of candidates and patterns stays on a low level for buffer sizes not exceeding 100 m. From the buffer sizes in the range 100–500 we can deduce that buffer sizes of about 1000 m may result in a similar increase in the number of candidates that would also considerably increase the computation time. The diagram in Fig. 7 shows an increase in the number of spatial joins relative to the buffer size.

It is difficult to unequivocally answer the question regarding the buffer size adequate for a given dataset. The choice of the right buffer size is usually preceded by trial and error experiments and is a separate problem to be analyzed in each case. The need to decide the value of an additional parameter does not occur for the DEOSP algorithm.

4.3. Patterns and rules quality. The goal of the next experiment was to compare the quality of the patterns and rules generated by DEOSP and EXCOM algorithms. Based on the results of previous tests, for the present experiment the buffer of size 500 m was selected. The reason for that was a considerable increase in the number of generated patterns and rules in comparison with a 300 m buffer with an acceptable processing time (below 1 minute). To realize the experiments, two datasets were used, similarly as in the previous experiment: roads and shopping centers.

The test encompassed a series of executions of both methods for the mentioned datasets with different parameters. Aggregate statistics considering these executions are given in Table 3.

Two essential things can be observed. Firstly, the number of frequent sets, as well as rules, with no

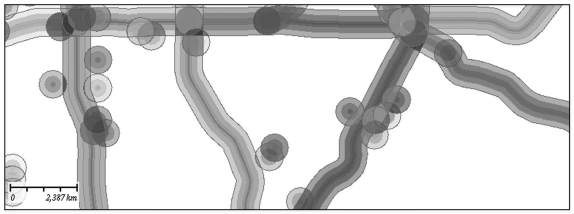


Fig. 5. Visualization of candidate buffers of size 20, 50, 100, 300 and 500 meters on a selected area for data concerning roads and shopping centers.

Table 1. Execution times of EXCOM implementations for different buffer sizes.

EXCOM			
Buffer size (m)	Time (version I) (s)	Time (version II) (s)	Time (version III) (s)
150m	421.8	75.411	22.885
500m	–	91.316	25.077
1000m	–	97.774	29.702

Table 2. Aggregate statistics for runs with buffer sizes of 20, 50, 100, 300 and 500 meters.

Aggregate statistics					
Buffer size (m)	20	50	100	300	500
Number of buffer intersections	207	231	272	688	1484
Number of candidates of size > 1	55	61	62	89	123
Number of frequent patterns of size > 1	13	15	17	55	93
Number of co-location rules	26	30	35	141	269

thresholds of prevalence and confidence is larger for the method EXCOM. This certainly is related to a large size of the buffer. Besides, it is easy to notice that there are differences in ways of computing confidence for both methods. The number of rules generated in the approach using a buffer for the minimum confidence values of 0.5 and 0.7 is lower than that for the DEOSP algorithm, even though the tendency was opposite with no limitations.

4.3.1. Patterns comparison. For pattern comparison we juxtaposed the pattern with the largest prevalence values for the methods under consideration. The results for DEOSP and EXCOM were separately sorted by decreasing prevalence values and then the patterns generated by DEOSP with prevalence of at least 0.2 were juxtaposed with their counterparts from EXCOM.

We only compared patterns of a size no larger than 3 as that is the maximum pattern size that can be discovered by DEOSP. Most patterns generated by both methods covered 2 and 3 elements, though a few patterns discovered with EXCOM were longer and comprised of 4 and 5 elements.

There is a significant difference in the distribution of the prev and coverage ratio values. Comparing these directly is impossible due to the significant dispersion of the coverage ratio. To mitigate this problem we: (i) reduced the influence of the square in the coverage ratio by calculating the square root from the value which significantly reduced the differences between the ratios, (ii) used proportion to achieve comparable orders of magnitude for both parameters (to do that we selected the pattern with the highest coverage ratio and based on the comparison with the prevalence value for the same pattern computed with DEOSP we assigned to it a proportional prevalence, which is summarized with the equation

$$PP_i = 0.9 \frac{\sqrt{CR_i}}{\sqrt{CR_{\{pwhCR\}}}}$$

where PP_i is proportional prevalence, CR_i is the coverage ratio, $pwhCR$ is the pattern with the highest coverage ratio.

The analysis of differences was based on the sequence of occurrence of a pattern in the results from both algorithms and on the values of the *Prev* and *Proportional Prevalence* parameters. The results are presented in Table 4.

We marked in gray the patterns that have high prevalence as computed by DEOSP, but much lower for EXCOM. One of the patterns was found using DEOSP and was not discovered with EXCOM. The cells marked

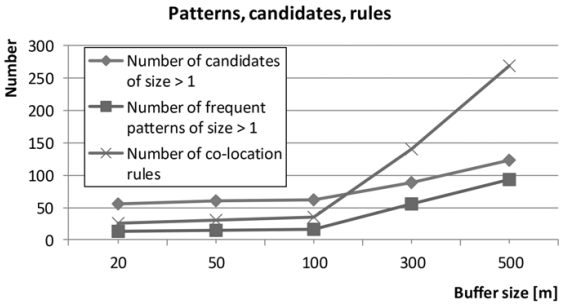


Fig. 6. Number of candidates, patterns and co-location rules.

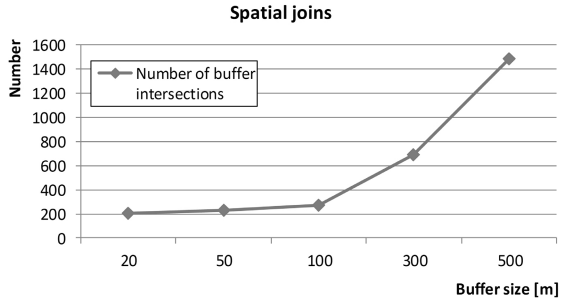


Fig. 7. Number of spatial joins relative to the buffer size.

Table 3. Aggregate statistics considering the number of frequent sets and the number of rules for the algorithms DEOSP and EXCOM executed with different parameters of prevalence and confidence.

DEOSP				EXCOM			
minPrev	minConf	# freq sets	# rules	minCov Ratio	minConf	# freq sets	# rules
1.00E-10	1.00E-10	89	204	1.00E-10	1.00E-10	103	269
0.2	1.00E-10	32	48	1.00E-06	1.00E-10	95	243
0.4	1.00E-10	23	26	1.00E-05	1.00E-10	57	115
0.5	1.00E-10	20	20	1.00E-04	1.00E-10	24	28
1.00E-10	0.5	89	53	1.00E-10	0.5	103	40
1.00E-10	0.7	89	26	1.00E-10	0.7	103	20

in light gray in the table have a high coverage ratio computed by EXCOM and a much lower prevalence computed by DEOSP.

The selected cases were further analyzed. For all cases marked in gray the cause was selection of too small a buffer size. For example, for the pattern Rl_Ctr, $Intst$ shown in Fig. 8, one can see that most bounding neighborhoods of objects of type Rl_Ctr do not overlap the buffers of the Interstate only two of eight dark gray circles overlap. For this pattern to be more frequent, one would have to enlarge the buffer size twice, then only three buffers would not overlap.

The patterns marked in light gray differ as far as the prevalence values are concerned in both methods for two reasons. The first one is a different approach to compute this parameter. In DEOSP the participation ratio is used for this purpose that takes into consideration the prevalence of objects of each type in the pattern in comparison to the total number of occurrences of the pattern in the area considered. In EXCOM there is no context of the sum of areas of bounding neighborhoods in the pattern. The only context is the total area of the plane under consideration. For this reason the participation ratio seems to reflect the real prevalence for the patterns much better.

Figure 9 shows the multitude of shopping centers of type Nbh_Ctr (light gray color) and relatively rare occurrences of the $County\_Rd$ type in their neighborhood, due to a small number of roads of such a type.

The second reason for the differences in prevalence is depicted in Fig. 10, where with X the objects of the type Nbh_Ctr are marked which will not have a common edge in DEOSP triangulation with edges of type $County\_Rd$ (due to more closely situated objects of different types

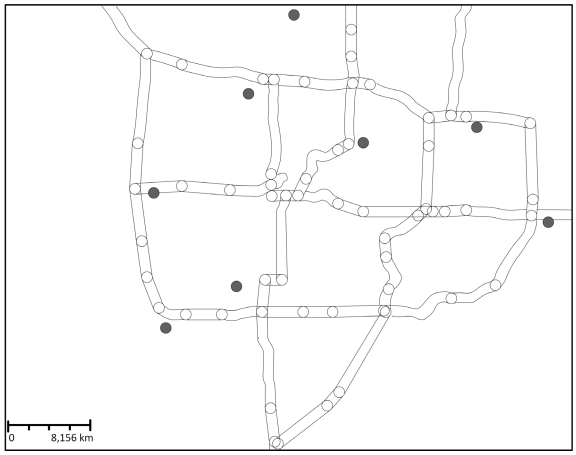


Fig. 8. Visualization of buffers around $Intst$ roads and buffers for shopping centers of type Rl_Ctr (dark gray points).

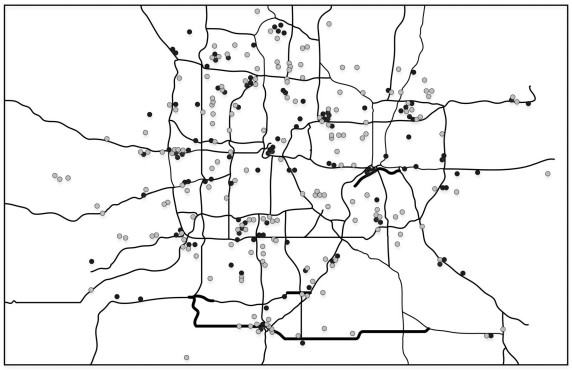


Fig. 9. General view of $County\_Road$ objects (thick line) and Nbh_Ctr (light gray).

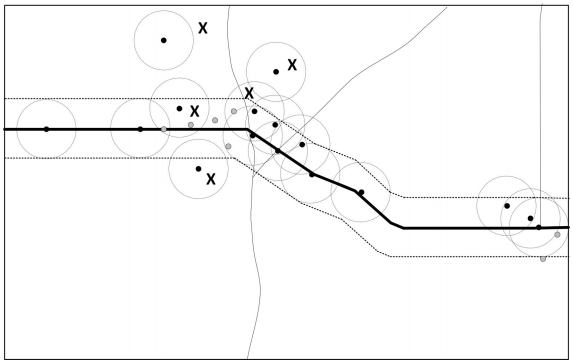


Fig. 10. Close-up on $County\_Road$ (black line) and Nbh_Ctr (light gray) objects with buffers.

Table 4. Juxtaposition of patterns with the highest values of prevalence and the coverage ratio.

Element 1	Element 2	Element 3	Prev (DEOSP)	Ordering (DEOSP)	Ordering (EXCOM)	Cov ratio (EXCOM)	Prop. prev (EXCOM)
$Intst$	US_Hwy		1	1	5	0.000383894	0.693287285
$State\_Hwy$	$Intst$		0.8125	2	1	0.000646948	0.9
$State\_Hwy$	US_Hwy		0.625	3	8	0.000346703	0.658850065
$Cmty\_Ctr$	$Intst$		0.6	4	4	0.000395316	0.703526193
Nbh_Ctr	$Intst$		0.5990099	5	2	0.000565091	0.841137905
Nbh_Ctr	$State\_Hwy$		0.5792079	6	3	0.000504439	0.7947161
Dtn_Ctr	$Intst$		0.5714286	7	22	6.39716E-05	0.283009792
$Intst$	$County\_Rd$		0.5714286	8	15	0.000125551	0.396477266
$Sub-Rl\_Ctr$	$Intst$		0.5555556	9	23	5.61332E-05	0.265104901
Rl_Ctr	$Intst$		0.5555556	10	30	3.39461E-05	0.206159237
$Cmty\_Ctr$	$State\_Hwy$		0.46	11	7	0.00035342	0.66520152
Nbh_Ctr	$Cmty\_Ctr$		0.4455445	12	6	0.000365155	0.676155431
US_Hwy	$County\_Rd$		0.4285714	13	14	0.000127961	0.400264596
$Cmty\_Ctr$	US_Hwy		0.38	14	11	0.00018656	0.483299975
Rl_Ctr	US_Hwy		0.3333333	15	71	4.9043E-06	0.078360351
Dtn_Ctr	$County\_Rd$		0.3333333	16	34	3.02221E-05	0.194522649
Rl_Ctr	$Intst$	US_Hwy	0.3333333	17	NULL	NULL	NULL
Nbh_Ctr	US_Hwy		0.3217822	18	9	0.000274965	0.586741599
$State\_Hwy$	$Intst$	US_Hwy	0.3125	19	41	1.60217E-05	0.14163225
$State\_Hwy$	$Sub-Rl\_Ctr$		0.25	20	29	3.46698E-05	0.208345212
Nbh_Ctr	$Intst$	$State\_Hwy$	0.2277228	21	19	8.59739E-05	0.328088622
Nbh_Ctr	$Intst$	$Cmty\_Ctr$	0.2277228	22	18	8.85542E-05	0.332975621
...	...	...	...	...	...	...	...
$State\_Hwy$	$County\_Rd$		0.1875	24	16	0.000110587	0.37210036
Nbh_Ctr	$State\_Hwy$	$Cmty\_Ctr$	0.1633663	25	21	7.18906E-05	0.30001565
...	...	...	...	...	...	...	...
Nbh_Ctr	$County\_Rd$		0.1188119	35	10	0.00020264	0.503697631
...	...	...	...	...	...	...	...
$Cmty\_Ctr$	$County\_Rd$		0.07	44	20	8.38294E-05	0.323970924

relative to the road), and thus will not be taken into consideration when computing the participation index. In this case, EXCOM considers 11 neighborhoods of objects, while DEOSP at most 8. If we increase, buffer 3 times the difference will be even bigger: 13 to 8 for EXCOM. In DEOSP, to compute neighborhoods, only first order neighborhoods are considered.

A detailed discussion of the differences in patterns of both methods explains the properties numbered 2 and 3 in Section 3.3. EXCOM, due to the use of buffers, does not adjust in any way to an uneven distribution and does not take into consideration objects lying outside of the buffer. DEOSP, on the other hand, considers objects unevenly distant and takes into consideration objects very distant as long as they are located in one cluster with the object of interest.

4.3.2. Rule comparison. The approach to qualitatively compare rules computed using both methods was similar to comparing patterns. In the first step, based on the analysis of executions of algorithms with different parameters, the controlling parameters for the rules computed with DEOSP were determined to be $minPrev = 0.1$ and $minConf = 0.5$. In the case of EXCOM, the minimum coverage ratio reflecting prevalence was determined using the equation computing proportional prevalence to be 3.14014×10^{-5} no restrictions were imposed on the confidence at this point. Both methods were run and the generated rules were sorted in descending order based on the confidence value. Next, all DEOSP rules were assigned corresponding ordering in the sorted results of EXCOM. Proportional prevalence values were computed additionally. Proportional confidence was computed in a similar way as the proportional prevalence computed earlier.

The juxtaposition of the discovered rules is presented in Table 5. The light gray color highlights rules found in DEOSP which were not found in EXCOM; the gray color highlights rules from EXCOM not discovered in DEOSP. The rules highlighted with the light gray color were not discovered by the EXCOM algorithm but they have very low prevalence. After computing proportional minimum prevalence, the resultant rules of the EXCOM structure could be found just below the threshold of

Table 5. Juxtaposition of rules with the largest confidence values for DEOSP and EXCOM.

Antecedent 1	Antecedent 2		Consequent	Prev.	Conf.	Ordering (DEOSP)	Ordering (EXCOM)	Coverage ratio (EXCOM)	Prop. prev. (EXCOM)	Condit. prob. (EXCOM)	Proporc. conf. (EXCOM)
Intst		→	Nbh_Ctr	0.599	1	1	23	0.0005650	0.8411366	0.0631283	0.3196941
Intst		→	Cmty_Ctr	0.6	1	2	30	0.0004037	0.7109859	0.0434814	0.2653227
Intst		→	Stat_Hwy	0.8125	1	3	22	0.0006517	0.9033054	0.0728049	0.3433229
Dtn_Ctr		→	Intst	0.57142	1	4	1	6.397E-05	0.2830100	0.6176681	1
Intst		→	US_Hwy	1	1	5	27	0.0004037	0.7109859	0.0451038	0.2702272
US_Hwy		→	Intst	1	1	6	30	0.0004037	0.7109859	0.0434814	0.2653227
Rl_Ctr	US_Hwy	→	Intst	0.33333	1	7	NULL	NULL	NULL	NULL	NULL
Rl_Ctr	County_Rd	→	Intst	0.11111	1	8	NULL	NULL	NULL	NULL	NULL
Dtn_Ctr	US_Hwy	→	Intst	0.11111	1	9	NULL	NULL	NULL	NULL	NULL
Dtn_Ctr	US_Hwy	→	County_Rd	0.11111	1	10	NULL	NULL	NULL	NULL	NULL
State_Hwy		→	Nbh_Ctr	0.57920	0.9375	11	25	0.0005044	0.7947166	0.0583167	0.3072693
US_Hwy		→	State_Hwy	0.625	0.85714	12	33	0.0003581	0.6696230	0.0385693	0.2498870
Intst		→	Rl_Ctr	0.55555	0.85714	13	46	3.394E-05	0.2061592	0.0037922	0.0783557
State_Hwy		→	Intst	0.8125	0.8125	14	20	0.0006517	0.9033054	0.0753422	0.3492541
Intst	US_Hwy	→	Nbh_Ctr	0.12871	0.77777	15	NULL	NULL	NULL	NULL	NULL
Intst		→	Sub-Rl_Ctr	0.55555	0.71428	16	43	5.613E-05	0.2651048	0.0062708	0.1007594
County_oad		→	Intst	0.57142	0.66666	17	18	0.0001255	0.3964772	0.0894798	0.3806142
State_Hwy	Intst	→	Nbh_Ctr	0.22772	0.66666	18	59	8.597E-05	0.3280882	0.1319204	0.4621452
Cmty_Ctr	State_Hwy	→	Nbh_Ctr	0.16336	0.64	19	53	7.189E-05	0.3000156	0.2034143	0.5738696
State_Hwy		→	US_Hwy	0.625	0.625	20	31	0.0003581	0.6696230	0.0414028	0.2589031
Cmty_Ctr		→	Intst	0.6	0.6	21	5	0.0003953	0.7035258	0.2122609	0.5862156
Nbh_Ctr		→	Intst	0.599	0.599	22	12	0.0005650	0.8411366	0.1525901	0.4970335
Nbh_Ctr		→	State_Hwy	0.5792	0.5792	23	14	0.0005044	0.7947166	0.1362128	0.4696036
Cmty_Ctr	Intst	→	Nbh_Ctr	0.22772	0.57692	24	50	8.855E-05	0.3329756	0.2240087	0.6022195
Intst		→	Dtn_Ctr	0.57142	0.57142	25	42	6.397E-05	0.28301	0.0071465	0.1075647
Intst		→	County_Rd	0.57142	0.57142	26	37	0.0001255	0.3964772	0.0140258	0.1506907
Sub-Rl_Ctr		→	Intst	0.55555	0.55555	27	2	5.613E-05	0.2651048	0.3141996	0.7132228
Rl_Ctr		→	Intst	0.55555	0.55555	28	9	3.394E-05	0.2061592	0.1900099	0.5546391
State_Hwy	US_Hwy	→	Cmty_Ctr	0.15	0.5	29	NULL	NULL	NULL	NULL	NULL
Dtn_Ctr	County_Rd	→	US_Hwy	0.11111	0.5	30	NULL	NULL	NULL	NULL	NULL
...	...	...	...	...	...	...	...	...	...	...	...
Rl_Ctr		→	Nbh_Ctr	NULL	NULL	NULL	3	4.762E-05	0.2441847	0.2665673	0.6569402
Rl_Ctr		→	State Hwy	NULL	NULL	NULL	4	4.628E-05	0.2407300	0.2590783	0.6476464
Rl_Ctr		→	Cmty_Ctr	NULL	NULL	NULL	6	3.755E-05	0.2168460	0.2102198	0.5833903
Cmty_Ctr		→	Nbh_Ctr	NULL	NULL	NULL	7	0.0003651	0.6761554	0.1960663	0.5634092
Sub-Rl_Ctr		→	State_Hwy	NULL	NULL	NULL	8	3.466E-05	0.2083452	0.1940604	0.5605197
Cmty_Ctr		→	State Hwy	NULL	NULL	NULL	10	0.0003534	0.6652012	0.1897650	0.5542816

acceptable prevalence. The usefulness of rules with such low prevalence is low.

To sum up: the most important rules discovered by DEOSP are also found by EXCOM. There also exists a reciprocal relationship.

5. Conclusions

In the paper we studied the properties of spatial data mining methods allowing one to create patterns among extended spatial objects. In particular, we made a detailed comparison of the properties of two event-centric methods called EXCOM and DEOSP. For that purpose we conducted a series of experiments with the use of data sets including real data, and the obtained results allowed us to draw some interesting conclusions concerning the methods under investigation. As far as the patterns are concerned, we found that few patterns discovered with DEOSP did not occur as a result of EXCOM, or were found with much lower prevalence. This was caused by too small a buffer size. We also found a reverse situation, a case where EXCOM returns a pattern with particular prevalence because it operates on buffers, while DEOSP significantly lowers the prevalence. The reason for that is a low number of edges between objects of two types in triangulation (objects of these types are separated by other objects—transitional neighbors). The experiments also let us conclude that the participation index is a better measure of prevalence than the coverage ratio relating the size of the bounding neighborhood of a candidate to the total area of space under consideration. The participation index refers not to all objects, but only to those of each type separately, finally choosing the minimum value. There is no significant discrepancy in confidence and the discovered rules are practically the same, excluding prevalence of the rules, which stems from different methods used to compute these values.

For the purpose of testing we implemented three versions of the EXCOM algorithm and selected the best one for further tests. This implementation, even though the most efficient of the three, was slower than the C++ implementation of DEOSP. The reasons for worse performance of EXCOM in our tests were twofold: (i) EXCOM was implemented in a spatial database environment which generates substantial overhead, and (ii) the buffer operation widely utilized in the approach is quite computationally expensive. We proved that finding the right size of the buffer for a particular dataset requires an additional step of initial data analysis.

To sum up, based on the obtained results, we can state that in the case of data with large differences in distances among objects, a better choice is DEOSP, whereas in the case of data for which it is easy to adjust the buffer size (e.g., for problems of influence of rivers on surrounding soils, the influence of noise or exhaust in the neighborhood of roads on the surroundings), EXCOM may be a first choice as by using it one may anticipate a small but expected set of patterns and rules.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments that helped improve the final version of the paper.

References

Adilmagambetov, A., Zaiane, O.R. and Osornio-Vargas, A. (2013). Discovering co-location patterns in datasets with extended spatial objects, *International Conference on Data Warehousing and Knowledge Discovery, Berlin, Germany*, pp. 84–96.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules, *20th International Conference Very Large Data Bases, VLDB, Santiago de Chile, Chile*, pp. 487–499.

Appice, A., Berardi, M., Ceci, M. and Malerba, D. (2005). Mining and filtering multi-level spatial association rules with ares, *International Symposium on Methodologies for Intelligent Systems, Saratoga Springs, NY, USA*, pp. 342–353.

Barua, S. and Sander, J. (2011). SSCP: Mining statistically significant co-location patterns, *International Symposium on Spatial and Temporal Databases, Minneapolis, MN, USA*, pp. 2–20.

Bembenik, R., Ruszczyk, A. and Protaziuk, G. (2014). Discovering collocation rules and spatial association rules in spatial data with extended objects using Delaunay diagrams, *International Conference on Rough Sets and Intelligent Systems Paradigms, Granada/Madrid, Spain*, pp. 293–300.

Bembenik, R. and Rybiński, H. (2009). FARICS: A method of mining spatial association rules and collocations using clustering and Delaunay diagrams, *Journal of Intelligent Information Systems* **33**(1): 41–64.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). From data mining to knowledge discovery in databases, *AI Magazine* **17**(3): 37.

Karamshuk, D., Noulas, A., Scellato, S., Nicosia, V. and Mascolo, C. (2013). Geo-spotting: Mining online location-based services for optimal retail store placement, *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA*, pp. 793–801.

Kim, S.K., Lee, J.H., Ryu, K.H. and Kim, U. (2014). A framework of spatial co-location pattern mining for ubiquitous GIS, *Multimedia Tools and Applications* **71**(1): 199–218.

Koperski, K. and Han, J. (1995). Discovery of spatial association rules in geographic information databases, *in* M.J. Egenhofer and J.R. Herring (Eds.), *Advances in Spatial Databases*, Springer, Berlin/Heidelberg, pp. 47–66.

Li, D., Wang, S. and Li, D. (2016). *Spatial Data Mining: Theory and Application*, Springer, Berlin/Heidelberg.

Li, J., Zaïane, O.R. and Osornio-Vargas, A. (2014). Discovering statistically significant co-location rules in datasets with extended spatial objects, *International Conference on Data Warehousing and Knowledge Discovery, Munich, Germany*, pp. 124–135.

Lisi, F. A. and Malerba, D. (2004). Inducing multi-level association rules from multiple relations, *Machine Learning* **55**(2): 175–210.

Loglisci, C., Ceci, M. and Malerba, D. (2010). Relational learning of disjunctive patterns in spatial networks, *1st Workshop on Dynamic Networks and Knowledge Discovery, Barcelona, Spain*, pp. 17–28.

Okabe, A., Boots, B., Sugihara, K. and Chiu, S.N. (2009). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, Chichester.

Oracle (2017). *Oracle Spatial Developer's Guide,* `https://docs.oracle.com/cd/B28359_01/appdev.111/b28400/sdo_objgeom.htm#SPATL120`.

PostGIS (2017). Buffer operation in PostGIS, `http://www.postgis.net/docs/ST_Buffer.html`.

Rineau, L. (2017). 2D conforming triangulations and meshes. CGAL user and reference manual, `https://doc.cgal.org/latest/Mesh_2/index.html`.

Shekhar, S. and Huang, Y. (2001). Discovering spatial co-location patterns: A summary of results, *International Symposium on Spatial and Temporal Databases, Redondo Beach, CA, USA*, pp. 236–256.

Shekhar, S. and Xiong, H. (2007). *Encyclopedia of GIS*, Springer Science & Business Media, New York, NY.

Xiong, H., Shekhar, S., Huang, Y., Kumar, V., Ma, X. and Yoo, J. S. (2004). A framework for discovering co-location patterns in data sets with extended spatial objects, *4th SIAM International Conference on Data Mining, Lake Buena Vista, FL, USA*, pp. 78–89.

Yang, X. and Cui, W. (2008). A novel spatial clustering algorithm based on Delaunay triangulation, *International Conference on Earth Observation Data Processing and Analysis, Wuhan, China*, pp. 728530–728530.

Zheng, Y., Capra, L., Wolfson, O. and Yang, H. (2014). Urban computing: Concepts, methodologies, and applications, *ACM Transactions on Intelligent Systems and Technology* **5**(3): 38.

Robert Bembenik received his PhD degree in information science from the Warsaw University of Technology (2007), where he works as an assistant professor. He is the author or a co-author of over 20 scientific papers. His fields of interest include data mining, text mining, spatial data mining and mobile applications.

Wiktor Jóźwicki received his MSc degree in computer science from the Warsaw University of Technology (2013). His interests encompass data mining, NoSQL databases, platforms and technologies processing large volumes of data. He has co-founded the Streetquest startup. Currently he works for well known companies as a senior Java/JVM languages programmer.

Grzegorz Protaziuk received his MSc (2001) and PhD (2006) degrees in information science from the Warsaw University of Technology, where he works as an assistant professor. He is the author or a co-author of over 20 scientific papers. His fields of interest include data and text mining, knowledge discovery, and spatial data mining.

Received: 6 December 2016
Revised: 5 June 2017
Re-revised: 25 July 2017
Accepted: 9 August 2017