A Language and a Visual Interface to Specify Complex Spatial Patterns

Xiaohui Li and Yan Huang

Computer Science and Engineering
University of North Texas
{xl0023, huangyan}@unt.edu

Abstract. The emerging interests in spatial pattern mining lead to the demand for a flexible spatial pattern mining language, on which easy to use and understand visual pattern language could be built. This motivates us to define a pattern mining language called CSPML to allow users to specify complex spatial patterns they are interested in mining from spatial datasets. We describe our proposed pattern mining language in this paper. Unlike general pattern languages proposed in literature, our language is specifically designed for specifying spatial patterns. An interface which allows users to specify the patterns visually is designed. The visual language is based on and goes beyond the visual language proposed in literature in the sense that users use CSPML to retrieve patterns instead of the results of a simple spatial query.

1 Introduction

Mining implicit, potentially useful, and previously unknown patterns from spatial data is becoming increasingly important due to the availability of large spatial database collected by inexpressive location enabled devices such as GPS and RFID. The applications of spatial data mining range from animal movement tracking, environmental monitoring, transportation, to national security [6, 14].

Mining *spatial co-location patterns* [9, 15, 7, 17] is an important spatial data mining task with broad applications. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ be a set of spatial features. A spatial feature categorizes or groups spatial objects that have the same characteristics together. Example spatial features include car accident, traffic jam, Chromium 6 polluted water source, West Nile disease, and deforestation. Consider a number of l spatial datasets $\{SD_1, SD_2, \ldots, SD_l\}$, such that $SD_i, i \in [1, l]$ contains all and only the objects that have the spatial feature f_i, e.g. the spatial objects of West Nile disease. Let $\mathcal{R}$ be a given spatial neighbor relation (e.g. distance less than 1.5 miles). A set of spatial features $X \subseteq \mathcal{F}$ is a co-location if its value $im(X)$ of an interesting measure, which is specified differently by variants of the mining problem, is above a threshold min_im. The problem of finding the complete set of co-location patterns is called the co-location mining problem.

Recently, co-location pattern has been extended to include positive relationships, self-co-location/ self-exclusion relationships, one-to-many relationships,

and multi-feature exclusive relationships [2]. Mining complex spatial patterns from large spatial datasets is an emerging area in data mining [12]. Many spatial features interact with each other through complex spatial relationships, e.g. topological and directional [16], other than metric ones. A complex spatial pattern refers to a subset of spatial features whose spatial objects tend to appear in some spatial configuration specified by some spatial predicate. An example complex spatial pattern may be (car accident, yield sign, service road) with the spatial predicate (yield sign *on* service road AND car accident *close* yield sign). These complex spatial patterns may be summarized by the interactions of their spatial objects. When happenings of this configuration are observed and the pattern is significant enough according to some *significant measure*, we report such pattern.

Most current spatial data mining algorithms follow an almost exhaustive exploration of the problem space and recommends a large set of non-trivial, potentially useful, and previously unknown patterns. This mining process entails costly computational cost. However, most of the time domain experts posses some basic knowledge about the data and this knowledge may be used to guide the mining process. So a spatial data mining language that is both expressive and easy to use will allow domain experts to incorporate their interests and knowledge to specify complex spatial patterns. In this patter, we describe such a language and a visual interface to accept visual query which will be translated into the proposed language.

We make the following two contributions in this paper. First, we provide a spatial pattern language called CSPML to specify complex spatial pattern. The language allows users to specify a large set of complex patterns that they are interested in. Second, we design a visual interface to relieve users from writing CSPML queries.

2 Related Work

Existent data mining query languages are not sufficient for spatial data mining. DBQL (Data Mining Query Language) [8] is a general purpose data mining language and does not allow expression of spatial patterns. SDMOQL (Spatial Data Mining Object Query Language) [11] is a spatial mining language designed for INGENS(Inductive Geographic Information System). It trains the inductive geographic information system by inductive queries first and creates a special user view. SDMOQL focuses more on querying spatial database than retrieving spatial patterns. ATLaS (Aggregate and Table Language and System) [10] is an native extension of SQL language which adds to SQL the ability of defining new User Defined Aggregates and Table Functions. It extends DBMS to support efficiently database-centric data mining, stream computation, and decision support applications. Snoop [3] is a model independent event specification language. It distinguishes events from conditions of event and allows the construction of complex events needed for a large class of applications.

A user friendly visual interface to alleviate users from the burden of writing sophisticated query language, is important for both spatial query language and

spatial pattern mining. Topological relationships among arbitrary spatial objects using 9-intersection model is discussed in [5]. Regions, lines and points and all the possible relations between them were defined [5] using point topology. Spatial-Query-by-Sketch tries to offer an efficient visual interface for retrieving spatial objects satisfying a spatial query in geographic information systems [4]. Twelve positional operators and a set of their specifications in object-oriented geographic database are considered in PQL(Pictorial Query Language) in [13]. The system developed in this context aims at providing a visual interface with the expressive power of a database query language such as SQL. It concerns less about spatial pattern mining where users are more interested in patterns than spatial objects.

3 Complex Spatial Pattern Query Language

In this section, we introduce the definition of the proposed pattern mining language for complex spatial patterns (CSPML). The formal definition of CSPML is of the form:

FIND t
WHERE $P(t)$
WITH *interestingness* θ c

where t is variable to present an ordered (alphabetically by spatial feature name) subset of all the spatial features in $\mathcal{F}$, P is a *formula*, *interestingness* specifies the interestingness measures used for the complex spatial pattern, θ is a comparison operator that includes $<, \leq, >$, and $\geq$, and c is a constant to specify the interestingness measure threshold. The formula $P(t)$ is built up using conjunctive forms, which may be one of the following:

1. $t.size\ \theta\ c'$ where $t.size$ is the cardinality of the spatial feature set t and c' is a constant integer;
2. $Q(t)$ where $Q(t)$ is a constraint a conjunctive form with components from the following format:
   (a) $t[i] \odot t[j], \forall i, j \in integer$
   (b) $t[i].type \in \{geo\_point, geo\_line, geo\_area, geo\_donotcare\}, \forall i \in integer$
   where $\odot$ is a spatial predicate, e.g. *overlap*, and $t[k]$ specifies the kth spatial feature in an ordered set t. A metric spatial predicate $\odot$ has an attribute attached which denotes distance information; There are three types of spatial features in a geography database, namely geo_point, geo_line and geo_area. Every spatial feature in a pattern has one attribute called *type*. If the *type* of a feature is known, user can provide the type. For unknown types, we introduce a new type called *geo_do_not_care*, which means any one of the three types.
3. $s \subseteq t\{\bigwedge Q'(s,t)]\}$, where s is any ordered (again alphabetically) subset of the spatial feature set $\mathcal{F}$ specified by a user; $[Q'(s,t)]$ is an optional constraint consisting one of the following form or their conjunctive combinations:
   (a) $(t-s)[i] \odot (t-s)[j], \forall, i, j \in integer$;
   (b) $(t-s)[i] \odot s[j], \forall, i, j \in integer$;

(c) $s[i] \odot s[j] \forall, i, j \in integer$;
where $\odot$ is a spatial predicate, e.g. overlap, and $t[k]$ specifies the kth spatial feature in an ordered set t

4. $t.pairwise(\odot)$, meaning pairwise spatial features in t satisfying spatial predicate $\odot$

Spatial predicate $\odot$ in the formula $P(t)$ may be metric, topological, and directional [16] relationships. The *interestingness* measure can be anything that is well defined for a complex spatial pattern, such as participation index [9], join cardinality, join selectivity and support [17]. We now describe a few examples using the proposed language to illustrate its expressiveness:

1. Find all patterns with less than 5 spatial features, spatial predicate pairwise distance less than 5 miles, and interestingness measure participation-index with threshold 0.7.
   FIND t
   WHERE t.size $\leq 5 \bigwedge$ t.pairwise(close(5))
   WITH $participation\_index \geq 0.7$

2. Find all patterns which contains 3 features and one feature is Chromium 6 polluted water. The Chromium 6 polluted water contains the other two features and the other two features are close to each other. The interestingness is participation-index and the threshold is 0.7. In this example, the pattern size is three with one as feature Chromium 6 polluted water, which belongs to set s. The other two features, which are the components of set $(t - s)$, are close to each other and inside the Chromium 6 polluted water.
   FIND t
   WHERE t.size $= 3 \bigwedge$ (Chromium 6 polluted water) $\subseteq$ t $\bigwedge$
   (t - s)[1] *containedBy* chromium 6 polluted water $\bigwedge$
   (t - s)[2] *containedBy* chromium 6 polluted water $\bigwedge$
   (t - s)[1] *close_to* (t - s)[2]
   WITH $participation - index \geq 0.7$

3. Find all patterns with three features two of which are forest and river. The other feature is to the north of the forest and close to the river. The interestingness is support and the threshold is 100. The pattern size in this example is three with two known features. So set s contains forest and river. The other feature has relationships with both of the two features through *north of* and *close* respectively.
   FIND t
   WHERE t.size $= 3 \bigwedge$ forest, river) $\subseteq$ t $\bigwedge$
   (t - s)[1] $northofforest \bigwedge$
   (t - s)[1] $close(defaultlength)$ river
   WITH $support \geq 100$

4. Find all the patterns of sizes from 3 to 5 and contain feature volcano. One feature is close to volcano in the pattern. The interestingness is *support* and the threshold is 25. The pattern size is between three to five with volcano in set s. One feature in the set $(t - s)$ must be close to the feature volcano.

FIND t
WHERE t.size $\geq 3 \bigwedge$ t.size $\leq 5 \bigwedge$ (volcano) $\subseteq$ t $\bigwedge$ (t - s)[1] *close_to* volcano
WITH *support* ≥ 25

4 Visual Pattern Expression

There are three kinds of spatial features in spatial databases, namely geo-point, polyline and geo-area [13]. Geo-point is a zero dimensional point to represent events such as location of an accident. Polylines are used to represent spatial objects such as roads and rivers. Geo-area can be used to represent forest stands or lakes. We consider the meaningful relationships among these three features. In all figures in this section, geo-point, polyline and geo-area are represented a dot, a straight line (we do not consider polylines that can turn around), and a rectangle (without holes) respectively. Due to space constraint, we could not describe the prototype system that we implemented based on JUMP [1] architecture and the four examples specified using our visual interface in this paper. Interested readers should refer to the full version of the paper available from the second author's website.

Geo-point Figure 1(a) and Figure 1(b) represent a geo-point *on* and *close to* a polyline respectively. A buffer is created around the polyline to indicate the *close to* relationship. The point may stand for an accident while the polyline may stand for a road. Figure1(a) shows the accident happened on the roads while Figure 1(b) shows that although geo-point is not on the polyline but is still within certain distance. We use the representation in Figure 1(c) to specify that we do not care about the relationship between the geo-point and polyline, i.e. we want all patterns with any spatial predicate, e.g. *on* or *close to*.

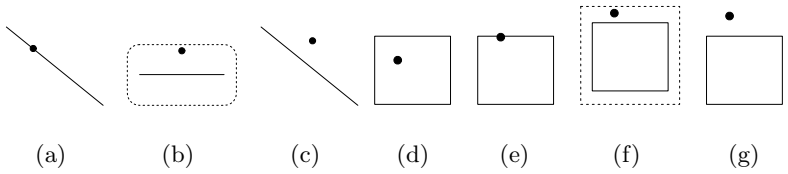

Fig. 1. The Expression of Relationships for Geo-point Objects

Figure 1(d), 1(e) and 1(f) are the three kinds of relationships between a geo-point and a geo-area. They are geo-point *in, on* and *close to* the geo-area respectively. If a user does not care exactly what kind of relationship is between the geo-point and the geo-area, users could use the representation of figure 1(g) to find the pattern whose features may be any one of the three relationships.

Polyline The relationship of a polyline and goe_area is based on the interiors, boundaries and exteriors of the objects [5]. A spatial relationship can be specified by a matrix representing if the interior(A°)/boundary(∂A)/exterior(A^-) of one object A intersects with the interior/boundary/exterior of another object B. Thirty-three possible relationships between two simple polylines were considered in [5]. We only consider straight lines and do not distinguish the boundaries from the interiors of the polyline, only five possible relationships between two simple polyline remain with four of them shown in figure 2 and the symmetric relationship of 2 (d) omitted due to space constraint.

The relationships specified in 2(b) to 2(d) can be further abstracted into a high level relationship called *intersect*, depending if the two polyline have any common points or not as shown in Figure 3(b). Users may define the a intersection relationship by using the specification as in figure 3(a). Otherwise, users may define specify the relationship by that in Figure 3(a). Figure 3(c) means the two polylines do not intersect while Figure 3(c) is used to express any kind of the relationships between two polylines.

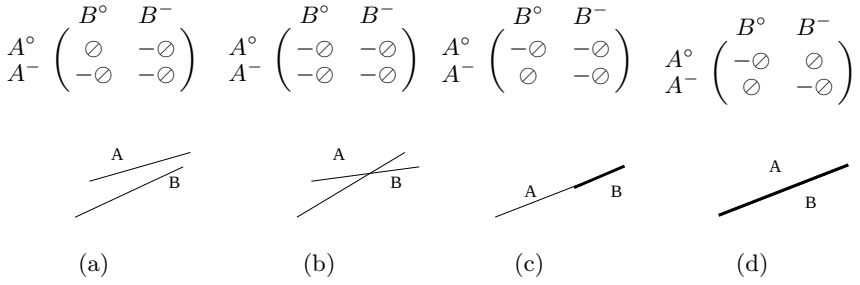

Fig. 2. Four Intersection Model between Two Straight Polylines

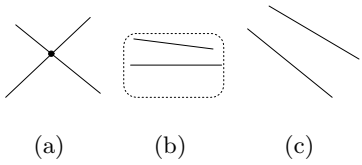

Fig. 3. The Expression of Relationships between Two Polylines

Based on the twenty relationships between a polyline and a geo-area [5], we found that the exteriors of a polyline always intersect with the interiors, boundaries and exteriors of a geo-area. We do not distinguish the boundary from the interior of a polyline, so we only consider the relationships between the interiors of a polyline with a geo-area. Figure 4 shows the matrix representation and the five relationships between a polyline and a geo-area. Figure 4(b) and 4(c) can be combined into a *touch* relationship as show in Figure 5(b).

The polyline could be *in, touch, across,* and *out of* a geo-area which are shown in Figure 5. If a user wants to find all relationships involving a polyline and a geo-area without providing a certain kind of relationship between them, they could just use the specification in figure 5(e).

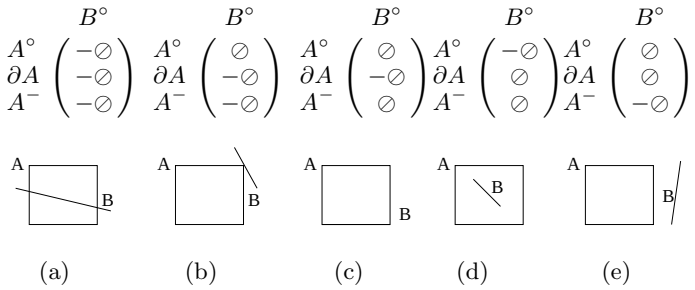

Fig. 4. Three intersection Model between a Polyline and a Geo-area

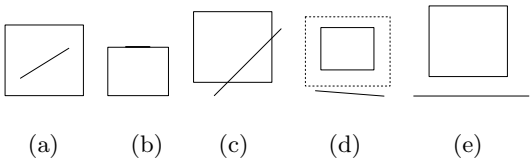

Fig. 5. The Expression of Relationships between a Polyline and a Geo-area

Geo-area We allow the specification of 5 relationships between two geo-areas out of the 8 relationships specified in [5] as shown from Figure 6(a) to Figure 6(d) (the symmetric relationship of Figure 6(a) is omitted). If the users do not care the exact relationship between two features, they could specify the relationship as in figure 6(e).

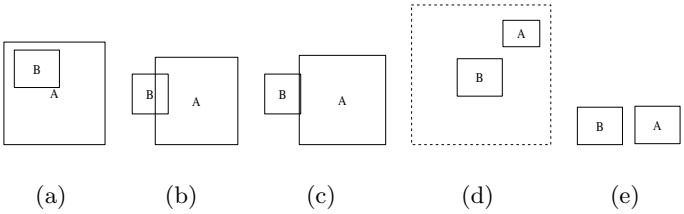

Fig. 6. The Expression of Relationships between Two Geo-areas

5 Conclusion and Future Work

In this paper, we designed and developed a pattern mining language called CSPML for specifying complex spatial patterns. CSPML allows users to choose between exhaustive pattern search and targeted pattern mining constrained by prior knowledge about the datasets. We also proposed a pictorial language which has the equivalent expressive power to the proposed CSPML language. Efficient mining algorithms utilizing as much as the JTS tools of JUMP [1] will be implemented in the future.

References

1. The JUMP project. http://www.jump-project.org/index.php.
2. B. Arunasalam, S. Chawla, P. Sun, and R. Munro. Mining complex relationships in the sdss skyserver spatial database. In *COMPSAC '04*, pages 142–145, Washington, DC, USA, 2004.
3. S. Chakravarthy and D. Mishra. Snoop: An expressive event specification language for active databases. *Data Knowledge Engineering*, 14(1):1–26, 1994.
4. M. J. Egenhofer. Query processing in spatial-query-by-sketch. *Journal of Visual Languages and Computing*, 8(4):403–424, 1997.
5. M. J. Egenhofer and J. Herring. Categorizing binary topological relationships between regions, lines and points in geographic databases, 1991.
6. M. Ester, H.-P. Kriegel, and J. Sander. Algorithms and applications for spatial data mining, 2001.
7. V. Estivill-Castro and A. Murray. Discovering Associations in Spatial Data - An Efficient Medoid Based Approach. In *Proc. of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1998.
8. J. Han, Y. Fu, W. Wang, K. Koperski, and O. Zaiane. Dmql: A data mining query language for relational databases. *SIGMOD'96 Workshop*, 1996.
9. Y. Huang, S. Shekhar, and H. Xiong. Discovering Co-location Patterns from Spatial Datasets:A General Approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(12), 2004. (A summary of results was published at SSTD 2001).
10. C. Luo, H. Thakkar, H. Wang, and C. Zaniolo. A native extension of sql for mining data streams. In *SIGMOD '05*, 2005.
11. D. Malerba, A. Appice, and N. Vacca. Sdmoql: An oql-based data mining query language for map interpretation tasks. In *Proc. of the Workshop on Database Technologies for Data Mining (DTDM'02),in conjunction with the VIII International Conference on Extending Database Technology (EDBT'02)*, 2002.
12. R. Munro, S. Chawla, and P. Sun. Complex spatial relationships. *ICDM*, 00:227, 2003.
13. E. Pourabbas and M. Rafanelli. A pictorial query language for querying geographic databases using positional and olap operators. *SIGMOD Rec.*, 31(2):22–27, 2002.
14. J. Roddick and M. Spiliopoulou. A Bibliography of Temporal, Spatial and Spatio-temporal Data Mining Research. *ACM SIGKDD Explorations*, 1999.
15. S. Shekhar and Y. Huang. Co-location Rules Mining: A Summary of Results. In *Proc. 7th Intl. Symposium on Spatio-temporal Databases*, 2001.
16. A. P. Sistla and C. Yu. Reasoning about qualitative spatial relationships. *J. Autom. Reason.*, 25(4):291–328, 2000.
17. X. Zhang, N. Mamoulis, D. W. Cheung, and Y. Shou. Fast mining of spatial collocations. In *KDD '04*, pages 384–393, 2004.