Discovery of Spatial Association Rules in Geographic Information Databases *

Krzysztof Koperski and Jiawei Han

School of Computing Science
Simon Fraser University
Burnaby, B.C., Canada V5A 1S6
e-mail: {koperski, han}@cs.sfu.ca

Abstract. Spatial data mining, i.e., discovery of interesting, implicit knowledge in spatial databases, is an important task for understanding and use of spatial data- and knowledge-bases. In this paper, an efficient method for mining strong spatial association rules in geographic information databases is proposed and studied. A spatial association rule is a rule indicating certain association relationship among a set of spatial and possibly some nonspatial predicates. A strong rule indicates that the patterns in the rule have relatively frequent occurrences in the database and strong implication relationships. Several optimization techniques are explored, including a two-step spatial computation technique (approximate computation on large sets, and refined computations on small promising patterns), shared processing in the derivation of large predicates at multiple concept levels, etc. Our analysis shows that interesting association rules can be discovered efficiently in large spatial databases.

1 Introduction

With wide applications of remote sensing technology and automatic data collection tools, tremendous amounts of spatial and nonspatial data have been collected and stored in large spatial databases. Traditional data organization and retrieval tools can only handle the storage and retrieval of explicitly stored data. The extraction and comprehension of the knowledge implied by the huge amount of spatial data, though highly desirable, pose great challenges to currently available spatial database technologies.

This situation demands new technologies for knowledge discovery in large spatial databases, or spatial data mining, that is, *extraction of implicit knowledge, spatial relations, or other patterns not explicitly stored in spatial databases.*

Recently, there have been a lot of research activities on knowledge discovery in large databases (data mining) [9, 16]. These studies led to a set of interesting techniques developed, including mining strong association and dependency

* This research was supported in part by the research grant NSERC-OGP003723 from the Natural Sciences and Engineering Research Council of Canada and an NCE/IRIS research grant from the Networks of Centres of Excellence of Canada.

rules [1, 2], attribute-oriented induction for mining characteristic and discriminant rules [12], etc. Such studies set a foundation and provide some interesting methods for the exploration of highly promising spatial data mining techniques.

Spatial data mining can be categorized based on the kinds of rules to be discovered in spatial databases. A spatial characteristic rule is a *general description of a set of spatial-related data.* For example, the description of the general weather patterns in a set of geographic regions is a spatial characteristic rule. A spatial discriminant rule is the *general description of the contrasting or discriminating features of a class of spatial-related data from other class(es).* For example, the comparison of the weather patterns in two geographic regions is a spatial discriminant rule. A spatial association rule is a *rule which describes the implication of one or a set of features by another set of features in spatial databases.* For example, a rule like "*most big cities in Canada are close to the Canada-U.S. border*" is a spatial association rule.

There have been some interesting studies related to the mining of spatial characteristic rules and spatial discriminant rules [14, 15]. However, there is lack of studies on mining spatial association rules. In this paper, we study the extension of the techniques for mining association rules in transaction-based databases to mining spatial association rules.

A spatial association rule is a rule of the form "$X \rightarrow Y$", where X and Y are sets of predicates and some of which are spatial ones. In a large database many association relationships may exist but some may occur rarely or may not hold in most cases. To focus our study to the patterns which are relatively strong, i.e., which occur frequently and hold in most cases, the concepts of *minimum support* and *minimum confidence* are introduced [1, 2]. Informally, the support of a pattern A in a set of spatial objects S is the probability that a member of S satisfies pattern A; and the confidence of $A \rightarrow B$ is the probability that pattern B occurs if pattern A occurs. A user or an expert may specify thresholds to confine the rules to be discovered to be *strong* ones.

For example, one may find that 92% of cities within British Columbia (bc) and adjacent to water are close to U.S.A., as shown in (1), which associates predicates *is_a*, *within*, and *adjacent_to* with spatial predicate *close_to.*

$$is\_a(X, city) \wedge within(X, bc) \wedge adjacent\_to(X, water) \rightarrow close\_to(X, us). \ (92\%) \quad (1)$$

Although such rules are usually not 100% true, they carry some nontrivial knowledge about spatial associations, and thus it is interesting to "mine" (i.e., "discover") them from large spatial databases. The discovered rules will be useful in geography, environmental studies, biology, engineering and other fields.

In this paper, efficient methods for mining spatial association rules are studied, with a top-down, progressive deepening search technique proposed. The technique firstly searches at a high concept level for large (i.e., frequently occurring) patterns and strong implication relationships among the large patterns at a coarse resolution scale. Then only for those large patterns, it deepens the search to lower concept levels (i.e., their lower level descendants). Such a deepening search process continues until no large patterns can be found. An important

optimization technique is that the search for large patterns at high concept levels may apply efficient spatial computation algorithms at a coarse resolution scale (such as generalized close_to (*g_close_to*), using approximate spatial computation algorithms, such as R-trees or plane-sweep techniques operating on minimum bounding rectangles (MBRs). Only the candidate spatial predicates, which are worth detailed examination, will be computed by refined spatial techniques(giving detailed predicates such as intersect, contain, etc.). Such multiple-level approach saves much computations because it is very expensive to perform detailed spatial computation for all the possible spatial association relationships.

In Sect. 2 of our paper, existing spatial data mining methods are surveyed. In Sect. 3, the concept of spatial association rules and its data mining methods are outlined. In Sect. 4, an algorithm for the discovery of spatial association rules is presented. In Sect. 5 we discuss the advantages of the algorithm and its possible extensions. The study is summarized in Sect. 6.

2 Previous Work Related to Spatial Data Mining

In this section, previous studies related to spatial data mining are overviewed, which provides a short survey of the topic and associates the previous work with our study.

2.1 Statistical Analysis

Until now statistical spatial analysis has been one of the most common techniques for analyzing spatial data [10]. Statistical methods handle well numerical data, contain a large number of algorithms, have a strong possibility of getting models of spatial phenomena, and allow optimizations. However, statistical analysis usually requires the assumptions regarding to statistical independence of spatially distributed data. Such assumptions are often unrealistic due to the influence of neighboring regions. To deal with such problems, spatial models can include trend surface or dummy variables. If data in one region are influenced by features of neighboring regions, the analyst may fit a regression model with a spatial lagged forms of the dependent variables. Statistical analysis also deals poorly with symbolic data like names.

$$expensive(condo) \leftrightarrow inside(condo, downtown) \wedge area(condo, large). \quad (2)$$

Nonlinear rules in the form of (2) cannot be described using standard methods in statistical spatial analysis. Statistical approach requires a lot of domain and statistical knowledge. Thus, it should be performed by domain experts with the experience in statistics. Another problem related to statistical spatial analysis is expensive computation of the results.

2.2 Generalization-based Spatial Data Mining

One major approach in spatial data mining is to apply generalization techniques to spatial and nonspatial data to generalize detailed spatial data to certain high level and study the general characteristics and data distributions at this level.

An attribute-oriented induction method has been proposed in [14]. It generalizes data to high level concepts and describes general relationships between spatial and nonspatial data. Two algorithms were proposed in the study: (1) *nonspatial-dominant generalization*, and (2) *spatial-dominant generalization.*

The nonspatial-dominant generalization algorithm first performs attribute-oriented generalization on task-relevant nonspatial data describing the properties of spatial objects. In this step, numerical data can be generalized to ranges or descriptive high level concepts (e.g., $-9°C$ to a range value "$-10\_to\_0°C$" or *cold*), and symbolic values to higher level concepts (e.g., *potatoes* and *beets* to *vegetables*). By doing so, low level distinctive values may be generalized to identical high level values, and such high-level identical values among different tuples can be merged together with their spatial pointers clustered into one slot in the spatial attribute. Finally, the map consists of a small number of regions with high level descriptions.

The spatial-dominant generalization first performs generalization on query-related spatial data. Data are generalized using spatial data hierarchies (such as geographic or administrative regions) provided by users/experts or hierarchical data structures (such as quad-trees [19] or R-trees [11]). The generalized spatial entities (such as the merged regions) cluster the related nonspatial data together. After generalization of non-spatial data, every region can be described at a high concept level by one or a set of predicates.

Spatial hierarchies are not always given *a priori*. It is often necessary to describe spatial behavior of similar objects or to determine characteristic features of distinct clusters. In [15], the attribute-oriented induction method was combined with some efficient spatial clustering algorithms, which can still be classified into *spatial-dominant* vs. *nonspatial-dominant* methods. The *spatial-dominant* method classifies task-relevant spatial objects (such as points) into clusters using an efficient clustering algorithm and then perform an attribute-oriented induction for each cluster to extract rules describing general properties of a cluster. The *nonspatial-dominant* method first generalizes nonspatial attributes of query-related objects to high concept levels and then cluster the spatial objects with the same nonspatial descriptions. Then one may find that "*expensive single houses in Vancouver area are clustered along the beach and around two city parks*".

2.3 Other Relevant Studies

Also knowledge mining in image databases, which can be treated as a special type of spatial databases, has been studied recently. Method for the classification of sky objects and another method for recogntion of volcanos on the surface of Venus are described in [8], where classification trees were used to make final decisions.

Sky objects were classified as stars or galaxies. In the first step of the algorithm, basic attributes describing each object were extracted. Attributes like area, sky brightness, positions of peak brightness, and intensity image moments,

etc. were produced. The training set was classified by astronomers, and attributes mentioned above were used to construct the decision tree.

In the study of volcanos attributes recognized by humans like diameters and central peaks are not sufficient for the classification. Thus, eigenvalues of matrices representing images of possible volcanos were used as attributes for the classification algorithm.

The studies on data mining in relational databases [1, 2, 12, 13, 16] are closely related to spatial data mining. In particular, the previous studies on mining association rules [1, 2, 13] are closely related to this study.

An *association rule* is a general form of dependency rule and is defined on transaction-based databases [1]. It is in the form of "$W \rightarrow B$ (c%)", explained as, "*if a pattern W appears in a transaction, there is c% possibility (confidence) that the pattern B holds in the same transaction*", where W and B are a set of attribute values. Moreover, to ensure that such rules are interesting enough to cover frequently encountered patterns in a database, the concept of the *support* of a rule "$W \rightarrow B$" is introduced, which is defined as the ratio that the patterns of W and B occurring together in the transactions vs. the total number of transactions in the database. For example, in a shopping transaction database one may find a rule like "*butter* $\rightarrow$ *bread* (90%)", which means that *90% of customers who buy butter also purchase bread.* Efficient algorithms for the discovery of such kind of rules in transaction-based databases have been studied [1, 2].

3 Spatial Association Rules

Generalization-based spatial data mining methods [14, 15] discover spatial and nonspatial relationships at a general concept level, where spatial objects are expressed as merged spatial regions [14] or clustered spatial points [15]. However, these methods cannot discover rules reflecting structure of spatial objects and spatial/spatial or spatial/nonspatial relationships which contain spatial predicates, such as *adjacent_to, near_by, inside, close_to, intersecting*, etc.

As a complementary, spatial association rules represents object/predicate relationships containing spatial predicates. For example, the following rules are spatial association rules.

- Nonspatial consequent with spatial antecedent(s).

$$is\_a(x, house) \wedge close\_to(x, beach) \rightarrow is\_expensive(x). \quad (90\%)$$

- Spatial consequent with non-spatial/spatial antecedent(s).

$$is\_a(x, gas\_station) \rightarrow close\_to(x, highway). \quad (75\%)$$

Various kinds of spatial predicates can be involved in spatial association rules. They may represent topological relationships [6] between spatial objects, such as *disjoint, intersects, inside/outside, adjacent_to, covers/covered_by, equal*, etc. They may also represent spatial orientation or ordering, such as *left, right, north, east*, etc., or contain some distance information, such as *close_to, far_away*, etc.

For systematic study the mining of spatial association rules, we first introduce some preliminary concepts.

Definition 1. A **spatial association rule** is a rule in the form of

$$P_1 \wedge \ldots \wedge P_m \rightarrow Q_1 \wedge \ldots \wedge Q_n. \quad (c\%) \tag{3}$$

where at least one of the predicates $P_1, \ldots, P_m, Q_1, \ldots, Q_n$ is a spatial predicate, and c% is the *confidence* of the rule which indicates that c% of objects satisfying the antecedent of the rule will also satisfy the consequent of the rule. □

Following this definition, a large number of spatial association rules can be derived from a large spatial database. However, most people will be only interested in the patterns which occur relatively frequently (i.e., with *large supports*) and the rules which have strong implications (i.e., with *high confidence*). The rules with large supports and high confidence are *strong rules.*

Definition 2. The **support** of a conjunction of predicates, $P = P_1 \wedge \ldots \wedge P_k$, in a set S, denoted as $\sigma(P/S)$, is the number of objects in S which satisfy P versus the cardinality (i.e., the total number of objects) of S. The **confidence** of a rule $P \rightarrow Q$ in S, $\varphi(P \rightarrow Q/S)$, is the ratio of $\sigma(P \wedge Q/S)$ versus $\sigma(P/S)$, i.e., the possibility that Q is satisfied by a member of S when P is satisfied by the same member of S. A single predicate is called **1-predicate**. A conjunction of k single predicates is called a **k-predicate**. □

Since most people are interested in rules with large supports and high confidence, two kinds of thresholds: *minimum support* and *minimum confidence*, can be introduced. Moreover, since many predicates and concepts may have strong association relationships at a relatively high concept level, the thresholds should be defined at different concept levels. For example, it is difficult to find regular association patterns between a *particular house* and a *particular beach*, however, there may be strong associations between many *expensive houses* and *luxurious beaches.* Therefore, it is expected that many spatial association rules are expressed at a relatively high concept level.

Definition 3. A set of predicates P is **large** in set S at level k if the support of P is no less than its minimum support threshold σ'_k for level k, and all ancestors of P from the concept hierarchy are large at their corresponding levels. The confidence of a rule "$P \rightarrow Q/S$" is **high** at level k if its confidence is no less than its corresponding minimum confidence threshold φ'_k. □

Definition 4. A rule "$P \rightarrow Q/S$" is **strong** if predicate "$P \wedge Q$" is *large* in set S and the *confidence* of "$P \rightarrow Q/S$" is high. □

Based on these definitions, an example is presented for the explanation of the process of mining strong spatial association rules in large databases. To facilitate the specification of the primitives for spatial data mining, an SQL-like spatial data mining query interface, which is designed based on a spatial SQL proposed in [7], has been specified for an experimental spatial data mining system prototype, GeoMiner, which is currently under implementation and experimentation.

Example 1. Let the spatial database to be studied adopt an extended-relational data model and a SAND (spatial-and-nonspatial database) architecture [3]. That is, it consists of a set of spatial objects and a relational database describing nonspatial properties of these objects.

Our study of spatial association relationships is confined to British Columbia, a province in Canada, whose map is presented in Fig. 1, with the following database relations for organizing and representing spatial objects.

1. *town*(*name*, *type*, *population*, *geo*, ...).
2. *road*(*name*, *type*, *geo*, ...).
3. *water*(*name*, *type*, *geo*, ...).
4. *mine*(*name*, *type*, *geo*, ...).
5. *boundary*(*name*, *type*, *admin_region_1*, *admin_region_2*, *geo*, ...).

Notice that in the above relational schemata, the attribute "*geo*" represents a spatial object (a point, line, area, etc.) whose spatial pointer is stored in a tuple of the relation and points to a geographic map. The attribute "*type*" of a relation is used to categorize the types of spatial objects in the relation. For example, the types for *road* could be {*national highway, local highway, street, back_lane*}, and the types for *water* could be {*ocean, sea, inlets, lakes, rivers, bay, creeks*}. The *boundary* relation specifies the boundary between two administrative regions, such as B.C. and U.S.A. (or Alberta). The omitted fields may contain other pieces of information, such as the area of a lake and the flow of a river.

Suppose a user is interested in finding within the map of British Columbia the strong spatial association relationships between large towns and other "near_by" objects including mines, country boundary, water (sea, lake, or river) and major highways. The GeoMiner query is presented below.

```
discover spatial association rules
inside British_Columbia
from road R, water W, mines M, boundary B
in relevance to town T
where g_close_to(T.geo, X.geo) and X in {R, W, M, B}
      and T.type = ''large'' and R.type in {divided_highway}
      and W.type in {sea, ocean, large_lake, large_river}
      and B.admin_region_1 in ''B.C.''
      and B.admin_region_2 in ''U.S.A.''
```

Notice that in the query, a relational variable `X` is used to represent one of a set of four variables {`R, W, M, B`}, a predicate `close_to(A, B)` says that a spatial objects A and B are close one to another, and *g_close_to* is a predefined generalized predicate which covers a set of spatial predicates: *intersect, adjacent_to, contains, close_to.*

Moreover, "close_to" is a condition-dependent predicate and is defined by a set of knowledge rules. For example, a rule in (4) states if X is a town and Y is a country, then X is close to Y if their distance is within 80 kms.

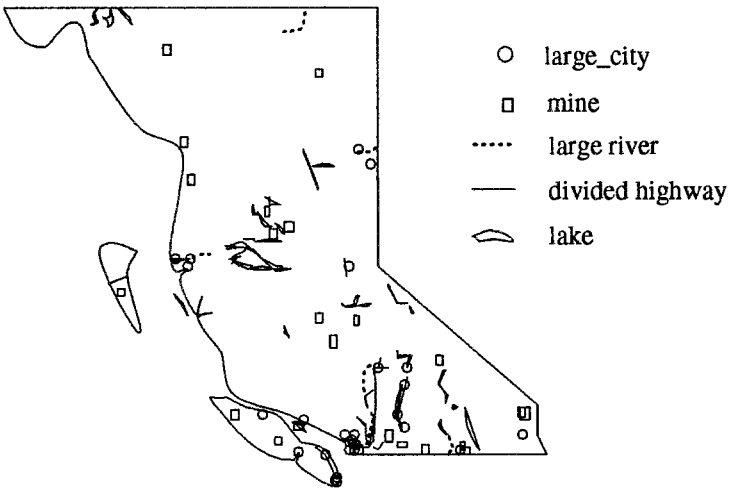

Fig. 1. The map of BC.

$$close\_to(X,Y) \leftarrow is\_a(X,town) \wedge is\_a(Y,country) \wedge dist(X,Y,d) \wedge d < 80\ km. \quad (4)$$

$$close\_to(X,Y) \leftarrow is\_a(X,town) \wedge is\_a(Y,road) \wedge dist(X,Y,d) \wedge d < 5\ km. \quad (5)$$

However, "close_to" between a town and a road will be defined by a smaller distance such as (5).

Furthermore, we assume in the B.C. map, admin_region_1 always contains a region in B.C., and thus "U.S.A." or its states must be in "B.admin_region_2". Since there is no constraint on the relation "mine", it essentially means, "M.type **in** ANY", which is thus omitted in the query.

To facilitate mining multiple-level association rules and efficient processing, concept hierarchies are provided for both data and spatial predicates.

A set of hierarchies for data relations are defined as follows.

- A concept hierarchy for *towns*:
  (town (large_town (big_city, medium_sized_city), small_town (...) ...) ...).
- A concept hierarchy for *water*:

  (water (sea (strait (Georgia_Strait, ...), Inlet (...), ...),
  river (large_river (Fraser_River, ...), ...),
  lake (large_lake (Okanagan_Lake, ...), ...), ...), ...)

- A concept hierarchy for *road*:

  (road (national_highway (route1, ...),
  provincial_highway (highway_7, ...),
  city_drive (Hasting St., Kingsway, ...),
  city_street (E_1st Ave., ...), ...), ...)

Spatial predicates (topological relations) should also be arranged into a hierarchy for computation of approximate spatial relations (like "*g_close_to*" in Fig.

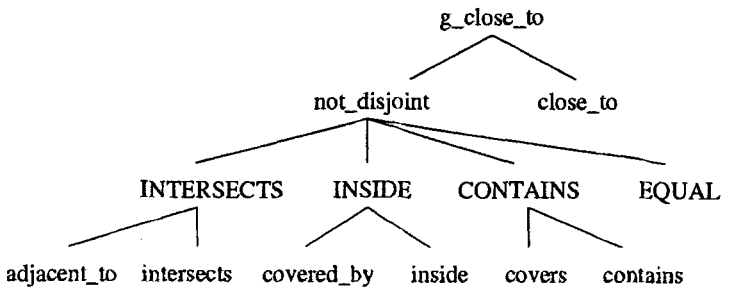

Fig. 2. The hierarchy of topological relations.

2) using efficient algorithms with coarse resolution at a high concept level and refine the computation when it is confined to a set of more focused candidate objects. □

4 A Method for Mining Spatial Association Rules

4.1 An Example of Mining Spatial Association Rules

Example 2. We examine how the data mining query posed in Example 1 is processed, which illustrates the method for mining spatial association rules.

Firstly, the set of relevant data is retrieved by execution of the data retrieval methods [3] of the data mining query, which extracts the following data sets whose spatial portion is inside B.C.: (1) towns: only large towns; (2) roads: only divided highways[2]; (3) water: only seas, oceans, large lakes and large rivers; (4) mines: any mines; and (5) boundary: only the boundary of B.C., and U.S.A.

Secondly, the "generalized close_to" (*g_close_to*) relationship between (large) towns and the other four classes of entities is computed at a relatively coarse resolution level using a less expensive spatial algorithm such as the MBR data structure and a plane sweeping algorithm [18], or R*-trees and other approximations [5]. The derived spatial predicates are collected in a "g_close_to" table (Table 1), which follows an extended relational model: each slot of the table may contain a set of entries. The support of each entry is then computed and those whose support is below the minimum support threshold, such as the column "mine", are removed from the table.

Notice that from the computed *g_close_to* relation, interesting large item sets can be discovered at different concept levels and the spatial association rules can be presented accordingly. For example, the following two spatial association rules can be discovered from this relation.

$$is\_a(X, large\_town) \rightarrow g\_close\_to(X, water). \quad (80\%)$$

$$is\_a(X, large\_town) \wedge g\_close\_to(X, sea) \rightarrow g\_close\_to(X, us\_boundary). (92\%)$$

[2] Not all the segments of national and provincial highways in Canada are divided ones, our computation only counts the divided ones. Also, "provincial divided highway" is abbreviated to "provincial highway" in later presentations.

Town	Water	Road	Boundary	Mine
Victoria	Juan_de_Fuca_Strait	highway_1, highway_17	US	
Saanich	Juan_de_Fuca_Strait	highway_1, highway_17	US	
Prince_George		highway_97		
Pentincton	Okanagan_Lake	highway_97	US	Alalla
...	...	...	...	...

Table 1. The computed "g_close_to" relation.

The detailed computation process is not presented here since it is similar to mining association rules for exact spatial relationships to be presented below.

Since many people may not be satisfied with approximate spatial relationships, such as *g_close_to*, more detailed spatial computation often needs to be performed to find the refined (or precise) spatial relationships in the spatial predicate hierarchy. Thus we have the following steps.

Refined computation is performed on the large predicate sets, i.e., those retained in the *g_close_to* table. Each *g_close_to* predicate is replaced by one or a set of concrete predicate(s) such as *intersect, adjacent_to, close_to, inside*, etc. Such a process results in Table 2.

Town	Water	Road	Boundary
Victoria	⟨adjacent_to, J.Fuca_Strait⟩	⟨intersects, highway_1⟩, ⟨intersects, highway_17⟩	⟨close_to, US⟩
Saanich	⟨adjacent_to, J.Fuca_Strait⟩	⟨intersects, highway_1⟩, ⟨close_to, highway_17⟩	⟨close_to, US⟩
Prince_George		⟨intersects, highway_97⟩	
Pentincton	⟨adjacent_to, Okanagan_Lake⟩	⟨intersects, highway_97⟩	⟨close_to, US⟩
...	...	...	...

Table 2. Detailed spatial relationships for large sets.

Table 2 forms a base for the computation of detailed spatial relationships at multiple concept levels. The level-by-level detailed computation of large predicates and the corresponding association rules is presented as follows.

The computation starts at the top-most concept level and computes large predicates at this level. For example, for each row of Table 2 (i.e., each large town), if the *water* attribute is nonempty, the count of water is incremented by one. Such a count accumulation forms 1-predicate rows (with $k = 1$) of Table 3 where the support count registered. If the (support) count of a row is smaller than the minimum support threshold, the row is removed from the table. For example, the minimum support is set to 50% at level 1, a row whose count is less than 20, if any, is removed from the table. The 2-predicate rows (i.e.,

$k = 2$) are formed by the pair-wise combination of the large 1-predicates, with their count accumulated (by checking against Table 2). The rows with the count smaller than the minimum support will be removed. Similarly, the 3-predicates are computed. Thus, the computation of large k-predicates results in Table 3.

k	large k-predicate set	count
1	⟨adjacent_to, water⟩	32
1	⟨intersects, highway⟩	29
1	⟨close_to, highway⟩	29
1	⟨close_to, us_boundary⟩	28
2	⟨adjacent_to, water⟩, ⟨intersects, highway⟩	25
2	⟨adjacent_to, water⟩, ⟨close_to, us_boundary⟩	23
2	⟨close_to, us_boundary⟩, ⟨intersects, highway⟩	26
3	⟨adjacent_to, water⟩, ⟨close_to, us_boundary⟩, ⟨intersects, highway⟩	22

Table 3. Large k-predicate sets at the top concept level (for 40 large towns in B.C.).

Spatial association rules can be extracted directly from Table 3. For example, since ⟨*intersects, highway*⟩ has a support count of 29, and ⟨*adjacent_to, water*⟩, ⟨*intersects, highway*⟩ has a support count of 25, and $25/29 \doteq 86\%$, we have the association rule (6).

$$is\_a(X, large\_town) \wedge intersects(X, highway) \rightarrow adjacent\_to(X, water). \ (86\%) (6)$$

Notice that a predicate "$is\_a(X, large\_town)$" is added in the antecedent of the rule since the rule is related only to large_town.

Similarly, one may derive another rule (7). However, if the minimum confidence threshold were set to 75%, this rule (with only 72% confidence) would have been removed from the list of the association rules to be generated.

$$is\_a(X, large\_town) \wedge adjacent\_to(X, water) \rightarrow close\_to(X, us\_boundary).(72\%) (7)$$

After mining rules at the highest level of the concept hierarchy, large k-predicates can be computed in the same way at the lower concept levels, which results in Tables 4 and 5. Notice that at the lower levels, usually the minimum support and possibly the minimum confidence may need to be reduced in order to derive enough interesting rules. For example, the minimum support of level 2 is set to 25% and thus the row with support count of 10 is included in Table 4; whereas the minimum support of level 3 is set to 15% and thus the row with support count of 7 is included in Table 5.

Similarly, spatial association rules can be derived directly from the large k-predicate set tables at levels 2 and 3. For example, rule (8) is found at level 2, and rule (9) is found at level 3.

$$is\_a(X, large\_town) \rightarrow adjacent\_to(X, sea) \ (52.5\%) \ (8)$$

$$is\_a(X, large\_town) \wedge adjacent\_to(X, georgia\_strait) \rightarrow close\_to(X, us).(78\%) \ (9)$$

k	large k-predicate set	count
1	⟨adjacent_to, sea⟩	21
1	⟨adjacent_to, large_river⟩	11
1	⟨close_to, us_boundary⟩	28
1	⟨intersects, provincial highway⟩	21
1	⟨close_to, provincial highway⟩	24
2	⟨adjacent_to, sea⟩, ⟨close_to, us_boundary⟩	15
2	⟨close_to, us_boundary⟩, ⟨intersects, provincial highway⟩	19
2	⟨adjacent_to, sea⟩, ⟨close_to, provincial highway⟩	11
2	⟨close_to, us_boundary⟩, ⟨close_to, provincial highway⟩	22
3	⟨adjacent_to, sea⟩, ⟨close_to, us_boundary⟩, ⟨close_to, provincial highway⟩	10

Table 4. Large k-predicate sets at the second level (for 40 large towns in B.C.).

k	large k-predicate set	count
1	⟨adjacent_to, georgia strait⟩	9
1	⟨adjacent_to, fraser_river⟩	10
1	⟨close_to, us_boundary⟩	28
2	⟨adjacent_to, georgia_strait⟩, ⟨close_to, us_boundary⟩	7

Table 5. Large k-predicate sets at the third level (for 40 large towns in B.C.).

Notice that only the descendants of the large 1-predicates will be examined at a lower concept level. For example, the number of large towns adjacent to a lake is small and thus ⟨*adjacent_to, lake*⟩ is not represented in Table 4. Then the predicates like ⟨*adjacent_to, okanagan_lake*⟩ will not be even considered at the third level. The mining process stops at the lowest level of the hierarchies or when an empty large 1-predicate set is derived.

As an alternative of the problem, large_towns may also be further partitioned into *big_cities* (such as towns with a population larger than 50,000 people), *other_large_towns*, etc. and rules like rule (10) can be derived by a similar mining process.

$$is\_a(X, big\_city) \wedge adjacent\_to(X, sea) \rightarrow close\_to(X, us\_boundary).(100\%) \quad (10)$$

4.2 An Algorithm for Mining Spatial Association Rules

The above rule mining process can be summarized in the following algorithm.

Algorithm 4.1 Mining the spatial association rules defined by Definition 1 in a large spatial database.

Input: The input consists of a spatial database, a mining query, and a set of thresholds as follows.

1. A database, which consists of three parts: (1) a spatial database, SDB, containing a set of spatial objects, (2) a relational database, RDB, describing nonspatial properties of spatial objects, and (3) a set of concept hierarchies,
2. a query, which consist of: (1) a reference class S, (2) a set of task-relevant classes for spatial objects $C_1, ..., C_n$, and (3) a set of task-relevant spatial relations, and
3. two thresholds: minimum support ($minsup[l]$) and minimum confidence ($minconf[l]$) for each level l of description.

Output: Strong multiple-level spatial association rules for the relevant sets of objects and relations.

Method: Mining spatial association rules proceeds as follows.

Step 1: $Task\_relevant\_DB$:= extract_task_relevant_objects(SDB, RDB);
Step 2: $Coarse\_predicate\_DB$:=
coarse_spatial_computation($Task\_relevant\_DB$);
Step 3: $Large\_Coarse\_predicate\_DB$:=
filtering_with_minimum_support($Coarse\_predicate\_DB$);
Step 4: $Fine\_predicate\_DB$:=
refined_spatial_computation($Large\_Coarse\_predicate\_DB$);
Step 5: Find_large_predicates_and_mine_rules($Fine\_predicate\_DB$);

Explanation of the detailed steps of the algorithm.

Step 1 is accomplished by the execution of a spatial query. All the task-relevant objects are collected into one database: $Task\_relevant\_DB$.

Step 2 is accomplished by execution of some efficient spatial algorithms at a coarse resolution level. For example, R-trees [4] or fast MBR technique and plane-sweep algorithm [18] can be applied to extract the objects which are approximately close to each other, corresponding to computing g_close_to for the $Task\_relevant\_DB$. The efficiency of the method is reasoned in the next subsection. Predicates describing spatial relations between objects are stored in an extended relational database, called $Coarse\_predicate\_DB$, which allows an attribute value to be either a single value or a set of values (i.e., in non-first-normal form).

Step 3 computes the support for each predicate in $Coarse\_predicate\_DB$, (and registers them in a predicate-support table), and filters out those entries whose support is below the minimum support threshold at the top level, i.e., $minsup[1]$. This filtering process results in a database which contains all large 1-predicates, which is called $Large\_Coarse\_predicate\_DB$. Notice that spatial association rules can also be generated at this resolution level, if desired. Since this process is similar to the process of Step 5, the detailed processing of Step 3 is not presented here.

Step 4 is accomplished by execution of some efficient spatial computation algorithms [5] at a fine resolution level on $Large\_Coarse\_predicate\_DB$ obtained in Step 3. Notice that although such computation is performed for the interesting portion of the spatial database, the computation is only on those pairs which have passed the corresponding spatial testing at a coarse resolution level. Thus, the

number of object pairs which need to be computed at this level is substantially smaller that the number of pairs computed at a coarse level. Moreover, as an optimization technique, one can use the support count of an approximate predicate in *Large_Coarse_predicate_DB* to predict whether there is still hope for a predicate at a fine level to pass the minimum support threshold. For example, if the current support for predicate P plus the remaining number of support for its corresponding predicate *P_coarse* is less than the minimum support threshold, no further test of P is necessary in the remaining processing.

Step 5 computes the large k-predicates for all the k's and generates the strong association rules at multiple concept levels. This step is essential for mining multiple-level association rules and is thus examined in detail.

This step is outlined as follows. First, obtain large k-predicates (for all the k's) at a top concept level. Second, for the large 1-predicates at level 1, get their corresponding large 1-predicates at level 2, and then get all large k-predicates at this level. This process repeats until an empty large 1-predicate set is returned or bottom level in the hierarchy was explored. A detailed study of such a progressive deepening process for mining multiple-level association rules in a transaction-based (but not spatial) database is presented in [13].

At each level, the computation of large k-predicates for all k's proceeds from computing large-1 predicates, then large-2 predicates (using the pair-wise combination of large 1-predicates as the candidate set), large-3 predicates (using the combinations of large 2-predicates as the candidate set), and so on, until an empty candidate set or an empty computed k-predicate set is obtained. Such a process of computing large k-predicate sets (called large k-itemsets in [1]) using previously computed $(k-1)$-predicate sets in a transaction-based database is studied in [1], and is called *Algorithm Apriori*.

Notice that this k-predicate sets computation algorithm is fairly efficient one since it generates candidate k-predicate sets by full exploration of the combination of $(k-1)$-predicate sets before testing the k-predicate pairs against the predicate database. For example, Table 4 contains large 2-predicates "⟨adjacent_to, sea⟩, ⟨close_to, us_boundary⟩" and "⟨close_to, us_boundary⟩, ⟨intersects, provincial highway⟩" but does not contain "⟨adjacent_to, sea⟩, ⟨intersects, provincial highway⟩". It cannot form a candidate 3-predicate "⟨adjacent_to, sea⟩, ⟨close_to, us_boundary⟩, ⟨intersects, provincial highway⟩". Thus the effort of testing such a 3-predicate against the predicate database can be saved.

After finding large k-predicates, the set of association rules for each level l can be derived based on the minimum confidence at this level, $minconf[l]$. This is performed as follows [1]. For every large n-predicate A, if m-predicate B is not a subset of A, the rule "$A \rightarrow B$" is added into the result of the query if $support(A \wedge B)/support(A) \geq minconf[l]$.

The process is summarized in the following procedure, where $\mathcal{LL}[l]$ is the large predicate set table at level l, and $\mathcal{L}[l,k]$ is the large k-predicate set table at level l. The syntax of the procedure is similar to C and Pascal.

```
(1)  procedure find_large_predicates_and_mine_rules(DB);
(2)      for (l := 1; L[l, 1] ≠ ∅ and l < max_level; l++) do begin
```

```
(3)        L[l,1] := get_large_1_predicate_sets(DB,l);
(4)        for (k := 2; L[l,k-1] ≠ ∅; k++) do begin
(5)           P_k := get_candidate_set(L[l,k-1]);
(6)           foreach object s in S do begin
(7)              P_s := get_subsets(P_k, s); {Candidates satisfied by s}
(8)              foreach candidate p ∈ P_s do p.support++;
(9)           end;
(10)          L[l,k] := {p ∈ P_k | p.support ≥ minsup[l]};
(11)       end;
(12)       LL[l] := ⋃_k L[l,k];
(13)       output := generate_association_rules(LL[l]);
(14)    end
(15) end □
```

In this procedure, line (2) shows that the mining of the association rules is performed level-by-level, starting from the top-most level, until either the large 1-predicate set table is empty or it reaches the maximum concept level. For each level l, line (3) computes the large 1-predicate sets and put into table $\mathcal{L}[l,1]$. Lines (4)-(11) computes the large k-predicate sets $\mathcal{L}[l,k]$ for all $k > 1$ at the level l progressively, essentially using the Apriori algorithm [1], as we discuss above. Line (12) collects all the large k predicate at each level l into one table $\mathcal{LL}[l]$, and finally line (13) generates the spatial association rules at each concept level from the large predicate table $\mathcal{LL}[l]$. □

The generated rules may need to be examined by human experts or pass through some automatic rule quality testing program [17] in order to filter out some obvious or redundant rules and output only those fairly new and interesting ones to the users.

4.3 A Discussion of the Algorithm

Algorithm 4.1 is an interesting and efficient algorithm for mining multiple-level strong spatial association rules in large spatial databases. Here we reason on the two essential properties of this algorithm: its correctness and its efficiency.

Correctness of the algorithm.

First, we show that Algorithm 4.1 discovers the correct and complete set of association rules given by the Definition 1.

Step 1 is a query processing process which extracts all data which are relevant to the spatial data mining process based on the completeness and correctness of query processing. Step 2 applies a coarse spatial computation method which computes the whole set of relevant data and thus still ensures its completeness and correctness. Step 3 filters out those 1-predicates whose support is smaller than the minimum support threshold. Obviously, predicates filtered out are those which has no hope to generate rules with support reaching the minimum support. Step 4 applies a fine spatial computation method which computes predicates from the set of derived coarse predicates and thus still ensures the completeness

and correctness based on the nature of the spatial computation methods. Finally, Step 5 ensures to find the complete set of association rules at multiple concept levels based on the previous studies at mining multiple-level association rules in transaction-based databases [1, 13]. Therefore, the algorithm discovers the correct and complete set of association rules.

Efficiency of the algorithm.

We have the following theorem for the efficiency of the algorithm.

Theorem 5. *Let the average costs for computing each spatial predicate at a coarse and fine resolution level be C_c and C_f respectively. The worst-case time complexity of Steps 2-5 of Algorithm 4.1 is $O(C_c \times n_c + C_f \times n_f + C_{nonspatial})$, where n_c is the number of predicates to be coarsely computed in the relevant spatial data sets, n_f is the number of predicates to be finely computed from the coarse predicate database, and $C_{nonspatial}$ is the total cost of rule mining in a predicate database.*

Proof sketch.

Step 1 applies a spatial database query processing method whose computational complexity has been excluded from the total cost of the computation according to the statement of the theorem.

Step 2 involves the computation of the largest set of spatial predicates since each pair of objects needs to be checked to see whether it potentially and approximately satisfies the predicate to be coarsely computed. Since there are totally n_c predicates with distinct object sets as variables to be coarsely computed in the relevant spatial data sets, and the cost of computing each spatial predicate at a coarse resolution level is C_c, the total processing cost at this step should be $O(C_c \times n_c)$.

To avoid checking the predicates which will not be used later in the fine computation, approximate computation can be performed at a coarse resolution level. To accelerate this process, every object can be described using its MBR and coarse predicates can be derived using R-tree technique for spatial join [4] or plane sweep technique [18].

Furthermore, to computations faster one may use the data generalized and approximated data. For example, sinusoity of lines can be reduced, and small regions can be converted to points, etc.

With a similar reasoning, Step 4 involves the computation of the spatial predicates at a refined level. More detailed spatial computation algorithms will be applied at this stage. Since there are totally n_f predicates with distinct object sets as variables to be finely computed in the relevant data sets, and the cost of computing each spatial predicate at a fine resolution level is C_f, the total processing cost at this step should be $O(C_f \times n_f)$. Notice in most cases, $C_f > C_c$, but $n_f \ll n_c$, which ensures that the total cost of computation is reasonable.

According to the algorithm, the computation of support counts, threshold testing, and rule generation will not involve further spatial computation. Thus the total computation cost for Steps 3 and 5 will be $O(C_{nonspatial})$, where $C_{nonspatial}$ is the total cost of rule mining in a nonspatial predicate database.

Adding all costs together, we have the formula presented in the theorem. □

Execution time of the above mining algorithm can be estimated using the results of spatial join computations based on real data [4, 5] and on our experience on mining multilevel association rules [13]. Time of finding multiple level association rules by algorithm 4.1 is presented by (11). Component $C_c^{'} \times N$ of this equation presents time of the execution of step 2 of the algorithm, $C_{filter} \times N_{nsp}$ is the time of filtering small coarse predicates, $C_f \times F_{ratio} \times N_c$ presents execution time of finding fine predicates and $C_{nsp} \times F_{ratio} \times N_{nsp}$ presents mining association rules from the set of fine predicates. Curve *"coarse+filter+fine"* on Fig. 3 shows the execution time of algorithm 4.1. In case when filtering in Step 2 of the algorithm is not used t_2 time is needed as it is shown by curve *"coarse+fine"*. Execution time of naive algorithm when no tree structure is used for finding coarse predicates can be computed by (13). This time is presented by curve *"naive+filter+fine"*. Table 6 lists some parameters used in the cost analysis. Estimated time shown in Fig. 3 indicates a substantial improvement of performance when tree structure is used to compute coarse predicates. It also shows large acceleration of computation process by filtering out coarse predicates not leading to large predicates, which avoids fine computations on such predicates.

$$t_1 = C_c^{'} \times N + C_{filter} \times N_{nsp} + C_f \times F_{ratio} \times N_c + C_{nsp} \times F_{ratio} \times N_{nsp} \quad (11)$$

$$t_2 = C_c^{'} \times N + C_f \times N_c + C_{nsp} \times N_{nsp} \quad (12)$$

$$t_3 = C_c^{''} \times N^2 + C_{filter} \times N_{nsp} + C_f \times F_{ratio} \times N_c + C_{nsp} \times F_{ratio} \times N_{nsp} \quad (13)$$

Name	Value	Meaning
$C_c^{'}$	0.5 ms	constant for finding coarse predicates using R-trees [4]
$C_c^{''}$	0.2 ms	constant for finding coarse predicates using naive algorithm
C_f	10 ms	cost of computing one fine predicate using TR*-trees [5]
C_{nsp}	1.5 ms	constant for finding association rules in a predicates database
C_{filter}	0.5 ms	constant for filtering out predicates in step 3 of the algorithm 4.1
N_{nsp}	0.2 × N	number of tuples in a predicates database
N_c	0.8 × N	number of coarse predicates from step 2 of the algorithm 4.1
F_{ratio}	0.1	ratio of coarse predicate possibly leading to large predicates

Table 6. Database parameters.

5 Discussion

5.1 Major Strengths of the Method

The spatial data mining method developed in the previous section has the following major strengths for mining spatial association rules.

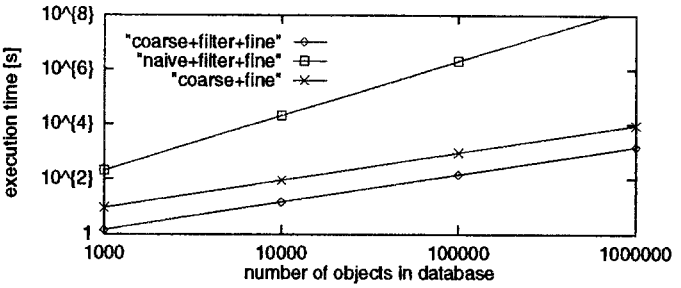

Fig. 3. Execution time.

- Focused data mining guided by user's query.
  The data mining process is directed by a user's query which specifies the relevant objects and spatial association relationships to be explored. This not only confines the mining process to a relatively small set of data and rules for efficient processing but also leads to desirable results.
- User-controlled interactive mining.
  Users may control, usually via a graphical user interface (GUI), minimum support and confidence thresholds at each abstraction level interactively based on the currently returned mining results.
- Approximate spatial computation: Substantial reduction of the candidate set.
  Less costly but approximate spatial computation is performed at an abstraction level first on a large set of data which substantially reduces the set of candidate data to be examined in the future.
- Detailed spatial computation: Performed once and used for knowledge mining at multiple levels.
  The computation of support counts at each level can be performed by scanning through the same computed spatial predicate table.
- Optimizations on computation of k-predicate sets and on multiple-level mining.
  These two optimization techniques are shared with the techniques for mining other (i.e., nonspatial) multiple-level association rules [13]. First, it uses the $(k-1)$-predicate sets to derive the candidate k-predicate sets at each level, which is similar to the *apriori* algorithm developed in [1]. Second, it starts at the top-most concept level and applies a progressive deepening technique to examine at a lower level only the descendants of the large 1-predicates, which is similar to the technique developed in [13].

5.2 Alternatives of the Method

Many variations and extensions of the method can be explored to enhance the power and performance of spatial association rule mining. Some of these are listed as follows.

- Integration with nonspatial attributes and predicates.
  The relevant set of predicates examined in our examples are mainly spatial ones, such as *close_to, inside*, etc. Such a process can be integrated with

the generalization and association of nonspatial data, which may lead to the rules, such as "*if a house is big and expensive, it is located in West Vancouver or Vancouver West-End (with 75% of confidence)*", etc.

- Mining spatial association rules in multiple thematic maps.
  In principle, the method developed here can be applied to handle the spatial databases with multiple thematic maps. The rule mining process will be similar to the one presented above since the judgement of $g\_close\_to(X, Y)$ or $intersect(X, Y)$ can be performed by an approximate or detailed map overlay. The mining algorithm itself will remain intact.
- Multiple and dynamic concept hierarchies.
  Our method can also handle the cases when there exist multiple concept hierarchies or when the concept hierarchies need to be adjusted dynamically based on data distributions. For example, *towns* can be classified into *large* or *small* according to an existing hierarchy, *coast* or *in-land* according to their distance to the ocean, or *southwest, southeast*, etc. according to their geographic areas. Different characteristics will be discovered based on different hierarchies or their adjustments, which is similar to execute the same algorithm based on different knowledge-bases.

6 Conclusion

Based on the previous studies on spatial data mining and mining association rules in transaction-based databases, we proposed and studied an interesting method in this paper for mining strong spatial association rules in large spatial databases. Discovery of spatial association rules may disclose interesting relationships among spatial and/or nonspatial data in large spatial databases and thus it represents a new and promising direction in spatial data mining.

The method developed in this paper explores efficient mining of spatial association rules at multiple approximation and abstraction levels. It proposes first to perform less costly, approximate spatial computation to obtain approximate spatial relationships at a high abstraction level and then refine the spatial computation only for those data or predicates, according to the approximate computation, whose refined computation may contribute to the discovery of strong association rules. Such a two-step spatial mining method facilitates mining strong spatial association rules at multiple concept levels by a top-down, progressive deepening technique.

Our study is based on the assumption that a user has reasonably good knowledge on what s/he wants to find, and that there exists good knowledge (such as concept or operation hierarchies) for nonspatial or spatial generalization. Such assumptions, though valid in many cases, may enforce some strong restrictions to naive users or to some complex spatial databases with poorly understood structures or knowledge. More studies are needed to overcome these restrictions.

The method investigated in this study is currently under implementation and experimentation as one of several spatial data mining methods being developed in the spatial data mining system prototype, GeoMiner. We plan to integrate this

technique with the generalization-based spatial data mining technique developed before [14, 15] and will report the prototype implementation and the experiments with reasonably large spatial databases in the future.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 1994 Int. Conf. VLDB*, pp. 487–499, Santiago, Chile, Sept. 1994.
2. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data*, pp. 207–216, Washington, D.C., May 1993.
3. W. G. Aref, and H. Samet. Optimization Strategies for Spatial Query Processing. In *Proc. 17th Int. Conf. VLDB*, Barcelona, Spain, pp. 81–90, Sept. 1991.
4. T. Brinkhoff, H. P. Kriegel, B. Seege. Efficient Processing of Spatial Joins Using R-trees. In *Proc. 1993 ACM-SIGMOD Conf. Management of Data*, Washington, D.C. pp. 237–246, May 1993.
5. T. Brinkhoff, H. P. Kriegel, R. Schneider, B. Seege. Multistep Processing of Spatial Joins. In *Proc. 1994 ACM-SIGMOD Conf. Management of Data*, Minneapolis, Minnesota, pp. 197–208, May 1994.
6. M. Egenhofer. Reasoning about Binary Topological Relations. In *Proc. 2nd Sympp. SSD'91*, Zurich, Switzerland, pp. 143–160, Aug. 1991.
7. M. Egenhofer. Spatial SQL: A Query and Presentation Language. In *IEEE Trans. Knowledge and Data Engineering*, 6:86-95, 1994.
8. U. Fayyad, and P. Smyth. Image Database Exploration: Progress and Challenges. In *Proc. 1993 Knowledge Discovery in Databases Workshop*, pp. 14-27, Washington, D.C..
9. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining* , AAAI/MIT Press, 1995.
10. S. Fotheringham, and P. Rogerson. *Spatial Analysis and GIS*, Taylor and Francis, 1994.
11. A. Guttman. R-trees: A Dynamic Index Structure for Spatial Searching. In *Proc. ACM SIGMOD Int. Conf. on Management of Data* , Boston, MA, 1984, pp. 47-57.
12. J. Han, Y. Cai, and N. Cercone. Data-driven Discovery of Quantitative Rules in Relational Databases. *IEEE Trans. Knowledge and Data Eng.*, 5:29–40, 1993.
13. J. Han, and Y. Fu. Discovery of Multiple-Level Association Rules from Large Databases in *Proc. 1995 VLDB*, Zurich, Switzerland, Sept. 1995.
14. W. Lu, J. Han, and B. C. Ooi. Discovery of General Knowledge in Large Spatial Databases. In *Proc. Far East Workshop on Geographic Information Systems* pp. 275-289, Singapore, June 1993.
15. R. Ng, and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. 1994 Int. Conf. VLDB*, pp. 144–155, Santiago, Chile, Sept. 1994
16. G. Piatetsky-Shapiro, and W. J. Frawley. Knowledge Discovery in Databases, AAAI/MIT Press, 1991.
17. G. Piatesky-Shapiro, C. J. Matheus. The Interestingness of Deviations. In *Proc. 1994 Workshop on Knowledge Discovery in Databases*, Seattle, WA, pp. 25-36.
18. F. P. Preparata, and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 1985.
19. H. Samet. *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1990.