

# Seminar Report

## Algorithm of Spatial Data Mining

Rajiv Gandhi  
Roll no. 05331002  
[rgandhi@iitb.ac.in](mailto:rgandhi@iitb.ac.in)



CSRE, IIT Bombay.

November 8, 2005

# Index

<i>Topic</i>	<i>Page</i>
1. Introduction	3
2. Pattern Discovery	4
2.1 The Data Mining Process	4
2.2 Spatial Measures	6
2.3 Spatial Statistical Models	7
3. The Data Mining Trinity	7
3.1 Classification	8
3.2 Linear Regression	8
3.2.1 Spatial Regression	9
3.3 Clustering	9
3.1.1 Categories of Clustering Algorithms	10
3.4 Association Rule	11
3.4.1 Method for Mining Spatial Association	16
3.4.2 Algorithm for Mining Spatial Association	20
3.4.3 Explanation of the detailed steps	21
3.4.5 Efficiency of the algorithm	24
3.4.4 Discussion of the Algorithm	26
3.4.5 Conclusion of Algorithm	28
4. Summary	29
5. References	29

# 1. Introduction

Today, there are countless terabytes of data processed in database systems, and we store a measurable portion of that data for analysis purpose, and it is increasing as new technologies are arriving, like with the wide applications of remote sensing technology and automatic data collection tools stored data in large spatial data bases. 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, which gives the need of data mining

Data mining is the process of discovering interesting and potentially useful patterns of information embedded in large database. The mining metaphor is meant to convey an impression that patterns are nuggets of precious information hidden with in large database, which is to be discovered based on need

If data mining is about extracting patterns from large database, then the spatial database are large database and have a strong need, for example the Earth Observation Satellites(by NASA), which are systematically mapping the entire surface of earth, collect about one terabyte of data every day

But the requirements of mining spatial database are different from those of mining classical relational database, in particular, the notation of spatial autocorrelation that similar object tend to cluster in geographic space, is central to spatial data mining

Here it is important to understand the distinction between spatial data mining and spatial data analysis. Spatial data analysis covers the broad spectrum of techniques that deal with both spatial and non spatial characteristic of spatial objects, on the other hand spatial data mining techniques are often derived from spatial statistics, spatial analysis, machine learning and database are customized to analyze massive data set

The complete data mining process is a combination of many sub processes. Some important are data extraction, data cleaning, feature selection, algorithm design, tuning and analysis of the output when the algorithm is applied to the data. For the spatial data, the issue of scale and the level of aggregation at which the data are being analyzed are also very important. It is well known in spatial analysis the identical experiments are at different levels of scale can some time lead to contradictory results

In this report, my focus is limited to the design of mining algorithms, in particular I surveyed association rule discovery algorithm. Firstly I introduce the data mining process and enumerate some well known techniques that are associated with data mining, then important concept of spatial auto-correlation, further I discuss about classification techniques, spatial regression, various clustering techniques and, then intensive dealing with association rules discovery. This report considers existing and upcoming theoretical models as well as models currently implemented in various spatial database systems.

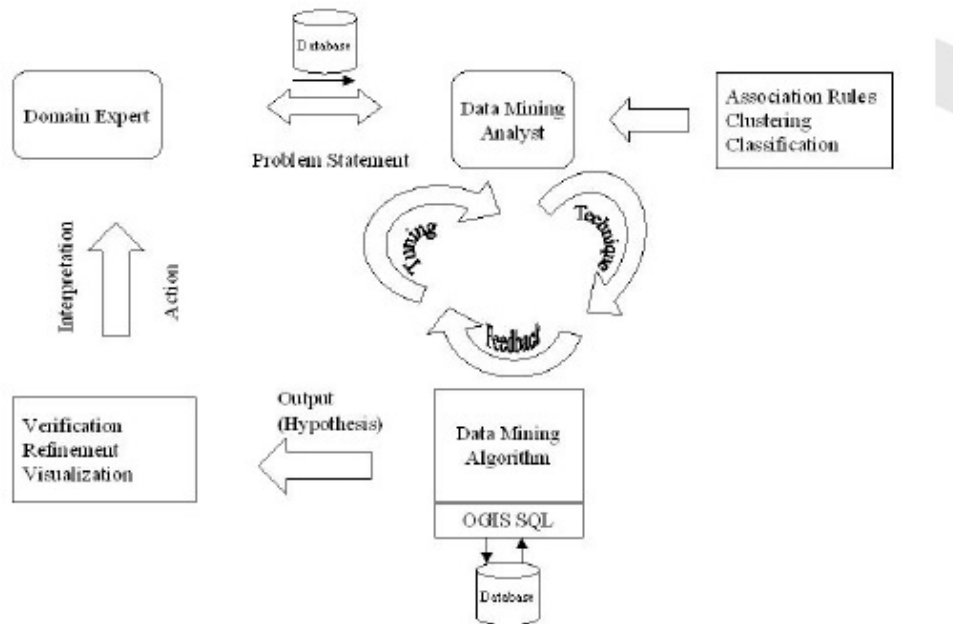
## 2. Pattern Discovery

A pattern is interesting, useful and hidden statistic like the mean, median or standard deviation of dataset or it can be a rule such that “the person who buys computer also buys UPS ” and much more. Data mining is the process of discovering potentially these hidden patterns, thus data mining encompasses a set of techniques to generate hypotheses, followed by their validation and verification via standard statistical tools. The promise of data mining is its ability to rapidly and automatically searches for local and potentially high utility patterns using computer algorithms

Spatial data mining can be categorized based on the kinds of rules to be discovered in spatial databases. A **spatial characteristics rule** is a general description of a set of spatial related data, for example the description of general forest pattern in a set of geographic regions is a spatial characteristics rule. A **spatial discriminant rule** is that which gives general description of the contrasting feature of a class of spatial related data from other class.

### ❖ The Data Mining Process

The entire data mining process is a typical scenario a domain expert (DE) consults a data-mining analyst (DMA) to solve a specific problem. For example a builder wants to know the available land which is suitable for his construction and meet with every specified requirement on huge geography area. Now DMA must decide which techniques or combination of techniques is required to address the problem, like DMA might want to use classification modal which predicts the most suitable land according to the requirements, and for classification, the DMA may decide to use linear regression instead of decision trees because the class attribute is continuous valued. The following is the figure of data mining process



The output of a data mining algorithm is typically a hypothesis which can be in the form of modal parameters, rules or labels. Thus the output is a potential pattern. The next step is verification, refinement and visualization of pattern. For spatial data this part of process is typically done with help of GIS software

The data mining process does not mean working with statistics, one way to view data mining is as a filter step before the application of rigorous statistical tools. The roll of the filter step is to literally plow through reams of data and generate some potentially interesting hypothesis which can be verified using statistics

The difference between non spatial and spatial data mining parallels the difference between non spatial and spatial statistics. One way of fundamental assumption that guide statistical analysis is that the data sample are independently generated, as with successive tosses of a coin or rolling of die. When it comes to the analysis of spatial data the assumption about the independence of sample is generally false. In fact spatial data tends to be highly self-correlated. For example people with similar characteristics, occupations and background tend to cluster in the same neighborhoods. The economics of a region tend to be similar. Changes in natural resource, wild life and temperature vary gradually over space. In fact this property of like things to cluster in space is so fundamental that geographers have elevated it to the status of the first law of geography "Every thing is related to every thing else, but nearby are more related then distant thing" [Tobler, 1979]

## ❖ Spatial Measures

### *Mean Centre*

It is the average location computed as the mean of X and mean of Y coordinate

### *Weighted Mean Center*

This is computed as the ratio between the sum of coordinates of each point multiplied by its weight (i.e. number of people in the block) it is appropriate measure for measuring like center of population

### *Central*

It is used to simplify the complex objects (e.g. to avoid storage requirements and complexity of digitization of boundaries, a geographic object can be represented by its center, or for identifying the most effective location for planned activity like distribution center should be located at a central point so that travel to it is minimized.

### *Dispersion*

It is a measure of the spread of a distribution around the center. It is summation over the ratio of the weigh of geographic objects and proximity between them

### *Shape*

It is multi-dimensional measures, and there is no measure to capture all the dimensions, many shape measures are based on comparison of the shape's perimeter with that of a circle of the same area

### *Spatial dependence*

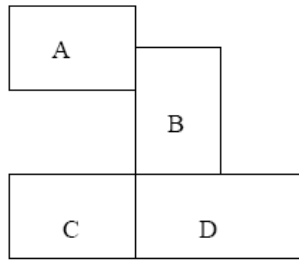
It can be defined as the propensity of a variable to exhibit similar (of different) values as a function of the distance between the spatial locations at which these are measured. Spatial auto correlation is used to measure spatial dependence

### *Moran's I : A Global Measure Of Spatial Autocorrelation*

Given a variable  $x = \{x_1, x_2, \dots, x_n\}$  which is sampled over  $n$  locations, Moran's I coefficient is defined as

$$I = \frac{zWz^t}{zz^t}$$

Where  $z = \{x_1 - X, \dots, x_n - X\}$ ,  $X$  is the mean,  $W$  is the  $n \times n$  row normalized contiguity matrix, and  $z^t$  is the transpose of  $z$ , the following is the example



(a) Map

	A	B	C	D
A	0	1	0	0
B	1	0	1	1
C	0	1	0	1
D	0	1	1	0

(b) Boolean  $W$

	A	B	C	D
A	0	1	0	0
B	0.3	0	0.3	0.3
C	0	0.5	0	0.5
D	0	0.5	0.5	0

(c) Row-normalized  $W$

## ❖ Spatial Statistical Models

This model is often used to represent the observation in terms of random variables. These models then can be used for estimation, description, and prediction based on probability

### **Point process:**

A point process is a model for the spatial distribution of the point in a point pattern. Several natural processes can be molded as spatial point pattern. The positions of trees in a forest, locations of gas stations in a city all are the examples of point pattern and it is defined as  $Z(t) = 1 ; \forall t \in T \text{ or } Z(A) = N(A), A \subset T$  Where both  $Z(\cdot)$  and  $T$  are random. Here  $T$  is the index set ( $T \subset \mathfrak{R}^d$ ) and  $Z(\cdot)$  is the spatial process spatial point process can grouped into two groups **random** (point generated by Poisson process using average distance between a point and its neighbor) and **non random** (can be either clustered by aggregated pattern or uniformly spaced using regular patterns)

### **Lattices:**

A lattice  $D$  is denoted by  $Z(s): s \in D$ , where the index set  $D$  is a countable set spatial sites at which data are observed. Here the lattice refers to a countable collection of regular or irregular spatial sites

### **Geo-statistics:**

It deals with the analysis of spatial continuity which is an inherent characteristic of spatial data sets. Geostatistics provides a set of statistical tool for modeling spatial variability and interpolation of attributes at un-sampled locations spatial variability can be analyzed using *variogram*. Spatial interpolation techniques are used to estimate the values at un-sampled place using known values at sampled locations **Kriging** as a well known estimation procedure used in Geostatistics, it uses known values and semi variogram generated from the data, it is different from conventional because it takes into account the spatial auto correlation

### 3. The Data Mining Trinity

Data mining is truly a multidisciplinary area, and there are many novel ways of extracting patterns from data, still, if one were to label data mining techniques, then the three most non controversial label would be **CLASSIFICATION**, **CLUSTERING**, and **ASSOCIATION RULES**

#### ❖ Classification

Classification is to find a function  $f : D \rightarrow L$  here  $D$ , the domain of  $f$ ;  $L$  is the set of labels. The goal of classification problem is to determine the appropriate  $f$  from a given finite subset  $T_r \subset D \times L$ . the success of classification is determined by the accuracy of  $f$  when applied to data set  $T_o$  which disjoint from  $T_r$ . The classification problem is known as predicate modeling because  $f$  is used to predicate the labels  $L$  when only data from set  $D$  is given.

There are many techniques available to solve the classification problem. They are following

- Classification trees
- Neural networks
- Bayesian learning or Maximum-likelihood
- Nearest neighbor
- Radial basis functions
- Support vector machines
- Meta learning methods

#### ❖ Linear regression

When class variable is real valued, it is more appropriate to calculate the conditional expectation rather than the conditional probability. Then the goal of classification is to compute

$$E[C|A_1, \dots, A_n]$$

Writing in a more familiar notation, with  $C$  replaced by  $Y$  and  $A_i^s$  by  $X_i^s$  and assuming that all the attributes are identical and independently generated standard normal random variables, the linear regression equation is

$$E[Y|X = x] \equiv \alpha + \beta x$$

Where  $X = (X_1, \dots, X_n)$ , this expression is equivalent to the more familiar expression.  $Y = X\beta + \varepsilon$ . Once again, the training data can be used to calculate the parameter vector  $\beta$  which in turn can be used to calculate the value of the class attribute in the test data set

## Spatial Regression

When variables are spatially referenced, they tend to exhibit spatial autocorrelation, thus the above assumption of identical independent distribution of random variables is not appropriate in the context of spatial data. Spatial statisticians have proposed many methods to extend the regression techniques that account for spatial autocorrelation. The simplest and most intuitive is to modify the regression equation with the help of the contiguity matrix  $W$ . thus the spatial autoregressive regression (SAR) equation is

$$Y = \rho WY + X\beta + \varepsilon$$

The solution procedure for the SAR equations is decidedly more complex than the classical regression equation because of the presence of the  $\rho WY$  term on the right side of the equation. Also notice that the  $W$  matrix is quadratic in terms of the data samples. Fortunately very few entries of  $W$  are non zero, and sparse matrix techniques are used, which exploit this fact, to speed up the solution process

## ❖ Clustering

Clustering is a process for discovering groups, in large database. Unlike classification clustering involve no a priori information either on the number of clusters or what cluster labels are. Thus there is no concept of training or test data in clustering, because of that clustering is also referred as *unsupervised learning*

The cluster is formed on the basis of a similarity criterion which is used to determine the relationship between each pair of tuples in the database. Tuples which are similar are usually grouped together and then the group is labeled, of course Domain Expert does have to examine, verify, and possibly refine the cluster

Clustering is very well known technique in statistics and the data mining role is to scale a clustering algorithm to deal with the large data sets which are now becoming the norm rather than exception. The size of the database is a function of the number of records in the table and also the number of attributes of each record. Beside the volume, the type of the data, whether it is numeric, binary, categorical, or ordinal is an important determinant in the choice of the algorithm employed

It is convenient to frame the clustering problem in a multi dimensional attribute space. Given  $n$  data objects described in term of  $m$  variables, each object can be represented as a point in an  $m$ -dimensional space. Clustering then reduces to determining high-density groups of points from a set of non uniformly distributed points. The search for potential within the multidimensional space is then driven by a suitably chosen similarity criterion

## Categories of Clustering Algorithms

Cluster analysis is one of the most often performed data analysis technique in many fields. This has resulted in a multitude of clustering algorithms, so it is useful to categorize them into groups, based on the technique adopted to define clusters, the clustering algorithms can be categorized into four broad categories:

1. **Hierarchical** clustering methods start with all patterns as a single cluster, and successively perform splitting or merging until a stopping criterion is met. This results in a tree of clusters, called dendograms. The dendogram can be cut at different levels to yield the desired clusters. Hierarchical algorithms can be further divided into agglomerative and divisive methods. Some of the hierarchical clustering algorithms are

- Balanced Iterative Reducing And Clustering Using Hierarchies
- Clustering Using Representative
- Robust Clustering Using Links

2. **Partitional** clustering algorithms start with each pattern as a single cluster and iteratively reallocate data point to each cluster until a stopping criterion is met. These methods tend to find clusters of spherical shape. K-means and K-medoids are commonly used partition algorithms. Squared error is the most frequently used criterion function in partitional clustering. Some of the recent algorithms in this category are

- Partitioning Around Clustering (PAM)
- Clustering Large Applications (CLARA)
- Clustering Large Application Based On Random Search (CLARANS)
- Expectation Maximization (EM)

3. **Density based** clustering algorithms tries to find the clusters based on density of data points in a region. These algorithms treat cluster as dense regions of objects in the data space. Some of the density based clustering algorithms are

- Density based spatial clustering of application with noise DBSCAN)
- Density based clustering (DENCLUE)

4. **Grid based** clustering algorithms first quantize the clustering space into a finite number of cells then perform the required operations on the quantized space. Cells that are more than certain number of points are treated as dense. The dense cells are connected to form the clusters. Grid based clustering algorithms are primarily developed for analyzing large spatial data sets. Some of the grid based clustering algorithms are

- Statistical Information Grid Based Method (STING)
- STING+
- Wave Cluster
- BANG- Clustering
- Clustering In Quest (CLIQUE)

## ❖ Association Rule

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, meaning which occur frequently and hold in most cases, this we measure by taking two variables minimum *support* and minimum *confidence* 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 threshold 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 USA, which associates predicates *is\_a()*, *within()* and *adjacent\_to()* with spatial predicate *close\_to()*

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

Although such rules are usually not 100% true, they carry some nontrivial knowledge about spatial association, and thus it is interesting to “Mine them from large spatial database. The discovered rules will be useful in geography, environmental studies, biology, engineering and other fields

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

$$P_1 \wedge P_2 \dots P_m \rightarrow Q_1 \wedge Q_2 \dots Q_n \quad (c\%)$$

Where at least one of the predicate  $P_1 \wedge P_2 \dots P_m$ ,  $Q_1 \wedge Q_2 \dots Q_n$  is a spatial predicate and  $c\%$  is a confidence of the rule which indicate that  $c\%$  of objects satisfying the antecedent of the rule will also satisfy the consequent of the rule

An association rule can also be seen in a general form of dependency rule and is defined on transactional based database. "It in the form of " $P \rightarrow Q (c\%)$  if a pattern  $P$  appears in a transaction, there is  $c\%$  possibility (confidence) that pattern  $Y$  holds in the same transaction" where  $P$  and  $Q$  are set of attribute values.

**Definition 2:** the **support** of a conjunction of predicates  $P = P_1 \wedge P_2 \dots P_k$  in a set of  $S$ , denoted by as  $\sigma(P/S)$ , is the number of objects in  $S$  which satisfy  $P$  versus the cardinality of  $S$ . the confidence of a rule  $P \rightarrow Q$  in  $S$ ,  $\vartheta(P \rightarrow Q/S)$  is the ratio of  $\sigma(P \rightarrow Q/S)$  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** and conjunction of  $k$  single predicate is called a **k-predicate**.

It is to ensure that such rules are interesting enough to cover frequently encountered patterns in a database the concept of **support** of a rule " $P \rightarrow Q$ " is there, which is defined as the ratio that the patterns of  $P$  and  $Q$  occurring together in the transaction vs. total number of transaction in the database, for example in 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.

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 association between 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  $P$  in no less than its minimum support threshold  $\sigma'_k$  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 at high level  $k$  if its confidence is no less than its corresponding minimum confidence threshold  $\vartheta'_k$ .

**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

In this part, efficient methods for mining spatial association rules are studied, with top-down, progressive deepening search techniques proposed. The technique firstly searches at a high concept level for large patterns and strong implication relationships among the large patterns at a coarse resolution scale. Then only for the 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 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 MBR(minimum bounding rectangle). Only the candidate spatial predicates, which are worth detailed examination, will be computed by spatial techniques (giving detailed predicates such as contain in or intersect etc.). Such multiple level approaches save much computation time because it is very expensive to perform detailed spatial computation for all the possible spatial association relationships

Generalization-based spatial data mining methods discover spatial and nonspatial relationship at a general concept level, where spatial objects are expressed as merged spatial regions or clustered spatial points, however there methods can not discover rules reflecting structure of spatial objects and spatial to spatial or spatial to non spatial relationship which contain spatial predicates, such as *near\_by()*, *inside()*, or *intersecting()* etc. as a complementary, spatial association rules represents object to predicate relationship containing spatial predicates. For example the following one more example says about spatial association rule.

- non spatial consequent with spatial antecedent

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

- Spatial consequent with non spatial to spatial antecedent

$$Is\_a(x, patrol\ pump) \rightarrow close\_to(x, highway) \quad (75\%)$$

Various kinds of spatial predicates can be involved in spatial association rules. They may represent topological relationship between spatial objects, such as *disjoint*, *inside/outside*, *covers/covered by*, *left*, *right*, *north*, *east* etc. can contain some distance information such that *close\_to* , *far\_away* etc.

**Example1.** Let the spatial database to be studied adopt an extended-relational data model and SAND (spatial-and-non spatial database) architecture. 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 species 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 assumed Geo-Miner 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.

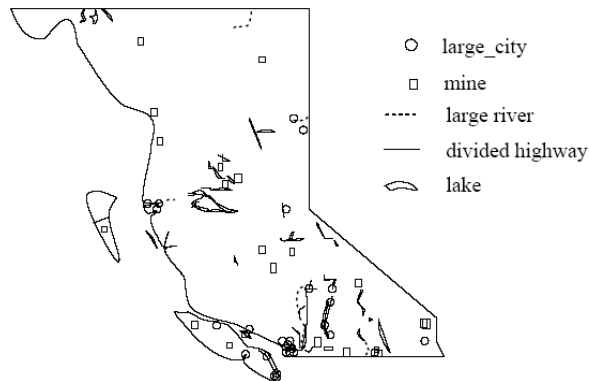


Fig. 1. The map of BC.

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

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

However, “close to” between a town and a road will be defined by a smaller distance. 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 (... ) ... ) ... ) ... )

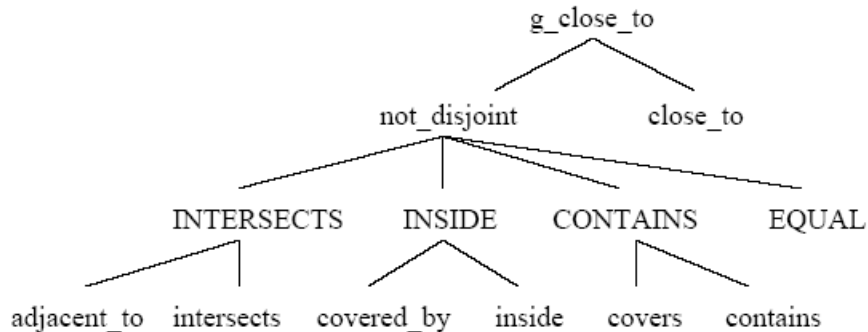
- A concept hierarchy for water:

(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 the following Fig.



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.

### A Method for Mining Spatial Association Rules

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 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\*
- (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, or R\*-trees and other approximations. The derived spatial predicates are collected in a “*g\_close\_to*” table, 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.

$is\_a(X, large\_town) \rightarrow g\_close\_to(X, water): (80\%)$   
 $is\_a(X, large\_town) \wedge g\_close\_to(X, sea) \rightarrow g\_close\_to(X, us\_boundary) : (92\%)$

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	$\langle adjacent\_to, J.Fuca\_Strait \rangle$	$\langle intersects, highway\_1 \rangle,$ $\langle intersects, highway\_17 \rangle$	$\langle close\_to, US \rangle$
Saanich	$\langle adjacent\_to, J.Fuca\_Strait \rangle$	$\langle intersects, highway\_1 \rangle,$ $\langle close\_to, highway\_17 \rangle$	$\langle close\_to, US \rangle$
Prince_George		$\langle intersects, highway\_97 \rangle$	
Pentincton	$\langle adjacent\_to, Okanagan\_Lake \rangle$	$\langle intersects, highway\_97 \rangle$	$\langle close\_to, US \rangle$
...	...	...	...

**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 Table3 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 pairwise 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	$\langle \text{adjacent\_to, water} \rangle$	32
1	$\langle \text{intersects, highway} \rangle$	29
1	$\langle \text{close\_to, highway} \rangle$	29
1	$\langle \text{close\_to, us\_boundary} \rangle$	28
2	$\langle \text{adjacent\_to, water} \rangle, \langle \text{intersects, highway} \rangle$	25
2	$\langle \text{adjacent\_to, water} \rangle, \langle \text{close\_to, us\_boundary} \rangle$	23
2	$\langle \text{close\_to, us\_boundary} \rangle, \langle \text{intersects, highway} \rangle$	26
3	$\langle \text{adjacent\_to, water} \rangle, \langle \text{close\_to, us\_boundary} \rangle, \langle \text{intersects, highway} \rangle$	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 Table3. For example, since  $(intersects, highway)$  has a support count of 29, and  $(adjacent\_to, water)$  has a support count of 25, and  $25=29 \div 86\%$ , 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	$\langle adjacent\_to, sea \rangle$	21
1	$\langle adjacent\_to, large\_river \rangle$	11
1	$\langle close\_to, us\_boundary \rangle$	28
1	$\langle intersects, provincial\_highway \rangle$	21
1	$\langle close\_to, provincial\_highway \rangle$	24
2	$\langle adjacent\_to, sea \rangle, \langle close\_to, us\_boundary \rangle$	15
2	$\langle close\_to, us\_boundary \rangle, \langle intersects, provincial\_highway \rangle$	19
2	$\langle adjacent\_to, sea \rangle, \langle close\_to, provincial\_highway \rangle$	11
2	$\langle close\_to, us\_boundary \rangle, \langle close\_to, provincial\_highway \rangle$	22
3	$\langle adjacent\_to, sea \rangle, \langle close\_to, us\_boundary \rangle, \langle close\_to, provincial\_highway \rangle$	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	$\langle adjacent\_to, georgia\_strait \rangle$	9
1	$\langle adjacent\_to, fraser\_river \rangle$	10
1	$\langle close\_to, us\_boundary \rangle$	28
2	$\langle adjacent\_to, georgia\_strait \rangle, \langle close\_to, us\_boundary \rangle$	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.

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

### **An Algorithm for Mining Spatial Association Rules**

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

*Algorithm for 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:
  - A spatial database, SDB, containing a set of spatial objects
  - A relational database, RDB, describing nonspatial properties of spatial objects
  - A set of concept the hierarchies,
2. a query, which consist of:
  - A reference class S
  - A set of task-relevant classes for spatial objects C1...Cn
  - A set of task-relevant spatial relations
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(SDBRDB)*

*Step 2: Coarse\_predicate\_DB :=  
coarse spatial computation(Task relevant DB)*

*Step 3: Large\_Coarse\_predicate\_DB :=  
filtering with minimumsupport(Coarse predicate DB)*

*Step 4: Fine\_predicate\_DB :=  
Refined\_spatial\_computation(Large\_Coarse\_predicate\_DB)*

*Step 5: Find\_large\_predicates and min\_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 or fast MBR technique and plane-sweep algorithm 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 computational algorithms 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 than 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-item sets in) using previously computed (k-1) predicate sets in a transaction-based database 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. For every large n-predicate A, if m-predicate B is not a subset of A, the rule "A→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  $LL[l]$  is the large predicate set table at level l, and  $L[l, k]$  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 ≤ 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)              Pk := get_candidate_set(L[l, k-1]);
(6)              foreach object s in S do begin
(7)                  Ps := get_subsets(Pk, s); {Candidates satisfied by s}
(8)                  foreach candidate p ∈ Ps do p.support++
(9)              end;
(10)         L[l, k] := {p ∈ Pk | 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 large1-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  $L[l,1]$ . Lines (4)-(11) computes the large  $k$ -predicate sets  $L[l, k]$  for all  $k > 1$  at the level  $l$  progressively, essentially using the Apriori algorithm, as we discuss above. Line (12) collects all the large  $k$  predicate at each level  $l$  into one table  $LL[l]$ , and finally line (13) generates the spatial association rules at each concept level from the large predicate table  $LL[l]$ .

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

### **Efficiency of the algorithm.**

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

**Theorem.** *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*

$$O(C_c \times n_c + C_f \times n_f + C_{non-spatial})$$

*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  $C_{non-spatial}$  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 or plane sweep technique. Furthermore, to computations faster one may use the data generalized and approximated data. For example, sinuosity 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  $O(C_c \times n_c)$ . 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  $O(C_{\text{nonspatial}})$ , where  $C_{\text{nonspatial}}$  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 and on our experience on mining multilevel association rules. 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,  $C_{\text{filter}} \times N_{\text{nsp}}$  is the time of filtering small coarse predicates,  $C_f \times F_{\text{ratio}} \times N_c$  presents execution time of finding fine predicates and  $C_{\text{nsp}} \times F_{\text{ratio}} \times N_{\text{nsp}}$  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 "naïve + 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
$N_{nsp}$	$0.2 \times N$	number of tuples in a predicates database
$N_c$	$0.8 \times N$	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.

## 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.

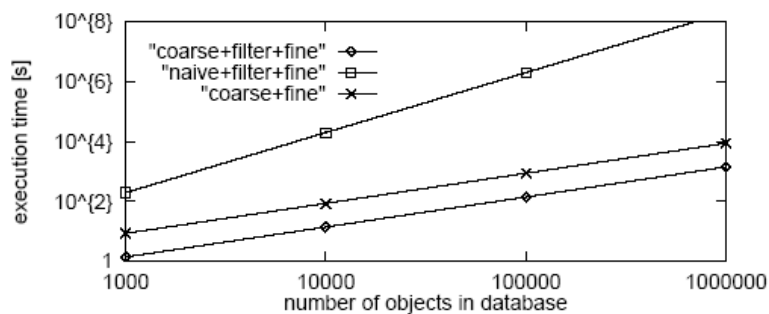
*Step 3* filters out those 1-predicates whose support is smaller than the minimum support threshold. Obviously, predicates filtered out are those which has no hope to generate rules with support reaching the minimum support.

*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.

*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.

### ***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 species 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. 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. Second, it starts at the topmost 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 above

### ***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 sets 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 judgment of *g\_close\_to(XY)* or *intersect(XY)* 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, *town* can be classified into large or *small* according to an existing hierarchy, *coast* or *inland* according to their distance to the ocean, or *south west*, *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.

## **Conclusion of Algorithm**

Based on the previous studies on spatial data mining and mining association rules in transaction-based databases, an interesting method in this report for mining strong spatial association rules in large spatial databases is studied. 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 surveyed in this report 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. Here study is based on the assumption that a user has reasonably good knowledge on what she/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.

## 4. Summary

Data mining is a rapidly developing area which lies at the intersection of database management, statistics and artificial intelligence. Data mining provides semi-automatic techniques for discovering unexpected patterns in very large quantities of data.

Spatial data mining is niche area with in data mining for rapid analysis of spatial data. Spatial data mining can potentially influence major scientific challenges including global change and genomics

The distinguishing characteristics of spatial data mining can be neatly summarized by the first law of geography: all the things are related but nearby things are more related than distinct things. The implication of this statement is that the standard assumption of independence and identically distributed random variables, which categorized non spatial data mining is not applicable for spatial data. Spatial statisticians have coined the word spatial-auto-correlation to capture this property of spatial data

The important techniques of data mining are association rules, clustering, classification and regression. Each of these techniques has to be modified before they can be used to mine spatial data. In general there are strategies available to modify the data mining techniques to make them more sensitive for spatial data, like spatial regression etc.

## 5. References

- ❖ Krzysztof Koperski and Jiawei Han, Discovery of spatial association rules in geographic information database
- ❖ R. Agrawal and R. Srikant. Fast algorithms for mining association rules. InProc.1994 Int. Conf. VLDB, pp. 487
- ❖ Shashi Shekhar, E- Book (7<sup>th</sup> chapter) at <http://www.cs.umn.edu/research/shashi-group/Book/>
- ❖ Jiawei Han , Book knowledge discovery from Geographic database