Spatial Data Mining

Introduction

The explosive growth of spatial data and widespread use of spatial databases emphasize the need for the automated discovery of spatial knowledge. Spatial data mining is the process of discovering interesting and previously unknown, but potentially useful patterns from spatial databases. The complexity of spatial data and intrinsic spatial relationships limits the usefulness of conventional data mining techniques for extracting spatial patterns.

Spatial data are the data related to objects that occupy space. A spatial database stores spatial objects represented by spatial data types and spatial relationships among such objects. Spatial data carries topological and/or distance information and it is often organized by spatial indexing structures and accessed by spatial access methods. These distinct features of a spatial database pose challenges and bring opportunities for mining information from spatial data. Spatial data mining, or knowledge discovery in spatial database, refers to the extraction of implicit knowledge, spatial relations, or other patterns not explicitly stored in spatial databases.

Till a few years back, statistical spatial analysis had been the most common approach for analyzing spatial data. Statistical analysis is a well studied area and therefore there exist a large number of algorithms including various optimization techniques. It handles very well numerical data and usually comes up with realistic models of spatial phenomena. The major disadvantage of this approach is the assumption of statistical independence among the spatially distributed data. This causes problems as many spatial data are in fact interrelated, i.e., spatial objects are influenced by their neighboring objects. Kriging (interpolation technique) or regression models with spatially lagged forms of the dependent variables can be used to alleviate this problem to some extent. Statistical methods also do not work well with incomplete or inconclusive data. Another problem related to statistical spatial analysis is the expensive computation of the results. With the advent of data mining, various methods for discovering knowledge from large spatial databases have been proposed and many such methods can be developed to the different kind of datasets.

Spatial Data Mining

Spatial data mining is a special kind of data mining. The main difference between data mining and spatial data mining is that in spatial data mining tasks we use not only non-spatial attributes (as it is usual in data mining in non-spatial data), but also spatial attributes. Spatial data mining is the process of discovering interesting and previously un-known, but potentially useful patterns from large spatial datasets. Extracting interesting and useful patterns from spatial datasets is more difficult than extracting the corresponding patterns from traditional numeric and categorical data due to the complexity of spatial data types, spatial relationships, and spatial autocorrelation. Specific features of geographical data that preclude the use of general purpose data mining algorithms are:

- rich data types (e.g., extended spatial objects)
- implicit spatial relationships among the variables
- observations that are not independent, and

- spatial autocorrelation among the features.

Preprocessing spatial data: Spatial data mining techniques have been widely applied to the data in many application domains. However, research on the preprocessing of spatial data has lagged behind. Hence, there is a need for preprocessing techniques for spatial data to deal with problems such as treatment of missing location information and imprecise location specifications, cleaning of spatial data, feature selection, and data transformation.

Unique features of Spatial Data Mining

The unique features that distinguish spatial data mining from classical data mining in the following four categories:

- **Data input:** The data inputs of spatial data mining are more complex than the inputs of classical data mining because they include extended objects such as points, lines, and polygons. The data inputs of spatial data mining have two distinct types of attributes: non-spatial attribute and spatial attribute. Non-spatial attributes are used to characterize non-spatial features of objects, such as name, population, and unemployment rate for a city. They are the same as the attributes used in the data inputs of classical data mining. Spatial attributes are used to define the spatial location and extent of spatial objects. The spatial attributes of a spatial object most often include information related to spatial locations, e.g., longitude, latitude and elevation, as well as shape.

  Relationships among non-spatial objects are explicit in data inputs, e.g., arithmetic relation, ordering, is instance of, subclass of, and membership of. In contrast, relationships among spatial objects are often implicit, such as overlap, intersect, and behind. One possible way to deal with implicit spatial relationships is to materialize the relationships into traditional data input columns and then apply classical data mining techniques. However, the materialization can result in loss of information. Another way to capture implicit spatial relationships is to develop models or techniques to incorporate spatial information into the spatial data mining process.

Non-spatial Relationship (Explicit)	**Spatial Relationship (Often Implicit)**
Arithmetic	Set-oriented: union, intersection, membership, …
Ordering	Topological: meet, within, overlap, …
Is_instance_of	Directional: North, NE, above, left, behind, …
Subclass_of	Metric: e.g., distance, area, destroy, …
Part_of	Dynamic: update, create, destroy, …
Membership_of	Shape-based and visibility

Table 1: Relationships among Non-spatial data and Spatial data

- **Statistical foundation**: Statistical models are often used to represent observations in terms of random variables. These models can then be used for estimation, description, and prediction based on probability theory. Spatial data can be thought of as resulting from observations on the stochastic process Z(s): s 2 D, where s is a spatial location and D is possibly a random set in a spatial framework. Presented below are three spatial statistical problems one might encounter: point process, lattice, and geo-statistics.

  - **Point Process:** A point process is a model for the spatial distribution of the points in a point pattern. Several natural processes can be modeled as spatial point patterns, e.g., positions of trees in a forest and locations of bird habitats in a wetland. Spatial point patterns can be broadly grouped into random or non-random processes. Real point patterns are often compared with a random pattern (generated by a Poisson process) using the average distance between a point and its nearest neighbor. For a random pattern, this average distance is expected to be density, where density is the average number of points per unit area. If for a real process, the computed distance falls within a certain limit, then we conclude that the pattern is generated by a random process; otherwise it is a non-random process.
  - **Lattice**: A lattice is a model for a gridded space in a spatial framework. Here the lattice refers to a countable collection of regular or irregular spatial sites related to each other via a neighborhood relationship. Several spatial statistical analyses, e.g., the spatial autoregressive model and Markov random fields, can be applied on lattice data.
  - **Geostatistics**: Geostatistics deals with the analysis of spatial continuity and weak stationarity, which is an inherent characteristics of spatial data sets. Geostatistics provides a set of statistics tools, such as kriging to the interpolation of attributes at unsampled locations.

One of the fundamental assumptions of statistical analysis is that the data samples are independently generated: like successive tosses of coin, or the rolling of a die. However, in the analysis of spatial data, the assumption about the independence of samples is generally false. In fact, spatial data tends to be highly self-correlated. For example, people with similar characteristics, occupation and background tend to cluster together in the same neighborhoods. The economies of a region tend to be similar. Changes in natural resources, wildlife, and temperature vary gradually over space. The 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: “Everything is related to everything else but nearby things are more related than distant things". In spatial statistics, an area within statistics devoted to the analysis of spatial data, this property is called spatial autocorrelation. In spatial statistics, spatial autocorrelation is quantified using measures such as Ripley's K-function and Moran's I. Another distinct property of spatial data is called *spatial hetetrogeneity* which implies that the variation in spatial data is a function of its location. Spatial heterogeneity is measured via local measures of spatial autocorrelation.

- **Output patterns:** Four important output patterns for spatial data mining: predictive models, spatial outliers, spatial co-location rules, and spatial clustering.

  - Predictive Models
  - Spatial outliers

- Spatial co-location rules
- Spatial clustering

- **Computational process:** Many generic algorithmic strategies have been generalized to apply to spatial data mining. For example, algorithmic strategies, such as divide-and-conquer, filter-and-refine, ordering, hierarchical structure, and parameter estimation, have been used in spatial data mining.

Generic	Spatial Data Mining
Divide-and-Conquer	Space Partitioning
Filter-and-Refine	Minimum-Bounding-Rectangle (MBR)
Ordering	Plane Sweeping, Space Filling Curves
Hierarchical Structures	Spatial Index, Tree Matching
Parameter Estimation	Parameter estimation with spatial autocorrelation

Table 2: Algorithmic Strategies in Spatial Data Mining

In spatial data mining, spatial autocorrelation and low dimensionality in space provide more opportunities to improve computational efficiency than classical data mining. spatial indexing approach exploits spatial autocorrelation to facilitate correlation-based queries. The approach groups similar time series together based on spatial proximity and constructs a search tree. The queries are processed using the search tree in a filter-and-refine style at the group level instead of at the time series level. Algebraic analyses using cost models and experimental evaluations showed that the proposed approach saves a large portion of computational cost, ranging from 40% to 98%.

The systematic structure of spatial data mining

The spatial data mining can be used to understand spatial data, discover the relation between space and the non- space data, set up the spatial knowledge base, excel the query, reorganize spatial database and obtain concise total characteristic etc. The system structure of the spatial data mining can be divided into three layer structures mostly, such as the Figure 1 show .The customer interface layer is mainly used for input and output, the miner layer is mainly used to manage data, select algorithm and storage the mined knowledge, the data source layer, which mainly includes the spatial database and other related data and knowledge bases, is original data of the spatial data mining.

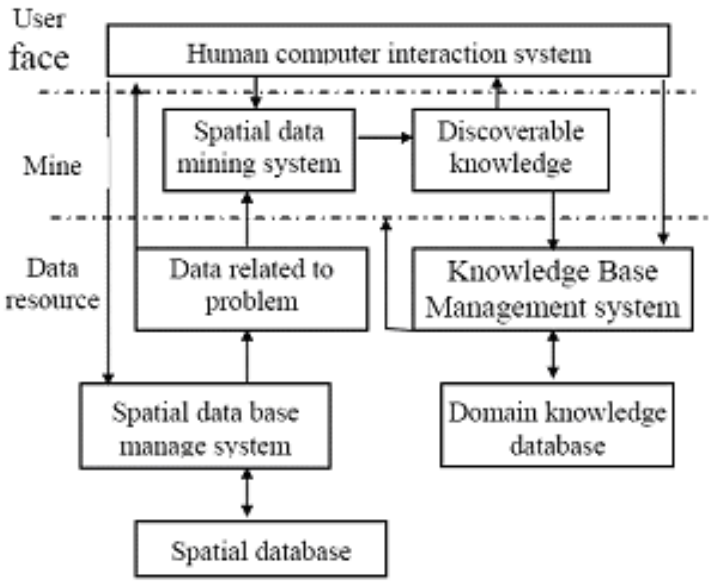


Figure 1: Systematic structure of spatial data mining

Spatial Data Mining: Exploratory Data Analysis Techniques

Spatial data analysis comprises a broad spectrum of techniques that deals with both spatial and non-spatial characteristics of spatial objects. Exploratory techniques allow to investigate first and second order effects of the data. First order variation informs about large scale global trend of phenomena which is spatially distributed through the area. The second order variation defines dependence between observations.

- **Global Autocorrelation (Moran's I):** Moran's I is a measure of global spatial autocorrelation. Global or local autocorrelation reveal feature similarity based location and attribute values to explore the pattern whether it is clustered, dispersed, or random.
- **Hot Spots (Getis-Ord):** The G-statistic is often used to identify whether hot spots or cold spots exist based on so-called distance statistics. Hot spots are regions that stand out compared with the overall behaviour prevalent in the space. Hot spots can be detected by visualizing the distribution in format of choropleth or isarithmic maps.
- **Local Autocorrelation (Anselin's Local Moran I):** The local Moran statistic is used as a local indicator of spatial association which is calculated for individual zones around each observation within a defined neighbourhood to identify similar or different pattern in nearby. Because the distribution of the statistic is not known, high positive or high negative standardized scores of *Ii* are taken as indicators of similarity or dissimilarity respectively.
- **Density (Kernel):** Kernel density estimation is a nonparametric unsupervised learning procedure (classifier). Kernel, *k* is bivarate probability density function which is symmetric around the origin.

Spatial data mining tasks

Basic tasks of spatial data mining are:

- ***classification*** – finds a set of rules which determine the class of the classified object according to its attributes e. g. ”IF population of city = high AND economic power of city = high THEN unemployment of city = low” or classification of a pixel into one of classes, e. g. water, field, forest.
- ***association rules*** – find (spatially related) rules from the database. Association rules describe patterns, which are often in the database. The association rule has the following form: A → B(s%; c%), where s is the support of the rule (the probability, that A and B hold together in all the possible cases) and c is the confidence (the conditional probability that B is true under the condition of A e. g. ”if the city is large, it is near the river (with probability 80%)” or ”if the neighboring pixels are classified as water, then central pixel is water (probability 80%).”
- ***characteristic rules*** – describe some part of database e. g. ”bridge is an object in the place where a road crosses a river.”
- ***discriminant rules*** – describe differences between two parts of database e. g. find differences between cities with high and low unemployment rate.
- ***clustering*** – groups the object from database into clusters in such a way that object in one cluster are similar and objects from different clusters are dissimilar e. g. we can find clusters of cities with similar level of unemployment or we can cluster pixels into similarity classes based on spectral characteristics.
- ***trend detection*** – finds trends in database. A trend is a temporal pattern in some time series data. A spatial trend is defined as a pattern of change of a non-spatial attribute in the neighborhood of a spatial object e. g. ”when moving away from Brno, the unemployment rate increases” or we can find changes of pixel classification of a given area in the last five years.

Spatial Data Mining: Clustering-Finding Patterns

Assigning each member of a large data set into homogeneous clusters is a fundamental operation in data mining. Clustering is the process of creating a group of data organized on some similarity among the members of a dataset. Each cluster consists of members that are similar in-between the cluster, however dissimilar to members of other clusters. There are four major clustering approaches. Partitional clustering algorithms construct *k* clusters - usually defined in advance by the user- due to an evaluation criterion. Hierarchical clustering algorithms create a hierarchical decomposition using some criterion which can be represented as dendograms. Density-based partitioning algorithms search for regions which are denser than a given threshold to form clusters from these dense regions by using connectivity and density functions. Grid-based algorithms are based on a multiple-level granularity by quantizing the search space into a finite number of cells.

- **k-Means :** K-means represents an attempt to define an optimal number of k locations where the sum of the distance from every point to each of the k centers is minimized what is called global optimization. In practice, (1) making initial

guesses about the k locations and (2) local optimization for cluster locations in relation to the nearby points is implemented. Thus, two k-means procedures might not produce the same results, even if k is identical because of several underlying local optimization methods.

The k-means algorithm is built upon four basic operations:

- selection of the initial k means for k clusters,
- calculation of the dissimilarity between an object and the mean of a cluster,
- allocation of an object to the cluster whose mean is nearest to the object,
- Re-calculation of the mean of a cluster from the objects allocated to it so that the intra cluster dissimilarity is minimised. Except for the first operation, the other three operations are repeatedly performed in the algorithm until the algorithm converges (until no points change clusters). The essence of the algorithm is to minimise the cost function which is a function of dissimilarity measure between each observation with mean of cluster. Dissimilarity is usually modelled as Euclidean Distance in k-means. The cost function is as follows;

$$Minimize \sum_{j=1}^{n}\sum_{k=1}^{k} a_j d_{jk} z_{jk}$$

where;

j , k denotes total number of observations and clusters
a_j denotes weight of observation j ,
d_{jk} denotes distance between observation j and centre of cluster k , and

$$z_{jk} = \begin{cases} 1, & \textbf{\textit{if the observation j is in cluster k,}} \\ 0 & \textbf{\textit{otherwise}} \end{cases}$$

z_{jk} is an indicator of belonging of an observation to a cluster, providing that z_{jk} takes value *1* only once for assigned cluster, and value *0* for other clusters.

ISODATA (Iterative Self-Organizing Data Analysis Techniques) algorithm:

As being a variety of k-means, the ISODATA clustering method uses the minimum feature distance to form clusters to identify statistical patterns in the data. It begins with either arbitrary cluster means or means of an existing signature set, and each time the clustering repeats, the means of these clusters are shifted. The new cluster means are used for the next iteration. The clustering of the data until either a maximum number of iterations is performed, or a maximum percentage of unchanged pixel assignments have been reached between two iterations. The optimal number of classes to specify is usually unknown. The resulting values of centers and their covariance matrix are used to do supervised classification. In this step any of classification methods such as decision trees (dendograms) or ML is used.

In case of no solid vulnerability model, clustering gives initial knowledge on distribution of factors/dimensions and estimates of the representative of the class centers. Then, the resultant clusters can be associated with vulnerability classes if prior knowledge is existing such as a risk or vulnerability map.

Generalized Density-Based Clustering

Clustering is the task of grouping the objects of a database into meaningful subclasses (that is, clusters) so that the members of a cluster are as similar as possible whereas the members of different clusters differ as much as possible from each other. Applications of clustering in spatial databases are, e.g., the detection of seismic faults by grouping the entries of an earthquake catalog or the creation of thematic maps in geographic information systems by clustering feature vectors. The clustering algorithms can be supported by the database primitives if the clustering algorithm is based on a "local" cluster condition, i.e. if it constructs clusters by analyzing a restricted neighbourhood of the objects. Examples are the density-based clustering algorithm DBSCAN as well as its generalized version GDBSCAN which is discussed in the following.

GDBSCAN (*Generalized Density Based Spatial Clustering of Applications with Noise*) relies on a density-based notion of clusters. The key idea of a density-based cluster is that for each point of a cluster its *Eps*-neighbourhood for some given *Eps* > 0 has to contain at least a minimum number of points, i.e. the "density" in the *Eps*-neighbourhood of points has to exceed some threshold. "Density-based clusters" can be generalized to density-connected sets in the following way:

First, any notion of a neighbourhood can be used instead of an *Eps*-neighbourhood if the definition of the neighbourhood is based on a binary predicate *NPred* which is symmetric and reflexive. Second, instead of simply counting the objects in a neighbourhood of an object, other measures to define an eqivalent of the "cardinality" of that neighbourhood can be used as well. For that purpose we assume a predicate *MinWeight* which is defined for sets of objects and which is *true* for a neighbourhood if the neighbourhood has the minimum weight (e.g. a minimum cardinality as for density- based clusters).

Whereas a distance-based neighbourhood is a natural notion of a neighbourhood for point objects, it may be more appropriate to use topological relations such as *intersects* or *meets* to cluster spatially extended objects such as a set of polygons of largely differing sizes. There are also specializations equivalent to simple forms of region growing, i.e. only local criteria for expanding a region can be defined by the weighted cardinality function. For instance, the neighbourhood may be given simply by the neighbouring cells in a grid and the weighted cardinality function may be some aggregation of the non-spatial attribute values. While region growing algorithms are highly specialized to pixels, density-connected sets can be defined for any data types.

The neighbourhood predicate *NPred* and the *MinWeight* predicate for these specializations are listed below and illustrated in figure 2 for a more detailed discussion of neighbourhood relations for different applications):

- Density-based clusters: *NPred*: "distanceδ *Eps*", *wCard*: cardinality, *MinWeight*(*N*): $|N|\varepsilon$ *MinPts*

- Clustering of polygons: *NPred*: "*intersects*" or "*meets*", *wCard*: sum of areas, *MinWeight*(*N*): sum of areas ε *MinArea*
- Simple region growing: *NPred*: "*neighbour and similar non-spatial attributes*",

MinWeight(*N*): *aggr*(non-spatial values) ε *threshold*.

Figure 2: Different specialization of density-connected sets.

Spatial Association Rules

Similar to the mining of association rules in transactional and relational databases, spatial association rules can be mined in spatial databases. A spatial association rule is of the form $A \Rightarrow B$ [*s%, c%*] , where A and B are sets of spatial or non spatial predictors, *s*% is the support of the rule and *c*% is the confidence of the rule. For example, the following is a spatial association rule:

$$is\_a(X, \text{“school”}) \wedge close\_to(X, \text{“sports\_center”}) \Rightarrow close\_to(X, \text{“park”}) \quad [0.5\%, 80\%]$$

This rule states that 80% of schools that are close to sports centers are also close to parks, and 0.5 % of the data belongs to such a case.

Various kinds of spatial predicates can be constitute a spatial association rule. Examples include distance information (such as *close_to* or *far_away*), topological relations (like *intersect*, *overlap*, and *disjoint*), and spatial orientation (like *left_of* and *west_of*).

Since spatial association mining needs to evaluate multiple spatial relationships among a large number of spatial objects, the process could be quite costly. An interesting mining optimization method called progressive refinement can be adopted in spatial association analysis. The method first mines large datasets roughly using a fast algorithm and then improves the quality of mining in a pruned data set using a more expensive algorithm.

To ensure that the pruned dataset covers the complete set of answers when applying the high quality data mining algorithms at later stage is the superset coverage property: that is, it preserves all of the potential answers. In other words, it should allow a *false_positive test*, which might include some data sets that do not belong to the answer sets, but it should not allow a *false-negative test* which might include some datasets that do not belong to the answers sets, but it should not allow a *false-negative test*, which might exclude some potential answers.

For mining spatial associations related to the spatial predicate *close_to*, we can first collect the candidates that pass the minimum support threshold by

- Applying certain rough spatial evaluation algorithms, for example, using an MBR structure (which registers only two spatial points rater than a set of complex polygons), and
- Evaluating the relaxed spatial predicate, *g_close_to*, which is a generalized *close_to* covering a broader context that includes *close_to, touch,* and *intersect*.

If two spatial objects are closely located, their enclosing MBRs must be closely located matching *g_close_to*. However, the reverse is not always true: if the enclosing MBRs are closely located, the two spatial objects may or may not be located so closely. Thus, the MBR pruning is a false-positive testing tool for closeness: only those that pass the rough test need to be further examined using more expensive spatial computation algorithms. With this preprocessing, only the patters that are frequent at the approximation level will need to be examined by more detailed and finer, yet expensive, spatial computation.

Besides mining spatial association rules, one may like to identify groups of particular features that appear frequently close to each other in a geospatial map. Such a problem is essentially the problem of mining **spatial co-locations**. Finding spatial co-locations can be considered as special case of mining spatial associations. However, based on the property of spatial autocorrelation, interesting features likely coexist in closely located regions. Thus spatial co-location can be just what one really wants to explore. Efficient methods can be developed for mining spatial co-locations by exploring the methodologies like Apriori and progressive refinement, similar to what has been done for missing spatial association rules.

Spatial Trend Detection

A *spatial trend* is defined as a regular change of one or more non-spatial attributes when moving away from a given start object *o* (Figure 3). Neighbourhood paths are used starting from *o* to model the movement and we perform a regression analysis on the respective attribute values for the objects of a neighbourhood path to describe the regularity of change.

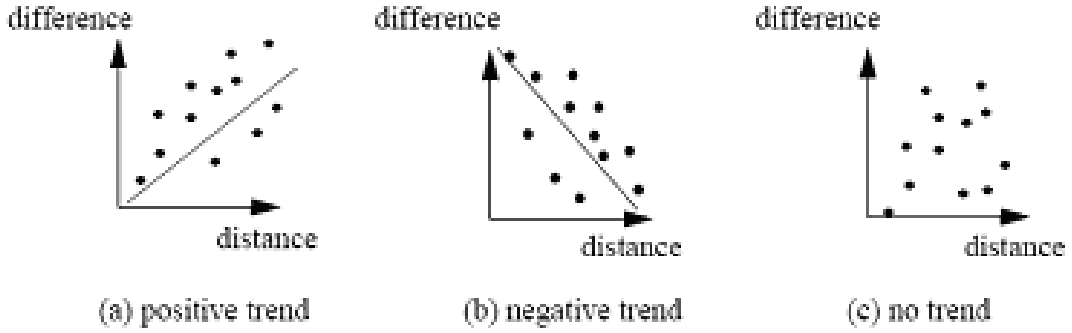


Figure 3: Sample linear trends

- **Algorithms**

First, we define the task of spatial trend detection for some source object $o_1 \in DB$. To detect regular changes of some non-spatial attributes, we perform a regression analysis as follows. The independent variable (X) yields the distance between any database object o_2 and the source object. The dependent variable (Y) measures the difference of the values of some non-spatial attribute(s) for o_1 and o_2. Then, the sets X and Y contain one observation for each element of a subset S of DB.

If the absolute value of the correlation coefficient is found to be large enough, S identifies a part of DB showing a significant spatial trend for the specified attributes(s) starting from o_1. In the following, we will use linear regression, since it is efficient and often the influence of some phenomenon to its neighbourhood is either linear or may be transformed into a linear model, e.g. exponential regression. Figure 3 illustrates a strong positive (correlation coefficient >> 0) and a strong negative (correlation coefficient << 0) linear trend as well as a situation where no significant (linear) trend is observed.

In a naive approach, one observation could be made for each element $o_2 \in DB$ and a regression analysis would be conducted once for this whole set of observations. This approach, however, would fail to discover a spatial trend in the following situations:

- a trend is present only locally in the neighbourhood but not globally in the database
- a trend is present not in all directions but only in some directions.

We argue that such situations are very common in real databases. We use the concept of neighbourhood paths to overcome these problems and consider only such objects o_2 which are located on one of the neighbourhood paths starting from o_1. We stop extending a neighbourhood path as soon as no more significant trend can be found and, thus, provide a means to restrict the search to the neighbourhood of the source object. Furthermore, it is enough that some, not all, neighbourhood paths show a spatial trend of the specified type.

Definition 1: (***spatial trend detection***): Let *g* be a neighbourhood graph, *o* an object (node) in *g* and *a* be a subset of all non-spatial attributes. Let *t* be a type of function, e.g. linear or exponential, used for the regression and let *filter* be one of the filters for neighbourhood paths. Let *min-conf* be a real number and let *min-length* as well as *max-length* be natural numbers. The task of *spatial trend detection* is to discover the set of all neighbourhood paths in g starting from *o* and having a trend of type *t* in attributes *a* with a correlation of at least *min-conf*. The paths have to satisfy the *filter* and their length must be between *min-length* and *max-length*.

Definition 1 allows different specializations. Either the set of all discovered neighbourhood paths or each of its elements must have a trend of the specified type. For each of these specializations, we present an algorithm to discover such spatial trends. Both algorithms require the same input parameters but they use different methods to search the set of all relevant neighbourhood paths.

The first algorithm discovers *global trends*, i.e. trends for the whole set of all neighbourhood paths with source *o* having a length in the specified interval. Algorithm *detect-global-trends* performs a breadth-first search of the set of all neighbourhood paths starting from *o*.

Beginning from *o*, the algorithm creates all neighbourhood paths of the same length simultaneaously - starting with *min-length* and continuing until *max-length*. The regression is performed once for each of these sets of all paths of the same length. If no trend of length *l* with abs(correlation) ε *min-conf* is detected, then the path extensions of length *l+1*, *l+2*, . . ., *max-length* are not created due to efficiency reasons. The parameter *min-length* has to be chosen large enough because a trend may become significant only at a certain minimum length of a neighbourhood path. The algorithm returns the set of neighbourhood paths of the maximum length having a trend with abs(correlation) ε *min-conf*. Furthermore, the slope of the regression function and its correlation coefficient are returned. The slope describes the degree of change of the specified attribute when moving away from the source object *o*: a large slope is found for a quickly changing attribute whereas a small slope indicates that the attribute changes only slowly in the neighbourhood of *o*. The correlation coefficient measures the confidence of the discovered trend - the larger the correlation coefficient the better can the observations be approximated by the regression function and the higher is the confidence of the discovered trend.

Spatial Classification

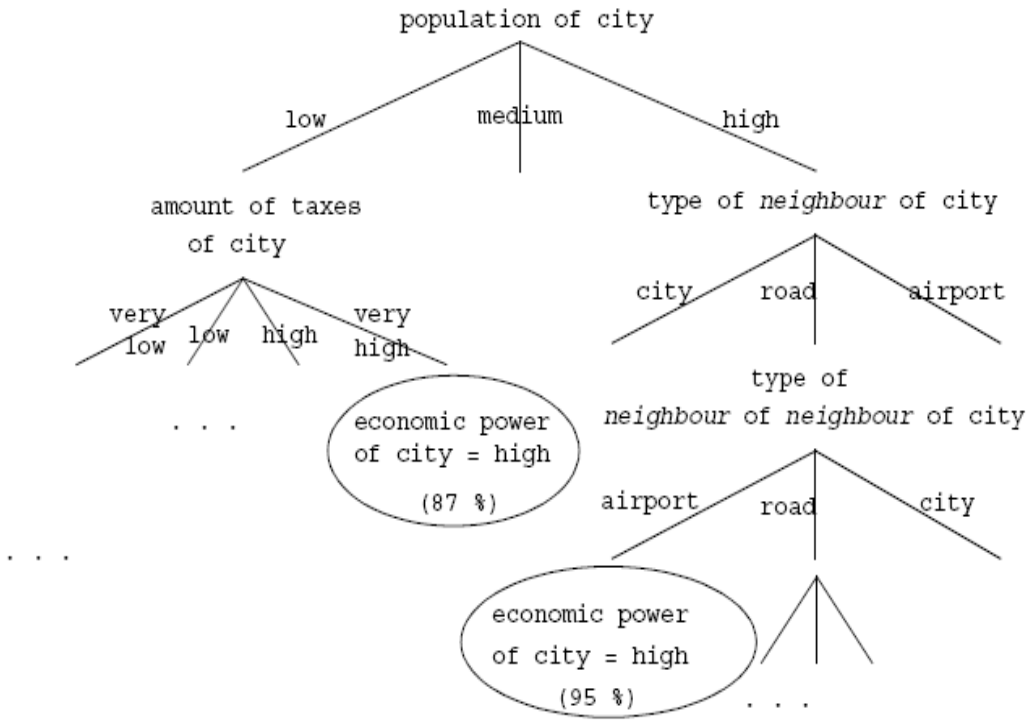
The task of *classification* is to assign an object to a class from a given set of classes based on the attribute values of the object. In *spatial classification* the attribute values of neighbouring objects may also be relevant for the membership of objects and therefore have to be considered as well.

Given a set of data (a training set) with one attribute as the dependent attribute, the classification task is to build a model to predict the unknown dependent attributes of future data based on other attributes as accurately as possible. A set of data (different from the training set) with dependent attributes known in advance is used to validate the model. In spatial classification, the attribute properties of neighbouring objects may also have an effect on the membership of objects.

The goal of spatial classification is to build a model for predicting the classes. Typically the model is built using a portion of the data, called the Learning or Training data, and then tested on the remainder of the data, called the Testing data. In the learning data, all the attributes are used to build the model and in the testing data, one value is hidden.

The extension to spatial attributes is to consider also the attribute of objects on a neighbourhood path starting from the current object. Thus, we define *generalized attributes* for a neighbourhood path $p = [o_1, \ldots, o_k]$ as tuples (attribute-name, index) where index is a valid position in *p* representing the attribute with attribute-name of object o_{index}. The generalized attribute (economic-power,2), e.g., represents the attribute economic-power of some (direct) neighbour of object o_1.

Because it is reasonable to assume that the influence of neighbouring objects and their attributes decreases with increasing distance, we can limit the length of the relevant neighbourhood paths by an input parameter max-length. Furthermore, the classification algorithm allows the input of a predicate to focus the search for classification rules on the objects of the database fulfilling this predicate. Figure 4 depicts a sample decision tree and two rules derived from it. Economic power has been chosen as the class attribute and the focus is on all objects of type city.



IF population of city = low AND amount of taxes of city = very high
THEN economic power of city = high (87 %)

IF population of city = high AND type of *neighbour* of city = road
AND type of *neighbour* of *neighbour* of city = airport
THEN economic power of city = high (95 %)

Figure 4: Sample decision tree and rules discovered by classification algorithm

Another algorithm for spatial classification works as follows:

The relevant attributes are extracted by comparing the attribute values of the target objects with the attribute values of their nearest neighbours. The determination of relevant attributes is based on the concept of the *nearest hit* (the nearest neighbour belonging to the same class) and the *nearest miss* (the nearest neighbour belonging to a different class). In the construction of the decision tree, the neighbours of target objects are not considered individually. Instead, so-called *buffers* are created around the target objects and the non-spatial attribute values are aggregated over all objects contained in the buffer. For instance, in the case of shopping malls a buffer may represent the area where its customers live or work. The size of the buffer yielding the maximum information gain is chosen and this size is applied to compute the aggregates for all relevant attributes.

Whereas the property of being a nearest neighbour cannot be directly expressed by our neighbourhood relations, it is possible to extend the set of neighbourhood relations accordingly. The proposed database primitives are, however, sufficient to express the creation of buffers for spatial classification by using a distance-based neigborhood predicate.

- **Spatial data classification: interest measures**

The interest measures of patterns in spatial data mining are different from those in classical data mining, especially regarding the four important output patterns. For a two-class problem, the standard way to measure classification accuracy is to calculate the percentage of correctly classified objects. However, this measure may not be the most suitable in a spatial context. *Spatial accuracy*- how far the predictions are from the actuals- is equally important in certain application domains. However, sometime the classification accuracy measure cannot distinguish between two classes since spatial accuracy is not incorporated in the classification accuracy measure. Hence, there is a need to investigate proper measures to improve spatial accuracy.

Conclusion

The spatial data mining is newly arisen area when computer technique, database applied technique and management decision support techniques etc. have been developed at certain stage. The spatial data mining gathered productions that come from machine learning, pattern recognition, database, statistics, artificial intelligence and management information system etc. Different theories, put forward the different methods of spatial data mining, such as methods in statistics, proof theories, rule inductive, association rules, cluster analysis, spatial analysis, fuzzy sets, cloud theories, rough sets, neural network, decision tree and spatial data mining technique based on information entropy etc. Spatial data mining, has established itself as a complete and potential area of research. This article tries to explain spatial data mining as well the its different tasks. It also explains how we explain a particular task in relation to the spatial data. The challenge is to take up spatial data mining as an important area and work on it using the various domains such as statistics, artificial intelligence, machine learning and geography etc.