ISSN No. 0976-5697

Volume 2, No. 5, Sept-Oct 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Preamble to Arrangement of Data in Spatial Data Mining

A.Srinivasa Reddy
Assoc Professor
Dept of IT, QISCET,
Ongole,India
Srinivas.asr@gmail.com

Abstract: - 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. This report deals with spatial data structures for indexing and with their usability for knowledge discovery in spatial data. Huge amount of data processed in spatial data mining (or in data mining generally) requires using some indexing structures to speed up the mining process. Typical data types and operations used in geographic information systems are described in this paper.

Keywords: **-** Spatial, Vector model, fuzzy, Tree, KDD, Mining

I. INTRODUCTION

The main difference between data mining in relational DBS and in spatial DBS is that attributes of the neighbors of some object of interest may have an influence on the object and therefore have to be considered as well. The explicit location and extension of spatial objects define implicit relations of spatial neighborhood (such as topological, distance and direction relations) [4,5] which are used by spatial data mining algorithms. Therefore, new techniques are required for effective and efficient data mining. A geographic information system is a special kind of information system, which allows manipulate, analyze, summarize, query, edit and visualize geographically related data. Geographically related data are composed of:

a. spatial attributes, e. g. coordinates, geometry
b. non-spatial attributes, e. g. name of town, number of inhabitants

Spatial data in GIS may be represented in raster or vector model. *Raster model* divides space into a regular grid of cells, usually called pixels. Each cell contains a single value and its position is defined by its indices in the grid. The resolution of the raster depends on its pixel size. The smaller the pixel size, the higher the resolution, but also the larger the data size. *Vector model* represents spatial objects with data Structures, whose basic primitive is a point, this allows precise representation of coordinates and it's useful for analysis. Data structures used to store spatial objects in the vector model are:

a. ***point*** **–** defined by its coordinates
b. ***chain*** **–** sequence of points connected with lines
c. ***polygon*** **–** sequence of connected chains

Fundamental operations used to manipulate vector data are:

a. determining the distance of two objects
b. determining the area of the object (if it is a polygon)
c. determining the length of the object (if it is a chain or polygon)
d. determining an intersection or union of the objects
e. determining a mutual position of two objects (Intersect, overlap, touch, one can contain the other etc.)

II. SPATIAL DATA MINING

Analysis is an important part of GIS which allows spatial operations [5,7] with data (e. g. network analysis or filtering of raster data), measuring functions (e.g. distance, direction between objects), statistic analyses or terrain model analysis (e. g. visibility analysis). 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.

A. Spatial data mining tasks:

Basic tasks of spatial data mining are:

a) ***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.

b) ***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%).”

c) ***Characteristic rules*** **–** describe some part of database e. g. “bridge is an object in the place where a road crosses a river.”

d) ***Discriminant rules*** **–** describe differences between two parts of database e. g. find differences between cities with high and low unemployment rate.

e) ***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.

f) ***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.

B. Spatial data mining systems:

a. GeoMiner:

The GeoMiner is a system for knowledge discovery in large spatial databases. It was developed at Simon Fraser University in Canada. The GeoMiner is an extension of and developed from DBMiner . The DBMiner is a relational data mining system, which uses Microsoft SQL Server 7.0 to store data. It contains the five following data mining modules: *Association*, *Classification*, *Clustering*, *3D Cube Explorer* (displays data cube in a 3D view) and *OLAP Browser* (generalizes data in a spreadsheet or graphical form). The Geominer is composed of the following modules: *Geo–characterizer*, *Geo–associator*, *Geo–cluster analyzer*, *Geo–classifier* (their functionis similar to the previous definition) and *Geo–comparator* (its function corresponds to the discriminant rules in the previous definition). The system contains its own language for knowledge discovery in spatial data and it uses graphical interface to communicate with the user and display results in the form of graphs, charts, maps, etc.

b. Descartes:

System Descartes supports the visual analysis of spatially referenced data. It uses two basic tools: automatic visualization (presentation of the data on the map) and interactive manipulation with maps. The system uses the following methods to visualize information: *area coloration*, *charts* and combination of both of them. The area coloration represents a numeric attribute as color: the greater the value, the darker the color of the region. Descartes offers various types of charts (bar, pie, etc.). The combination of both the methods allows visualizing one attribute with area coloration and another attribute (or attributes) with charts. In the integration of Descartes with Kepler is used to classify spatial related data. Kepler is a knowledge discovery system with a plug in architecture.

c. Fuzzy Spatial OQL for Fuzzy KDD:

A fuzzy spatial object query language [6, 9] and fuzzy decision tree are introduced. This language is designed to select process and mine data from Spatial Object–Oriented Databases and it is based on a fuzzy set theory. In the classical theory, given a set S and an element e, we can decide whether this element belongs or does not belong to S. In the fuzzy set theory, the probability that e belongs to S can vary from 0 to 1. In figure 1, a numerical attribute value can have the fuzzy values of Low, Medium and High. In the fuzzy spatial OQL, fuzzy values are used in the where clause of a fuzzy query and the answer is a fuzzy set of elements defined by these fuzzy values. In the fuzzy decision tree, each node is associated with a test on the values of some attribute. All the edges of the node are labeled by fuzzy values. This enhances the comprehensibility of the decision tree.

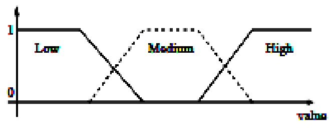

Figure 1: Fuzzy values for numerical attribute

III. SPATIAL DATA STRUCTURES IN GIS

A. Quad tree:

The quad tree [2] is used to index 2D space. Each internal node of the tree splits the space into four disjunct subspaces (called NW, NE, SW, SE) according to the axes. Each of these subspaces is split recursively until there is at most one object inside each of them (see Figure 2). The quad tree is not balanced and its balance depends on the data distribution and the order of inserting the points.

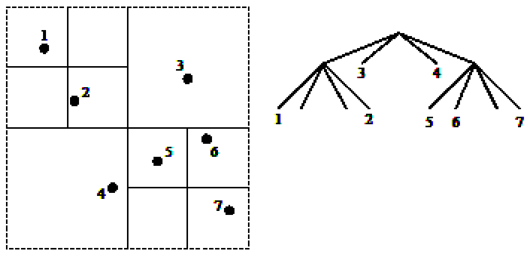

Figure 2: Quad tree

B. k–d–tree:

This method uses a binary tree to split k–dimensional space [1,8]. This tree splits the space into two subspaces according to one of the coordinates of the splitting point (see Figure 3). Let level(nod) be the length of the path from the root to the node nod and suppose the axes are numbered from 0 to k − 1. At the level level(nod) in every node the space is split according to the coordinate number (level(nod) mod k). Inserting and searching are similar to the binary trees. We only have to compare nodes according to the coordinate number (level(nod) mod k). This structure has one disadvantage: it is sensitive to the order in which the objects are inserted.

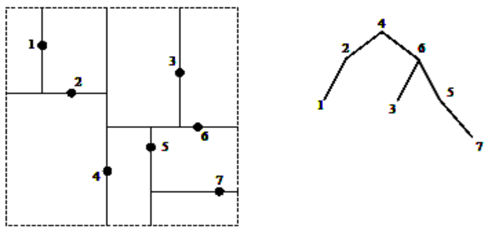

Figure 3: k–d–tree

C. R–tree:

The R–tree [1] is the modification of B-tree for spatial data. This tree is balanced and splits the space into the rectangles which can overlap. Each node except root contains from m to M children, where 2<=m<=M/2. The root contains at least 2 children unless it is a leaf. Figure 4 shows an example of r-tree of the order 3. The node is represented by the minimum bounding rectangle containing all the objects of its sub tree. Each of children of the node is split recursively. Pointers to the data objects are stored in the leafs. Because of the overlapping of bounding rectangles it could be necessary to search more than one branch of the tree. Therefore, it is important to separate the rectangles as much as possible.

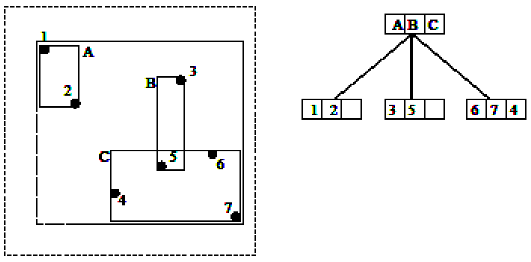

Figure 4: R–tree

This problem must be solved by the operation INSERT which uses some kind of heuristic. It finds such a leaf that inserting a new object into it will cause as small changes in the tree as possible. The splitting operation is also important. We want to decrease the probability that we will have to search both new nodes. The R-tree is one of the most cited spatial data structures and it is very often used for comparison with new structures.

IV. DATA STRUCTURES FOR METRIC SPACES

A. *Vp–tree:*

The vp–tree (or vantage point tree) partitions the data space around selected points and forms a hierarchical tree structure (see Figure 5). These selected points are called vantage points.

Each internal node of the tree is of the form (Pv,M,R,L) where:

a. Pv is the vantage point
b. M is the median distance among the distances of all the points (belonging to the subtree of the node) from Pv
c. R, L are the pointers to the sons of the node.
   Left (right) son contains the points whose distances from Pv are less than or equal (greater than or equal) to M.

The leafs contain pointers to the data points.

The vp-tree is used to find the objects whose distance from Q is less than or equal to r:

i. If d(Q,Pv) <= r, then Pv is in the answer set.
ii. If d(Q, Pv) + r >= M, then recursively search the right subtree.
iii. If d(Q, Pv) − r <= M, then recursively search the left subtree.

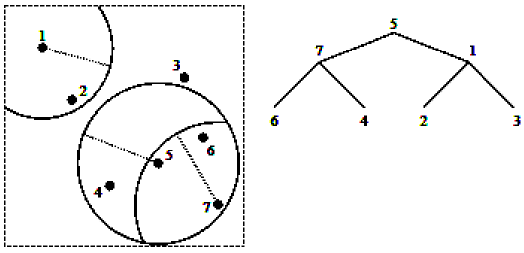

Figure 5: vp–tree

B. *M–way vp–tree:*

M–way vp–tree (or multi–way vp–tree) [5] is one of the modifications of the vp-tree which decreases the height of the tree. The structure of m–way vp-tree of order m is very similar to the vp-tree. The main difference is that it splits objects into m groups according to their distances from the vantage point. The splitting values, called *cutoff* values, are stored in a node. Figure 6 shows an example of m-way vp-tree of order 3.

The construction of the tree requires O(nlogmn) distance computations. That is log2m times better than binary vp-trees. M–way vp–tree has one disadvantage if it partitions high–dimensional spaces. In this case, spherical cuts of one node are very thin and search operation will have to search more than one branch very often.

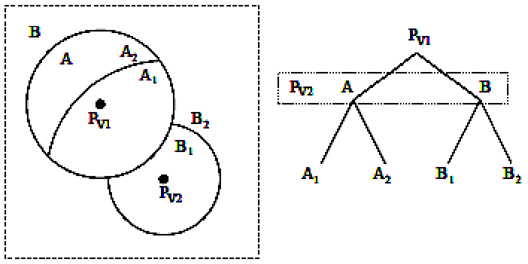

Figure 6: One node of mvp–tree

V. DATA STRUCTURES IN SPATIAL DATA STRUCTURES

A. *Neighborhood graphs:*

a. *Neighborhood graphs and neighborhood paths:*

Definition: *Neighborhood graph* G for spatial relation neighbor=2 is a graph G(U,H) where U is a set of nodes and H is a set of edges. Each node represents an object and two nodes N1, N2 are connected by edge iff the objects corresponding to N1 and N2 are in the relation neighbor.

The relation neighbor can be:

i. *topological relation*, e. g. two objects touch, cover, are equal
ii. *metric relation*, e. g. distance of the objects is less than d
iii. direction relation, e.g. north, south, east, west
iv. any conjunction or disjunction of previous relations

Neighborhood graph is oriented. Thus it can happen that object A is a neighbor of the object B but object B is not a neighbor of the object A.

Definition: *Neighborhood path* for the neighborhood graph G is an ordered list of nodes from G where every two following nodes from the path are connected by some edge from G, i. e. for the path [n0, n1,...., n_{k-1}] there must be edges (n_i,n_{i+1}) for every 0<= i < k−1. *Length of the path* is a sum of edges in the path.

b. *Elementary operations on the neighborhood graphs:*

Elementary operations on the neighborhood graphs are:

a) ***Get_Graph(Data, Neighbor)*** – returns the neighborhood graph G representing the relation neighbor on the objects from the table data. The relation neighbor can be one of the spatial relations listed in the definitionof the neighborhood graph.

b) ***Get_Neighborhood(G, o, Pred)*** – returns the set of the objects connected to the object o by some of the edges from the graph G. The predicate pred must hold for these objects. This condition is used if we want to get only some specific neighbors of the object o. The predicate pred may not necessarily be spatial.

c) ***Create_Path(G, Pred, i)*** – returns the set of all paths which consist of the nodes and edges from the graph G, their length is less than or equal to i and the predicate pred holds for them. Moreover these paths must not contain any cycles, i. e. every node from G can appear at most once in each path.

c. Neighborhood graphs in spatial data mining:

In neighborhood graphs are used to represent topology of data objects and their neighborhood. Four spatial data mining tasks that use neighborhood graphs are described: spatial association rules, spatial clustering, spatial trend detection and spatial classification. In GeoKD system (see section 2.2) spatial data are stored in structures described in section 1 – points, chains and polygons. Program uses operation get Graph from previous definition (implemented in PostgreSQL) to create neighborhood graph from these data. The graph is stored in database and is used to find neighbors of an object during the knowledge discovery process.

B. Amalgamation:

In two polygon amalgamation algorithms are described. Operation of polygon amalgamation can be used in spatial data mining where spatial objects (represented by polygons) need to be amalgamated when the user wants to aggregate them. Both these algorithms identify boundary rectangles which are important for the resulting boundary. The first of these algorithms, adjacency method, uses information about objects adjacency to identify boundary rectangles. The adjacency of two objects can be presented with the neighborhood graph. The second algorithm, occupancy method, uses *z–ordering* to partition the space. Z–ordering can be seen as a quad tree with paths described with numbers: The space is divided recursively into four quadrants. Each of these quadrants is identified by a number from 1 to 4. Numbering these quadrants recursively, as they are divided, we get the so called z–value which can be maintained by the one– dimensional access method *B+–tree*. The z–ordering method is modified and each quadrant contains information about its occupancy by amalgamated polygons. Bordering quadrants are not totally occupied.

VI. CONCLUSION

In the application of classification and trend detection in satellite raster images is described. The result of this application is an identification of the trends in the land use changes in the dynamic urban area of Brno city. The source images were preprocessed and then unsupervised classification (based on pixels clusterization) was applied to them. The results of the classification were used to find differences in the land use between images. The methods for finding differences are: subtracting digital values of corresponding pixels, dividing values or comparing the classifications of the pixels.

Another application of the raster images classification is described in , where treetops are detected in an airborne image of a forest. The first step of image processing is smoothing out with different filters (both linear and nonlinear). In the next step, treetops are detected. This step is based on the information that treetops have a greater reflectance than their neighborhood. In the final steps, duplicate identifications of tops are filtered.

In the analysis of traffic accessibility with respect to public transport accessibility is described. This analysis requires the information about road infrastructure and time schedules of public transport.

VII. REFRENCES

[1]. Beckmann N., Kriegel H. P., Schneider R. and Seeger B:The R-tree:an efficient and robust access method for points and rectangles Proceedings of the 1990 ACM SIGMOD international conference on Management of data, 1990, Atlantic City, NJ USA, pp. 322-331.

[2]. Finkel R., Bentley J.: Quad Trees: A Data Structure for Retrieval on Multiple Keys. Acta Informatica, Vol. 4, No. 1, 1974, pp. 1-9.

[3]. Bentley J.: Multidimensional Binary Search Trees used for Associative Searching. Communications of ACM, 18, 9, Sep. 1975, pp. 509-517.

[4]. Gaede V., G¨ unther O.: Multidimensional Access Methods. ACM Computing Surveys, Vol. 30, No. 2, June 1998, pp. 170 - 231.

[5]. Guttman, A.: R-trees: a dynamic index structure for spatial searching. Proc. of SIGMOD Int. Conf. of Management of Data, 1984, pp. 47-54.

[6]. Han et al.: DMQL: A Data Mining Query Language for RelationalDatabases. In: ACM-SIGMOD’96Workshop on Data Mining.

[7]. Knowledge Discovery in Spatial Data by Means of ILP.In Zytkow J.M., Quafafaou M.(eds.): Principles of Data Mining and Knowledge Discovery. Proc. of 2nd Eur. Symposium, PKDD’98, Nantes, France. LNCS 1510, Springer Verlag 1998.

[8]. Bozkaya T., Ozsoyoglu M.: Indexing Large Metric Spaces for SimilaritySearch Queries. ACM Trans. Database Syst. 24, 3 (Sep. 1999), Pages 361-404.

[9]. Ester M., Kriegel H. P., Sander J.: Spatial Data Mining: A Database Approach. Proc. of the Fifth Int. Symposium on Large Spatial Databases (SSD ’97), Berlin, Germany, Lecture Notes in Computer Science, Springer, 1997.

Short Biodata of the Author

Mr. A.Srinivasa Reddy currently working as Associate Professor in Dept of Information Technology. He Received his B.Tech from GVP College of Engineering, Visakhapatnam and M.Tech from Sathyabama University, Chennai both with Distinctions. Prior to joining in Academic Profession, he had been worked as Software Engineer in Satyam Computers, Hyderabad. He has published several papers at various national and International Journals and Conferences. His Research area includes Data mining, Network Security and Software Engineering.