Development of Efficient & Optimized Algorithm for Knowledge Discovery in Spatial Database Systems

Kapil AGGARWAL, India

Key words: KDD, SDBS, neighborhood graph, neighborhood path, neighborhood index

SUMMARY

Knowledge discovery in databases (KDD) is an important task in spatial databases since both, the number and the size of such databases are rapidly growing. The automated discovery of knowledge in databases is becoming increasingly important as the world's wealth of data continues to grow exponentially. The main contribution of this paper is to introduce a set of basic operations, which should be supported by a spatial database system (SDBS) to express algorithms for KDD in SDBS. The definition of such a set of basic operations and their efficient support by an SDBS will speed up the development of new spatial KDD algorithms and their performance. For this purpose, the concept of neighborhood graphs and paths and a small set of operations for their manipulation have been used. These operations are sufficient for KDD algorithms considering spatial neighborhood relations by presenting the implementation of typical spatial KDD algorithms based on the proposed operations. A wide variety of algorithms have been proposed for KDD. This involves evaluation of algorithms for optimizing the performance of the KDD operations. These algorithms are classified and identified certain generic tasks like cluster, classification, dependency analysis and deviation detection. While a lot of algorithms have been developed for KDD in relational databases, the area of KDD in spatial databases has only recently emerged. Furthermore, the efficient support of operations on large neighborhood graphs and on large sets of neighborhood paths by the SDBS is discussed. Neighborhood indices are introduced to materialize selected neighborhood graphs in order to speed up the processing of database operations. For that, firstly, the algorithms for optimizing the performance of the KDD operations using the available indices are to be evaluated. Secondly, the materialization of neighborhood paths has to be investigated. This seems to be feasible if appropriate filters are used to create reasonably small sets of paths. A materialization of relevant paths may further speed-up the overall performance of KDD tasks because different KDD algorithms may use the same set of paths and each algorithm may this scan set of paths many times.

Development of Efficient & Optimized Algorithm for Knowledge Discovery in Spatial Database Systems

Kapil AGGARWAL, India

1. INTRODUCTION

During the last decade, the management of spatial data in Geographic Information Systems (GIS), which are used in application areas such as Geography, CAD/CAM, Biology, Medicine, etc. has gained lot of importance. Due to the high complexity of objects and queries and also due to the extremely large data volumes, geographic database systems impose stringent requirements on the employed storage and access structures. Spatial database systems offer the underlying database technology for geographic information. Both, the number and the size of spatial databases are rapidly growing in applications such as traffic control, city planning and environmental studies. Advances in database technologies and data collection techniques including barcode reading, remote sensing, satellite telemetry, etc., have collected huge amounts of data in large databases. This explosively growing data creates the necessity of knowledge/information discovery from data, which leads to a promising emerging field, called data mining *or* knowledge discovery in databases. Data mining represents the integration of several fields, including machine learning, database systems, data visualization, statistics, and information theory. As, the number and the size of spatial databases, such as geographic or medical databases, are rapidly growing because of the large amount of data obtained from satellite images, computer tomography or other scientific equipment. Knowledge discovery in databases (KDD) is the process of discovering valid, novel, and potentially useful patterns from large databases. Knowledge discovery in databases revolves around the investigation and creation of knowledge, processes, algorithms, and the mechanisms for retrieving potential knowledge from data collections. Related issues include data collection, database design, the description of entries in the database using the most appropriate representation, and data quality. Spatial Database Systems are database systems for the management of spatial data. A geographic information system is an information system for data representing aspects of the surface of the earth together with relevant facilities such as roads or houses. In GIS, knowledge discovery helps in finding out the correlations between different characteristics of certain areas. The concept of neighborhood graphs explicitly represents implicit neighborhood relations relevant for KDD tasks. It follows a similar approach for modeling networks such as roads or telephone lines for the purpose of spatial query processing.

1.1 Knowledge Discovery in Geographic Information System

A geographic information system is an information system for data representing aspects of the surface of the earth together with relevant facilities such as roads or houses. In this section, one can introduce a sample geographic database providing spatial and non-spatial information such as communities, its natural facilities such as the mountains and its infrastructure such as roads. In GIS, knowledge discovery helps in finding out the

correlations between different characteristics of certain areas. E.g. one can find that areas with a high value for the attribute rate of retired people are highly correlated with neighboring mountains and lakes. This KDD task is performed in two steps like by finding the areas of spatial objects, i.e. clusters or neighboring objects, which are homogeneous with respect to some attribute values and then finding associations with other characteristics of these areas i.e. by correlating them with reference maps or with other attribute values.

2. OBJECTIVES

In the present work, I propose to work on the analysis of set of basic operations used for the knowledge discovery in spatial database systems and to optimizing propose algorithm for the performance of these operations. The main objectives of my paper are to study & analyze the set of basic operations used for the knowledge discovery in spatial database systems.

Design and develop KDD algorithm using basic set of operations whose performance is optimized in spatial database system by materializing the relevant neighborhood paths in order to speed-up the overall performance of KDD tasks. Finally, simulate a Cost Model through which comparison of the performance of database operation with a neighborhood index can be made versus without the neighborhood index.

3. REVIEW OF LITERATURE

Data mining, which supports knowledge discovery in databases is the automated extraction of patterns representing knowledge implicitly stored in large databases, data warehouses, and other massive information repositories. KDD is the non-trivial extraction of implicit, previously unknown, and potentially useful information from databases [Frawley 1991]. A wide variety of algorithms have been proposed for Knowledge discovery in Databases [Koperski 1996]. Proposed algorithms for knowledge discovery in databases have been classified and identify generic tasks like class identification, classification, dependency analysis, deviation detection [Matheus 1993]. Lots of algorithms have been developed for KDD in relational databases. Most of the queries in a relational database can be expressed using the five basic operations of relational algebra. Agrawal, Imielinski and Swami follow a similar approach for KDD in relational databases [Agarwal 1993]. One can use an extended relational model and the SAND (Spatial and Non-spatial Database) architecture [Aref 1991]. The spatial extension of the objects (i.e. polygons or lines) is stored and manipulated using an R*-tree [Beckmann 1990]. There are various approaches to extract knowledge from spatial databases. Attribute-oriented induction is applied to spatial and non-spatial attributes using (spatial) concept hierarchies to discover relationships between spatial and non-spatial attributes [Lu 1993]. Various algorithms are available to detect properties of clusters using reference maps [Han 1994]. One can use the concept of explicit neighborhood graphs for representing the neighborhood relations relevant for KDD tasks [Knorr 1996]. Erwig M. & Gueting follows a similar approach for modeling networks such as roads or telephone lines for the purpose of spatial query processing [Erwig 1994]. Clustering algorithms group a given set of objects into classes, i.e. clusters, such that objects in one class show a high degree of similarity, while objects in different classes are as dissimilar as possible. Several clustering

algorithms for large spatial databases have been designed [Ester 1995]. In an SDBS, a spatial trend is defined as a pattern of change of some non-spatial attribute in the neighborhood of some database object. While using the concept of neighborhood indices, one should assume the existence of some spatial index such as an R*-tree to support spatial query processing.

4. PROPOSED WORK

It is proposed to analyze the set of basic operations used for the knowledge discovery in spatial database systems and develop a new algorithm whose performance is optimized and efficient when applied to Spatial Databases. Furthermore, the efficient support of operations on large neighborhood graphs and on large sets of neighborhood paths by the SDBS will be analyzed. This involves evaluation of algorithms for optimizing the performance of the KDD operations. The effect of neighborhood indices is analyzed to materialize selected neighborhood paths in order to speed up the processing of various operations. A materialization of relevant paths may further speed-up the overall performance of KDD tasks because different KDD algorithms may use the same set of paths and each algorithm may scan set of paths many times.

5. METHODOLOGY

5.1 Knowledge Discovery in Spatial Database System (SDBS)

Spatial Database Systems are relational databases plus a concept of spatial location and spatial extension. The explicit location and extension of objects define implicit relations of spatial neighborhood. KDD algorithms make use of those neighborhood relationships, because it is the main difference between KDD in relational Database System and in Spatial Database system 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. Efficiency of many KDD algorithms for SDBS depends heavily on an efficient processing of these neighborhood relationships. This novel approach to KDD in spatial databases aiming at an extension of SDBSs with data structures and operations for efficient processing of implicit relations of spatial neighborhoods. This approach allows a tight integration of spatial KDD algorithms with the database management system of a SDBS, speeds-up the development and execution of spatial KDD algorithms.

5.1.1 Neighborhood Relations

Attributes of the neighbors of some object of interest may have an influence on the object itself. For instance, a new industrial plant may pollute its neighborhood depending on the distance and on the major direction of the wind. Figure 1 depicts a map used in the assessment of a possible location for a new industrial plant. The map shows three regions with different degrees of pollution (indicated by the different colors) caused by the planned plant. Furthermore, the influenced objects such as communities and forests are depicted.

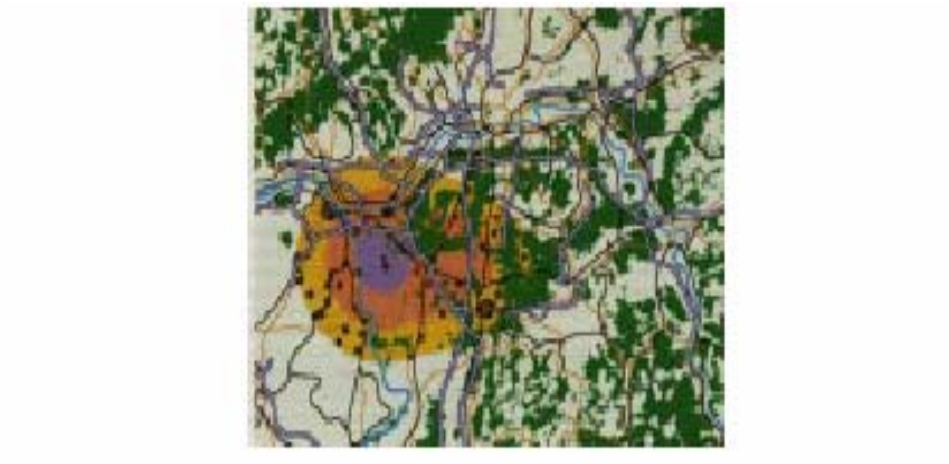


Figure 1: Regions of pollution around a planned industrial plant

Most mining algorithms for spatial databases will make use of spatial neighborhood relationships, because objects are influenced by the attributes of their neighbors depending on the type of neighborhood relation. In this section, three basic types of spatial relations i.e. topological, distance and direction relations are used. These relations are binary relations, i.e. relations between pairs of objects.

Spatial objects may be either point objects or spatially extended objects such as lines, polygons or polyhedrons. Spatially extended objects may be represented by a set of points at its surface, e.g. by the edges of a polygon or by the points contained in the object, e.g. the pixels of an object in a raster image. Therefore, sets of points as a generic representation of spatial objects are used.

5.1.2 Neighborhood Graphs and Neighborhood Paths

Concept of neighborhood graphs explicitly represents implicit neighborhood relations relevant for KDD tasks. It follows a similar approach for modeling networks such as roads or telephone lines for the purpose of spatial query processing.

A neighborhood graph $G_{neighbor}$ for some spatial relation "neighbor" is a graph (N, E) with the set of nodes N and the set of edges E. Each node corresponds to an object of the database and two nodes n_1 and n_2 are connected via some edge if neighbor (object (n_1), object (n_2)) holds. The predicate neighbor may be one of the following neighborhood relations:

Topological-Relations	e.g. {meet, overlap, covers, covered-by}
Metric-Relations	e.g. {distance < d}
Direction-Relations	e.g. {north, south, west, east}

Based on the neighborhood graphs, one can define a neighborhood path in some graph G as a list of nodes of G with an edge of G connecting each pair of successors in the list, e.g. (n_1, n_2, …n_k) where neighbor (n_i, n_{i+1}) holds for each i, $1 \leq i \leq k-1$. The length of a path is defined as the number of its nodes.

5.2 Database Operations for KDD in Spatial Database System

Some of the basic database operations, which are supported by the spatial database systems are as follows:-

S. No.	Name of the Operation
1.	select (db: Set-Of-Objects; pred: Predicate)
2.	get-value (o:Object; attr: Attribute)
3.	get_nGraph (db: Set-Of-Objects, rel)
4.	get_neighborhood (graph, o, pred)
5.	create_nPaths (objects, graph, pred, i)
6.	extend (set_of_paths, graph, pred, i)

Table 1: Basic database operations

5.3 KDD Algorithms using Basic Operations

The applicability of the proposed basic operations is shown in this paper by presenting spatial KDD Spatial Trend Detection algorithm.

5.3.1 Spatial Trend Detection

A trend may be defined as a temporal pattern in some time series data such as network alarms. In an SDBS, one can define a spatial trend as a pattern of change of some non-spatial attribute (attributes) in the neighborhood of some database object, e.g. “when moving away from Delhi, the economic power decreases”.

In the following, an algorithm is used, which discovers trends in SDBS starting from some object *o*. In each step, the algorithm computes both the local changes of the specified attribute when moving to the neighbors as well as the distance to these neighbors. A linear regression is applied to these pairs of values (change of attribute value, distance). If the resulting correlation coefficient is larger than a specified threshold, the slope of the resulting linear function is returned as the trend for *o*. If the correlation coefficient is not large enough, no trend is discovered for *o*.

5.4 Efficient **SDBS Support for Neighborhood Graphs and Paths**

In this section, the efficient support of the operations on neighborhood graphs and paths by an SDBS is taken up. Concept of neighborhood indices is used to materialize selected neighborhood graphs and show how they can be used to speed up the processing of basic operations. Furthermore, a cost model is presented which allows comparing the expected execution time of a get-neighborhood operation with vs. without a neighborhood index.

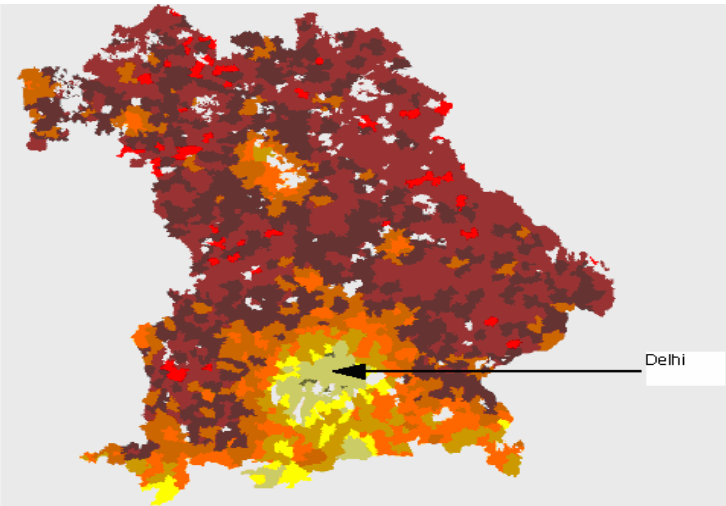


Figure 2: Depicts average rent from geographic database

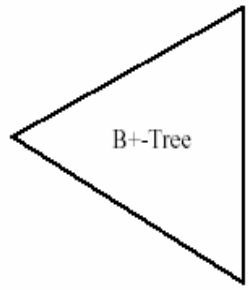
Figure 2. depicts a significant trend can be observed for the city of Delhi: the average rent decreases quite regularly when moving away from Delhi.

5.4.1 Neighborhood indices

Concept of spatial join indices has been introduced as a materialization of a spatial join result with the goal of speeding up spatial query processing. This paper, however, does not deal with the questions of efficient implementation of such indices. It extends the concept of spatial join indices by associating their distance with each pair of objects (distance associated join indices). Thus, the join index can be used to support not only queries concerning a single spatial predicate but the index is applicable to a large number of queries. In its naive form, however, this index requires $O(n^2)$ space because it needs one entry not only for pairs of neighboring objects but for each pair of objects. Therefore, a hierarchical version of distance associated join indices is proposed. These indices assume a spatial hierarchy of objects, e.g. countries > cities > houses. Entries in the index are only generated for pairs of objects contained in the same object of the next higher level of the hierarchy, e.g. only for pairs of houses of the same city and only for pairs of cities of the same country. The hierarchical approach significantly reduces the space requirements of the join index but also prevents its application to databases if a spatial hierarchy is either not available or the spatial hierarchy is not relevant for the purpose of KDD.

If many operations are performed on the same neighborhood graph and if this graph is relatively stable, an index should be constructed for this neighborhood graph. This seems to be especially important if neighborhood paths are to be constructed from some neighborhood graph. Note that many SDBS are rather static since there are not many updates on objects such as geographic maps or proteins. One can define a neighborhood index as an index explicitly representing a neighborhood graph, i.e. a neighborhood index supports the processing of all operations on its corresponding neighborhood graph without accessing the database itself. A simple implementation of a neighborhood index using a B+-tree is

illustrated in figure 5. In general, a neighborhood graph is undirected implying a double representation of each edge in the neighborhood index.



Object	Neighbors
o_1	o_2, o_3, o_7, o_9
o_2	o_1, o_3, o_9
. . .	. . .

Figure 3: Sample neighborhood index

6. PERFORMANCE EVALUATION

The database primitives have been implemented on top of the commercial RDBMS Oracle 9.1 as a client process. A neighborhood index was realized as a relational table indexed by a B-tree on the Object-ID attribute. Whenever no neighborhood index is applicable, the implementation of the database primitives uses an R-tree which is provided by the Oracle 9.1 2D spatial database. A cost model is developed to compare the performance with and without neighborhood indices for arbitrary parameter values. A geographic database was used for an experimental performance evaluation and validation of the cost model.

6.1 Cost Model

A cost model is developed to predict the cost of performing a get_neighborhood (graph, object, filter) operation with vs. without a neighborhood index. In the database community, usually the number of page accesses is chosen as the cost measure. However, the amount of CPU time required for evaluating a neighborhood relation on spatially extended objects such as polygons may very large so that model both, the I/O time and the CPU time for an operation. Use t_{page}, i.e. the execution time of a page access, and t_{float}, i.e. the execution time of a floating-point comparison, as the units for I/O time and CPU time, respectively.

Some of the parameters of the cost model and list typical values for each of them:

name	meaning	values
n	number of nodes in the neighborhood graph	$[10^3 . . 10^5]$
e	number of (directed) edges in the neighb. graph	$[10^3 . . 10^6]$
v	average number of vertices of a polygon	$[10 . . 10^4]$

Table 2: Parameters of the cost model

7. CONCLUSIONS

In this paper, neighborhood graphs, neighborhood paths and a small set of operations as database primitives for spatial data mining algorithm is introduced. An introduction of neighborhood indices also support efficient processing of the database primitives by a data base management system. Database primitives are implemented on top of a commercial spatial database management system. The effectiveness and efficiency of the proposed approach is evaluated by using an analytical cost model and an extensive study on a geographic database. Finally, the proposed database primitives will inspire the development of new spatial data mining algorithms based on neighborhood graphs and paths.

8. PROSPECTS

Spatial databases have applications in various areas as wide as geography, biology, computer aided design, medicine etc. The algorithms available as of now have many drawbacks, which is a major hurdle in advancements of this field. Optimizing these algorithms will definitely contribute a lot to it.

REFERENCES

Frawley W.J., Piatetsky-Shapiro G., Matheus J.: "Knowledge Discovery in Databases: An Overview", in: Knowledge Discovery in Databases, AAAI Press, Menlo Park, 1991, pp. 1-27

Koperski K., Adhikary J., Han J.: "Knowledge Discovery in Spatial Databases: Progress and Challenges", Proc. SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Technical Report 96-08, University of British Columbia, Vancouver, Canada, 1996

Matheus C.J., Chan P.K.: "Systems for Knowledge Discovery in Databases", IEEE Transactions on Knowledge and Data Engineering, Vol.5, No.6, 1993, pp. 903-913

Agrawal R., Imielinski T. Swami A.: "Database Mining: A Performance Perspective", IEEE Transactions on Knowledge and Data Engineering, Vol.5, No.6, 1993, pp. 914-925

Aref W.G. Samet H.: "Optimization Strategies for Spatial Query Processing", Proc. 17th Int. Conf. VLDB, Barcelona, Spain, 1991, pp. 81-90

Beckmann N., Kriegel H.: 'The R*-tree: An Efficient and Robust Access Method for Points and Rectangles', Proc. ACM SIGMOD Int. Conf. On Management of Data, Atlanta City, NJ, 1990, pp. 332-331

Lu W., Han J., Ooi B.C.: "Discovering of General Knowledge in Large Spatial Databases", Proc. Far East Workshop on Geographic Information Systems, Singapore, 1993, pp. 275-289

Han J., Ng R.T.: "Efficient and Effective Clustering Methods for Spatial Data Mining", Proc. 20th Int. Conf. on Very Large Data Bases, Santiago, Chile, 1994, pp. 144-155

Ng R.T.: "Spatial Data Mining: Discovering Knowledge of Clusters from Maps", Proc. SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Technical Report 96-08, University of British Columbia, Vancouver, Canada, 1996

Knorr E.M., Ng R.T.: “Finding Aggregate Proximity Relationships and Commonalities in Spatial Data Mining”, IEEE Transactions on Knowledge and Data Engineering, Vol.8, No.6, 1996, pp. 884-897

Erwig M., Gueting R.H.: “Explicit Graphs in a Functional Model for Spatial Databases”, IEEE Transactions on Knowledge and Data Engineering, Vol.6, No.5, 1994, pp.787-803

Ester M., Kriegel H., Xu X.: “Knowledge Discovery in Large Spatial Databases: Focusing Techniques for Efficient Class Identification”, Proc. 4th Int. Symp. on Large Spatial Databases, Portland, ME, 1995, pp.67-82

BIOGRAPHICAL NOTES

Kapil Aggarwal

Academic experience: Master of Technology, Bachelor of Engineering, Nagpur University
Current position: Lecturer (Computer Science), Banasthali Vidyapith 2003-
Practical experience: Remote sensing and geographic information system expert, computer programming, Geo-informatics
Activities in home and International relations:
Member, Indian Society of Remote Sensing 2003-
Member, Computer Society of India 2000-
Member, Institution of Electronics & Telecommunication Engineers, India 2000-
International conference (ICYCS), Beijing, delegate 2005-

CONTACT

Kapil Aggarwal
Banasthali Vidyapith
Deemed University
AIM & ACT
P.O. Banasthali Vidyapith
Rajasthan-304022.
INDIA
Tel. + 91 1438 228647
Mob + 91 9414543868
Fax + 91 1438 228649
Email: kapil594@rediffmail.com