A Reuse-based Spatial Data Preparation Framework for Data Mining

Vania Bogorny, Paulo Martins Engel, Luis Otavio Alvares
Instituto de Informática - Universidade Federal do Rio Grande do Sul
Av. Bento Goncalves, 9500 - Porto Alegre - Brazil
{vbogorny, engel, alvares }@inf.ufrgs.br

Abstract *The constant increase in use of geographic data in different application domains has resulted in large amounts of data stored in spatial databases and in the desire of data mining. Many solutions for spatial data mining have been proposed. Most create data mining languages or extend existing query languages to support data mining operations. This paper presents an interoperable framework for spatial data preparation for data mining. The approach is based on reuse of standard definitions such as Open GIS Consortium specifications, SQL query language, and well-established data mining toolkits. The proposed framework was implemented in the Java programming language and validated with real spatial databases and the Weka data mining toolkit.*

Keywords: software reuse, spatial databases, data mining, data preparation framework

1. Introduction

Large amounts of spatial data have been used more and more in many areas in different application domains such as urban planning, transportation, telecommunication, marketing, and so on. These data are stored and manipulated in Spatial Database Management Systems (SDBMS), and Geographic Information Systems (GIS) is the technology which provides a set of operations and functions for spatial data analysis. However, within the large amount of data stored in spatial databases there is implicit, nontrivial and previously unknown knowledge that cannot be detected by GIS. Specific techniques are necessary to find this kind of knowledge, which is the objective of Knowledge Discovery in Databases (KDD) research.

KDD is an interactive process which consists of five steps: *selection, preprocessing, transformation, data mining and evaluation/interpretation* [1]. *Selection, preprocessing* and *transformation* are the steps in which data are rearranged to the format required by data mining algorithms. It is stated that between 60 and 80 percent of time and effort in the whole KDD process is required for data preparation [2].

Data Mining (DM) is the step of applying discovery algorithms that produce an enumeration of patterns over the data. Most of these algorithms were created to deal with small amounts of data and with a restrictive *single table* input format. This limitation causes a *gap* between spatial databases and data mining algorithms.

Many solutions for spatial data mining have been proposed in the literature, but only a few consider aspects of data preparation. Most approaches extend query languages with new functions and operations for data mining. Han [3] proposed a geo mining query language (GMQL) implemented in the GeoMiner software prototype. Ester [4] defined a set of new operations such as *get_nGraph, get_neighborhood* and *create_nPaths* to compute spatial neighbors. Sattler [5] proposed a multi-database language to support the KDD steps. Malerba [6] proposed an object-oriented data mining query language named SDMOQL, implemented in the INGENS software prototype.

In those approaches it is expected that the SDBMS will implement the proposed languages and operations. However, most SDBMS follow the Structured Query Language (SQL), which became the standard language to manipulate databases. As most SDBMS do not implement those approaches, and most spatial data mining software prototypes are no longer available outside academic areas, we propose an interoperable *reuse-based* framework to prepare spatial data for classical DM. The objective is to automate part of the KDD steps in order to reduce data preparation time.

The remainder of the paper is organized as follows: Section 2 describes the components of reuse and interoperability. Section 3 shows the transformation model to convert spatial data into the single table format. Section 4 presents the framework for KDD in spatial databases. Section 5 outlines experiments with artificial and real geographic databases and Section 6 presents the conclusion and future work.

2. Specifications for Reuse and Interoperability

Our approach is based on four well-established components of reuse and interoperability: Open GIS Consortium (OGC) specifications [7], SQL (Structured Query Language), java database connectivity (JDBC) and classical DM toolkits.

2.1 OGC Spatial Operations and Database Schema

The GIS implement specific operations and functions to manipulate and visualize spatial data. The OGC is an organization dedicated to develop patterns for spatial operations and spatial data integration, providing

interoperability for GIS. Among many specifications established by the OGC, there are two that will be considered in this approach: functions and operations to compute *spatial relations* and the *database schema* for storage of geographic data.

Spatial relations are relationships among real world entities (or features) which are usually not explicitly stored in spatial databases, so they have to be computed with spatial operations. There are basically 3 spatial relations to consider: *topological, distance* and *order. Topological* relations characterize the type of intersection between two geographic features and they can be classified in *Equal, Disjoint, Touches, Within, Overlaps, Crosses, Contains* and *Covers*. Figure 1 (a) shows an example of the topological relations *Touches, Contains* and *Crosses.*

Distance relations are based on the Euclidean distance between two spatial features, as shown in Figure 1(b).

Direction relations deal with the order as spatial features are located in space. Figure 1(c) shows an example of direction/order relations.

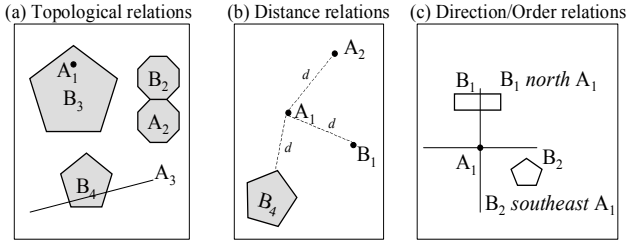


Figure 1 – Spatial relationships

The OGC defines a standard set of operations to compute spatial relations for SQL [7], which are implemented by most SDBMS. These operations are sufficient enough to compute relations among spatial features.

The OGC also defines a database schema for storage of spatial data. This schema defines a database table named GEOMETRY_COLUMNS, which stores all database characteristics, as shown in Figure 2(d). This table consists of a row for each feature table in the spatial database with geometric attributes.

2.2 Query Language and Database Connectivity

The SQL became the standard data definition and manipulation language for relational databases [8]. This language is implemented by most commercial and open source SDBMS and was extended with spatial functions and operations to manipulate spatial data. With this standard, it is possible to write queries to access data stored in different databases, without changing the statements.

JDBC is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases. The JDBC API provides a call-level API for SQL database access. With a JDBC API it is possible to establish a connection with a spatial database, to send SQL statements to manipulate data, and to process the results.

2.3 Spatial Data Mining

Spatial DM can be performed in two ways: with spatial data mining algorithms - which by themselves compute the spatial relations, or with classical data mining algorithms, if spatial relations are previously materialized in non-spatial data.

Spatial data mining algorithms are not available in many toolkits, while classical data mining algorithms are implemented in many well-established commercial and open-source toolkits such as Weka, Intelligent Miner, DBMiner and others. In our approach, we follow the second concept to maximize the reuse of available data mining toolkits.

3. Transformation Model for the Single Table Representation

In the single table format shared by most classical data mining algorithms, each row represents an independent unit (target feature) and each column is a relevant feature characterizing this unit. Figure 2 shows a sample of real data to illustrate the transformation model to obtain the single table format. These data were used in a case study in the context of a public health application domain of the Brazilian government. The data were stored in a SDBMS constructed under OGC specifications. *Hospitalization*, *Factory* and *Cellular Antenna* are feature types stored as database tables. Each instance of hospitalization, factory or antenna are the feature instances, stored as rows of those tables. *Hospitalization* represents patients with some kind of disease, age, gender and address spatial position. *Factory* contains all industries with its name, address, kind of activity and its spatial position. *Cellular Antenna* stores the company name, power level, installation date and spatial position.

Considering *Hospitalization* as the target feature type and *Factory* and *Cellular Antenna* as the relevant features types, the single table representation is created as follows: first, the spatial relations between the geographic position of each patient and all factories and antennas are materialized; second, two different transposed tables can be created according to the granularity level.

In the feature type granularity level, the resultant single table is created with the patient's non-spatial attributes and with the addition of a new column for each pair of different spatial relation and feature type, as shown in Figure 3 (a). The value of each new column is the materialized spatial relation.

Figure 3 (b) shows the feature instance granularity level, where a new column is created for each pair of feature type and feature instance identifier. The value of each column is the materialized spatial relation.

(a) Hospitalization

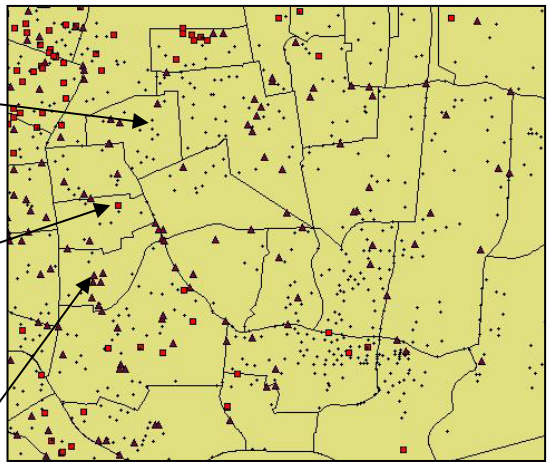
Gid	Disease	Age	Gender	The_geom
1	Cancer	23	M	Point...
2	Infarct	45	M	Point...
...	...	...	...	...

(b) Factory

Gid	Name	Address	Business	The_geom
1	Gerdau	Xxxxxx	Siderurgical	Point...
2	Triunfo	Zzzzzzz	Metallurgical	Point...
...	...	...	...	...

(c) Cellular Antenna

Gid	Owner	Power level	Date	The_geom
1	Vivo	2	03/01/1999	Point...
2	Vivo	1	10/10/1995	Point...
...	...	...	...	...



(d) GEOMETRY_COLUMNS

F_table_schema	F_table_name	F_geometry_column	.....	SRID
Public	Hospitalization	The_geom	.....	-1
Public	Factory	The_geom	.....	-1
Public	Cellular_Antenna	The_geom	.....	-1

Figure 2 – Spatial database table format

(a) feature type granularity level

Gid	Disease	Age	Gender	Factory	Cellular Antenna
1	Cancer	23	M	Close	Far
2	Infarct	45	M	Far	Close
...	...	...	...	...	...

(b) feature instance granularity level

Gid	Disease	Age	Gender	Factory 1	Factory 2	Cellular Antenna 1	Cellular Antenna 2
1	Cancer	23	M	Close	Close	Far	Far
2	Infarct	45	M	Far	Far	Close	...
...	...	...	...	...	...	...	...

Figure 3 – The transformation model for the single table format

4. A Framework for Spatial Data Mining

In order to automate the spatial data preparation for data mining, we propose a reuse-based framework, shown in Figure 4. It is composed of three conceptual levels: data mining, data preparation and data repository. At the bottom are the spatial databases stored in SDBMS constructed under OGC specifications. At the top are the data mining toolkits to be used in the KDD process. In the center is the spatial data preparation level which covers the *gap* between data mining tools and spatial databases. In this level the data repositories are accessed through JDBC connections and data are retrieved and transformed into the single table format with SQL statements, according to the user's specifications. There are three main modules to implement the tasks of spatial data preparation for data mining: *feature selection, spatial join* and *transformation*.

The *Feature Selection* module defines and retrieves all relevant information from the databases, including the database schema, the target feature type, the target feature non-spatial attributes and the relevant feature types. The feature types are retrieved through the Open GIS database schema, stored in the table GEOMETRY_COLUMNS.

Spatial Join is the module which computes and materializes the user-specified spatial relation(s) between the features retrieved by the *Feature Selection* module. *Spatial Join* computes the spatial relations with SQL statements and spatial operations defined by the OGC.

The *Transformation* module is responsible to transpose the *Spatial Join* module output into the single table

representation, understandable by data mining algorithms. This module requires two user-specified parameters: the granularity level and the classical data mining toolkit. This step is based on SQL statements, where the single table *S* is created with the target feature non-spatial attributes and all its instances. Then for each instance in *S*, a new attribute is created for each different spatial relation between the target feature and the relevant feature.

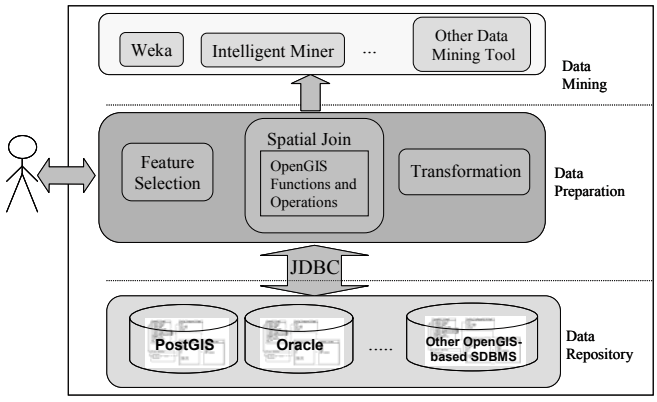


Figure 4 – Framework design

5. Validation and Experiments

The proposed framework was implemented in the java programming language and tested with the Weka data mining toolkit. To evaluate the framework, the best way should be with real geographic databases, but sometimes it might be difficult to judge the quality of the discovered results, without knowing a priori what the DM algorithm is supposed to find. Thus, we did experiments with artificial and real databases.

The artificial dataset was created in the same application domain as the real geographic database and some patterns were intentionally introduced. For example, patients with cancer live close to cellular antennas; patients with respiration problems live close to factories; and patients with ear problems live close to noise places (airports and highways).

The artificial dataset gave us support to figure out the appropriate model for spatial relations materialization. Clustering and association rules were performed over the data and the input format was adapted until the DM algorithms were able to find the patterns introduced in the artificial dataset. Experiments to find the appropriate input data model are described in [9].

6. Conclusion and Future work

The data preparation is an arduous task which usually is manually accomplished by the end-user, consuming the most effort in the KDD process. Aiming to reduce the data preparation time, we studied the problem conducted by a case study with real problems and presented a solution based on reuse of well-established and available components.

To show the feasibility of our framework it was implemented and validated with *Weka* (data mining tool) and *PostGIS* (spatial database). This framework reduces the data preparation time to mine large spatial databases with different techniques. In addition, it has the advantage that the preparation step can be easily performed many times, which is a common task in the KDD process.

The *Feature Selection* and *Spatial Join* module were implemented and the next step is to extend the *Transformation* module to prepare large spatial databases for other classical data mining tools. We will also evaluate the performance of *Spatial Join* module to compute and materialize the spatial relations.

Acknowledgments

We would like to thank Shashi Shekhar and José Palazzo Oliveira for their comments in earlier drafts of this paper. This research was partially supported by CAPES and CNPQ.

References

[1] Fayyad U, Piatetsky-Shapiro G and Smyth, P (1996). From data mining to discovery knowledge in databases**.** *AI Magazine*, 3(17): 37-54.

[2] Adriaans, P. and Zantinge, D (1996). *Data mining.* Addison Wesley Longman, Harlow, England.

[3] Han J, Koperski K., Stefanvic N (1997) GeoMiner: a system prototype for spatial data mining. In *Proceedings of the ACM-SIGMOD international conference on Management Of Data (SIGMOD'97)* (May 13-15*,*1997). ACM Press, Tucson, AR, 553-556.

[4] Ester M, Kriegel H-P, Sander J (1997). Spatial Data Mining: A Database Approach. *In Proceedings 5th Int. Symposium on Large Spatial Databases (SSD*), Berlin, Germany, pp. 47-66.

[5] Sattler K, Schallehn E (2001). A Data Preparation Framework based on a Multidatabase Language. In Proceedings of International Database Engineering and Applications Symposium (IDEAS).

[6] Malerba D, Esposito F, Lanza A, et al (2000). Discovering spatial knowledge: the INGENS system. In *Foundations of Intelligent Systems, 12th International Symposium, (ISMIS)*, Lecture Notes in Artificial Intelligence, 1932, 40-48, Springer, Berlin, Germany.

[7] Open GIS Consortium (1999). OpenGIS simple features specification for SQL. In URL: http://www.opengeospatial.org/docs/99-054.pdf

[8] Elmasri R, Navathe S (2003). Fundamentals of Database Systems. (4) Addison-Wesley.

[9] Bogorny V, Alvares, L O. *Geographic Data Representation for Knowledge Discovery.* Technical Report. UFRGS-RP 349, Porto Alegre, Brazil, 2005.