Spatial Data Mining: From Theory to Practice with Free Software

Vania Bogorny[1], Andrey Tietbohl[2], Bart Kuijpers[1], Luis Otavio Alvares[1,2]

[1] Hasselt University & Transnational University of Limburg
Agoralaan Gebouw D, 3590 Diepenbeek - Belgium
(vania.bogorny,bart.kuijpers)@uhasselt.be

[2]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS),
Av. Bento Gonçalves, 9500, Porto Alegre, RS, Brasil
{andrey,alvares}@inf.ufrgs.br

***Abstract**. One of the main challenges in spatial data mining is to automate the data preparation tasks, which consume more than 60% of the effort and time required for knowledge discovery in geographic databases. In this paper we present an extension of the classical open source data mining toolkit Weka to support automatic geographic data preprocessing. We propose Weka-GDPM+, which is interoperable with all geographic information systems constructed under OGC specifications. We tested Weka-GDPM+ with geographic databases stored into PostGIS, which is an open source GDBMS and is implemented according to OGC standards.*

1. Introduction

There is a growing interest in spatial data mining in many different areas such as transportation, ecology, epidemiology, etc. Many algorithms to extract knowledge from such data have been proposed in the literature. One of the main problems is that only a few of these algorithms are implemented in toolkits which support the whole discovery process.

In spatial/geographic databases data are stored in different relations (e.g. cities, rivers, roads), which need to be spatially joined in order to find novel and useful patterns in data mining. This step is normally performed in data preprocessing, and basically manually, since there is no available free and open source toolkit which automates this step. The result is that effort and time is required from the data mining user, which is usually neither an expert in geographic databases nor in data mining. It is stated that between 60 and 80 percent of the time and effort in the whole discovery process is required for data preparation [Addrians and Zantinge 1996].

Another problem is that data mining algorithms produce large amounts of outputs, especially the association rule mining technique. In spatial data mining this problem increases significantly. Besides the large number of patterns, many are well known a priori, because of the natural geographic dependences intrinsic to geographic data [Bogorny et al 2006a, 2006b].

Many solutions for spatial data mining have been proposed in the literature, but only a few consider data preparation aspects. Most approaches extend query languages with new functions and operations for data mining. Han et al (1997) proposed a geo-mining query language (GMQL) implemented in the GeoMiner software prototype. Malerba et al (2000) proposed an object-oriented data mining query language named SDMOQL, implemented in the INGENS software prototype. In these approaches it is expected that the GDBMS will implement the proposed languages and operations. However, most GDBMS follow the Structured Query Language (SQL), which became the standard language to manipulate databases, and do not implement data mining functionalities to either automate or semi-automate geographic data preprocessing. Indeed, most geographic data mining software prototypes are no longer available outside academic institutions.

Aiming to make a contribution for the spatial data mining field and to facilitate the practice of knowledge discovery in geographic databases, we developed Weka-GDPM+, which is an extension of Weka [Witten and Frank 2005] [Weka 2007] to support spatial data mining from geographic data stored under PostGIS [PostGIS 2007].

Weka is a free and open source classical data mining toolkit which provides friendly graphical user interfaces to perform the whole discovery process. It implements a variety of data mining algorithms and has been widely used for mining non-spatial databases. A version of the extended Weka is available for download at [Bogorny 2007].

The main objective of this work is to automate geographic data preprocessing steps into the Weka data mining toolkit. Additionally, in this extension an interface is provided such that the user may define well known dependences which are used in data preprocessing to reduce the number of spatial joins and non-interesting patterns. The main contributions include: (1) Interoperability with any GDBMS constructed under OpenGIS Simple Features Implementation Specification [OGC 1999a]; (2) Generate new datasets with different spatial relationships and attributes automatically; (3) Allow the use of different classical data mining algorithms in the data mining step; (4) Allow the user to choose the relevant spatial features, the reference feature type, different spatial relations, as well as different granularity levels [Han 1995]; (5) Geographic dependences can be specified by the user. These dependences are eliminated into Weka between the target feature type and any relevant feature type. This results in more efficient spatial joins and the generation of less well known geographic domain patterns, as explained in [Bogorny et al 2006a].

The remaining of this paper is organized as follows: Section 2 presents background knowledge about geographic databases. Section 3 presents the general framework for spatial data mining. Section 4 presents Weka-GDPM+, and Section 5 concludes the paper and suggests directions of future work.

2. Geographic Databases

Geographic databases (GDB) store real world entities, also called spatial features, located in a specific region [OGC 1999b]. Spatial features (e.g. Belgium, Brazil) belong to a feature type (e.g. country), and have both non-spatial (e.g. name, population) and spatial attributes (geographic coordinates *x,y*). In GDB every different feature type is usually stored in a different relation, because most geographic databases follow the relational or object-relational approach. Figure 1 shows an example of geographic data stored in relational databases, where the spatial feature types street, water resource, and gas station are different relations with spatial (shape) and non-spatial attributes.

The spatial attributes of geographic object types, represented by *shape* in Figure 1, have intrinsic spatial relationships (e.g. close, far, contains, intersects). Because of these relationships real world entities can affect the behavior of other features in the neighborhood. This makes spatial relationships be the main characteristic of geographic data to be considered for data mining and knowledge discovery. It is also the main characteristic which differs geographic/spatial data mining from non-spatial data mining.

Spatial relationships are usually not explicitly stored in geographic databases, so they have to be computed with spatial operations. There are basically 3 spatial relationships to consider [Gutting 1994]: *distance, order,* and *topological. Distance* relationships are based on the Euclidean distance between two spatial features. *Order* relationships deal with the order as spatial features are located in space. *Topological* relationships characterize the type of intersection between two spatial features and can be classified in *Equal, Disjoint, Touches, Within, Overlaps, Crosses, Contains, Covers,* and *CoveredBy*.

(a) Street

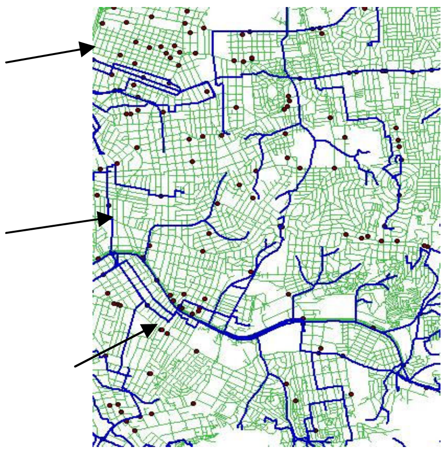
Gid	Name	Shape
1	Azenha	Multiline [$(x_1,y_1),(x_2,y_2),..$]
2	Lajeado	Multiline [$(x_3,y_3),(x_4,y_4),..$]

(b) WaterResource

Gid	Name	Shape
1	Jacui	Multiline [$(x_5,y_5),(x_2,y_2),..$]
2	Guaiba	Multiline [$(x_6,y_6),(x_2,y_2),..$]
3	Uruguai	Multiline [$(x_5,y_5),(x_7,y_7),..$]

(c) GasStation

Gid	Name	VolDiesel	VolGas	Shape
1	BR	20000	85000	Point[(x_8,y_8)]
2	IPF	30000	95000	Point[(x_9,y_9)]
3	Elf	25000	120000	Point[(x_3,y_3)]



(d) GEOMETRY_COLUMNS

F_table_schema	F_table_name	F_geometry_column	Type	SRID
Public	Street	Shape	Multiline	-1
Public	WaterResource	Shape	Multiline	-1
Public	GasStation	Shape	Point	-1

Figure 1. Geographic data storage structure in OGC based GDBMS

Considering the spatial relationships in the map shown in Figure 1 we can observe that all gas stations intersect (touches) streets. This relationship is well known since there exist no gas stations that are not located on streets. In data mining such relationships will generate non-interesting patterns such as *is_a(gasStation)→intersects(Street)*. However, notice that there is no standard spatial relationship between water bodies and gas stations. Their spatial relationships might be interesting for data mining.

Geographic database management systems (GDBMS) and Geographic information systems (GIS) implement specific functions to manipulate geographic data.

The OGC (Open GIS Consortium) is an organization dedicated to develop standards for geographic operations and geographic data integration, aiming to provide interoperability for GIS. Among many specifications established by the OGC, two are of fundamental importance for this work: operations to compute *spatial relationships* and the *database schema* metadata.

The database schema metadata are stored in a database table named GEOMETRY_COLUMNS, which is automatically created in a GDBMS that follows OGC specifications. An example is illustrated in Figure 1 (d). This table consists of a row for each feature type in the geographic database with spatial attributes. It is instantiated automatically when geographic data are loaded to the database the first time, and stores all database characteristics, including the database schema name, all geographic table names (f_table_name), the name of the geometry column (f_geometry_column), and its type (type).

3. The proposed framework

Figure 2 shows the proposed framework which is interoperable with any geographic information system developed under Open GIS Consortium specifications [OGC 1999a]. The framework is composed of three abstraction levels: data mining, data preparation, and data repository.

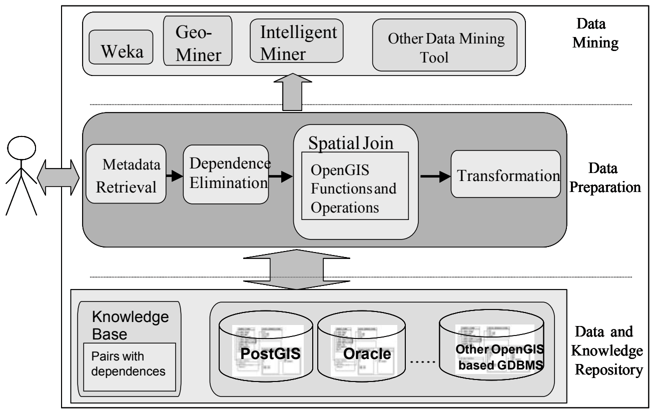


Figure 2. Integrated spatial data mining framework

At the bottom are the geographic data repositories, stored in GDBMS, constructed under OGC specifications. There is also a knowledge repository which stores well known geographic associations, extracted from geographic database schemas, geo-ontologies, or provided by the user. On the top are the data mining toolkits or algorithms for knowledge discovery in databases.

In the center is the spatial data preparation level which covers the *gap* between data mining tools and geographic databases. In this level the data repositories are accessed through JDBC connections and data are retrieved, preprocessed, and transformed into the single table format, according to the user specifications. This level we implemented into Weka, which integrates the data mining tasks with the operations and functionalities of PostGIS. It is important to observe that this module could also be implemented as module into a GIS or a GDBMS such as PostGIS, but Weka already provides support for different data mining tasks.

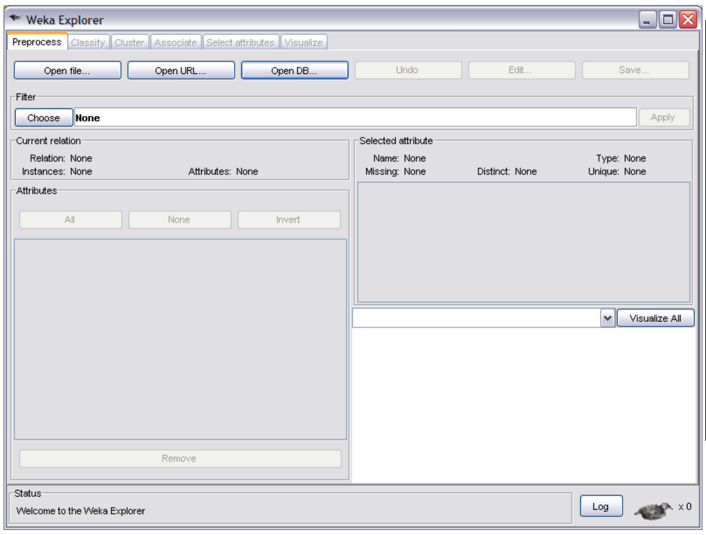
The *Metadata retrieval* module retrieves all relevant information from the database, including the target feature type, the target feature non-spatial attributes and the set of relevant feature types that may have some influence on the target feature type. The feature types are retrieved through the OpenGIS database schema, stored in the table GEOMETRY_COLUMNS.

The *Dependence Elimination* module verifies all associations between the target feature type and all relevant feature types. It searches the knowledge base and if the target feature has a dependence with a relevant feature, then the relevant feature type is eliminated from the set of relevant feature types. Notice that for each relevant feature type removed from the set, no spatial join is required to extract spatial relationships. By consequence, no well known pattern with this feature type will be generated.

The *Spatial Join* module computes and materializes the user-specified spatial relationships between the reference feature type and the relevant feature types, retrieved by the *Metadata Retrieval* module and filtered by *Dependence Elimination* module. The *Transformation* module transposes as well as discretisizes the *Spatial Join* module output into the single table representation, understandable by data mining algorithms.

4. Weka-GDPM+

Weka is a free and open source non-spatial data mining toolkit developed in Java. It provides a non-spatial data preprocessing module named weka.Explorer, shown in Figure 3 (left), in which it is possible to establish a database connection, open a web site, or an *arff* (input text file in the format required by Weka) file.



None
Relation: None
Instances: None
Attributes: None
Name: None
Type: None
Missing: None
Distinct: None
Unique: None
Welcome to the Weka Explorer

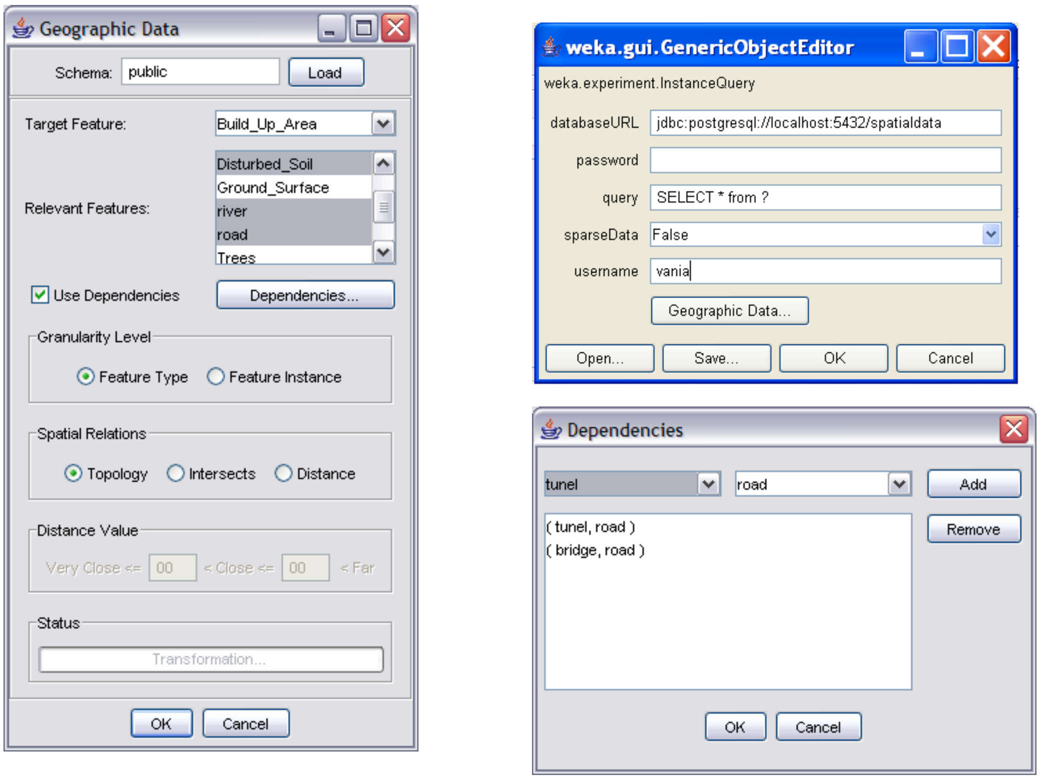
```
@relation geographic_data
@attribute bairro {Restinga, Moinhos,
                    Mont_Serrat, Azenha}
@attribute dengue {null,NAO,SIM}
@attribute postos_comb_CONTAINS {yes}
@attribute hospitais_CONTAINS {yes}
@attribute arroios_CROSSES {yes}

@data
Restinga,NAO,yes,?,yes
Moinhos,NAO,yes,yes,?
Mont_Serrat,NAO,yes,?,yes
Azenha,SIM,yes,yes,yes
```

Figure 3. (left) Weka Explorer GUI (right) *arff* format

Weka implements more than twenty different algorithms for classification, clustering, and association rules. The algorithms implemented in toolkits usually have a single table input format, which can vary a little among different toolkits, but can easily be adapted. When organized into the single table format, different techniques and algorithms can be applied to the same dataset. Figure 3 (right) shows an example of a spatial dataset preprocessed and transformed into the single table format to be mined with Weka.

For spatial DM, the single table/file represents the *target feature type* on which discovery will be performed. Each row in the table/file is an independent unit, i.e. a different instance of *target feature type* and each column is an item characterizing this unit. In the example in Figure 3 (right) the target feature type is *bairro,* such that each row of the file is a different *bairro*. The attributes *dengue, posto_comb_contains, hospitais_contains,* and *arroios_crosses* are relevant feature types spatially related to *bairro*. Details of the this transformation process can be found in [Bogorny et al 2006a]

The button *OpenDB* on the interface shown in Figure 3 (left), calls the interface shown in Figure 4 (right top), to connect a database. From this interface Weka connects to PostGIS through JDBC and opens the module GDPM through the new button “Geographic Data”. GDPM is a new java class completely independent, named GeographicData.java, stored into the weka/gui/gdpm directory. Weka-GDPM+ is an evolution of Weka-GDPM [Bogorny 2006c] which besides automatic geographic data preprocessing supports the definition as well as the elimination of well known geographic dependences, as will be explained later in this section.


Geographic Data

Schema: public

Target Feature: Build_Up_Area

Relevant Features: Disturbed_Soil, Ground_Surface, river, road, Trees

[✓ Use Dependencies]

Granularity Level: [✓ Feature Type] Feature Instance

Spatial Relations: [✓ Topology] Intersects Distance

Distance Value: Very Close <= 00 < Close <= 00 < Far

Status: Transformation...

weka.gui.GenericObjectEditor

weka.experiment.InstanceQuery

databaseURL	jdbc:postgresql://localhost:5432/spatialdata
password	
query	SELECT * from ?
sparseData	False
username	vania

Dependencies

tunel road

(tunel, road)
(bridge, road)

Figure 4. (left) GDPM GUI, (right top) GUI for database connection, (right bottom) dependence definition

The data preprocessing tasks start with the name of the database schema. “Load” is the button that retrieves all spatial feature types from the database table *geometry_columns* which belong to the given schema, as shown in the example in Figure 1. This relation stores all spatial relations which contain spatial attributes in a geographic database. Since the spatial feature types are loaded, the user may select the target feature type on which discovery will be performed and all relevant feature types.

To define geographic dependences, the user has to mark the check box which enables the button to specify geographic dependences. After clicking on the *dependence definition* button the interface shown in Figure 4 (right bottom) is called.

In the dependence definition windows there are two combo boxes that contain all database spatial feature types with an instance in the table *geometry_columns*. The user may choose one spatial feature type in each combo, and by pressing the *add* button, a new dependence will be created with the respective selected pair. The pair will then appear in the list of dependences in the same window. To remove any dependence already defined, the user must select the pair and click on the button *remove.*

After dependences have been defined, the button *OK* must be clicked in order to store the dependences into a database table named *knowledgeConstrains.*

The following step is to provide the type of spatial relationships and the granularity level in the window shown in Figure 4 (left). Our prototype automatically generates data at two granularity levels (feature instance and feature type), for distance, topological, and high level topological relationships (intersects), without any concept hierarchy. For distance relationships one or two distance parameters must be provided to generate "very_close", "close", and "far" relationships.

Since all parameters have been provided to the window shown in Figure 4 (left) and button *OK* is pressed, the first step performed is the dependence elimination. Before perform the spatial join step between the target feature type and each relevant feature type, GDPM searches for a dependence in the table *knowledgeConstraints.* When a dependence is found, the relevant feature type is removed from the set and the next relevant feature type is tested. After all dependences have been removed the spatial join and transformation modules start. Details about dependence elimination and the kind of non-interesting patterns generated can be found in [Bogorny et al 2006a, 2006b].

4.1 Weka Modifications

The Weka Toolkit was projected to work with JavaBeans. As a consequence, the correct point to insert the new code is not a trivial task. The PropertySheetPanel class is responsible to create the panels with their respective properties, which is the case of the panel shown in Figure 4 (right top), where the new button called "Geographic Data" was added. Weka creates all instances dynamically at the beginning when it is loaded, calling the PropertySheetPanel class for each instance. To create the new button only in the specific panel we added an if command that asks for an InstanceQuery object. The InstanceQuery is the bean used to create the panel, and it contains all information necessary to connect to the database. When the InstanceQuery is found, the new button is created. Additionally, it needed to know the InstanceQuery object, because the new module GDPM needs to get the information about the user´s connection of this object. Thus, a listener adapter inner class was created to maintain this object. The adapter class calls the module GDPM with the information of its InstanceQuery object.

4.2 New Classes and Methods

A new class named GeographicData.java was created into the weka\gui\gdpm directory. It is independent and does not affect any other class or the Weka original code. The main methods of this class are: topology, intersects, distance, and transformation, which respectively implement spatial relationships. The dependence elimination is implemented in another class, named Dependences.

5. Conclusions and Future Work

This work presented a solution for the problem of automatic geographic data preprocessing for data mining. The dependence elimination between the target feature type and relevant feature types reported in [Bogorny et 2006a, 2006b] has also been addressed and solved with Weka-GDPM+. The main advantage of extending an open source data mining toolkit is that all data mining algorithms implemented in this toolkit as well as the visualization GUI to evaluate for post-processing can be reused, without the necessity to develop a new tool for mining geographic databases. The next steps include the implementation of the algorithm Apriori-KC [Bogorny et 2006b] into Weka to eliminate geographic dependences among relevant feature types using the spatial association rule mining technique.

ACKNOWLEDGMENT

This research has been funded by CAPES, CNPQ, and the European Union (FP6-IST-FET program, Project n. FP6-14915, GeoPKDD: Geographic Privacy-Aware Knowledge Discovery and Delivery (http://www.geopkdd.eu).

6. References

Addrians, P. and Zantinge, D. (1996) "Data mining". Addison Wesley Longman, Harlow, England.

Bogorny, V. (2007). www.inf.ufrgs.br/~vbogorny/weka_gdpm.

Bogorny, V., Engel, P. M. and Alvares, L.O (2006a) "GeoARM – an interoperable framework to improve geographic data preprocessing and spatial association rule mining". In 18th SEKE', San Francisco, California, pages 79-84.

Bogorny, V.; Camargo, S.; Engel, P.; Alvares, L. O. (2006b) "Mining Frequent Geographic Patterns with Knowledge Constraints" In 14th ACM-GIS, Arlington, pages 139-146.

Bogorny, V.; Palma, A; Engel. P. ; Alvares, L.O. (2006c) "Weka-GDPM: Integrating Classical Data Mining Toolkit To Geographic Information Systems" In: SBBD Workshop WAAMD, Florianopolis, Brazil, pages 9-16.

Gutting, R. H. (1994). "An Introduction to Spatial Database Systems". The International Journal on Very Large Data Bases, V3 (4), (October), pages. 357 - 399.

Han, J. (1995). "Mining Knowledge at Multiple Concept Level". In: 4th CIKM, Baltimore, Maryland, Nov. 1995, pages 19-24.

Han, J., Koperski, K. and Stefanvic, N. (1997) "GeoMiner: a system prototype for geographic data mining". In ACM-SIGMOD, ACM Press, Arizona, pages 553-556.

Malerba, D. et al. (2000). "Discovering geographic knowledge: the INGENS system". In Foundations of Intelligent Systems, 12th International Symposium, (ISMIS), Lecture Notes in Artificial Intelligence, 1932, 40-48, Springer, Berlin, Germany.

OGC (1999a). "OpenGIS simple features specification for SQL". http://www.opengeogeographic.org/docs/99-054.pdf, August 2005.

OGC (1999b). "Topic 5, the OpenGIS abstract specification – OpenGIS features – Version 4". http://www.OpenGIS.org/techno/specs.htm. August 2005.

PostGIS (2007) http://postgis.refractions.net/

Weka (2007) http://www.cs.waikato.ac.nz/ml/weka/

Witten, I. and Frank, E. (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco.