10 Leveraging the Power of Spatial Data Mining to Enhance the Applicability of GIS Technology

Donato Malerba
Dipartimento di Informatica, Università
degli Studi di Bari, Bari, Italy

Antonietta Lanza
Dipartimento di Informatica, Università
degli Studi di Bari, Bari, Italy

Annalisa Appice
Dipartimento di Informatica, Università
degli Studi di Bari, Bari, Italy

CONTENTS

ABSTRACT

The strength of a geographic information system (GIS) is in providing a rich data infrastructure for combining disparate data in meaningful ways, by using a spatial arrangement (e.g., proximity). As a toolbox, a GIS allows planners to perform spatial analysis using geo-processing functions, such as map overlay, connectivity measurements, or thematic map coloring. Although this makes the geographic visualization of individual variables effective, complex multi-variate dependencies are easily overlooked. The required step to take GIS beyond a tool for automating cartography is to incorporate the ability of analyzing and condensing a large number of geo-referenced variables into a single forecast or score. This is where spatial data mining promises great potential benefits and the reason why there is such a hand-in-glove fit between GIS and data mining facilities. INGENS 2.0 is a prototype GIS which resorts to emerging spatial data mining technology to support geographers, geologists, and town planners in discovering (descriptive and predictive) patterns, which are never explicitly represented in topographic maps or in a GIS-model and are useful in the task of topographic map interpretation. In spatial data mining, spatial dimension adds a substantial complexity to the data mining task. First, spatial objects are characterized by a geometrical representation and relative positioning with respect to a reference system, which implicitly define spatial properties. Modeling these implicit spatial properties (attributes and relations) in order to associate them with clear semantics and a set of eficient procedures for their computation is the first challenge to be met when facing a spatial data mining problem. Second, spatial phenomena are characterized by autocorrelation, i.e., observations of spatially distributed random variables are not location-independent. Third, spatial objects can be considered at different levels of abstraction (or granularity). Spatial data mining facilities in INGENS deal with these challenges in both inducing classification rules and discovering association rules from spatial data. The spatial data mining process is aimed at a user who controls the parameters of the process by means of a query written in SDMOQL, a spatial data mining query language that permits the specification of the task-relevant data, the kind of knowledge to be mined, the background knowledge and the hierarchies and the interestingness measures. Some constraints on the query language are identified by the particular mining task. An application to a real repository of topographic maps is briefly illustrated.

10.1 INTRODUCTION

In a large number of application domains (e.g., traffic and fleet management, environmental and ecological modeling), collected data are measurements of one or more attributes of objects that occupy specific locations with respect to the Earth's surface. Collected geographic objects are characterized by a geometry (e.g., point, line, or polygon) which is formulated by means of a reference system and stored under a geographic database management system (GDBMS). The geometry implicitly defines both spatial properties, such as orientation, and spatial relationships of a different nature, such as topological (e.g., intersects), distance, or direction (e.g., north of) relations.

A GIS, is the software system that provides the infrastructure for editing, storing, analyzing, and displaying geographic objects, as well as related data on geoscientific, economic, and environmental situations [11]. Popular GISs (e.g., ArcView, MapInfo, and Open GIS) have been designed as a toolbox that allows planners to explore geographic data by means of geo-processing functions, such as zooming, overlaying, connectivity measurements, or thematic map coloring. Consequently, these GISs are provided with functionalities that make the geographic visualization of individual variables effective, but overlook complex multi-variate dependencies. Traditional GIS technology does not address the requirement of complex geographic libraries which search for relevant information, without any *a priori* knowledge of data set organization and content. In any case, GIS vendors and researchers now recognize this limitation and have begun to address it by adding spatial data interpretation capabilities to the systems.

A first attempt to integrate a GIS with a knowledge-base and some reasoning capabilities is reported in [43]. Nevertheless, this system has a limited range of applicability for a variety of reasons. First, providing the GIS with operational definitions of some geographic concepts (e.g., morphological environments) is not a trivial task. Generally only declarative and abstract definitions, which are difficult to compile into database queries, are available. Second, the operational definitions of some geographic objects are strongly dependent on the data model adopted for the GIS. Finding relationships between density of vegetation and climate is easier with a raster data model, while determining the usual orientation of some morphological elements is simpler in a topological data model [15]. Third, different applications of a GIS will require the recognition of different geographic elements in a map. Providing the system in advance with all the knowledge required for its various application domains is simply illusory, especially in the case of wide-ranging projects like those set up by governmental agencies.

The solution to these difficulties can be found in spatial data mining [22], which investigates how interesting, but not explicitly available, knowledge (or pattern) can be extracted from spatial data. This knowledge may include classification rules, which describe the partition of the database into a given set of classes [22], clusters of spatial objects [19, 42], patterns describing spatial trends, that is, regular changes of one or more non-spatial attributes when moving away from a given start object [26], and subgroup patterns, which identify subgroups of spatial objects with an unusual, an unexpected, or a deviating distribution of a target variable [21].

Following the mainstream of research in spatial data mining, there have been several atttempts to enhance the applicability of GIS technology by leveraging the power of spatial data mining [6, 16, 18, 32, 34]. In all these cases, the GIS users are not interested in processing the geometry of geographic objects collected in spatial database, but in working at higher conceptual levels, where human-interpretable properties and relationships between geographic objects are expressed.[1] To bridge the gap between geometrical representation and conceptual representation of

[1] A typical example is represented by the possible relations between two roads, which either cross each other, or run parallel, or can be confluent, independently of the fact that they are geometrically represented as "lines" or regions in a map.

geographic objects, GISs are provided with facilities to compute the properties and relationships (features), which are implicit in the geometry of data. In most cases, these features are then stored as columns of a single double entry data table (or relational table), such that a classical data mining algorithm can be applied to transformed data within the GIS platform. Unfortunately, the representation in a single double entry data table offers inadequate solutions with respect to spatial data analysis requirements. Indeed, information on the original heterogeneous structure of geographic data is partially lost: for each unit of analysis, a single row is constructed by considering the geographic objects which are spatially related to the unit of analysis. Properties of objects of the same type are aggregated (e.g., by sum or mode) to be represented in a single value.

In this chapter, we present a prototype of GIS, called INGENS 2.0, that differs from most existing GISs in the fact that the data mining engine works in a first-order logic, thus providing functionalities to navigate relational structures of geographic data and generate potentially new forms of evidence. Originally built around the idea of applying the classification patterns induced from georeferenced data to the task of topographic map interpretation [31], INGENS 2.0 now extends its predecessor INGENS [32] by combining several technologies, such as spatial DBMS, spatial data mining, and GIS within an open extensible Web-based architecture. Vectorized topographic maps are now stored in a spatial database [40], where mechanisms for accessing, filtering, and indexing spatial data are available free of charge for the GIS requests. Data mining facilities include the possibility of discovering operational definitions of geographic objects (e.g., fluvial landscape) not directly stored in the GIS database, as well as regularities in the spatial arrangement of geographic objects stored in the GIS database. The former are discovered in the form of classification rules, while the latter are discovered in the form of association rules. The operational definitions can then be used for predictive purpose, that is, to query a new map and recognize instances of geographic objects not directly modeled in the map itself. Efficient procedures are implemented to model spatial features not explicitly encoded in the spatial database. Such features are associated with clear semantics and represented in a first-order logic formalism. In addition, INGENS 2.0 integrates a spatial data mining query language, called SDMOQL [28], which interfaces users with the whole system and hides the different technologies. The entire spatial data mining process is condensed in a query written in SDMOQL and run on the server side. The query is graphically composed by means of a wizard on the client side. The GUI (graphical user interface) is a Web-based application that is designed to support several categories of users (administrators, map managers, data miners, and casual users) and allows them to acquire, update, or navigate vectorized maps stored in the spatial database, formulate SDMOQL queries, explore data mining results, and so on. Logging data and the history of users are maintained in the database.

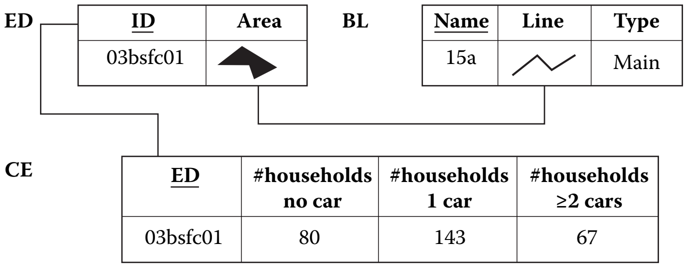
The chapter is organized as follows. In the next section, we discuss issues and challenges of leveraging the power of spatial data mining to enhance the applicability of GIS technology. We present the architecture and data model of INGENS 2.0 in Section 10.3 and the spatial data mining process in Section 10.4. The syntax of SDMOQL is described in Section 10.5. An application of INGENS 2.0 is reported and discussed in Section 10.6. Finally, Section 10.7 gives conclusions and presents ideas for further work.

10.2 SPATIAL DATA MINING AND GIS

Empowering a GIS with spatial data mining facilities presents some difficulties, since the design of a spatial data mining module depends on several aspects. The first aspect is the representation of spatial objects. In the literature, there are two types of data representations for the spatial data, that is, tessellation and vector [39]. They differ in storage, precision, and complexity of the spatial relation computation. The second aspect is the implicit definition of spatial relationships among objects. The three main types of spatial relationships are topological, distance, and directional relationships, for which several models have been proposed for the definition of their semantics (e.g., "9-intersection model" [14]). The third aspect is the heterogeneity of spatial objects. Spatial patterns often involve different types of objects (e.g., roads or rivers), which are described by completely different sets of features. The fourth aspect is the interaction between spatially close objects, which introduces different forms of spatial autocorrelation: spatial error (correlations across space in the error term), and spatial lag (the dependent variable in space *i* is affected by the independent variables in space *i*, as well as those, dependent or independent, in space *j*).

Classical data mining algorithms, such as those implemented in Weka [45], offer inadequate solutions with respect to these aspects. In fact, they work under the single table assumption [46], that is, units of analysis are represented as rows of a classical double-entry table (or database relation), where columns correspond to elementary (nominal, ordinal, or numeric) single-valued attributes. In any case, this representation neither deals with geographic data characterized by geometry, nor handles observations belonging to separate relations, nor naturally represents spatial relationships, nor takes them into account when mining patterns. Differently, geographic (or spatial) data are naturally modeled as a set of relations $R_1,...,R_n$, such that each R_i has a number of elementary attributes and possibly a geometry attribute (in which case a relation is a layer). In this perspective, a (multi-)relational data mining approach seems the most suitable for spatial data mining tasks, since (multi)relational data mining tools can be applied directly to data distributed on several relations and since they discover relational patterns [13].

Example. To investigate the social effects of public transportation in a British city, a geographic data set composed of three relations is considered (see Figure 10.1). The first relation, ED, contains information on enumeration districts, which are the smallest areal units for which census data are published in the U.K. In particular, ED has two attributes, the identifier of an enumeration district and a geometry attribute (a closed polyline), which describes the area covered by the enumeration district. The second relation, BL, describes all the bus lines which cross the city. In this case, relevant attributes are the name of a bus line, the geometry attribute (a line) represents the route of a bus and the type of bus line (classified as main or secondary). The third relation, CE, contains some census data relevant for the problem, namely, the number of households with 0, 1, or "more than 1" cars. This relation also includes



ED

ID	Area
03bsfc01	

BL

Name	Line	Type
15a		Main

CE

ED	#households no car	#households 1 car	#households ≥2 cars
03bsfc01	80	143	67

FIGURE 10.1 Representation of geographic data on the social effects of public transportation in a British city.

the identifier of the enumeration district, which is a foreign key for the table ED. A unit of analysis corresponds to an enumeration district (the target object), which is described in terms of the number of cars per household and crossing bus lines (bus lines are the task-relevant objects). The relationship between reference objects and task-relevant objects is established by means of a spatial join, which computes the intersection between the two layers ED and BL.

Although several spatial data mining methods have already been designed by resorting to the multi-relational approach [4, 7, 21, 29, 30], most GISs which integrate data mining facilities [6, 16, 18] continue to frame the requests made by the spatial dimension within the classical data mining solution. Spatial properties and relationships of geographic objects are computed and stored as columns of a classical double-entry table, such that a classical data mining algorithm can be applied to the transformed data table.

At present, only two of the GISs reported in the literature integrate spatial data mining algorithms designed according to the multi-relational approach. They are SPIN! [34] and INGENS [32]. SPIN! is the spatial data mining platform developed within the EU research project of the same name. SPIN! assumes an object-relational data representation and offers facilities for multi-relational sub-group discovery and multi-relational association rule discovery. Subgroup discovery [21] is approached by taking advantage of a tight integration of the data mining algorithm with the database environment. Spatial relationships and attributes are then dynamically derived by exploiting spatial DBMS extension facilities (e.g., packages, cartridges, or extenders) and used to guide the subgroup discovery. Association rule discovery [4] works in first-order logic and is only loosely integrated with a spatial database by means of some middle layer module that extracts spatial attributes and relationships independently of the mining step and represents these features in a first-order logic formalism. INGENS is our first attempt to empower a GIS with inductive learning capabilities. Indeed, it integrates the inductive learning system, ATRE, which can induce first-order logic descriptions of some concepts from a set of training examples. INGENS assumes an object-oriented representation of data organized in topographic maps. The geographic data collection is organized according to an object-oriented data model and is stored in the object store object oriented DBMS. Since object store does not provide automatic facilities for storing, indexing, and retrieving geographic objects, these facilities are completely managed by the GIS. In addition, INGENS integrates a Web-based GUI, where the user is simply asked to provide a set of (counter-) examples of geographic concepts of interest and a number of parameters that define the classification task more precisely. First-order descriptions learned by ATRE are only visualized in a textual format. The data mining process is condensed in a query written in SDMOQL [28], but the textual composition of the query is completely managed by the user.

10.3 INGENS 2.0 ARCHITECTURE AND SPATIAL DATA MODEL

The architecture of INGENS 2.0 is illustrated in Figure 10.2. It is designed as an open, highly extensible, Web-based architecture, where spatial data mining services are integrated within a GIS environment. The GIS functionalities are distributed among the following software components:

- a Web-based *GUI* for supporting users in all activities, that is, user log-in and log-out, acquisition and editing of a topographic map, visualization and exploration of a topographic map, execution of a data mining request formulated by means of a spatial data mining query;
- the *User Management* module for managing the access to the GIS (user creation, authentication, and history) for the different categories of users;
- the *Map Management* module for managing requests of map creation, acquisition, update, delete, visualization, and exploration;
- the *Query Interpreter* module for running user-composed SDMOQL queries and performing a spatial data mining task of classification or association rule discovery;
- the *Feature Extractor* module for automatically generating conceptual descriptions (in first-order logic) of geographic objects, by making

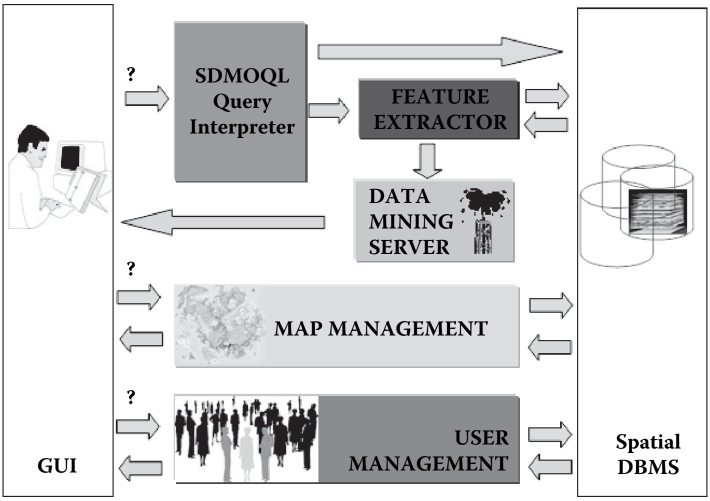


FIGURE 10.2 INGENS 2.0 software architecture.

explicit (spatial) properties and relationships, which are implicit in the spatial dimension of data;
- the *Data Mining Server* for running data mining algorithms;
- the *Spatial Database* for storing both map data and information on the user history (logging user identifier and password, privileges, and spatial data mining queries executed in the past).

The GUI can be accessed by four categories of users, namely, the GIS administrators, the map maintenance users, the data miners, and the casual end users. User profiles (e.g., authentication information, list of privileges) are stored in the database. The profile lists the topographic maps and the GIS functionalities to be accessed by the user. The administrator is the only user authorized to create, delete, or modify profiles of all other users of the GIS. The map maintenance user is in charge of upgrading the map repository stored in the spatial database by creating, updating, or deleting a map. The data miner can ask the GIS to discover either the operational definition of a geographic object or a spatial arrangement of geographic objects that are frequent on the topographic map under analysis. Finally, the casual end user is provided with geo-processing functionalities to navigate the topographic map, visualize geographic objects, belonging to one or more map layers (roads, parcels, and so on), and perform zooming operations.

The user management module is in charge of the activities of creating, modifying, or deleting a user profile. Users are authorized to use only the GIS functionalities that match the privileges provided in their profiles.

The map management module executes the requests of the map maintenance users. This component interfaces with the spatial database in order to create or drop an instance of a topographic map, as well as retrieve and display geographic objects belonging to one or more layers of a map.

The query interpreter runs the SDMOQL queries composed by data miners. A query refers to one of the topographic maps accessible to the data miner and specifies the set of objects relevant to the task at hand, the kind of knowledge to be discovered (classification or association rules), the set of descriptors to be extracted from the map, the set of descriptors to be used for pattern description and optionally the background knowledge to be used in the discovery process, the geographic hierarchies, and the interestingness measures for pattern evaluation. The query interpreter's responsibility is to ask the feature extractor to generate conceptual descriptions of the geographic objects extracted from the spatial database and then to invoke the inference engine of the data mining server. The conceptual descriptions are conjunctive formulae in a first-order logic language, involving both spatial and non-spatial descriptors specified in the query. SDMOQL queries are maintained in the user workspace and can be reconsidered in a new data mining process. Due to the complexity of the SDMOQL syntax, a user-friendly wizard is designed on the GUI side to graphically support data miners in formulating SDMOQL queries.

The data mining server provides a suite of data mining systems that can be run concurrently by multiple users to discover previously unknown, useful patterns in geographic data. Currently, the data mining server provides data miners with two systems, ATRE [27] and SPADA [24]. ATRE is an inductive learning system that generates models of geographic objects from a set of training examples and counter-examples. SPADA is a spatial data mining system to discover multi-level spatial association rules, that is, association rules involving spatial objects at different granularity levels. In both cases, discovered patterns are returned to the GUI to be visualized and interpreted by data miners.

The spatial database (SDB) can run on a separate computational unit, where topographic maps are stored according to an object-relational data model. The object-relational DBMS used to store data is a commercial one (Oracle 10*g*) that includes spatial cartridges and extenders, so that full use is made of a well-developed, technologically mature spatial DBMS. Moreover, the object-relational technology facilitates the extension of the DBMS to accommodate management of geographic objects.

At a conceptual level, the geographic information is modeled according to an object-based approach [41], which sees a topographic map as a surface littered with distinct, identifiable, and relevant objects that can be punctual, linear, or surfacic. Interactions between geographic objects are then described by means of topological, directional, and distance-based operators. In addition, geographic objects are organized in a three-level hierarchy expressing the semantics of geographic objects independently of their physical representation (see Figure 10.3). The entity object is a total generalization of eight distinct entities, namely, hydrography, orography, land administration, vegetation, administrative (or political) boundary, ground transportation network, construction, and built-up area. Each of these is in turn a generalization, for example, administrative boundary generalizes the entity's city, province, county, or state.

At a logical level, geographic information is represented according to a hybrid model, which combines both a tessellation and a vector model [39]. The tessellation

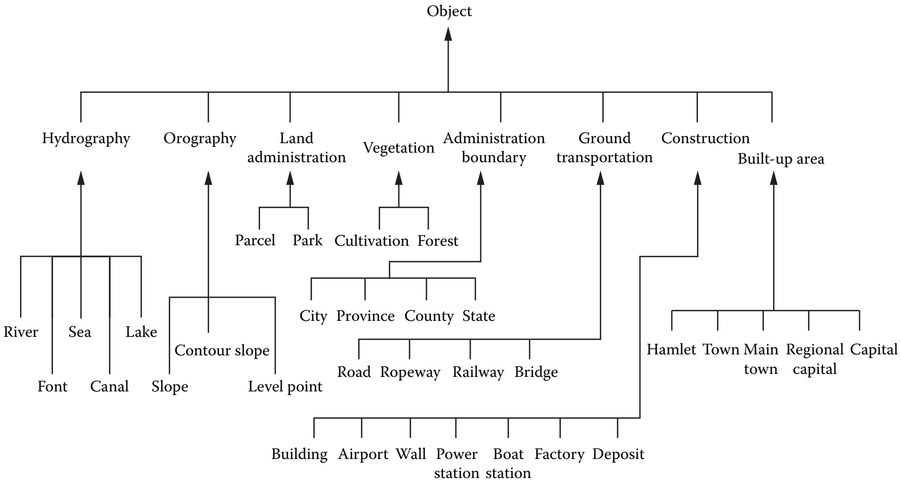


FIGURE 10.3 Hierarchical representation of geographic objects at different levels of granularity.

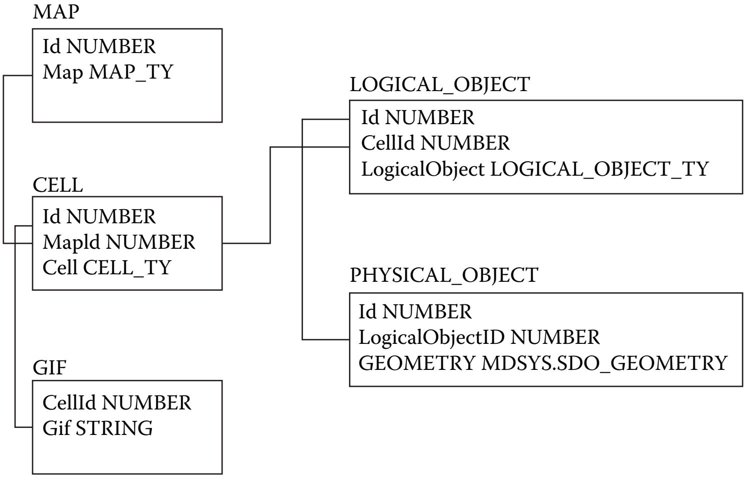


FIGURE 10.4 Spatial data schema.

model partitions the space into a number of cells, each of which is associated with a value of a given attribute. No variation is assumed within a cell and values correspond to some aggregate function (e.g., average) computed on the original values in the cell. A grid of square cells is a special tessellation model called raster. In the vector model the geometry is represented by a vector of coordinates, which define points, lines, or polygons. Both data structures are used to represent geographic information in INGENS 2.0. The partitioning of a map into a grid of square cells simplifies the localization and indexing process. For each cell, the raster image in GIF format is stored, together with its coordinates and component geographic objects. These are represented by a vector of coordinates stored in the field *Geometry* of the database relation PHYSICAL OBJECT (see Figure 10.4), while their semantics are defined in the field *LogicalObject* of the database relation LOGICAL OBJECT. A foreign key constraint relates each tuple of PHYSICAL OBJECT to one tuple of LOGICAL OBJECT. Type inheritance is exploited to represent the conceptual hierarchy in Figure 10.3 at the logical level. Indeed, the type of the attribute *LogicalObject* (LOGICAL_OBJECT_TY) has eight subtypes, namely, HYDROGRAPHY_TY, OROGRAPHY_TY, LAND_ADMINISTRATION_TY, VEGETATION_TY, ADMINISTRATIVE_BOUNDARY_TY, GROUND_TRANSPORTATION_TY, CONSTRUCTION_TY, and BUILDUP_AREA_TY. Each of these is in turn a generalization of new types according to the conceptual hierarchy.

Spatial and non-spatial features can be extracted from geographic objects stored in the SDB. Feature extraction requires complex data transformation processes to make spatial properties and relationships explicit. This task is performed by the feature extractor module, which makes possible a loose coupling between data mining services and the SDB. The feature extractor module is implemented as an Oracle package of PL/SQL functions to be used in the spatial SQL queries.

10.4 SPATIAL DATA MINING PROCESS IN INGENS 2.0

In INGENS 2.0 the spatial data mining process is activated and controlled by means of a query expressed in SDMOQL (see Figure 10.5). Initially, the query is syntactically and semantically validated. Then the feature extractor generates the conceptual representation of the geographic objects selected by the query. This representation, which is in a first-order logic language, is input to multi-relational data mining systems, which return spatial classification rules or association rules. Finally, the results of the mining process are presented to the user.

10.4.1 Conceptual Description Generation

A set of descriptors used in INGENS 2.0 is reported in Table 10.1. They are either spatial or non-spatial. According to their nature, spatial descriptors can be classified as follows:

1. Geometrical, if they depend on the computation of some metric/distance. Their domain is typically numeric, for example, "extension."
2. Topological, if they are invariant under the topological transformations (translation, rotation, and scaling). The type of their domain is nominal, for example, "region_to_region" and "point_to_region."
3. Directional, if they concern orientation. The type of their domain can be either numerical or nominal, for example, "geographic direction."
4. Locational, if they concern the location of objects. Locations are represented by numeric values that express coordinates. There are no examples of locational descriptors in Table 10.1.

Some spatial descriptors are hybrid, in the sense that they merge properties of two or more of the above categories. For instance, the descriptor "line_to_line" that

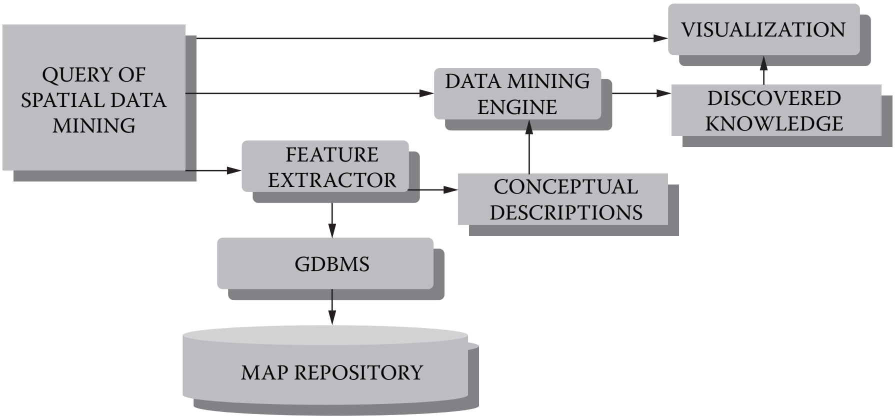


FIGURE 10.5 Spatial data mining process in INGENS 2.0.

TABLE 10.1
Set of Descriptors Extracted by the Feature Extractor

Feature	Meaning	Value
contain(C,L)	Cell C contains a logical object L	{true, false}
part_of(L,F)	Logical object L is composed of physical object F	{true, false}
type_of(L)	Type of L	33 nominal values (e.g., river, road, ...)
color(L)	Color of L	{blue, brown, black}
area(F)	Area of F	[0..MAX_AREA]
extension(F)	Extension of F	[0..MAX_EXT]
geographic_direction(F)	Geographic direction of F	{north-east, north-west, east, north}
line_shape(F)	Shape of the linear object F	{straight, curvilinear, cuspidal}
altitude(F)	Altitude of F	[0.. MAX_ALT]
line_to_line(F1,F2)	Spatial relation between lines F1 and F2	{almost parallel, al most perpendicular}
distance(F1,F2)	Distance between lines F1 and F2	[0..MAX_DIST]
region_to_region(F1,F2)	Spatial relation between regions F1 and F2	{disjoint, contains, in side, equal, meet, covers, covered by, over lap}
line_to_region(F1,F2)	Spatial relation between a line F1 and a region F2	{along edge, intersect}
point_to_region(F1, F2)	Spatial relation between a point F1 and a region F2	{inside, outside, on boundary, vertex (i.e., F1 is a vertex of F2)}

expresses conditions of parallelism and perpendicularity is both topological (it is invariant with respect to translation, rotation, and scaling) and geometrical (it is based on the angle of incidence).

In INGENS 2.0, geographic objects can also be described by two non-spatial descriptors, namely “type_of” and “color.” The former describes the type of a geographic object, according to the layer (street, parcel, river, and so on) it belongs to, while the latter describes the color (blue, black, or brown) used in the visualization of a geographic object. The descriptor “part_of” describes the structure of complex geographic objects, i.e., a geographic object can be formed by physical component objects, represented by separate geometries.

There is no common mechanism to express the semantics of such different features. The semantics of topological relationships are based on the 9-intersection model [14], while the semantics of other features are based on mathematical methods of 2D-graphics [37] as described in [23].

Example (Geographic Direction). Let *o* be a geographic object associated with a line, that is,

$$o : \{P_1 = (x_1, y_1), \ldots, P_n = (x_n, y_n)\}.$$

If α is the angle defined by the straight line L connecting P_1 and P_n, that is:

$$\alpha = arctg\frac{x_n - x_1}{y_n - y_1},$$

then the geographic direction of o is computed as follows:

$$\begin{array}{ll} \text{north} & \text{if } \alpha > \left(\frac{\pi}{2} - \frac{\pi}{8}\right) \vee \alpha \leq -\left(\frac{\pi}{2} - \frac{\pi}{8}\right) \\ \text{north east} & \text{if } \alpha \leq \left(\frac{\pi}{2} - \frac{\pi}{8}\right) \wedge \alpha > -\frac{\pi}{8} \\ \text{east} & \text{if } \alpha \leq \frac{\pi}{8} \wedge \alpha > -\frac{\pi}{8} \\ \text{northwest} & \text{if } \alpha \leq -\frac{\pi}{8} \wedge \alpha > -\left(\frac{\pi}{2} - \frac{\pi}{8}\right). \end{array}$$

This feature is computed only for geographic objects physically represented as lines.

10.4.2 Classification Rule Discovery

Classification of geographic objects is a fundamental task in spatial data mining and GIS, where training data consist of multiple target geographic objects (reference objects), possibly spatially related with other non-target geographic objects (task-relevant objects). The goal is to learn the concept associated with each class on the basis of the interaction of two or more spatially referenced objects or space-dependent attributes [22].

While a lot of research has been conducted on classification, only a few works deal with geographic classification. GISs empowered with classification facilities are reported in [6, 18]. These systems allow the learning of a classifier from data stored in a classical double-entry table (single-table assumption [46]). This is a severe restriction in GIS applications, where different geographical objects have different features (properties and relationships), which are properly modeled by as many data relations as the number of object types. To map the natural multi-relational form of geographic data into a single double-entry data table, GISs must integrate a transformation module that is in charge of computing the spatial features of geographic objects (e.g., a street crosses a river) and store them as columns of the double-entry table. This table can then be input to a wide range of robust and well-known classification methods which operate on a single table. This transformation (known as propositionalization) presents some drawbacks. In fact, the full equivalence between the original and the transformed training sets is possible only in special cases. However, even when possible, the output table size is unacceptable in practice [10] and some form of feature selection is required. Therefore, the transformed problem is different from the original one, for pragmatic reasons [7].

On the other hand, INGENS 2.0 overcomes the limitations of single table assumption by integrating a classification system, named ATRE [27], which resorts to a multi-relational data mining approach [13] to classify geographic objects. Indeed, a multi-relational approach to data mining (or MRDM) looks for patterns that involve multiple relations of a relational data representation. Thus, data taken as input by these approaches typically consist of several relations and not just a single one, as is the case in most existing data mining approaches. Patterns found by these approaches are called relational and are typically stated in a more expressive language than patterns defined in a single data table. Typically, subsets of first-order logic, which is also called predicate calculus or relational logic, are used to express relational patterns. In this way, the expressive power of predicate logic is exploited to represent both spatial relationships and background knowledge, thus providing functionalities to navigate relational structures of geographic data and generate potentially new forms of evidence, not readily available in flattened single double entry data table representation.

The problem solved by ATRE is formalized as follows:

Given

- a set of concepts $C_1, C_2, \ldots, C_r$ to be learned;
- a set of units of analysis (or observations) O described in a language $\mathcal{L}_O$;
- a background knowledge BK described in a language $\mathcal{L}_{BK}$;
- a language of hypotheses $\mathcal{L}_H$ that defines the space of hypotheses S_H;
- a user's preference criterion PC.

Find a logical theory $T \in S_H$, defining the concepts $C_1, C_2, \ldots, C_r$, such that T is complete and consistent with respect to the set of observations and satisfies the preference criterion PC.

The logical theory T is a set of first-order definite clauses [25], such as:

cell(X1)=fluvial_landscape ←
    contain(X1,X2)=true, type_of(X2)=river, part_of(X2,X3)=true,
    line_to_line(X4,X3)=almost_parallel, part_of(X5,X4), type_of(X5)=street

This clause can be interpreted easily as follows: If a cell X1 contains a river X2 with X2 represented by the line X3 and X3 almost parallel to the line X4 that represents a street X5, then the cell X1 can be classified as a "fluvial landscape." This clause contains an operational definition of the fluvial landscape morphology. This definition can be used to recognize the unknown morphology for the cells of a new topographic map.

The units of analysis are represented by means of a ground clause[2] called objects. For example, if the units of analysis are the cells (reference objects) of a topographic map, then the body of an object describes the spatial arrangement of the geographic objects (task-relevant objects) within the cell, while the head may describe the landscape morphology (class) associated with the cell. The literal in the head of the clause is an example (either positive or negative) of the concepts $C_1, C_2, \ldots, C_r$.

[2] A ground clause contains no variables.

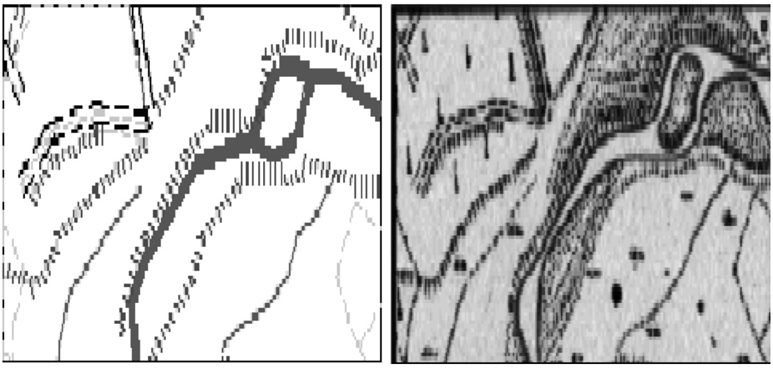


FIGURE 10.6 Raster and vector representation (above) and symbolic description of a cell (below). The cell is an example of a territory where there is a fluvial landscape. The cell is extracted from a topographic chart (Canosa di Puglia 176 IV SW—Series M891) produced by the Italian Geographic Military Institute (IGMI) at scale 1:25,000 and stored in INGENS 2.0.

An instance of an object is reported in Figure 10.6, where the constant c8 denotes the whole cell, while the remaining constants (e.g., rv1_8, pc473_0, x20_8,…) denote the logical (river, street, parcel) or geometrical (line, point or polygon) component of the geographic objects in the cell. The descriptor cell(X) in the head denotes the known value of the morphology of the territory covered by the cell.

The background knowledge *BK* can be defined in the form of first-order definite clauses, which allow the definition of new descriptors not explicitly encoded in a conceptual description of objects. An example of a clause that is part of a *BK* is the following:

```
parcel_to_parcel(A,B)=C ←type_of(A)=parcel,
            type_of(B)=parcel, part_of(A,D)=true,
            part_of(B,E)=true, region_to_region(D,E)=C
```

This clause allows the relationship *C* between two regions *D* and *E* to be automatically renamed as "parcel_to_parcel," when *D* and *E* are parts of two parcels *A* and *B*.

The completeness property of the output theory *T* holds when *T* explains all observations in *O* of the *r* concepts C_i, while the consistency property holds when *T* explains no counter-example in *O* of any concept C_i. The satisfaction of these properties guarantees the correctness of the induced theory with respect to *O*, but not necessarily with respect to new unseen observations. The selection of the clause in *T* is made on the grounds of an inductive bias [35], expressed in the form of preference criterion (*PC*). For example, clauses that explain a high number of positive examples and a low number of negative examples can be preferred to others.

At the high-level, the learning strategy implemented in ATRE is sequential covering (or separate-and-conquer) [35], that is, one clause is learned (conquer stage), covered examples are removed (separate stage), and the process is iterated on the remaining examples. The conquer stage of this algorithm aims to generate a clause that covers a specific positive example, called *seed*. The most important novelty of the learning strategy implemented in ATRE is embedded in the design of the conquer stage. Indeed, the separate-and-conquer strategy is traditionally adopted by single concept learning systems that generate clauses with the same literal in the head at each step. In ATRE, clauses generated at each step may have different literals in their heads. In addition, the body of the clause generated at the *i*-th step may include all literals corresponding to those target concepts C_1, C_2,..., C_r for which at least a clause has been added to the partially learned theory in previous steps. In this way, dependencies between target concepts can be automatically discovered. An example of a logical theory, where the dependency between concepts "downtown" and "residential" is handled, is reported in the following:

class(X)=downtown ←
  on_the_sea(X)=true, business_activity(X)=high.

class(X)=residential ←
  contain(X,Y)=true, type_of(Y)=kindergarten, shopping_activity(X)=high.

class(X)=residential ←
  close to(X,Y)=true, *class(Y)=downtown,* business_activity(X)=low.

The order in which clauses of distinct target concepts have to be generated is not known in advance. This means that it is necessary to generate clauses with different literals in the head and then to pick one of them at the end of each step of the separate-and-conquer strategy. Since the generation of a clause depends on the chosen seed, several seeds have to be chosen, such that at least one seed per incomplete concept definition is kept. Therefore, the search space is actually a forest of as many search-trees (called specialization hierarchies) as the number of chosen seeds. A directed arc from a node (clause) C to a node C' exists if C' is obtained from C by adding a literal (single refinement step).

The forest can be processed in parallel by as many concurrent tasks as the number of search-trees (hence, the name of separate-and-parallel-conquer for this search strategy). Each task traverses the specialization hierarchy top-down (or general-to-specific), but synchronizes traversal with the other tasks at each level. Initially, some clauses at depth one in the forest are examined concurrently. Each task is actually free to adopt its own search strategy, and to decide which clauses are worth testing. If none of the tested clauses is consistent, clauses at depth two are considered. The search proceeds toward deeper and deeper levels of the specialization hierarchies until at least a user-defined number of consistent clauses is found. Task synchronization is performed after all "relevant" clauses at the same depth have been examined. A supervisor task decides whether the search should carry on or not, on the basis of the results returned by the concurrent tasks. When the search is stopped, the supervisor selects the "best" consistent clause according to the user's preference criterion. This separate-and-parallel-conquer search strategy provides us with a solution to the problem of interleaving the induction process for distinct concept definitions. It has the advantage that simpler consistent clauses are found first, independently of the predicates to be learned. Moreover, the synchronization allows tasks to save much computational effort when the distribution of consistent clauses in the levels of the different search-trees is uneven. A more detailed description of the search strategy implemented in ATRE and its optimization through caching techniques is reported in [5, 27].

10.4.3 Association Rule Discovery

Association rules are a class of regularities introduced by Agrawal and Srikant [1], which can be expressed by an implication of the form:

$$A \Rightarrow C(s, c),$$

where A(antecedent) and C(consequent) are sets of atoms, called *items*, with $A \cap B = \phi$. s is called support and estimates the probability $p(A \cup C)$, while c is called confidence and estimates the probability $p(C|A)$. A pattern P $(s\%)$ is *frequent* if $s \geq minsup$. An association rule $A \Rightarrow C$ $(s\%, c\%)$ is *strong* if the pattern $A \cup C$ $(s\%)$ is frequent and $c \geq minconf$. We call an association rule $A \Rightarrow C$ spatial, if $A \cup C$ is a spatial pattern, that is, it expresses a spatial relationship among spatial objects.

The problem of mining spatial association rules was originally tackled by Koperski [22], who implemented the module geo-associator of the spatial data mining system GeoMiner [18]. Similar to the classification task, the method implemented in geo-associator suffers from the limitations due to adapting the restrictive single-table data representation to the case geographic data. Weka-GPDM [6] is a further example of a GIS that includes facilities to discover spatial association rules. Once again, spatial features are extracted in a preprocessing step and stored as features of a single double-entry data table. Association rules are discovered in another step by applying the conventional association rule discovery algorithm included in Weka [45] to the single double-entry data table.

Similar to the classification case, INGENS 2.0 overcomes limitations of single table assumption by integrating an association rule discovery system, named SPADA [24], which exploits the expressive power of a predicate logic to deal with spatial relationships in the original relational form. In addition, SPADA automatically supports a multiple-level analysis of geographic data. Indeed, geographic objects are organized in hierarchies of classes. By descending or ascending through a hierarchy, it is possible to view the same geographic object at different levels of abstraction (or granularity). Confident patterns are more likely to be discovered at low granularity levels. On the other hand, large support is more likely to exist at higher granularity levels. In general, the discovery of multi-level *patterns* (e.g., the most supported and confident) can be performed by forcing users to repeat independent experiments on different representations. In this way, results obtained for high granularity levels are not used at low granularity levels (or vice versa). Conversely, SPADA is able to explore altogether the search space at different granularity levels, such that patterns obtained for high granularity levels are used to control search at low granularity levels.

The problem solved by SPADA is formalized as follows:
Given

- a set S of reference objects, which is the main subject of the analysis,
- some sets R_k, $1 \leq k \leq m$ of task-relevant objects,
- a background knowledge *BK* including spatial hierarchies H_k on objects in R_k,
- M granularity levels in the descriptions (1 is the highest, while M is the lowest),
- a set of granularity assignments ψ_k, which associate each object in H_k with a granularity level to deal with several hierarchies at once,
- a couple of thresholds *minsup*[*l*] and *minconf*[*l*] for each granularity level *l*,
- a language bias *LB* which constrains the search space.

Find strong spatial association rules for each granularity level.

The reference objects are the main subject of the description, while task-relevant objects are geographic objects that are relevant for the task at hand and are spatially related to the reference objects. For example, the cells may be the reference objects of our analysis, while the geographic objects within the cells are the task-relevant objects. In this case, properties and relationships of task relevant objects within each cell are computed by the feature extractor and stored as ground atoms, e.g., the spatial perpendicularity between the geographic objects *g*1 and *g*2 is represented by the ground atom *almost_perpendicular*(*g*1, *g*2). If *g* is a task-relevant object of the set R_k, then $is\_a(g, n_j)$ establishes the association between a geographic object *g* and corresponding objects at the level j ($j = 1, \ldots, M$) of the hierarchy H_k. Finally, for each cell *c*, the ground atom *cell*(*c*) identifies the unique reference object in the units of analysis.

The task of spatial association rule discovery performed by SPADA is split into two sub-tasks: find frequent spatial patterns and generate highly confident spatial association rules. The discovery of frequent patterns is performed according to the levelwise method described in [33], that is, a breadth-first search in the lattice of patterns spanned by a generality order between patterns. In SPADA the generality order is based on θ substitution [38]. The pattern space is searched one level at a time, starting from the most general patterns and iterating between candidate generation and evaluation phases. Once large patterns have been generated, it is possible to generate strong spatial association rules. For each pattern *P*, SPADA generates antecedents suitable for rules being derived from *P*. The consequent, corresponding to an antecedent, is simply obtained as the complement of atoms in *P* and not in the antecedent. Rule constraints are used to specify literals which should occur in the antecedent or consequent of discovered rules. In a more recent release of SPADA (3.1) [3], new pattern (rule) constraints have been introduced in order to specify exactly both the minimum and maximum number of occurrences for a literal in a pattern (antecedent or consequent of a rule). An additional rule constraint has been introduced to eventually specify the maximum number of literals to be included in the consequent of a rule. In this way, we are able to constrain the consequent of a rule requiring the presence of only the literal representing the class label and obtain useful patterns for classification purposes. Finally, the generation of patterns also takes into account a BK expressed in the form of first-order definite clauses. In this way, it is possible to simulate inferential mechanisms defined within a spatial reasoning theory. Moreover, by specifying both a BK and some suitable pattern constraints, it is possible to change the representation language used for spatial patterns, making it more abstract (human-comprehensible) and less tied to the physical representation of geographic objects.

An example of a spatial pattern discovered by SPADA is the following:

cell(A), contain(A,B), contain(A,C), is a(B,object),
is_a(C,object), extension(C,[100..200.5]) (40%),

which expresses a spatial containment relation between a cell *A* and some geographic objects *B* and *C*, where *C* is represented by a line with an extension between 100 and 200.5 m. This pattern occurs in 40% of the cells. The following spatial association rule:

cell(A), contain(A,B), contain(A,C), is_a(B,object),
    is_a(C,object) ⇒ extension(C,[100..200.5]) (40%, 60%),

states that "in 60% of the cells (A), containing two geographic objects B and C, C is a line whose extension is between 100 and 200.5." Since SPADA, like many other association rule mining algorithms, cannot process numerical data properly, these are discretized in equal-width intervals which are treated as ground terms.

By taking into account hierarchies on task-relevant objects, we obtain descriptions at different granularity levels. For instance, by considering a portion of the logical hierarchy on geographic objects, in which both hydrography and administrative boundary are considered, specialization of objects is as follows:

hydrography ↘
object
administrative boundary ↗

A finer-grained spatial association rule can be the following:

cell(A), contain(A,B), contain(A,C),
    is_a(B,administrativeBoundary), is_a(C,hydrography)
    ⇒ extension(C,[100..200.5]) (35%, 70%),

which provides better insight into the nature of the geographic objects *B* and *C*.

10.5 SDMOQL

The syntax of SDMOQL is designed according to a set of data mining primitives designed to facilitate efficient, fruitful spatial data mining in INGENS 2.0. Seven primitives have been considered as guidelines for the design of SDMOQL. They are:

1. the set of geographic objects relevant to a data mining task,
2. the kind of knowledge to be discovered,
3. the set of descriptors to be extracted from a digital map (primitive descriptors),

4. the set of descriptors to be used for pattern description (pattern descriptors),
5. the background knowledge to be used in the discovery process,
6. the concept hierarchies,
7. the interestingness measures and thresholds for pattern evaluation.

These primitives correspond directly to as many non-terminal symbols of the definition of an SDMOQL statement, according to an extended BNF grammar. Indeed, the SDMOQL top-level syntax is the following:

<SDMOQL> ::= <SDMOQLStatement>;
{<SDMOQLStatement>;}

<SDMOQLStatement> ::= <SDMStatement>
|<BackgroundKnowledge>
|<Hierarchy>

<SDMStatement> ::= <ObjectQuery>
mine <KindOfPattern>
analyze <PrimitiveDescriptors>
with descriptors <PatternDescriptors>
[<BackgroundKnowledge>]
{<Hierarchy>}
[**with** <InterestingnessMeasures>],

where "[]" represents 0 or one occurrence and "{}" represents 0 or more occurrences, and words in bold type represent keywords. In Sections 10.5.1 to 10.5.5 the detailed syntax for each data mining primitive is both formally specified and explained through various examples of possible mining problems.

10.5.1 Data Specification

The first step in defining a spatial data mining task is the specification of the geographic objects on which mining is to be performed. Geographic objects are selected by means of a query with a **SELECT-FROM-WHERE** structure, that is:

<Object_Query> ::= <Query_Statement>
{**UNION** <Query_Statement>}

<Query_Statement> ::=
SELECT <Object> {, <Object>}
FROM <Class> {, <Class>}
[**WHERE** <Conditions>]

The **SELECT** clause should return a cell or objects of a layer (hydrography, orography, and so on), or logical objects of a specific type (river, street, and so on). Hence, the selected geographic objects must belong to the same symbolic level, namely, cell, layer, or logic object. More formally the **FROM** clause can contain either a group of cells, a set of layers, or a set of logic objects, but never a mixture of them. Whenever the generation of the descriptions of objects belonging to different symbolic levels is necessary, the user can obtain it by means of the **UNION** operator. The following are examples of valid data queries:

Example (Cell-level query). The user selects cell 26 from the topographic map of Canosa (Apulia) and the feature extractor generates the description of all the geographic objects in this cell.

```
SELECT x
FROM x in Cell
WHERE x->num_cell = 26 AND x->part map->map_name = “Canosa”
```

Example 2 (Layer-level query). The user selects the orography layer from the topographic map of Canosa and the construction layer from any map. The feature extractor generates the description of the objects in these layers for all cells of the map of Canosa.

```
SELECT x, y
FROM x in Orography, y in Construction
WHERE x->part_map->map_name = “Canosa”
```

Example 3 (Logical object-level query). The user selects the objects of the logic type river, from cell 26 of the topographic map of Canosa. The feature extractor generates the description of the rivers in this cell.

```
SELECT x
FROM x in River
WHERE x->part_map->map_name = “Canosa”
      AND x->log_incell->num_cell = 26
```

10.5.2 The Kind of Knowledge to be Mined

The kind of knowledge to be discovered determines the data mining task in hand. Currently, SDMOQL supports the generation of either classification rules or association rules. The former are used for a predictive task, while the latter are used for a descriptive task. The top-level syntax is defined as follows:

```
<KindOfPattern> ::= <ClassificationRules>|<AssociationRules>

<ClassificationRules> ::= classification as <PatternName>
      for <ClassificationConcept>
      {, <ClassificationConcept>}

<AssociationRules> ::= association as <PatternName>
      key is <Descriptor>
```

The <PatternName> denotes the name to be associated to the set of (classification or association) patterns to be discovered in the data mining task formulated within the SDMOQL statement. In a classification task, the user may be interested in inducing a set of classification rules for a subset of the classes (or concepts) to which training examples belong. In this case, the subset of interest for the user is specified in the <ClassificationConcept> list.

As pointed out, spatial association rules define spatial patterns involving both reference objects and task-relevant objects [4]. For instance, a user may be interested in describing a given area by finding associations between large towns (reference objects) and geographic objects in the road network, hydrography, and administrative boundary layers (task-relevant objects). The atom denoting the reference objects is called the key atom. The predicate name of the key atom is specified in the key is clause.

10.5.3 Specification of Primitive and Pattern Descriptors

The **analyze** clause specifies which descriptors, among those automatically generated by the feature extractor, can be used to describe the geographic objects extracted by means of the first primitive. The syntax of the **analyze** clause is the following:

analyze <PrimitiveDescriptors>,

where:

<PrimitiveDescriptors> ::= <Descriptor>{, <Descriptor>}
    parameters <ParameterSpecs>{, <ParameterSpecs>}

<Descriptor> ::= <Predicate>/<Arity>
<ParameterSpecs> ::= <ParameterName> threshold <Integer>.

The specification of a set of parameters is required by the feature extractor to automatically generate some primitive descriptors. The language used to describe generated patterns is specified by means of the following clause: with descriptors <PatternDescriptors> where:

<PatternDescriptors> ::= <DescriptorSpecification>{; <DescriptorSpecification>}
<DescriptorSpecification> ::= <Descriptor> [cost <Integer>] | <Descriptor>
    [**with** <TermsSpec>]
<TermsSpec> ::= <TermSpec>{, <TermSpec>}
<TermSpec> ::= <ConstantType> | <VariableType>
<ConstantType> ::= **constant** [<Value>]
<VariableType> ::= **variable mode** <VariableMode> role <VariableRole>
<VariableMode> ::= **old** | **new** | **diff**
<VariableRole> ::= **ro** | **tro**

The specification of descriptors to be used in the high-level conceptual descriptions can be of two types: either the name of the descriptor and its relative cost, or the name of the descriptor and the full specification of its arguments. The former is appropriate for classification.

The (classification or association) rules are expressed by means of descriptors specified in the **with descriptors** list. They are specified by Prolog programs on the basis of descriptors generated by the feature extractor. For instance, the descriptor “font_to_parcel/2” has two arguments which denote two logical objects, a font and a parcel. The topological relation between the two logical objects is defined by means of the clause:

font_to_parcel(Font,Parcel) = TopographicRelation :-
    type_of(Font) = font, part_of(Font,Point) = true,
    type_of(Parcel) = parcel, part_of(Parcel,Region) = true,
    point_to_region(Point,Region) = TopographicRelation.

In association rule mining tasks, the specification of pattern descriptors corresponds to the specification of a collection of atoms: "predicateName(t_1, …, t_n)," where the name of the predicate corresponds to a <Descriptor>, while <TermSpec> describes each term ti, which can be either a constant or a variable. When the term is a variable, the mode and role clauses indicate, respectively, the type of variable to add to the atom and its role in a unification process. Three different modes are possible: **old** when the introduced variable can be unified with an existing variable in the pattern, **new** when it is not already present in the pattern, or **diff** when it is a new variable but its values must be different from the values of a similar variable in the same pattern. Furthermore, the variable can fill the role of reference object (**ro**) or task-relevant object **(tro**) in a discovered pattern during the unification process. The **is key** clause specifies the atom that has the key role during the discovery process. The first term of the key object must be a variable with mode **new** and role **ro**. The following is an example of specification of pattern descriptors defined by an SDMOQL statement:

with descriptors
    contain/2 **with variable mode old role ro,**
        variable mode new role tro;
    type_of/2 **with variable mode old role tro,**
        constant;
    fluvial_landscape/1 **with is key with variable mode new role ro;**

This specification helps to select only association rules where the descriptors fluvial_landscape/1, contain/2, and type_of/2 occur. The argument of "cell" is a **new** variable that plays the role of **ro**. The argument of the predicate "fluvial landscape" is always a new variable that plays the role of **ro**. The predicate "contain" links the **ro** with other geographic objects contained in the "fluvial_landscape." Finally, the first argument of the predicate "type_of" is always an **old** variable, denoting a geographic object that plays the role of **tro**, whereas the second argument is a constant value that denotes the type of object (e.g., river, street, parcel). The following association rule:

fluvial_landscape(X), contain(X,Y), type_of(Y,river), X≠Y ⇒
    contain(X,Z), type_of(Z,font), X≠Z, Y≠X

satisfies the constraints of the specification and expresses the co-presence of both a river and a font in a cell classified as a fluvial landscape.

10.5.4 Syntax for Background Knowledge and Concept Hierarchy Specification

Many data mining algorithms use background knowledge or concept hierarchies to discover interesting patterns. Background knowledge is provided by a domain expert on the domain to be discovered. This can be useful in the discovery process.

The SDMOQL syntax for background knowledge specification is the following:

<BackgroundKnowledge> ::= [<NewKnowledge>] {<UseKnowledge>}
<NewKnowledge> ::= **define knowledge** <Clause> {; <Clause>}
<UseKnowledge> ::= **use background knowledge of users** <User> {, <User>}
        on <Descriptor> {, <Descriptor>}

In INGENS 2.0, the user can define a background knowledge expressed as a set of definite clauses; alternatively, the user can specify a set of rules explicitly stored in a deductive database and possibly discovered in a previous step. An example of a background knowledge definition is reported in the following:

Example (Definition of close_to).
close_to(X,Y)=true :_region_to_region(X,Y)=meet.
close_to(X,Y)=true :_close_to(Y,X)=true.

while an example of the use of this background knowledge is reported in the following:

Example (Import of close_to).
use background knowledge of users UserName1 **on** close_to/2.

Concept hierarchies allow knowledge mining at multiple abstraction levels [17]. In SDMOQL, a specific syntax is defined for the hierarchy:

<Hierarchy> ::= [<NewHierarchy>] [<UseHierarchy>]
<NewHierarchy> ::= **define hierarchy** <Schema_Hierarchy> |
define hierarchy for <SetGroupingHierarchy>
<UseHierarchy> ::= **use hierarchy** <NameHierarchy> **of user** <User>.

The following example shows how to define some hierarchies in SDMOQL:

Example (Logical hierarchy on geographic objects).
define hierarchy LogicalObject **as**
    level1: {Hydrography,Orography, ...} < level0: Object;
    level2: {River,Lake,See,Font,Canal...} <level1:Hydrography;
    level1: {Slope,Contour slope, Level Point ...} < level0: Orography;
    …

In INGENS 2.0, this hierarchy is automatically extracted from the GIS data model and used to discover multi-level spatial association rules.

10.5.5 Syntax for Interestingness Measure Specification

The user can control the data mining process by specifying interestingness measures for data patterns and their corresponding thresholds. The SDMOQL syntax is the following:

<InterestingnessMeasures> ::= [<Criteria>] [<Settings>]

<Criteria> ::= **criteria**
(intermediate | final)(minimize | maximize) <Parameter>
with tolerance <Value> **{,(intermediate | final)**
(minimize | maximize) <Parameter> **with tolerance** <Value>}

<Settings> ::= <Parameter> := <StringValue>

Interestingness measures may include: threshold values, weights, search biases in the hypotheses space, and algorithm-specific parameters. In particular the user can bias the search in the hypotheses space by a number of preference criteria, such as the maximization of the number of covered examples or the minimization of the number of variables in the body of a learned clause. The user can also set thresholds such as confidence, support, or number of learned concepts. Finally, the user can set the value of a generic input parameter of a data mining algorithm.

10.6 MINING SPATIAL PATTERNS: A CASE STUDY

To show the potential of the integration of spatial data mining tools with GIS technology, we extend and elaborate on the case study on topographic map interpretation reported in [31]. The goal is to characterize and recognize some morphologies, which are not explicitly represented in the GIS data model.

The area considered in this application covers 90 km^2 in the surrounding area of the Ofanto River of Apulia, Italy (see Figure 10.7). The map of this area, stored in INGENS 2.0, is produced at a scale of 1:25000 by the Italian Military Geographic

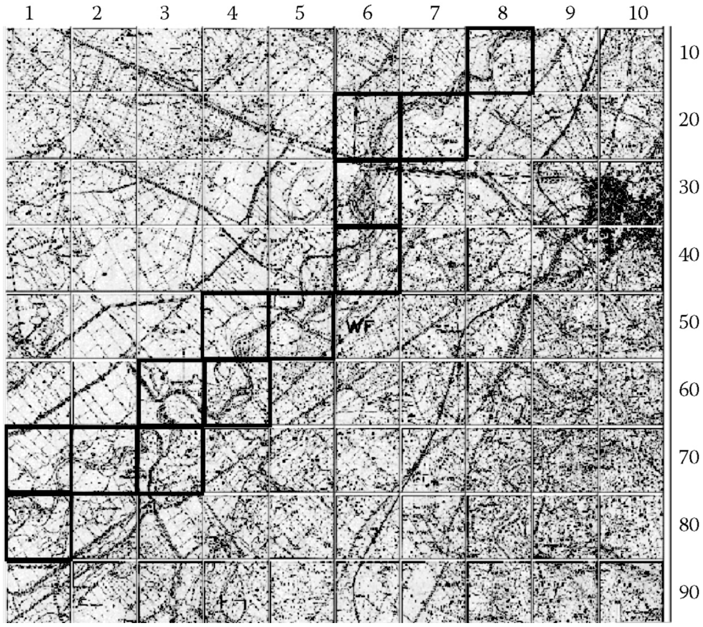


FIGURE 10.7 Surroundings of the Ofanto River. The boundary of fluvial landscape cells is blue.

Institute (IGMI). The map is segmented into 90 square observation units of 1 km^2. A map maintenance user has created the vectorized map and stored it in the SDB, according to the data model reported above.

The geomorphology considered in the following sections is the fluvial landscape, which is characterized by the presence of waterways, fluvial islands, and embankments. The classification rule provides an operational definition which can be used to retrieve this geomorphology in other similar topographic maps, while spatial association rules can be used to describe the area and support the implementation of an environmental policy.

10.6.1 Mining Classification Rules

The data miner user graphically composes an SDMOQL query to mine the concept of a fluvial landscape, by using, as training data, all the cells of the map. The query interpreter analyzes the SDMOQL query and verifies its syntactic and semantic correctness. The feature extractor generates a symbolic description for each cell by computing descriptors listed in the **analyze** clause. In this study, all descriptors in Table 10.1 are extracted. The data miner then associates the conceptual description of each cell with a concept (fluvial landscape or others), thus completely defining the training data. Association is made by binding variable terms of one of the concepts to be discovered to the constants that represent the cells. This binding function is supported by the GUI of the system (see Figure 10.8).

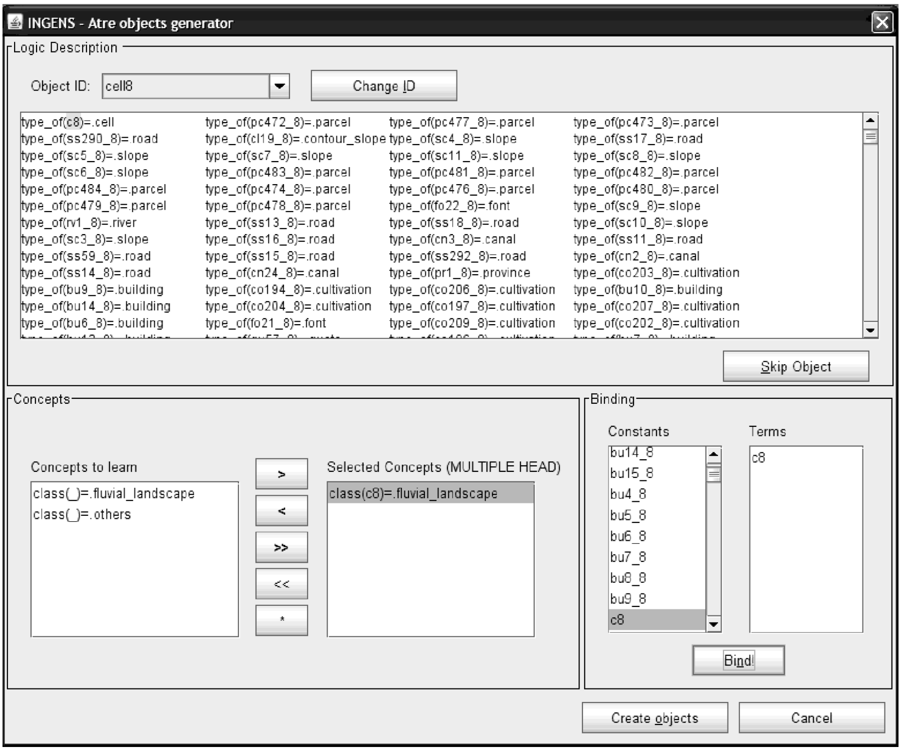
The classification rules induced by the learning system ATRE are reported as follows:

R1: class(X1)=fluvial_landscape ←type_of(X1)=cell,
contain(X1,X2)=true, color(X2)=blue,
type_of(X2)=river, part_of(X2,X3)=true,
extension(X3)∈[653.495..1642.184],
line_to_line(X4,X3)=almost_perpendicular,
extension(X4)∈[325.576..1652.736].

R2: class(X1)=fluvial_landscape ←type_of(X1)=cell,
contain(X1,X2)=true, type_of(X2)=province,
part_of(X2,X3)=true,
line_to_line(X4,X3)=almost_parallel,
part_of(X5,X4)=true, type of(X5)=contour_slope.

R1 covers 10 examples, while R2 covers 5 examples, two of which are different from those covered by R1.

According to R1, a cell is an instance of fluvial landscape if it contains geographic objects in blue classified as river, which is represented as a line (X3) with an extension between 653.495 and 1642.184 m. This line is almost perpendicular to another line (X4) with an extension between 325.576 and 1652.736 m. Unfortunately, the logical type of X4 is not specified by the rule. This is because the representation of a cell is related to the physical objects that it contains. To move from a physical



Object ID: cell8

```
type_of(c8)=.cell              type_of(pc472_8)=.parcel          type_of(pc477_8)=.parcel          type_of(pc473_8)=.parcel
type_of(ss290_8)=.road         type_of(cl19_8)=.contour_slope    type_of(sc4_8)=.slope             type_of(ss17_8)=.road
type_of(sc5_8)=.slope          type_of(sc7_8)=.slope             type_of(sc11_8)=.slope            type_of(sc8_8)=.slope
type_of(sc6_8)=.slope          type_of(pc483_8)=.parcel          type_of(pc481_8)=.parcel          type_of(pc482_8)=.parcel
type_of(pc484_8)=.parcel       type_of(pc474_8)=.parcel          type_of(pc476_8)=.parcel          type_of(pc480_8)=.parcel
type_of(pc479_8)=.parcel       type_of(pc478_8)=.parcel          type_of(fo22_8)=.font             type_of(sc9_8)=.slope
type_of(rv1_8)=.river          type_of(ss13_8)=.road             type_of(ss18_8)=.road             type_of(sc10_8)=.slope
type_of(sc3_8)=.slope          type_of(ss16_8)=.road             type_of(cn3_8)=.canal             type_of(ss11_8)=.road
type_of(ss59_8)=.road          type_of(ss15_8)=.road             type_of(ss292_8)=.road            type_of(cn2_8)=.canal
type_of(ss14_8)=.road          type_of(cn24_8)=.canal            type_of(pr1_8)=.province          type_of(co203_8)=.cultivation
type_of(bu9_8)=.building       type_of(co194_8)=.cultivation     type_of(co206_8)=.cultivation     type_of(bu10_8)=.building
type_of(bu14_8)=.building      type_of(co204_8)=.cultivation     type_of(co197_8)=.cultivation     type_of(co207_8)=.cultivation
type_of(bu6_8)=.building       type_of(fo21_8)=.font             type_of(co209_8)=.cultivation     type_of(co202_8)=.cultivation
```

Concepts to learn
class(_)=.fluvial_landscape
class(_)=.others

Selected Concepts (MULTIPLE HEAD)
class(c8)=.fluvial_landscape

Constants
bu14_8
bu15_8
bu4_8
bu5_8
bu6_8
bu7_8
bu8_8
bu9_8
c8

Terms
c8

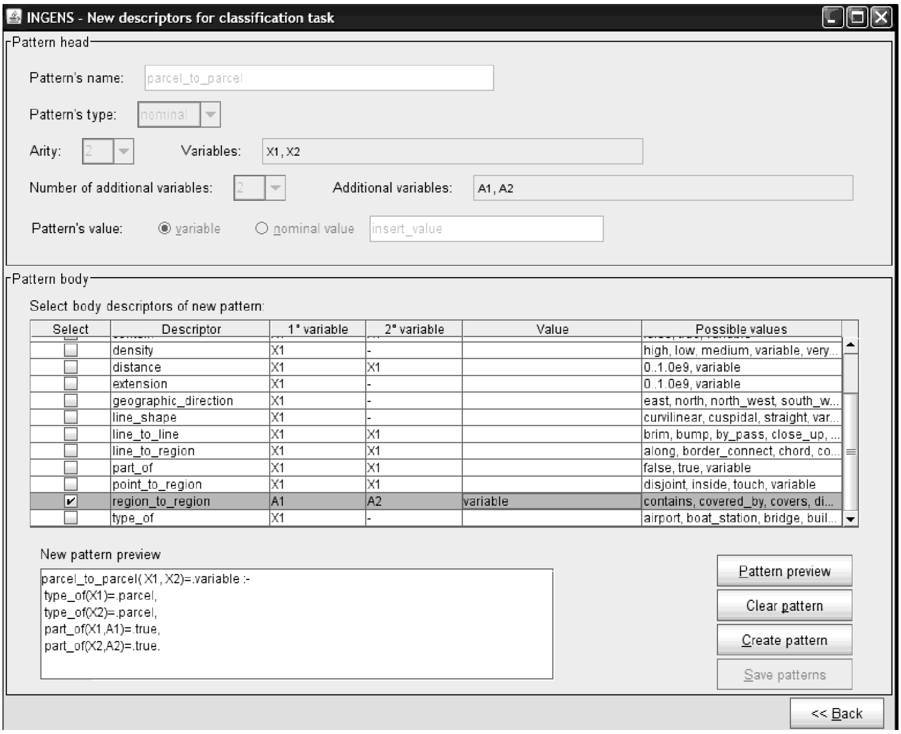
FIGURE 10.8 Associating a cell with a concept in INGENS 2.0.

to a logical level in the conceptual descriptions of the cells, some new descriptors are defined as background knowledge (see Figure 10.9). For example, the following <BackgroundKnowledge> statement:

parcel_to_parcel(A,B)=C ←type_of(A)=parcel,
  type_of(B)=parcel, part_of(A,D)=true,
  part_of(B,E)=true, region_to_region(D,E)=C

describes the topological relation between the regions that physically represent the "parcels" here referred to as the variables A and B, respectively. This BK statement can be stored in the GIS repository and re-used in a new data mining task. By defining other similar descriptors and then constraining the search space only to the definite clauses including these new descriptors, it is possible to discover a more abstract, human-interpretable operational definition of a fluvial landscape:

R3: class(X1)=fluvial_landscape ←
  contain(X1,X2)=true,
  river_extension(X2)∈[653.495..1642.184],
  river direction(X2)=north east.



INGENS - New descriptors for classification task

Pattern head

Pattern's name: parcel_to_parcel

Pattern's type: nominal

Arity: 2 Variables: X1, X2

Number of additional variables: 2 Additional variables: A1, A2

Pattern's value: ● variable ○ nominal value insert_value

Pattern body

Select body descriptors of new pattern:

Select	Descriptor	1° variable	2° variable	Value	Possible values
□	density	X1	-		high, low, medium, variable, very...
□	distance	X1	X1		0..1.0e9, variable
□	extension	X1	-		0..1.0e9, variable
□	geographic_direction	X1	-		east, north, north_west, south_w...
□	line_shape	X1	-		curvilinear, cuspidal, straight, var...
□	line_to_line	X1	X1		brim, bump, by_pass, close_up, ...
□	line_to_region	X1	X1		along, border_connect, chord, co...
□	part_of	X1	X1		false, true, variable
□	point_to_region	X1	X1		disjoint, inside, touch, variable
☑	region_to_region	A1	A2	variable	contains, covered_by, covers, di...
□	type_of	X1	-		airport, boat_station, bridge, buil...

New pattern preview

```
parcel_to_parcel( X1, X2)=.variable :-
type_of(X1)=.parcel,
type_of(X2)=.parcel,
part_of(X1,A1)=.true,
part_of(X2,A2)=.true.
```

FIGURE 10.9 Specifying a new pattern descriptor in INGENS 2.0.

R4: class(X1)=fluvial_landscape ←
contain(X1,X2)=true,
road_to_province(X2,X3)= almost_perpendicular,
road_to_river(X2,X4)= almost_perpendicular,
river_extension(X4) in [653.495..1642.184].

Rule R3 covers eight examples, while R4 covers five examples, four of which are different from those covered by R1. Both rules capture the presence of a river as a characterizing geographic object. In addition, rule R4 describes the spatial arrangement of other logical objects (road and administrative boundary) in the surroundings. The presence of an administrative boundary in this rule is not surprising because the River Ofanto partially overlaps the boundary between the provinces of Bari and Foggia in Apulia.

A different analysis is done by randomly selecting only four positive examples (8, 16, 17, 53) and nine negative examples (5, 11, 15, 27, 29, 34, 84, 88, 89) of the fluvial landscape concept and using only this training data to discover an operational definition of a fluvial landscape. By ignoring the BK, the following rule is discovered:

R5: class(X1)=fluvial_landscape ←type_of(X1)=cell,
contain(X1,X2)=true, type_of(X2)=river,

part_of(X2,X3)=true,
line_to_line(X4,X3)=almost_perpendicular,
part_of(X5,X4)=true, type_of(X5)=road,

while considering new descriptors defined in the BK, the following rule is discovered:

R6: class(X1)=fluvial_landscape ←
contain(X1,X2)=true,
road_to_river(X2,X3)= almost_perpendicular,
river_extension(X3) in [141.623..1642.184].

Discovered rules are used to query the entire map and recognize fluvial landscape cells. Several statistics are collected in Table 10.2. "TP" is the number of true positives (correctly classified cells). "FP" is the number of false positives. "FN" is the number of false negatives. "Prec" is the precision of the concept (Prec = TP/(TP + FP)). "Recall" is the recall of concepts (Recall = TP/(TP + FN)).

10.6.2 Association Rules

A purely descriptive analysis of the fluvial landscape is performed when the data miner extracts the frequent spatial association rules which compactly describe the morphology of the fluvial landscape cells in the topographic map. Similar to the classification case, INGENS 2.0 GUI offers facilities to graphically compose the SDMOQL query. In addition, INGENS 2.0 allows users to visualize the portion of the logical hierarchy matching at least one of the geographic objects extracted within the <ObjectQuery>statement (see Figure 10.10) and to translate it in a <NewHierarchy> statement to be added to the user-composed SDMOQL query. The logical hierarchy is then exploited to discover association rules at multiple levels of granularity without forcing data miners to repeat independent experiments on different representations. Once again, the BK is defined to move from a physical description to a logical description of the reference objects.

SPADA is run by setting *min_sup*=0.9 and *min_conf* = 0.9 for each granularity level, and the maximum pattern length is set to eight.

TABLE 10.2
Classification of the Surroundings of the Ofanto River Map (90 cells)

Rule	Time (sec)	TP	FP	FN	Prec	Recall
R5	832	12	5	1	0.706	0.923
R6	68	12	4	1	0.750	0.923

Note: The experiments are performed on Intel Pentium 4 -2.00 GHz CPU RAM 532Kb running Windows Professional 2000.

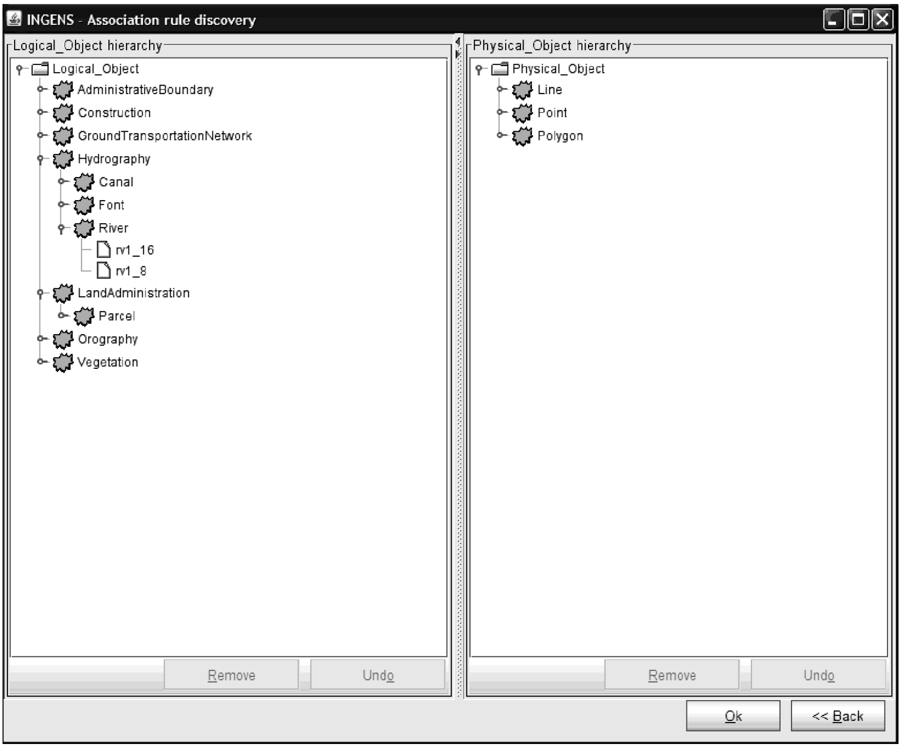


FIGURE 10.10 A portion of logical hierarchy that is automatically derived from a database. The hierarchy is visualized in the GUI of INGENS 2.0.

Despite the above constraints, SPADA generates 25830 confident rules from a set of 15048 candidate patterns, in 1819 sec. Confident rules and frequent patterns are visualized to data miners in separate views: one view for each hierarchy level and pattern length.

An association rule discovered by SPADA at the second level of granularity is the following:

fluvial_landscape(A) ⇒
  contain(A,B), is_a(B,administration_boundary),
  almost_perpendicular(B,C), C\=B ,is_a(C,hydrography)
(92.3%, 92.3%)

At a granularity level 3, SPADA specializes the task-relevant objects B and C by generating the following rule, which preserves both support and confidence values:

fluvial_landscape(A) ⇒
  contain(A,B), is a(B,province),
  almost_perpendicular(B,C), C\=B, is_a(C,river)
(92.3%, 92.3%)

The rule states that A is an instance (a cell) of a fluvial landscape, then A is crossed by a province boundary B that is almost perpendicular to a river C. Once again, the frequent pattern underlying this rule suggests a correlation between a fluvial landscape and a province boundary.

10.7 CONCLUDING REMARKS AND DIRECTIONS FOR FURTHER RESEARCH

Empowering a GIS with spatial data mining capabilities is not a trivial task. First, the geometrical representation and relative positioning of geographic objects implicitly define spatial properties and relationships, whose efficient computation requires an integration of the data mining system with the GDBMS. Second, the interactions between spatially close objects introduce different forms of autocorrelation, whose effect should be considered to improve predictive accuracy of induced models and patterns. Third, the units of analysis are typically composed of several geographic objects with different properties, and their structure cannot be easily accommodated by classical double entry tabular data. In INGENS 2.0, these challenges have been dealt with by integrating (multi-)relational data mining systems, which are able to navigate the relational structure of data and to generate relational patterns expressed in first-order logic or expressively equivalent formalisms. In particular, INGENS 2.0 integrates the MRDM systems ATRE and SPADA, which discover spatial classification rules and association rules, respectively. Different technologies, such as spatial database, data mining, and GIS, are hidden from users by means of a spatial data mining query language, SDMOQL, that permits condensing a data mining task in a query. Some constraints on the query language are identified by the particular mining task.

Although resorting to MRDM enables the INGENS 2.0 users to perform a sophisticated topographic map process, there are still several challenges that must be overcome and issues that must be resolved before the relational approach can effiectively enhance GIS applicability.

First, several MRDM methods exploit knowledge on the data model (e.g., foreign keys), which is obtained free of charge from the database schema, in order to guide the search process. However, this approach does not fit spatial databases well, because the database navigation is also based on the spatial relationships which are not explicitly modeled in the schema. To solve this problem, a feature extraction module is implemented in INGENS 2.0 to precompute spatial properties and relationships which are converted into Prolog facts used by ATRE and SPADA. The pre-computation is justified by the fact that geographic maps are rarely updated. However, the number of spatial relationships between two layers can be very large and many of them might be unnecessarily extracted. The alternative is to dynamically perform spatial joins only for the part of the hypothesis space that is really explored during the search by a data mining algorithm. This approach has been implemented in two MRDM systems, namely SubgroupMiner for subgroup mining [21] and Mrs-SMOTI for regression analysis [30]. Both systems realize a tight integration with a spatial DBMS (namely, Oracle Spatial), but have been applied to datasets where few spatial relationships are actually computed. Hence, scalability remains a problem when many spatial predicates have to be computed.

Second, the presence of autocorrelation in spatial phenomena strongly motivates an MRDM approach to spatial data mining. In any case, it also introduces additional challenges. In particular, it has been proven that the combined effect of autocorrelation and concentrated linkage (i.e., high concentration of objects linked to a common neighbor) can bias feature selection in relational classification [20]. In fact, the distribution of scores for features formed from related objects with concentrated linkage presents a surprisingly large variance when the class attribute has a high autocorrelation. This large variance causes feature selection algorithms to be biased in favor of these features, even when they are not related to the class attribute, that is, they are randomly generated. Most MRDM algorithms, such as ATRE, do not account for this bias. A solution to be investigated in INGENS 2.0 is the generation of pseudo samples from the relational data by retaining the linkage present in the original sample and the autocorrelation among the class labels, and, at the same time, by destroying the correlation between the original attributes and the class labels [36].

Third, an inductive learning algorithm designed for the predictive tasks typically requires large sets of labeled data. However, a common situation in geographic data mining is that many unlabeled geographic objects (e.g., map cells) are available and manual annotation is fairly expensive. Inductive learning algorithms would actually use only the few labeled examples to build a prediction model, thus discarding a large amount of information potentially conveyed by the unlabeled instances. The idea of transductive inference (or transduction) [44] is to analyze both the labeled (training) data and the unlabeled (working) data to build a classifier and classify (only) the unlabeled data as accurately as possible. Transduction is based on a (semi-supervised) smoothness assumption, according to which if two points in a high-density region are close, then the corresponding outputs should also be so [9]. In spatial domains, where closeness of points corresponds to some spatial distance measure, this assumption is implied by (positive) spatial autocorrelation. Therefore, the transductive setting seems especially suitable for classification and regression in GIS, and more in general, for those relational learning problems characterized by autocorrelation on the dependent variables. Only recently, a work on the transductive relational learning has been reported in the literature [8], and some preliminary results on spatial classification tasks show the effectiveness of the transductive approach [2]. No results are available on another class of predictive tasks, namely spatial regression.

Fourth, a large amount of knowledge is available in the case of geographic knowledge discovery, where relationships among geographic objects express natural geographic dependencies (e.g., a port is adjacent to a water body). These dependencies are expressed in non-novel or uninteresting patterns but with a very high level of support and confidence. If this geographic knowledge were used to constrain the search for new patterns, the scalability of the spatial data mining algorithms would greatly increase. Actually, these dependencies are represented either in geographic database schema, through one-to-one and one-to-many cardinality constraints, or in geographic ontologies. Therefore, their usage can be done at no additional cost in MRDM perspective, thus moving a step forward toward knowledge-rich data mining [12]. In INGENS 2.0, SPADA uses knowledge to constrain the search space for spatial association rules. In any case, the use of background knowledge can be investigated in several data mining tasks.

A final consideration on spatial reasoning can be made on spatial data mining methods in general. Spatial reasoning is the process by which information about objects in space and their relationships is gathered through measurement, observation, or inference, and is used to reach valid conclusions regarding the objects' relationships. For instance, in spatial reasoning, the accessibility of a site A from a site B can be recursively defined on the basis of the spatial relationships of adjacency or contiguity. Principles of spatial reasoning have been proposed for both quantitative and qualitative approaches to spatial knowledge representation. Embedding spatial reasoning in spatial data mining is crucial to make the right inferences, either when patterns are generated or when patterns are evaluated. Surprisingly, there are few examples of data mining systems that support some form of spatial reasoning. In INGENS 2.0, SPADA supports a limited form of spatial inference if rules of spatial reasoning are encoded in the background knowledge. However, although a general-purpose theorem prover for predicate logic can be used for spatial reasoning (as in SPADA), constraints that characterize spatial problem solving have to be explicitly formulated in order to make the semantics consistent with the target domain space. Therefore, embedding specialized spatial inference engines in the GIS seems to be the most promising, but still unexplored, solution.

REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Very Large Databases, VLDB 1994*, pp. 487–499. Morgan Kaufmann, San Francisco, CA, 1994.

[2] A. Appice, N. Barile, M. Ceci, D. Malerba, and R.P. Singh. Mining geospatial data in a transductive setting. In A. Zanasi, C.A. Brebbia, and N.F.F. Ebecken, editors, *Data Mining VIII,* pp. 141–150. WIT Press, Southampton, UK, 2007.

[3] A. Appice, M. Berardi, M. Ceci, and D. Malerba. Mining and filtering multi-level spatial association rules with ares. In M.S. Hacid, Z.W. Ras, and S. Tsumoto, editors, *15th International Symposium On Methodologies for Intelligent Systems, ISMIS 2005,* volume 3488 of *LNCS*, p. 342353. Springer-Verlag, Berlin, 2005.

[4] A. Appice, M. Ceci, A. Lanza, F.A. Lisi, and D. Malerba. Discovery of spatial association rules in georeferenced census data: A relational mining approach. *Intelligent Data Analysis*, 7(6):541–566, 2003.

[5] M. Berardi, A. Varlaro, and D. Malerba. On the effect of caching in recursive theory learning. In R. Camacho, R.D. King, and A. Srinivasan, editors, *14th International Conference on Inductive Logic Programming, ILP 2004,* volume 3194 of *Lecture Notes in Computer Science,* pp. 44–62. Springer, Berlin, 2004.

[6] V. Bogorny, A.T. Palma, P. Engel, and L.O. Alvares. Weka-gdpm: Integrating classical data mining toolkit to geographic information systems. In *SBBD Workshop on Data Mining Algorithms and Aplications, WAAMD 2006*, pp. 9–16, 2006.

[7] M. Ceci and A. Appice. Spatial associative classification: propositional vs structural approach. *Journal of Intelligent Information Systems*, 27(3):191–213, 2006.

[8] M. Ceci, A. Appice, N. Barile, and D. Malerba. Transductive learning from relational data. In P. Perner, editor, *5th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM 2007*, volume 4571 of *Lecture Notes in Computer Science*, pp. 324–338. Springer, Berlin, 2007.

[9] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[10] L. De Raedt. Attribute-value learning versus inductive logic programming: The missing links (extended abstract). In D. Page, editor, *Proceedings of the 8th International Workshop on Inductive Logic Programming, ILP 1998 (extended abstract),* volume 1446 of *Lecture Notes in Computer Science*, pp. 1–8, London, UK, 1998. Springer-Verlag.

[11] P. Densham. Spatial decision support systems. *Geographical Information Systems: Principles and Applications*, pp. 403–412, 1991.

[12] P. Domingos. Toward knowledge-rich data mining. *Data Mining Knowledge Discovery*, 15(1):21–28, 2007.

[13] S. Džeroski and N. Lavrač. *Relational Data Mining*. Springer-Verlag, Berlin, 2001.

[14] M. J. Egenhofer. Reasoning about binary topological relations. In *Proceedings of the 2nd Symposium on Large Spatial Databases, VLDB 1991*, pp. 143–160, Zurich, Switzerland, 1991.

[15] A.U. Frank. Spatial concepts, geometric data models, and geometric data structures. *Computers and Geosciences*, 18(4):409–417, 1992.

[16] S. Haehnel, J. Hauf, and T. Kudrass. Design of a data mining framework to mine generalized association rules in a web-based gis. In S.F. Crone, S. Lessmann, and R. Stahlbock, editors, *Proceedings of the 2006 International Conference on Data Mining, DMIN2006*, pp. 114–117. CSREA Press, 2006.

[17] J. Han, Y. Fu, W. Wang, K. Koperski, and O.R. Zaiane. DMQL: a data mining query language for relational databases. In *Proceedings of the Workshop on Research Issues on Data Mining and Knowledge Discovery*, pp. 27–34, Montreal, Quebec, 1996.

[18] J. Han, K. Koperski, and N. Stefanovic. GeoMiner: a system prototype for spatial data mining. pp. 553–556, 1997.

[19] A.K.H. Tung, J. Han, and M. Kamber. Spatial clustering methods in data mining. *Geographic Data Mining and Knowledge Discovery*, pp. 188–217. Taylor & Francis, London, 2001.

[20] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004,* pp. 593–598. ACM, 2004.

[21] W. Klosgen and M. May. Spatial subgroup mining integrated in an object-relational spatial database. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2002*, volume 2431 of LNAI, pp. 275–286. Springer-Verlag, Berlin, 2002.

[22] K. Koperski. *Progressive Refinement Approach to Spatial Data Mining.* PhD thesis, Computing Science, Simon Fraser University, British Columbia, Canada, 1999.

[23] A. Lanza, D. Malerba, F.A. Lisi, A. Appice, and M. Ceci. Generating logic descriptions for the automated interpretation of topographic maps. In D. Blostein and Y.B. Kwon, editors, *Graphics Recognition: Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pp. 200–210. Springer-Verlag, Berlin, 2002.

[24] F.A. Lisi and D. Malerba. Inducing multi-level association rules from multiple relations. *Machine Learning*, 55:175–210, 2004.

[25] L. Lloyd. *Foundations of Logic Programming,* 2nd ed. Springer-Verlag, Berlin,1987.

[26] H.P. Kriegel J. Sander M. Ester, S. Gundlach. Database primitives for spatial data mining. In *Proceedings of the International Conference on Database in Office, Engineering and Science, BTW 1999,* Freiburg, Germany, 1999.

[27] D. Malerba. Learning recursive theories in the normal ilp setting. *Fundamenta Informaticae,* 57(1):39–77, 2003.

[28] D. Malerba, A. Appice, and M. Ceci. A data mining query language for knowledge discovery in a geographical information system, *Database Support for Data Mining Applications*, pp. 95–116. Number 2682 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2003.

[29] D. Malerba, A. Appice, A. Varlaro, and A. Lanza. Spatial clustering of structured objects. In S. Kramer and B. Pfahringer, editors, *ILP,* volume 3625 of *Lecture Notes in Computer Science*, pp. 227–245. Springer, Berlin, 2005.

[30] D. Malerba, M. Ceci, and A. Appice. Mining model trees from spatial data. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 3721 of *Lecture Notes in Computer Science,* pp. 169–180. Springer, Berlin, 2005.

[31] D. Malerba, F. Esposito, A. Lanza, and F. A. Lisi. Machine learning for information extraction from topographic maps, *Geographic Data Mining and Knowledge Discovery*, pp. 291–314. Taylor & Francis, London, 2001.

[32] D. Malerba, F. Esposito, A. Lanza, F.A. Lisi, and A. Appice. Empowering a GIS with inductive learning capabilities: The case of ingens. *Journal of Computers, Environment and Urban Systems,* 27:265–281, 2003.

[33] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining Knowledge Discovery*, 1(3):241–258, 1997.

[34] M. May and A.A. Savinov. Spin!-an enterprise architecture for spatial data mining. In V. Palade, R.J. Howlett, and L.C. Jain, editors, *KES*, volume 2773 of *Lecture Notes in Computer Science*, pp. 510–517. Springer, Berlin, 2003.

[35] T. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[36] J. Neville and D. Jensen. Collective classification with relational dependency networks, 2003.

[37] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Springer, Berlin, 1982.

[38] G.D. Plotkin. A note on inductive generalization. 5:153–163, 1970.

[39] D. Thompson and R. Laurini. Fundamentals of spatial information systems. *APIC Series*, 37, 1992.

[40] P. Rigaux, M. Scholl, and A. Voisard. *Spatial Databases with Application to GIS.* Morgan Kaufmann, San Francisco, CA, 2002.

[41] S. Chawla and S. Shekhar. *Spatial Databases: A Tour.* Prentice Hall, Upper Saddle River, NJ, 2003.

[42] J. Sander, M. Ester, H.P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998.

[43] M. Sudhakar A. Pankaj, T. Smith, and P. Donna. KBGIS-II: A knowledge-based geographic information system. *International Journal of Geographic Information Systems*, 1(2):149–172, 1997.

[44] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[45] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, CA, 2000.

[46] S. Wrobel. Inductive logic programming for knowledge discovery in databases, *Relational Data Mining*, pp. 74–101. LNAI. Springer-Verlag, Berlin, 2001.