Geographic Knowledge Discovery in INGENS: an Inductive Database Perspective

Annalisa Appice Anna Ciampi Antonietta Lanza Donato Malerba
Antonella Rapolla Luisa Vetturi
Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
appice@di.uniba.it, aciampi@di.uniba.it, lanza@di.uniba.it, malerba@di.uniba.it

Abstract

INGENS is a prototype of GIS which integrates a geographic knowledge discovery engine to mine several kinds of spatial KDD objects from the topographic maps stored in a spatial database. In this paper we describe the main principles of an inductive spatial database in INGENS. Inductive database allows to keep permanent KDD objects and integrate database technology with systems for the geographic knowledge generation. In contrast to traditional spatial database technology, inductive database allows to answer queries which require synthesizing and applying plausible knowledge which is generated by (inductive) inference from both spatial objects and KDD objects (prior knowledge) stored in the same database.

1 Introduction

The huge amount of geographic data currently collected in geoscientific, economic and environmental situations has promoted the current generation of spatial database systems which is appositely designed to provide the database infrastructure for adequately storing and searching geographic data. Similarly to conventional database systems, spatial database systems provide answers only if the answers are in the database or are computed by the spatial primitives in database. A deductive database overcomes this limitation by adding to a spatial database the capabilities for drawing deductive consequences from data using a rule base. The rule base (or knowledge base) can be constructed by spatial data mining methods. Research in spatial data mining and geographic knowledge discovery is very active in last years [15, 14]. Anyway if spatial data mining methods remain not integrated with database systems, employing domain knowledge in the geographic knowledge discovery process makes data miner in charge of integrating different technologies such as data mining, spatial databases, knowledge bases and GIS. This is the spatial counterpart of a Knowledge and Data Discovery Management System (KDDMS) [6] to manage spatial KDD applications.

The objects queried in spatial KDDMS are not only the spatial objects stored in the spatial database, but also the spatial KDD objects (Knowledge Discovery Objects) of the knowledge base. KDD objects may not exist a-priori, thus querying them may require their run time generation. Alternatively they may be simply retrieved in the case they are previously generated and stored in a database. This view gives rise to an inductive database (inductive queries, data mining language and inductive solvers) [3] in geographic knowledge discovery process in order to answer to spatial queries which require drawing spatial inductive inferences and deriving plausible conclusions.

Research on inductive database provides several proposals for inductive query languages, such as Mine Rule [2], MSQL [7] and DMQL [5] and introduces constraint-based mining as a key mechanism for inductive query evaluation and thus knowledge discovery [4]. Anyway these works do not provide a general purpose solution to the specific issues posed by spatial dimension (i.e. computing spatial relationships and properties, e.g. area or perpendicularity, which are not explicitly encoded in database, spatial autocorrelation, spatial objects which can be considered at different levels of abstraction). For example, MO-DMQL [13] is a spatial data mining query language to support knowledge discovery from movement data only.

In this paper, we present the concepts and an implementation of an inductive spatial database in INGENS that is a prototype of GIS which offers data mining facilities to support tasks of topographic map interpretation. Originally built [11, 12] around the idea of inducing classification rules from geo-referenced data, INGENS is extended by integrating a spatial data mining query language, called SDMOQL [10], and by supporting the discovery of association rules

[1]. The answer to an SDMOQL is a collection of spatial KDD objects (classification or association rules) which are induced from the spatial database but not stored in it. The introduction of an inductive spatial database in INGENS allows to store spatial KDD objects to be available for subsequent queries. In this way, the geographic knowledge discovery process is performed by answering spatial KDD queries formulated in SDMOQL. A spatial KDD object can be intended as a spatial constraint to be satisfied in retrieving spatial objects which cover it or learning new spatial KDD objects which are spatially related to.

INGENS is briefly presented in the next Section. Principles of the inductive spatial database are described in Section 3, while in Section 4 we present an extension of SDMOQL to query the inductive database. Finally, in Section 5, some examples of SDMOQL queries are described.

2 INGENS: Concepts and Implementations

INGENS is a software system that provides the GIS infrastructure for editing, storing, interpreting, and displaying topographic maps. Topographic maps are partitioned into a number of cells. Spatial objects located in a cell are vectorized and stored an Oracle 10 g Spatial Database. The spatial data mining engine works in a first-order logic and allows to discover spatial patterns which capture a spatial arrangement of data. This arrangement is expressed in terms of spatial properties and relations. At present data mining engine integrates ATRE [9] to induce spatial classification rules and SPADA [8] to discover spatial association rules.

The engine works on a conceptual representation of data, but topographic maps are stored in a vector format. INGENS integrates spatial primitives to compute primitive descriptors of spatial objects located in a cell. Some descriptors are *contain*(C,L) (cell C contain a geographic object L), *type_of* (L) (type of L), *part_of*(L,F) (the geographic object L is composed by the geometry F), *line_to_line*(F1,F2) (spatial relations between lines F1 and F2), *line_to_region*(F1,F2) (spatial relations between line F1 and polygon F2), *region_to_region*(F1,F2) (spatial relations between polygons F1 and F2) and so on. Descriptors which involve lines (e.g. line_to_line) cannot be computed on polygons or points. Similarly descriptors like region_to_region can be computed only on polygons. INGENS allows to compute descriptors to model direction relations between cells (e.g. *north*(C1,C2) - cell C2 is at north of cell C1).

The knowledge discovery process is condensed in a query written in SDMOQL and run on the server side.

3 Inductive Spatial Database

An inductive spatial database aims at transforming INGENS in a Spatial Knowledge and Spatial Data Management System where the entire geographic knowledge discovery process is viewed as querying a database which contains both spatial data and generalization of spatial data induced by means of spatial data mining methods. The definitions of inductive spatial database schema and inductive spatial database instance are adapted from [3].

Definition 1. An *inductive spatial database schema* is a pair $\Re = (R, (Q_R, e, V))$, where R is a spatial database schema, Q_R is a collection of spatial KDD objects, V is a set of result values and e is the evaluation function that defines how a spatial KDD object occurs in some spatial data. The function e maps each element (r, p) to an element of V, where r denotes some spatial database with schema R and p is a spatial KDD object from Q_R.

Definition 2. An *instance of an inductive spatial database schema* $\Re$ is the pair (r, s) where r is a spatial database with schema R and q is a subset of Q_R.

In INGENS, R corresponds to the object-relational data schema of the spatial database where vectorized topographic maps are stored and Q_R represents the collection of spatial KDD objects which are not necessarily available a-priori in the database, but they can be retrieved by running SDMOQL queries. Two kind of spatial KDD objects are defined, that is, the spatial classification objects and the spatial association objects.

Definition 3 Let p be a *spatial classification object* of the concept c, p is in the form $\langle head \leftarrow body, bk\rangle$, where: (1) $head$ is the classification literal which represents c. (2) $body$ is the conjunction of literals which provides an operational definition of c. (3) bk is the (possibly empty) set of first-order definite clauses which provides the operational definitions of which literals in the $body$ are not computed as primitive descriptors in INGENS.

The literal in the $head$ is in the form $class(V) = c$ with V the key variable that identifies the spatial object to be classified and $class$ the classification literal. To satisfy the linkedness, V occurs as argument in at least one literal of the $body$. Each literal in the $body$ may be either a primitive literal (i.e. contain, type_of, area, ...) whose computation is demanded to the spatial primitives integrated in INGENS or a new literal whose operational definition is provided in the bk. The operational definition of a new literal may include both primitive literals and new literals which are defined in the same bk. Example 1 describes a spatial classification object which provides an operational definition for the concept fluvial landscape.

Example 1. Let us consider the spatial classification object p defined as follows:

$p.head$: class(C)=fluvial_landscape.

$p.body$: cell(C) $\wedge$ contain(C,Rd)=true $\wedge$
road_to_river(Rd,Rv)=roughly_perpendicular.

$p.bk$: (1) cell(C) $\leftarrow$ type_of(C)=cell.
(2) road_to_river(A,B)=C $\leftarrow$ type_of(A)=road,

type_of(B)=parcel, part_of(A,D)=true,
part_of(B,E)=true, line_to_line(D,E)=C.

The *head* identifies fluvial landascape as the classification concept of this spatial KDD object. The *body* encodes the operational definition of fluvial landscape that is learned as a classification rule by ATRE. In this case, the operational definition in the *body* classifies a cell C (key variable of this rule) as a fluvial landscape if C is crossed by a road Rd that is almost perpendicular to a river Rv. The bk includes the definition of the literals *cell* and *road_to_river* which are not computed by means of primitive descriptors automatically computed by INGENS.

Definition 3 can be extended in order to allow the representation of a (possible) dependency between several classification concepts which are learned as part of the same *recursive* logical theory.

Example 2. A recursive logical theory which models the dependency between the concepts of fluvial landscape and system of farms is the following:

R1: class(C)=system_of_farms ← type_of(C)=cell,
contain(C,A), type_of(A, cultivation),
point_to_region(A,B), type_of(B,parcel),
R2: class(C)=fluvial_landscape ← close(C,A)=true,
class(A)=system_of_farms,
contain(C,B)=true, type_of(B)=river

Definition 4. A *recursive* spatial classification object p is a spatial classification rule whose body includes one or more classification literals. p is in the form $\langle head \leftarrow body, bk, F\rangle$ with F a conjunction of disjunctive spatial classification objects: one disjunction for each distinct concept which appears as a classification literal in $p.body$. A spatial classification object is a special case of recursive spatial classification object where $F = \oslash$.

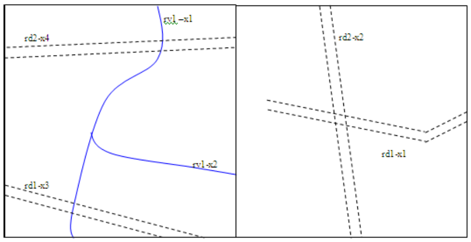
According to Definition 4, the logical theory $T = \{R1, R2\}$ presented in Example 2 is stored in the inductive spatial database as two spatial classification objects, that is, $p1$ and $p2$, respectively. $p2$ is a recursive spatial classification object where $p2.F$ includes $p1$. From a practical point of view, only the object references to the base spatial KDD objects are stored in F thus avoiding to store the same spatial classification object several times in database.

To complete the definition of spatial classification object, we define the evaluation function e.

Definition 5. Let (i) r be a spatial database with schema R and (ii) p be a spatial classification object, the evaluation function $e(r,p)$ returns a sequence of boolean values $[v_1, \ldots, v_n]$, one boolean value v_i for each distinct spatial object (reference object) r_i of r that matches the key variable of p in r. $e(r,p)$ is computed as $e(r,p) = [E(r_1,p), \ldots E(r_n,p)]$. $E(r_i,p) = true$ iff p covers $desc(r_i|r) \cup bk$, that is, there is a substitution θ such that $p\theta \subseteq \{desc(r_i|r) \cup bk\}$. $desc(r_i|r)$ denotes the first-order conceptual description of both the reference object r_i and the task-relevant spatial objects which are stored in r and are (transitively) spatially related to r_i according to the primitive literals which appear in the $body$ and bk of p.

Example 3 describes how to compute e for a spatial classification object p.

Figure 1. Vector representation of the cells a (left) and b (right) used in Example 2.



Example 3. Let us consider the spatial classification object p defined in Example 1 and the spatial database r where the cells a and b of Figure 1 are stored. The conceptual description of r is obtained by extracting the literals type_of, contain, part_of and line_to_line for the spatial objects (e.g. roads and rivers) located in the cells.

type_of(**a**)=cell, contain(a,rv1_a), type_of(rv1_a)=river,
part_of(rv1_a,x1_a),part_of(rv1_a,x2_a),contain(a,rd1_a),
type_of(rd1_a)=road, part_of(rd1_a,x3_a),
contain(a_a,rd2_a), type_of(rd2_a)=road,
part_of(rd2_a,x4_a), line_to_line(x3_a,x4_a)=roughly_parallel,
line_to_line(x3_a,x1_a)=roughly_perpendicular
type_of(**b**)=cell, contain(b,rd1_b), type_of(rd1_b)=road,
part_of(rd1_b,x2_b), contain(b,rd1_b),
type_of(rd1_b)=road, part_of(rd1_b,x2_b),
line_to_line(x1_b,x2_b)=roughly_perpendicular

$e(r,p) = [E(a,p), E(b,p)] = [true, false]$, since there is the substitution $\theta = \{A \leftarrow a, Rd \leftarrow rd1, Rv \leftarrow rv1\}$ such that $p\theta \subseteq desc(a|r) \cup bk$, while there is no substitution θ' such that $p\theta' \subseteq desc(b|r) \cup bk$.

Only the values $e(r,p)$ that the user's query requires are really computed. Hence, SDMOQL syntax must be extended by providing a new primitive to invoke the computation of e in both inductive and deductive requests.

Similarly to classification, we can provide the definition of a spatial association object and its evaluation function.

Definition 6 Let p be a spatial association object that is a spatial association rule in the form $A \Rightarrow C(supp, conf)$, p is in the form $\langle A, C, supp, conf, bk\rangle$, where: (1) A (antecedent of the rule) and B (consequent of the rule) are conjunctions of literals such that $A \cap B = \oslash$, $A \cup C$ is a spatial pattern that satisfies linkedness. (2) $supp$ (support of the rule) and $conf$ (confidence of the rule) are estimates of the probabilities p($A \cap C$) and $p(C|A)$ on the spatial data mined to discover the rule, (3) bk is a possibly empty set of first-order definite clauses which provide the definitions of non primitive literals which appear in $A \cup C$.

Definition 7 By assuming $V = \{true, false\}$, for each spatial database r with schema R and spatial association object p, $e(r,p) = true$ iff the support of $p.A \Rightarrow p.B$ computed in $r \cup p.bk$ is greater than $p.sup$ and the confidence of $p.A \Rightarrow p.B$ computed in $r \cup p.bk$ is greater than $p.conf$.

4 SDMOQL

SDMOQL is a spatial data mining query language to formulate spatial KDD queries in INGENS. The SDMOQL top-level syntax [10] is the following:

⟨SDMOQL⟩ ::= ⟨SDMOQLStatement⟩;
{⟨SDMOQLStatement⟩; }
⟨SDMOQLStatement⟩ ::= ⟨InductiveStatement⟩
⟨InductiveStatement⟩ ::= ⟨SpatialObjectQuery⟩
mine ⟨KindOfPattern⟩ **analyze** ⟨DataDescriptors⟩
with descriptors ⟨PatternDescriptors⟩
[⟨BackgroundKnowledge⟩] {⟨Hierarchy⟩}
[**with** ⟨InterestingnessMeasures⟩]

The syntax is extended with (1)language primitives to retrieve spatial objects which cover one or more spatial KDD objects stored in the database, and (2)language primitives to retrieve possibly recursive spatial KDD objects which have a spatial dependency with spatial KDD objects already stored in database. The novel top level syntax of a SDMOQL statement is reported below:

⟨SDMOQLStatement⟩ ::= ⟨DeductiveStatement⟩
|⟨InductiveStatement⟩

4.1 Deductive SDMOQL queries

The top-level syntax of ⟨DeductiveStatement⟩ is reported in th following:

⟨DeductiveStatement⟩ ::= ⟨SpatialObjectQuery⟩
evaluate ⟨SpatialKDDObjectQuery⟩
analyze ⟨DataDescriptors⟩
with descriptors ⟨PatternDescriptors⟩
[⟨BackgroundKnowledge⟩]

The **analyze** clause specifies which primitive descriptors, among those automatically generated by INGENS, must be used to describe the spatial objects extracted by means of the ⟨SpatialObjectQuery⟩. The **with descriptors** clause specifies the list of descriptors which appear as literals in the spatial KDD objects selected by ⟨SpatialKDDObjectQuery⟩. They are either primitive descriptors which are specified in the **analyze** clause or new descriptors whose definition is provided in the ⟨BackgroundKnowledge⟩. The ⟨SpatialKDDObjectQuery⟩ depends on the kind of knowledge.

⟨SpatialKDDObjectQuery⟩ ::= ⟨SpatialClassObjQuery⟩}
|⟨SpatialAssObjQuery⟩}

The SDMOQL syntax for ⟨SpatialClassObjQuery⟩} is defined in the following.

⟨SpatialClassObjQuery⟩ ::=
⟨SpatialClassObject⟩{, ⟨SpatialClassObject⟩}
⟨SpatialClassObject⟩ ::= **class** ⟨concept⟩ **as** ⟨rule body⟩
with key variable ⟨key variable⟩

The **class** clause specifies the classification concept whose operational definition is provided in the

To compose the ⟨SpatialClassObjQuery⟩, INGENS users select the set of spatial classification objects from those stored in the database. The descriptors in both the **analyze** clause and the **with descriptors** clause are automatically extracted from both the $body$ part and the bk part of these classification objects. The **define knowledge** clause provides the definition of all non primitive descriptors. This definition is automatically extracted from the bk of the selected classification objects.

In INGENS the execution of the query is in charge of the SDMOQL interpreter. First, the query is syntactically and semantically validated. Second, INGENS generates the conceptual representation of the spatial objects selected by computing the descriptors in the **analyze** clause and representing them in first-order formalism. Third, each spatial KDD object selected by the query is evaluated over the conceptual representation of selected spatial data by taking into account the background knowledge.

Similarly to the classification case, the syntax for ⟨SpatialAssocObjectQuery⟩ is defined in the following.

⟨SpatialAssocObjQuery⟩ ::=
⟨SpatialAssocObject⟩{⟨SpatialAssocObject⟩}
⟨SpatialAssocObject⟩ ::=
association rule⟨association rule statement⟩
with support ⟨supp value⟩ **confidence** ⟨conf value⟩

The **association rule** clause specifies the association rule defined in ⟨association rule statement⟩, while the clause **support** and the clause **confidence** specify the minimum support and confidence, respectively. Each association rule object is selected from those stored in database and mapped into the **association rule** clause, **support** clause and **confidence** clause. Although the syntax of ⟨SpatialAssocObjQuery⟩ is defined, it is not currently supported by the SDMOQL Interpreter in INGENS.

4.2 Inductive SDMOQL queries

The top-level syntax of ⟨InductiveStatement⟩ is reported below. Extensions are in italics.

⟨InductiveStatement⟩ ::= ⟨SpatialObjectQuery⟩
mine ⟨KindOfPattern⟩
***[using** ⟨SpatialClassObjectQuery⟩]*
***[recursion mode** ⟨ON/OFF⟩]*
analyze ⟨DataDescriptors⟩ ...

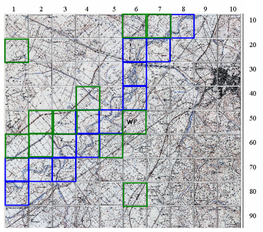
The **using** clause specifies which spatial classification objects among those stored in the database must be used as base classification objects for answering this inductive query. If **recursion mode** is on, spatial classification objects to be retrieved by the query are induced as part of a logical (recursive) theory. This theory originally includes the first-order rules which belong to the base spatial classification objects selected from database by means of ⟨SpatialClassObjectQuery⟩. The theory is then extended by adding the first-order rules which are induced by ATRE for the class concepts defined in the **mine** clause.

Recursion ON allows the induction of rules whose body may involve literals which are the class concepts already defined in the theory. A (recursive) classification rule is stored in database as a (recursive) classification object.

5 Query Examples

We present some examples of geographic knowledge discovery process performed by running SDMOQL queries over the inductive spatial database of INGENS. This database currently contains a map sheet (M81 NW) on a scale of 1:25000 produced by the Italian Military Geographic Institute (IGMI). The map covers 90 km^2 in the surroundings of the Ofanto River, spanning from the zone of Canosa to the Ofanto mouth (see Figure 2).

Figure 2. Surroundings of Ofanto River. The boundary of cells labeled as system of farms (fluvial landscape) is green (blue).



First, we present an inductive SDMOQL query that is formulated according to the original SDMOQL syntax. A portion of the query is reported in the following.

select x.* **from** Cell as x
where x.id in (34, 43, 44, 51, 52, 53, 54, 61, 62, 63)
mine classification as fluviallandscape
for class(_)=fluvial_landscape
analyze area/2, contain/2, line_to_line/2, ... type_of/1
with descriptors contain/2 cost 5 type boolean,
  area_parcel/1 cost 0 type real,
  cultivation_to_parcel/2 cost 0 type nominal, ...
define knowledge area_parcel(A,B)=C **as**
  [type_of(A)=parcel, part_of(A,B)=true, area(B)=C], ...

The data miner user graphically composes the SDMOQL query and associates the cells with a concept (fluvial landscape or system of farms) thus completely defining the training data. To answer the query INGENS data mining engine runs ATRE on the conceptual description generated for the cells under analysis. ATRE learns the classification object:

class(X1)=fluvial_landscape ← contain(X1,X2)=true,
  road_to_river(X2,X3)=roughly_perpendicular,
  road_to_river(X4,X3)=roughly_perpendicular

which discriminates training cells labeled as fluvial landscape from other cells. This classification object is now kept permanent in database and available for selection in subsequently SDMOQL queries.

Second, we present an SDMOQL query that is formulated to retrieve the (possibly recursive) spatial classification objects which provide an operational definition of "system of farms" by exploiting a (possibly) spatial relationship between system of farms and fluvial landscape. The training data are exactly the same selected in the previous query. In addition, spatial classification objects which are stored in database for the fluvial landscape concept are now selected in the query. A portion of the query is reported in the following:

select x.* **from** Cell as x
where x.id in (34, 43, 44, 51, 52, 53, 54, 61, 62, 63)
mine classification as systemoffarms
for class(_)=system_of_farms
using class(_)=fluvial_landscape **as**
  [contain(X1,X2)=true,
  road_to_river(X2,X3)=roughly_perpendicular,
  road_to_river(X4,X3)=roughly_perpendicular]
  with class X1
recursion mode on
analyze area/2, contain/2, direction/1, line_to_line/2, ...,
  type_of/1, north/2
with descriptors contain/2 cost 5 type boolean,
  north/2 cost 5 type boolean,
  area_parcel/1 cost 0 type real, ...
  road_to_river/2 cost 0 type nominal
define knowledge area_parcel(A,B)=C **as**
  [type_of(A)=parcel, part_of(A,B)=true, area(B)=C], ...

To answer the query, ATRE learns a disjunction of recursive classification rules which provide the operational definition of "system of farm" by taking into account spatial relations with the classification objects which model the "fluvial landscape" in the database. The rules are:

class(X1)=system_of_farms← north(X2,X1)=true,
  class(X2)=fluvial_landscape,contain(X1,X3)=true,
  area_parcel(X3)in[105050,152450].
class(X1)=system_of_farms←north(X2,X1)=true,
  class(X2)=fluvial_landscape,contain(X2,X3)=true,

geographic_direction_river(X3)=east.

Both rules are stored in the inductive spatial database in the form of recursive spatial classification objects.

Finally, we present a consider an SDMOQL query to locate the occurrence of fluvial landscape (whose definition is stored as KDD object in database) on a portion of a map stored in the same database (in this example, the cells 6, 7, 8, 16, 17, 18, 26, 27 and 28).

```
select x.* from Cell as x
where x.id in (6, 7, 8, 16, 17, 18, 26, 27, 28)
evaluate class(_)=fluvial_landscape as
   [contain(X1,X2)=true,
   road_to_river(X2,X3)=roughly_perpendicular,
   road_to_river(X4,X3)=roughly_perpendicuar]
   with class X1
analyze contain/2, type_of/1, part_of/2, line_to_line/2
with descriptors contain/2 cost 0 type boolean,
   road_to_river/2 cost 0 type nominal
define knowledge road_river(A,B)=C as [type_of(A)=road,
   part_of(A,D)=true, type_of(B)=river,
   part_of(B,E)=true, line_to_line(D,E)=C].
```

This query correctly retrieves the cells 8, 16, 17 and 26 as fluvial landscape. The query also returns the cell 7 as a fluvial landscape, while the domain expert recognizes it as a system of farm. This is a case of false positive.

6 Conclusions

In this paper, we introduce principles and architecture of an inductive spatial database in INGENS which allows to view the geographic knowledge discovery process as querying a database which contains both spatial data and generalization of spatial data induced by means of spatial data mining. Spatial KDD queries are formulated in SDMOQL which is extended in order to query both the spatial objects and the spatial KDD objects As future work, we plan to extend SDMOQL syntax to select, manipulate and query ”interesting” KDD objects which satisfy constraints w.r.t. accuracy, size, and so on. In addition, we intend to take into account a possible temporal dimension of geographic data.

Acknowledgments

This work is in partial fulfillment of research objectives of Strategic Project: “Telecommunication Facilities and Wireless Sensor Networks in Emergency Management” and ATENEO-2008 project “Knowledge Discovery in Relational Domains”.

References

[1] A. Appice, A. Lanza, and D. Malerba. An integrated platform for spatial data mining within a gis environment. In *ICDE Workshop on Spatio-Temporal Data Mining*, pages 507–516. IEEE Computer Society, 2007.

[2] J.-F. Boulicaut, M. Klemettinen, and H. Mannila. Querying inductive databases: A case study on the mine rule operator. In J. M. Zytkow and M. Quafafou, editors, *Principles of Data Mining and Knowledge Discovery, PKDD 1998*, volume 1510 of *LNCS*, pages 194–202. Springer, 1998.

[3] J.-F. Boulicaut, M. Klemettinen, and H. Mannila. Modeling kdd processes within the inductive database framework. In M. K. Mohania and A. M. Tjoa, editors, *International Conference on Data Warehousing and Knowledge Discovery, DAWAK 1999*, volume 1676 of *Lecture Notes in Computer Science*, pages 293–302. Springer, 1999.

[4] J.-F. Boulicaut, L. D. Raedt, and H. Mannila, editors. *Constraint-Based Mining and Inductive Databases*, volume 3848 of *LNCS*. Springer, 2005.

[5] J. Han, Y. Fu, W. Wang, K. Koperski, and O. Zaiane. DMQL: A data mining query language for relational databases. In *SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD 1996*, Montreal, Canada, 1996.

[6] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Communication of the ACM*, 39(11):58–64, 1996.

[7] T. Imieliński and A. Virmani. Msql: A query language for database mining. *Data Mining Knowledge Discovery*, 3(4):373–408, 1999.

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

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

[10] D. Malerba, A. Appice, and M. Ceci. A data mining query language for knowledge discovery in a geographical information system. In R. Meo, P. Lanzi, and M. Klemettinen, editors, *Database Support for Data Mining Applications*, number 2682 in LNCS, pages 95–116. Springer-Verlag, 2003.

[11] D. Malerba, F. Esposito, A. Lanza, and F. A. Lisi. Discovering geographic knowledge: The ingens system. In Z. W. Ras and S. Ohsuga, editors, *Foundations of Intelligent Systems, ISMIS 2000*, volume 1932 of *LNCS*, pages 40–48. Springer, 2000.

[12] 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, Elsevier Science*, 27:265–281, 2003.

[13] R. Ortale, E. Ritacco, N. Pelekis, R. Trasarti, G. Costa, F. Giannotti, G. Manco, C. Renso, and Y. Theodoridis. Daedalus: A knowledge discovery analysis framework for movement data. In S. Gaglio, I. Infantino, and D. Saccà, editors, *Italian Symposium on Advanced Database Systems, SEBD 2008*, pages 191–198, 2008.

[14] S. Rinzivillo, F. Turini, V. Bogorny, C. Krner, B.Kuijpers, and M. May. Knowledge discovery from geographical data. *In: F.Giannoni and D. Pedreschi (eds). Mobility, Data Mining and Privacy privacy Geographic Knowledge Discovery*, 2007.

[15] K. Zeitouni. A survey of spatial data mining methods databases and statistics point of views. *Data warehousing and web engineering*, pages 229–242, 2002.