SPADA: A Spatial Association Discovery System*

F. A. Lisi & D. Malerba

Dipartimento di Informatica, Università degli Studi di Bari, Italy.

Abstract

This paper presents a spatial association discovery system, named SPADA, which has been developed according to the theoretical framework of inductive databases. Our approach considers inductive databases as deductive databases with an integrated inductive component and relies on techniques borrowed from the field of Inductive Logic Programming (ILP). In SPADA, an ILP module supports the processing of spatial association mining queries and accesses spatial databases via a middle-layer module for feature extraction. The resulting architecture is illustrated by commenting spatial association mining sessions in detail and showing that ILP is a viable and promising solution to the development of inductive databases.

1 Introduction

In recent times several extensions of data mining methods and techniques have been attempted to deal with advanced databases. Progress in spatial databases and in geographic information systems (GIS) has paved the way to one of the most active strands of research in knowledge discovery in databases (KDD), spatial knowledge discovery. It aims at the extraction of implicit knowledge, spatial relations, or other patterns not explicitly stored in spatial databases [12]. A spatial pattern is a pattern showing the interaction of two or more spatial objects or space-depending attributes according to a particular spacing or set of arrangements. E.g., cities across nations are often clustered near lakes, oceans and streams. Actually such an arrangement reveals a spatial association, meaning that one spatial pattern

*This work has been supported by the MIUR COFIN-2001 project on "Methods of knowledge discovery, validation and representation of the statistical information in decision tasks".
is totally or partially related to some other spatial pattern. Spatial associations can be eventually presented to the user in the form of rules. The problem of discovering spatial association rules in geographic data has been originally formulated and solved by Koperski and Han [13]. They propose a top-down progressive refinement method which has been implemented in the module GeoAssociator of the spatial data mining system GeoMiner [10].

Contemporary to the development of new interesting KDD applications, database-theoretic foundations of data mining have been laid. In [11] KDD has been presented as a querying process to a data mining system. The resulting framework, called inductive database, defines a database which consists of the raw data and some more or less inductively obtained rules [17]. In this framework the user interacts with the database by issuing inductive queries such as "Find any possible association between P, Q, R and S" that return intensional instead of extensional answers. It is straightforward to notice that performance is an issue in inductive databases. Developing an inductive spatial database requires the solution to critical inefficiencies due to the inherent multi-relational nature of spatial data. Indeed the explicit location and/or extension of spatial objects also define implicit relations of spatial neighborhood that make knowledge discovery in spatial databases more difficult than in relational database. This applies to both the efficiency of algorithms and the complexity of patterns. In GeoMiner inductive queries are expressed by means of the data mining query language GMQL and spatial relations are computed off-line in two steps then reduced to a single relation according to the traditional approach to multi-relational data mining. Ester et al. [6] propose an approach based on database primitives which exploits the semantic information in the database schema to prune the search space.

In this paper we present SPADA, a spatial association discovery system that has been designed according to the theoretical framework of inductive databases. The adopted approach relies on techniques borrowed from the field of Inductive Logic Programming (ILP), a form of inductive learning based on computational logic and recently applied to KDD [4]. An inductive database according to the ILP approach can be perceived as a deductive database (DDB) with an integrated inductive component [8]. In SPADA the DDB is set up by a preliminary step of feature extraction from the spatial database and the inductive component is an ILP module that supports the processing of spatial association mining queries. Their integration requires data and patterns to be represented in a logical language for relational databases. We resort to Datalog [2], whose expressive power allows us to specify also rich domain knowledge such as spatial hierarchies, spatial constraints and rules for spatial qualitative reasoning.

The paper is organized as follows. Section 2 states the function of the system SPADA by commenting the processing of an spatial association mining query. Section 3 presents the software architecture of SPADA, providing details of the components. Section 4 draws conclusions and outlines possible future directions.
2 System function statement

We want the system SPADA to support the processing of spatial association mining queries, namely queries that activate the discovery of frequent associations between reference objects and task-relevant objects in a given spatial database. Reference objects are the main subject of the description. Task-relevant objects are spatially related to the reference ones. E.g. the following spatial association mining query in GMQL-like:

MINE ASSOCIATIONS DESCRIBING "large_towns"
WITH RESPECT TO topology(T.geo, R.geo), R.name, topology(T.geo, W.geo), W.name, topology(T.geo, B.geo), B.name
FROM town T, road R, water W, boundary B
WHERE T.type="large" AND distance(T.geo, R.geo)<"5 km" AND distance(T.geo, W.geo)<"5 km" AND distance(T.geo, B.geo)<"30 km"
SET SUPPORT minsup[1]=0.7 AND minsup[2]=0.6 AND minsup[3]=0.5
SET CONFIDENCE minconf[1]=0.8 AND minconf[2]=0.7 AND minconf[3]=0.6

asks for associations between large towns and spatial objects belonging to the map layers of roads, water bodies and boundaries within a given geographic area. In our context, intensional answers can be either patterns or rules. A pattern is an expression in some language describing a subset of data or a model applicable to that subset [7]. A spatial pattern is a pattern \( P(s\%) \) that contains at least one spatial relationship. The percentage \( s \) is called the support of the pattern. As usual in data mining, frequent patterns can be presented in the form of rules by filtering out those with low confidence. We call \( Q \rightarrow R(s\%, c\%) \) a spatial association rule if \( Q \) is a spatial pattern. The percentages \( s \) and \( c \) are called the support and the confidence of the rule respectively. Furthermore, analogously to GeoAssociator, SPADA discovers multiple-level patterns/rules. Minimum thresholds of support and confidence are given for each concept level of interest and influence the definitions of frequency and strength as shown later.

An example of intensional answer to the query above is the following logical conjunctive formula:

\[
is.a(X,\text{large.town}), \text{intersects}(X,Y), is.a(Y,\text{road}), \text{intersects}(X,Z), is.a(Z,\text{road}), Z \neq Y
\]

to be read as "\( X \) is a large town \text{ and } X \text{ intersects } Y \text{ and } Y \text{ is a road and } X \text{ intersects } Z \text{ and } Z \text{ is a road and } Z \text{ is distinct from } Y \)." Suppose that it is frequent with 91% support. Thus we can say that "91% of large towns intersect two distinct roads" in the given geographic area. Another possible intensional answer to the same query is the following strong rule:

\[
is.a(X,\text{large.town}), \text{intersects}(X,Y), is.a(Y,\text{road}) \rightarrow \text{intersects}(X,Z), is.a(Z,\text{road}), Z \neq Y
\]

(91%, 100%)

It says that: "\text{If a large town } X \text{ intersects a road } Y \text{ then } X \text{ intersects a road } Z \text{ distinct from } Y \text{ with 91% support and 100% confidence}." Since multiple concept levels are dealt with, finer-grained intensional answers to the query above are also expected, such as:
Data Mining III

is-a(X,large.town), intersects(X,Y), is-a(Y,regional.road), intersects(X,Z),
is-a(Z,main.trunk.road), Z≠ Y (75%)

which supplies more insight into the nature of the task-relevant objects Y and Z.

In general, answering spatial association rule mining queries in SPADA should solve instances of the following problem P:

Given
- a spatial database \( D \), a set \( S \) of reference objects and some sets \( R_k \), \( 1 \leq k \leq m \), of task-relevant objects,
- a taxonomy \( T \) involving objects in \( R_k \),
- a set \( \Psi \) of description granularity levels,
- two thresholds, \( \text{minsup}[l] \) and \( \text{minconf}[l] \), for each level \( l \) in \( \Psi \),

find strong multi-level spatial association rules.

It is noteworthy that specifying \( S \) and \( R_k \) enables the application of an Apriori-like algorithm [1]. Indeed, task-relevant objects are like landmarks. They break the continuity of space and define "transactions" around reference objects. In the case of geo-referenced data each \( R_k \) is typically a map layer and the taxonomy \( T \) is a collection of spatial hierarchies to be exploited to get descriptions of a given geographic area at different granularity levels. Spatial hierarchies capture is-a relations among locations on the basis of their geometry. To deal with several spatial hierarchies at once in a uniform manner, we map each concept of these hierarchies to one or more description granularity levels of \( \Psi \) so that a layering of \( T \) is induced. We denote by \( T[l] \) the \( l \)-th layer, \( l \in \Psi \), of \( T \) and by \( \mathcal{L}[l] \) the language of patterns built on an alphabet \( A \) and involving concepts in \( T[l] \). Frequency of patterns and strength of rules depend on the level \( l \) of description granularity. To be more precise, a pattern \( P(s%) \) in \( \mathcal{L}[l] \) is frequent if \( s \geq \text{minsup}[l] \) and all ancestors of \( P \) with respect to \( T \) are frequent at their corresponding levels. An association rule \( Q \rightarrow R(s%,c%) \) in \( \mathcal{L}[l] \) is strong if the pattern \( Q \cup R(s%) \) is frequent and \( c \geq \text{minconf}[l] \).

3 The software architecture

An inductive database via ILP can be perceived as a DDB with an integrated inductive component [8]. In particular, the architecture that envisages an ILP module as an additional back-end to the query engine would enable the processing of inductive queries. Actually such a tight integration of inductive components within DDBs is not so easily achievable with current ILP systems since they are mostly designed as stand-alone, user-driven applications for analyzing data. In our case, the application of the ILP approach to spatial databases is made possible by a middle-layer module for feature extraction as shown in Figure 1. The next subsections give details of the Spatial Database, the Feature Extractor and the DM Engine. The Query Interpreter is still under development according to the requirements specified in Section 2.
3.1 The spatial database

A spatial database (SDB) is a database system that offers spatial data types in its data model and query language and supports them in its implementation, providing at least spatial indexing and efficient algorithms for spatial join [9]. Thus spatial databases supply an adequate representation of both single objects and spatially related collections of objects. In particular, the abstraction primitives for spatial objects are point, line and region. Among the operations defined over spatial objects, spatial relationships are the most important because they make it possible, e.g., to ask for all objects in a given relationship with a query object. Egenhofer et al. [5] have proposed the 9-intersection model to categorize binary topological relations between arbitrary spatial objects. Examples are the relation meet between two regions and the relation crosses between a region and a line.

In our implementation of SPADA we have adopted Oracle Spatial, a well-known object-relational DBMS extended with spatial data handling facilities. It allows the storing of maps in vector format and supports the 9-intersection model for the computation of topological relations.

3.2 The feature extractor

The basic idea in the ILP approach of SPADA is that a SDB can be boiled down to a DDB. From now on, we denote by $D(S)$ the DDB portion that has been obtained by extracting features of interest to the problem $P$ from $D$.

Features in spatial representation are either properties of a spatial object, or relations that hold between spatial objects. We distinguish between spatial and a-spatial features according to whether they depend or not on spatial location. Spatial features can be classified in the following five categories:

1. Geometric, i.e. based on the principles of Euclidean geometry.
2. Directional, i.e. regarding relative spatial orientation in 2 or 3D.
3. Topological, i.e. binary relations that preserve themselves under topological transformations as translation, rotation, and scaling.
4. Hybrid, i.e. features which merge properties of two or more of the previous three categories.
Table 1: Map features in SPADA.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Semantics</th>
<th>Type</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>almost_parallel(X,Y)</td>
<td>Parallelism between X and Y</td>
<td>Hybrid relation</td>
<td>Boolean</td>
</tr>
<tr>
<td>almost_perpendicular(X,Y)</td>
<td>Perpendicular between X and Y</td>
<td>Hybrid relation</td>
<td>Boolean</td>
</tr>
<tr>
<td>area(X)</td>
<td>Area of X</td>
<td>Geometric attribute</td>
<td>Real</td>
</tr>
<tr>
<td>density(X,Y)</td>
<td>area(X)/area(Y)</td>
<td>Hybrid relation</td>
<td>Real</td>
</tr>
<tr>
<td>direction(X)</td>
<td>Geographic direction of X</td>
<td>Directional attribute</td>
<td>{north, east, north.west, north.east}</td>
</tr>
<tr>
<td>distance(X,Y)</td>
<td>Distance between X and Y</td>
<td>Geometric relation</td>
<td>Real</td>
</tr>
<tr>
<td>extension(X)</td>
<td>Length of X</td>
<td>Geometric attribute</td>
<td>Real</td>
</tr>
<tr>
<td>layer_name(X)</td>
<td>Type of X</td>
<td>A-spatial attribute</td>
<td>Layer name</td>
</tr>
<tr>
<td>line_shape(X)</td>
<td>Shape of X</td>
<td>Geometric attribute</td>
<td>{straight, curvilinear}</td>
</tr>
<tr>
<td>relate(X,Y)</td>
<td>Topological relation between X and Y</td>
<td>Topological attribute</td>
<td>Type of the topological relation</td>
</tr>
</tbody>
</table>

A-spatial features are, e.g., census data. Indeed, population data is usually geo-referenced with respect to areal spatial objects such as census zones, electoral constituencies, local government areas, or regular grid squares. See [15] for an application of SPADA to geo-referenced census data of Manchester Stockport, UK. Map features in SPADA are listed in Table 1.

In our implementation of SPADA the Feature Extractor enables the coupling of the DM Engine and Oracle Spatial databases. Algorithms for map feature extraction are implemented as PL-SQL functions so that complex SQL queries can be composed. The execution of these queries encompasses not only data selection but also data transformation. As for data selection we are interested in either relations between reference objects and task-relevant objects or properties of both kinds of objects. As for data transformation we need numerical features with a large domain to be discretized. For the purpose we have implemented the relative unsupervised discretization algorithm RUDE [14] which proves to be suitable for dealing with numerical data in the context of association rule mining. At the end of all this data processing, query results are stored in temporary Oracle
Spatial tables. An ad-hoc PL-SQL function is responsible for transforming these tuples into ground Datalog facts of $D(S)$. E.g., with reference to the experiment reported in [15], the Feature Extractor returns a *.db text file containing facts such as  
\[
\text{relate\_meet(edb03\_ed#4404, edb03\_ed#4487), relate\_crosses(edb03\_ed#4404, m\_way#1)}
\]
and  
\[
\text{table3\_census1(edb03\_ed#4487, [0.1111..0.1576])}
\]
The first fact states that the topological relation meet holds between the areal spatial objects #4404 and #4487 of the map layer stored in the database under the name edb03.ed. These two adjacent objects are enumeration districts (EDs) of the Stockport district. The second fact states that the topological relation crosses holds between the areal object #4404 of the edb03.ed map layer and the linear object #1 of the m.way map layer. Since m.way#1 is the motor-way M63, we can say that the M63 intersects the mentioned ED. The third fact states that the value of the census attribute census1 for the Stockport ED #4487 falls in the interval [0.1111..0.1576]. This value represents the percentage of persons aged 16 and over who reside in ED #4487, work out of the district of usual residence and drive to work. The real-valued attribute census1 has been derived from the integer-valued database-resident attribute s820161 as follows: s820161 has been first normalized with respect to the total number of residents aged 16 and over (i.e. the attribute s820001), then discretized by RUDE with respect to the other two census attributes of interest to this data mining problem.

### 3.3 The DM Engine

The DM Engine is an ILP module that discovers patterns to be intended as conjunctive queries in Datalog. Thus the alphabet $\mathcal{A}$ is a set of Datalog atoms and the language of patterns $\mathcal{L}[l]$ is the set of well-formed atomsets generated starting from a given atom $K \in \mathcal{A}$, called key atom, and combining atoms from $\mathcal{A}$ with concepts from $\mathcal{T}[l]$. Well-formedness encompasses properties which guarantee the correct evaluation of patterns and rules. In particular, the counting procedures for support and confidence are based on the coverage test of spatial observations, being it the ILP counterpart of counting the number of reference objects that satisfy a certain pattern/rule. Spatial observations are portions of $D(S)$, each of which concerns one and only one reference object. Thus the rule $Q \rightarrow R(s^%, c^%)$ means that $s^\%$ of spatial observations in $D(S)$ are covered by $Q \cup R$ and $c^\%$ of spatial observations in $D(S)$ that are covered by $Q$ are also covered by $Q \cup R$ respectively.

From the algorithmic viewpoint, the DM Engine discovers patterns/rules according to an increasing order of description granularity, i.e. from coarser-grained to finer-grained. The generation of frequent patterns implements the levelwise method [17], which is based on a breadth-first search in the lattice spanned by a generality order between patterns. The space is searched one level at a time till a maximum depth, starting from the most general patterns and iterating between candidate generation and candidate evaluation phases. In SPADA the levelwise method is adapted to deal with a Datalog representation of both data and patterns. Thus, the pattern space is structured according to $\theta$-subsumption [18]. The candidate generation phase consists of a refinement step followed by a pruning step. The former applies a refinement operator under $\theta$-subsumption to patterns that
have been previously found frequent. The latter involves verifying that candidate patterns do not \( \theta \)-subsume any infrequent pattern. The candidate evaluation phase checks whether the generated patterns satisfy the condition of frequency. If a pattern turns out not to be a large one, it is rejected. As for the support count, the candidate is transformed into a query whose answer set supplies the bindings for the variables with which the pattern is true in \( D(S) \). In particular, the number of different bindings of the variable which is the place-holder for reference objects is assumed as absolute frequency of the pattern in \( D(S) \). The support is obtained as relative frequency of the pattern in \( D(S) \). The generation of strong association rules post-processes frequent patterns into rules in the following way. For each frequent pattern \( P \), rules are built by putting one of the already computed sub-patterns of \( P \) in the left hand side and what remains of \( P \) in the right hand side. Their strength is then verified. Rules that do not exceed the minimum threshold for minimum confidence at the level currently being explored are rejected. Further details of the algorithm can be found in [16].

In our implementation of SPADA the DM Engine exploits a bias on the language of patterns and the available domain knowledge to constrain the search in the pattern space. Directives for generating patterns are found in the language bias specification. The \( '*'\.lb \) file mainly contains Datalog facts that supply atom templates to the DM Engine. They allow the user to define the set \( A \) intensionally rather than extensionally. E.g., in the experiment reported in [15], the fact \( lb\text{-atom}(sparelate\_meet(old\_ro, cliff\_tro)) \) specifies all the Datalog atoms with binary predicate \( relate\_meet \), an input variable in the domain of reference objects as first argument, and an output variable in the domain of task-relevant objects as second argument. Conversely, the fact \( lb\text{-atom}(table3\_census1(old\_tro, [0.1111..0.1576])) \) specifies all the Datalog atoms with binary predicate \( table3\_census1 \), an input variable in the domain of reference objects as first argument, and the term \([0.1111..0.1576]\) as second argument. Note that this term represents one of the intervals computed by RUDE for the feature \( table3\_census1 \). Domain knowledge is also given as input to the DM Engine. The \( '*'\.bk \) file mainly includes Datalog facts that represent the spatial hierarchies of \( T \). E.g., continuing on the example above, the Prolog fact hierarchy(\( ed, 2, ed, \{ bredbury\_ED, binnington\_ED, cale.green\_ED, cheadle\_ED, cheadle.hulme.north\_ED, davenport\_ED, cheadle.hulme.south\_ED, east.bramhall\_ED\}) represents the portion of \( T[2] \) which concerns the is-a hierarchy defined on the EDs map layer and rooted in the node labeled as ed. The \( '*'\.bk \) file can also contains rules for spatial qualitative reasoning. E.g. the relation of closeness in [15] is defined in terms of the extracted feature \( relate\_meet \) by means of the rule:

\[
\text{close\_to}(X, Y) \leftarrow relate\_meet(X, Z), relate\_meet(Z, Y).
\]

The DM Engine returns as many \( '*'\.pat \) and \( '*'\.rul \) files as the number of description granularity levels. These files report frequent patterns and strong rules either in text format or in XML format according to the user-defined setting.

4 Conclusions and future work

In this paper we have discussed some implementation issues in inductive databases and presented an inductive spatial database via ILP that can answer spatial asso-
Association mining queries. In particular we have given an insight into the Feature Extractor and DM Engine components. The latter discovers multi-level association patterns/rules in the DDB that the former sets up by extracting features from spatial databases and representing them as Datalog facts. SPADA can tackle applications that cannot be handled by similar systems, e.g. WARMR [3] and GeoAssociator. WARMR is an ILP system for mining association rules from multiple relations. But although it has been presented as a system able to use is-a hierarchies, it lacks mechanisms for dealing properly with structural knowledge. SPADA differs from GeoAssociator for the expressive power of the representation language. In particular, DDBs offer effective reasoning and representation means for the spatial domain. Of major interest is the possibility of embedding rules for the inference of implicit spatial relationships that are too numerous to be either stored in the spatial database or computed by computational geometry algorithms.

For the future, we plan to deliver a more user-friendly Query Interpreter. There is a need for both a spatial data mining query language and a presentation method of spatial association patterns/rules. Furthermore we intend to carry on optimizing and testing SPADA on real-world data sets to meet the requirements of efficiency, scalability and robustness that are of great interest to the data mining community. In particular, noise handling in spatial data mining requires techniques for dealing with approximated spatial relations. Finally we would like to investigate measures of interestingness for spatial association rules which can overcome the limits of the support-confidence problem setting in dealing with dense data. That would provide us with more effective pruning conditions for the rule space.

References


