Mining interesting spatial association rules: two case studies

Annalisa Appice, Margherita Berardi, Michelangelo Ceci, Michele Lapi,
Donato Malerba, Antonio Turi

Dipartimento di Informatica – Università degli Studi di Bari
via Orabona 4 - 70126 Bari
{appice, berardi, ceci, lapi, malerba, turi}@di.uniba.it

Abstract. In spatial data mining, a common task is the discovery of spatial association rules from spatial databases. We propose a distributed system named ARES that assists data miners in the complex process of extracting the association rules from a spatial database. We also face a common problem of association rule mining, that is the high number of discovered rules. This affects both efficiency of the data mining process and quality of the discovered rules. We propose some criteria to bias the search and to filter the discovered rules according to user's interests. Finally, we show the applicability of our proposal to two different real world domains, namely, document image processing and geo-referenced analysis of census data. We illustrate and comment experimental results on a set of multi-page documents extracted by IEEE PAMI and on North-West England 1998 census data.

1. Introduction

Spatial data mining investigates the problem of extracting pieces of knowledge from data describing *spatial objects*, which are characterized by a geometrical representation (e.g. point, line, and region in a 2D context) and a position with respect to some reference system. The relative positioning of spatial objects defines implicitly *spatial relations* of different nature, such as directional and topological. The goal of spatial data mining methods is to extract *spatial patterns*, that is, patterns involving spatial relations between mined objects such that they are certain, previously unknown, and potentially useful for the specific application [9].

Spatial data mining demands for the development of specific techniques that, differently from traditional data-mining techniques, do take the spatial dimension of the data into account when exploring the pattern space. Moreover, data to be mined are generally stored in spatial databases that provide powerful, flexible model data structures to serve multiple tasks including storage and sophisticated treatment of real-world geometry. Therefore, it is important that spatial data mining algorithms do explicitly consider these data structures in order to make the exploration of the pattern space more efficient.

Knowledge discovered from spatial data can be in various forms including classification rules, which describe the partition of the database into a given set of classes [10], clusters of spatial objects [16,17], patterns describing spatial trends, that is, regular changes of one or more non-spatial attributes when moving away from a given start object [4], and subgroup patterns, which identify subgroups of spatial objects with an unusual, an unexpected, or a deviating distribution of a target variable [8].

In a recent work, Lisi and Malerba [12] have proposed an algorithm, named SPADA (Spatial Pattern Discovery Algorithm), that discovers spatial association rules, that is, association rules involving spatial objects and relations. It is based on an ILP approach to (multi-)relational data mining and permits the extraction of multi-level spatial association rules, that is, association rules involving spatial objects at different granularity levels. For each granularity level, SPADA operates in three different phases: i) pattern generation; ii) pattern evaluation; iii) rule generation and evaluation.

SPADA has been loosely coupled with a spatial database and has been integrated into a system, named ARES (Association Rules Extractor from Spatial data), in order to assist data miners in the complex process of extracting the units of analysis from the spatial database, specifying the background knowledge on the application domain and defining some form of search bias. The last aspect is particularly relevant, since the number of discovered patterns or association rules is usually high and the interest of most of them does not fulfil user expectations. The presentation of thousands of rules can discourage users from interpreting them in order to find *nuggets of knowledge*. Furthermore, it is very difficult to evaluate which patterns could be interesting for the end users by means of some simple statistics, such as support and confidence. Therefore, an additional processing step is necessary in order to clean, order or filter interesting patterns/rules.

In this paper, we describe the integration of SPADA in the ARES system, then we explain the additional processing step of the rules, and finally we show the application of the spatial mining tool to two different application domains, namely, document image processing and geo-referenced analysis of census data. The paper is organized as follows. The problem of mining spatial association rules is briefly reported in Section 2. Section 3 describes the ARES distributed architecture that supports the interface of SPADA with a spatial database by generating high-level logic descriptions of spatial data. Some filtering mechanisms implemented in SPADA are described in Section 4. Finally, the application of ARES to two case studies is described in Sections 5. Some experimental results are reported and conclusions are drawn.

2. Mining Spatial Association Rules

The discovery of spatial association rules is a descriptive mining task aiming to detect associations between *reference objects* (*ro*) and some *task-relevant objects* (*tro*). The former are the main subject of the description, while the latter are spatial objects that are relevant for the task in hand and are spatially related to the former.

In general, association rules are a class of regularities that can be expressed by the implication: $P \rightarrow Q$ (s, c), where P and Q are a set of literals, called *items*, such that $P \cap Q = \varnothing$, the support s estimates the probability $p(P \cup Q)$, and the confidence c,

estimates the probability p($Q \mid P$). The conjunction P∧Q is called *pattern*. In classic data mining, patterns are mined from data represented in a single relation of a relational database, such that each tuple represents an independent unit of the sample population and columns correspond to properties of units. In the case of spatial patterns, this assumption turns out to be a great limitation. Indeed, a spatial pattern expresses a spatial relationship among spatial objects. The recently promoted (multi-) *relational* approach to data mining [2] looks for patterns that involve multiple relations of a relational database. Patterns found by these approaches are called *relational* and are typically stated in a more expressive language than patterns defined in a single relation. Typically, subsets of *first-order logic* are used to express relational patterns. Hence, a spatial association may be represented as conjunctive formula as follows:

$$is\_a(X, large\_town), intersects(X,Y), is\_a(Y, road) \Rightarrow$$
$$intersects(X, Z), is\_a(Z, road), Z \neq Y\ (91\%, 100\%)$$

to be read as "If a large town X intersects a road Y then X intersects a road Z distinct from Y with 91% support and 100% confidence". Since some kind of taxonomic knowledge on task-relevant objects may also be taken into account to obtain descriptions at different granularity levels (*multiple-level association rules*), finer-grained association rules are also expected, such as:

$$is\_a(X, large\_town), intersects(X, Y), is\_a(Y, regional\_road) \Rightarrow$$
$$intersects(X, Z), is\_a(Z, main\_trunk\_road), Z \neq Y\ (65\%,71\%)$$

It is noteworthy that the support and the confidence of the last rule have changed. Generally, the lower the granularity level, the lower the support of association rules. Therefore, different thresholds of support and confidence for different granularity levels should be used [5].

In general, the problem of mining association rules can be formally stated as follows: *Given* a spatial database (SDB), a set of reference objects S, some sets R_k, $1 \leq k \leq m$, of task-relevant objects, a background knowledge BK including some hierarchies H_k on objects in R_k , M granularity levels in the descriptions (1 is the highest while M is the lowest), a language bias LB that constrains the search space and a couple of thresholds $minsup[l]$ and $minconf[l]$ for each granularity level; *Find* strong multi-level spatial association rules.

Each R_k is typically a layer of the spatial database while hierarchies define *is-a* (i.e., taxonomical) relations of spatial objects in the same layer. To deal with several hierarchies at once in a uniform manner, objects in them are mapped to one or more of the M user-defined description granularity levels so that frequency of patterns as well as strength of rules depend on the level l of granularity with which patterns/rules describe data. To be more precise, a pattern P ($s\%$) at level l is *frequent* if $s \geq minsup[l]$ and all ancestors of P with respect to H_k are frequent at their corresponding levels. An association rule $Q \rightarrow R$ ($s\%$, $c\%$) at level l is *strong* if the pattern $Q \cup R$ ($s\%$) is frequent and $c \geq minconf[l]$.

A data mining tool that solves the problem stated above is ARES whose architecture is explained in the next section.

3. The architecture of ARES

ARES has a distributed architecture based on a client-server model (see Fig. 1). The spatial association rule miner SPADA is on the server side, so that several data mining tasks can be run concurrently by multiple users. SPADA is implemented in Prolog and fully exploits the flexibility of this logic programming language to specify both the background knowledge *BK* (hierarchies are expressed by a collection of ground atoms that define the binary predicate *is_a*, while domain specific knowledge is expressed as sets of definite clauses) and a language bias *LB* that constrains the search for patterns.

On the client side, the system includes a Graphical User Interface (GUI) implemented as Java application, which provides the user with facilities for controlling all parameters of the data mining process. More precisely, a wizard supports the user in the selection of layers (for spatial objects), tables (for aspatial properties) and attributes involved in the query to the SDB (Oracle Spatial). Conditions on both aspatial attributes (simple comparisons between two fields) and spatial features (simple comparisons with a field or a constant) can also be specified. Once the query is performed, the GUI allows the user to discretize some numerical attributes, to define the spatial hierarchies and their mappings into the *M* granularity level, to specify the declarative bias, and finally to run SPADA on the server.

SPADA, like many other association rule mining algorithms, cannot process numerical data properly, so it is necessary to perform a discretization of numerical features with a relatively large domain. For this purpose, ARES includes in the client side the module RUDE (relative unsupervised discretization algorithm) which discretizes a numerical attribute of a relational database in the context defined by other attributes [13].

In ARES, the SDB can run on a third computation unit. Many spatial features (relations and attributes) can be extracted from spatial objects stored in the SDB. The feature extraction requires complex data transformation processes to make spatial relations explicit and representable as ground Prolog atoms. Therefore, a middle layer module is required to make possible a loose coupling between SPADA and the SDB by generating features (e.g. *area, contains, on_top*) of spatial objects. The module, named FEATEX (Feature Extractor), is implemented as an Oracle package of procedures and functions, each of which computes a different feature [1].

On the client side, the system WISDOM++ [15] can be used to extract spatial data from document image and store them in the SDB. The process performed by WISDOM++ consists of the preprocessing of the raster image of a scanned paper

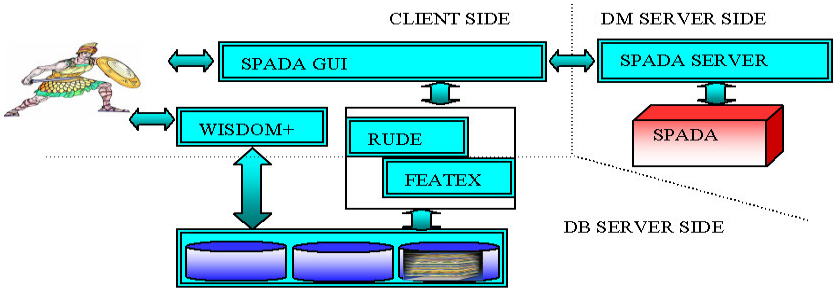

Fig. 1. ARES architecture.

document, the segmentation of the preprocessed raster image into basic layout components, the classification of basic layout components according to the type of content (e.g., text, graphics, etc.), the identification of a more abstract representation of the document layout (layout analysis), the classification of the document in one of predefined categories (e.g. business letter, scientific paper) on the basis of its layout and content, and the identification of semantically relevant layout components (e.g. title, abstract of a scientific paper) called logical components (*document image understanding* [18]). The final representation includes both layout structure (extracted in the layout analysis) and logical structure (semantic information extracted by means of document classification and understanding) computed on the original image. A further processing step stores the output structures in the SDB. WISDOM++ makes use of an Oracle Database to store intermediate data.

In order to handle spatial data provided by WISDOM++, FEATEX has been extended to allow features of layout components to be extracted (see section 5.1).

4. Filtering patterns and association rules

The efficiency of the data mining process is very important to tackle real-world problems. In order to improve the efficiency of the search process, SPADA associates each candidate pattern with backward pointers to parent patterns both at the same granularity level (intra-space parenthood) and at higher granularity levels (inter-space parenthood). Backward pointers are profitably exploited in the pattern generation phase to prevent the generation of some infrequent patterns [11]. In a more recent release of SPADA (3.0), backward pointers are also exploited in the pattern evaluation phase. Indeed, by associating each pattern with the list of support objects, it is possible to perform the evaluation of each pattern solely on the support objects of its intra-space parenthood instead of the whole set S of reference objects. An additional caching technique compensates the overhead in looking for the parenthood of each pattern, since it has a cost, which increases with the number of stored patterns.

The above mentioned efficiency improvements are based on the monotonicity property of the generality order that is defined for spatial patterns with respect to the support of the patterns themselves. This is a nice example of an "intelligent" exploitation of general properties to prune the search space and reduce the number of expensive tests. However, this approach does not take into account user preferences and expectations. In real-world applications, such as the characterization of areas crossed by motorways [14], a large number of spatial patterns can be generated even for a few hundred spatial objects. Nevertheless, most of discovered patters are useless for the application at hand. Therefore, it is important to allow the user to specify his/her bias for interesting solutions, and then to exploit this bias to improve both the efficiency of the system and the quality of the discovered rules. In SPADA, the language bias LB is expressed as a set of constraint specifications for either patterns or association rules. Users may specify the following pattern constraint:

pattern_constraint(AtomList, Min_occur, Max_occur)

where *AtomList* is a list of atoms (for atomic constraints) or a list of atom lists (for conjunctive constraints), while *Min_occur* (*Max_occur*) is positive number which

specifies the minimum (maximum) number of constraints in the *AtomList* that must be satisfied. When Max_occur = '_' no limitation is imposed on the maximum number of constraints. For instance, the following:

pattern_constraint(crossed_by_green_area(_,_), crossed_by_urban_area(_,_)],1,_)

specifies that at least one of the binary spatial predicates *crossed_by_green_area*, and *crossed_by_urban_area* must occur in the patterns filtered by SPADA, while the following:

pattern_constraint([[crossed_by_green_area(_,_), crossed_by_urban_area(_,_)] , [crossed_only_by_road(_)]], 1, _).

specifies that at least one of either the binary spatial predicates *crossed_by_green_area* and *crossed_by_urban_area* or the unary spatial predicate *crossed_only_by_road* must occur in the patterns filtered by SPADA. It is noteworthy that this simple specification allows users to define both conjunctive and disjunctive constraints.

During the rule generation phase, patterns that do not satisfy a pattern constraint are filtered out. This means that they are generated and evaluated anyway. This late exploitation of pattern constraints is due to the fact that if a pattern *P* does not satisfy a constraint (e.g. the lack of the predicate *crossed_by_green_area*), it is still possible that *P* descendants (i.e., more specific patterns) satisfy it. Therefore, pattern constraints do not prune the pattern space, but improve the efficiency of the mining process, since they prevent the generation of useless rules, and hence their evaluation.

A further pattern constraint takes into account the typing mechanism of the variables to be included in the rules. A variable *X* is untyped when it does not appear as first argument of a binary is-a atom in the rule. In some applications, the occurrence of untyped variables in a rule is undesirable. Therefore, users can specify the constraint *max_rules_untyped_vars(n)*, where *n* denotes the maximum number of untyped variables in the rules being generated. As in the previous case, the specification of this constraint affects the rule generation phase.

Users may specify constraints either on the antecedent or on the consequent of spatial association rules through one of the following facts:

body_constraint(AtomList, Min_occur, Max_occur).
head_constraint(AtomList, Min_occur, Max_occur).

where *AtomList*, *Min_occur* and *Max_Occur* have the same meaning as in the pattern constraint described above. For instance, the constraint *head_constraint([mortality_rate(_)], 1, 1)* specifies that a single occurrence of the unary predicate *high_mortality* must be in the head of the rules. As for pattern constraints, head and body constraints affect the rule generation phase. The main property of all described constraints is that they do not prevent the generation of candidate rules but only the evaluation of their confidence.

The LB described in this Section is a revisited version of the language bias proposed for SPADA in [1]. In particular, both pattern and rule constrains have been extended by introducing *Max_Occur* parameter that allows users to eventually specify the maximum number of constraints in the list to be satisfied. In addition, since association rules discovered by SPADA can have several conditions in the head, we have extended LB allowing users to specify the fact: *rule_head_length (Min_occur,*

Max_occur) in order to fix the minimum (*Min_occur*) and the maximum number (*Max_occur*) of predicates to be included in the head of generated rules.

By combining the rule filters *head_constraint([mortality_rate(_)], 1, 1)* and *rule_head_length(1, 1)* users is able to ask for rules containing only the predicate *mortality_rate* in the head. Rules in this form may be employed for spatial subgroup mining that is the discovery of interesting group of spatial objects with respect to a certain property of interest, as well as for classification purposes.

5. The Application: two case studies

In this section, we describe the application of SPADA to two distinct real-world problems, namely mining document images and mining geo-referenced census data. In the former problem, spatial objects are layout components extracted by means of a sophisticated document image segmentation algorithm. Layout components are in the same page of a document and have a common geometrical representation: they are all rectangles with edges parallel to the axes associated to the left and top border of a page. As result of the document understanding process, layout components may be associated with some components of the document logical structure, whose hierarchical organization defines the hierarchy of task-relevant objects. Discovered spatial association rules can be used in a generative way. For instance, if a part of the document is hidden or missing, strong spatial association rules can be used to predict the location of missing layout/logical components [7]. This problem is also related to document reformatting [6].

In the second problem, the goal is to perform a joint analysis of both socio-economic factors represented in census data and geographical factors represented in topographic maps. The discovery of interesting association rules on geographically distributed socio-economic phenomena can be a valuable support to good public policy. In this case, spatial objects are territorial units for which census data are collected as well as entities of the transport network (roads and rails), while the hierarchies are either based on layers of the topographic map or defined on the basis of a conceptual categorization or urban areas.

5.1 Document image processing

In this application SPADA takes in input a collection of ground facts describing both the layout and the logical structures of the documents processed by WISDOM++. Spatial features (relations and attributes) are used to describe the logical structure of a document image. In particular, we mention *locational* features such as the coordinates of the centroid of a logical component, *geometrical* features such as the dimensions of a logical component, and *topological* features such as relations between two components. We use the *aspatial* feature *type_of* that specifies the content type of a logical component (e.g. *image*, *text*, *horizontal line*). In addition there are other aspatial features, called *logical* features which define the label associated to the logical components. They are: *affiliation, page_number, figure, caption, index_term,*

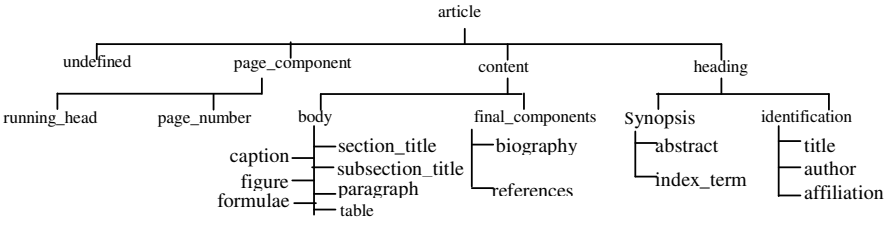

Fig. 2. Hierarchy of logical components

running_head, author, title, abstract, formulae, subsection_title, section_title, biografy, references, paragraph, *table, undefined*.

The specification of the domain specific knowledge allows SPADA to automatically associate information on page order to layout components, since the presence of some logical components may depend on the order page (e.g. *author* is in the first page). An example related to the first page is *at_page(X,first) :- part_of(Y,X), page(Y,first)*.

The specification of the hierarchy (Figure 2) allows the system to extract spatial association rules at different granularity levels.

In this task, the *ro* are the logical components associated with logical feature different from *undefined*. The *tro* are all the logical components. This is specified by means of the language bias *LB*. In particular, we ask for rules containing at least one binary spatial predicate:

pattern_constraint([only_middle_col(_,_),only_left_col(_,_),only_right_col(_,_),only_middle_row(_,_),only_upper_row(_,_),only_lower_row(_,_), to_right(_,_), on_top(_,_)],1).

Furthermore, we are interested in rules containing the *ro* in the antecedent. For instance, if we use *abstract* as *ro* the constraint is: *body_constraint([abstract(_)], 1).*

We investigate the applicability of the proposed solution on 19 real-world documents, which are scientific papers published as either regular or short in the IEEE Transactions on Pattern Analysis and Machine Intelligence in the January and February 1996 issues. Each paper is a multi-page document and has a variable number of pages and layout components per page, for a total of 179 document images and 2998 layout components. Eight-hundred and eleven layout components with no clear logical meaning are labelled as *undefined*. All logical labels belong to the lowest level of the hierarchy reported in the previous section. In Table 1 (second column) the average number of logical components (labels) is reported. The number of features to describe the documents presented to SPADA is 78,789, about 440 features for each page document. Average running time per document image is 1.32 secs (237.52/179), therefore this application of SPADA to document images seems scalable to larger collections of documents.

The number of mined association rules for each logical component at different granularity levels is also reported in Table 1. Many spatial patterns involving logical components (e.g., affiliation, title, author, abstract and index term) in the first page of an article are found. SPADA has found several spatial associations involving all logical components, references and biography excluded. This can be explained by the observation that the first page generally has a more regular layout structure and contains several distinct logical components.

An example of association rule discovered by SPADA is:

author(A) $\Rightarrow$ *on_top(A,B) , is_a(B,heading) , height(B,[1..174]) , type_text(A)*
(82.6%, 82.6%)

This means that 19 logical components which represent the *author* of some paper are textual components on top of a logical component B that is the *heading* of the

paper, with height between 1 and 174. At a lower granularity level, a similar rule is found where the logical component *B* is specialized as *abstract*:

author(A) $\Rightarrow$ *on_top(A,B) , is_a(B,abstract) , height(B,[1..174]) , type_text(A)*
(82.6%, 82.6%)

The rule has the same confidence and support reported for the rule inferred at the first granularity level.

Table 1. Number of rules.

	Tot No of Labels	No of Rules Level 1	No of Rules Level 2	No of Rules Level 3	No of Rules Level 4	Running Time (secs)
min_conf		*0.7*	*0.6*	*0.5*	*0.4*	
min_supp		*0.5*	*0.4*	*0.3*	*0.2*	
Affiliation	20	20	20	24	28	13.71
Page_Number	162	8	12	12	12	14.01
Figure	312	12	12	12	12	14.44
Caption	161	8	10	10	8	13.37
Index_term	10	29	41	74	74	15
Running_head	184	11	12	23	18	17.26
Author	23	48	56	56	56	14.24
Title	20	143	155	223	240	13.71
Abstract	19	31	42	74	94	16.51
Formulae	283	12	12	12	12	14.10
SubsectionTitle	25	14	26	26	30	14.03
Section_Title	59	26	26	38	38	14.14
Biografy	19	0	0	0	0	12.91
Paragraph	822	29	31	40	39	20.13
Table	39	7	13	14	13	17.52
References	19	0	0	0	0	12.44
TOTAL	**2177**	**398**	**468**	**638**	**674**	**237.52**

5.2 Geo-referenced exploratory data analysis

In this study we describe a practical example that shows how it is possible to employ ARES in performing data analysis on geo-referenced census data concerning Greater Manchester, one of the five counties of North West England, that is divided into censual sections or wards, for a total of two hundreds and fourteen wards. For this application, spatial analysis is enabled by the availability of vectorized boundary of wards as well as vector geographical data about transport network, waters, green and urban areas that allow us to investigate the mortality rate (i.e. percentage of deaths with respect to the number of inhabitants) from a spatial viewpoint according to some deprivation indices. Geographical layers are taken from the Meridian product of the Ordnance Survey.

In particular, we decide to mine spatial association rules relating wards, which play the role of reference objects, with topological related road network (i.e. motorways, primary roads, A- and B- roads), rail network, water network (i.e. rivers, canals and waters), green area (i.e. parks and woods) and urban area (i.e. small and large areas) as task relevant objects.

Therefore, by using FEATEX we extract facts concerning topological relationships between wards and roads, rails, waters, green areas and urban areas reported in the

spatial database for that area. An example of fact extracted by FEATEX is *crosses(ward_135, urbareaL_151)*. The number of facts is 784,107. Despite the complexity of the spatial computation performed by FEATEX to extract these facts, the results are still not appropriate for the goals of our data analysis tasks. Therefore, a domain specific knowledge should be expressed in form of a set of rules. Some of the rules used in this data mining task are:

crossed_by_urbanarea(X,Y) :- crosses(X,Y), is_a(Y,urban_area).
crossed_by_urbanarea(X,Y) :- inside(X,Y), is_a(Y,urban_area).
not_crossed_by_urbanarea(X) :- is_a(X,ward), \+ crossed_by_urbanarea(X,_).

Here the use of the predicate *is_a* hides the fact that a hierarchy has been defined for spatial objects belonging to urban area layer (see Figure 3). Similarly, four different hierarchies have been defined to describe road network, rail network, water network and green area. The hierarchies have depth three and are straightforwardly mapped into three granularity levels. Hence, these hierarchies are used to complete the domain specific knowledge by adding rules describing topological relationships and/or not-relationschips between wards and green area, transport and water net.

Until now, all extracted data and user-defined background knowledge are purely spatial. However, we can observe that the mortality rate of an area cannot be defined on the basis of the geographical environment alone. We select four deprivation indecies, namely Townsend index, Carstairs index, Jarman index and DoE index, we discretize them with RUDE and generate the following four binary predicates for SPADA: *townsend_idx, carstairs_idx, jarman_idx* and *doe_idx*. The first argument of the predicate refers to a ward, while the second argument is an interval returned by RUDE. The Townsend index is a measure of multiple deprivation that is computed at ward level according to the percentage of households that are not owner occupied, percentage of households with no car, percentage of households with more than one person per room and percentage of persons who are unemployed. Similarly, Carstairs index, Jarman index and DoE index are calculated using census data to measure socio-economical deprivation of a ward.

To complete the problem statement we specify a declarative bias both to constrain the search space and to filter out some uninteresting spatial association rules. In particular, we rule out all spatial relations directly extracted by means of FEATEX.

Moreover, by specifying the rule filters *head_constraint([mortality_rate(_)], 1, 1)* and *rule_head_length(1, 1)* we ask for rules containing only the predicate *mortality_rate* in the head. After some tuning of the parameters *min_sup* and *min_conf* for each granularity level, we decide to run the system with the following parameter values:

$$min\_sup[1]=0.1,\ min\_sup[2]=0.1,\ min\_sup[3]=0.05,$$
$$min\_conf[1]=0.3,\ min\_conf[2]=0.2,\ min\_conf[3]=0.1.$$

Despite the above constraints, SPADA generates 413 rules from a set of 100791 candidates. A rule returned by SPADA at the first level is the following:

is_a(A, ward), crossed_by_urbanarea(A, B), is_a(B, ***urban_area****),*
townsendidx_rate(A, high) $\Rightarrow$ *mortality_rate(A, high)* *(40.72%, 72.47%)*

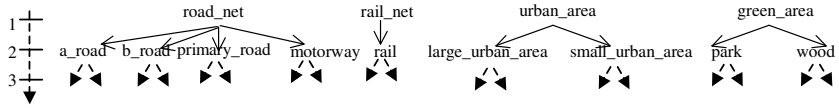

Fig. 3. Spatial hierarchies defined for four Greater Manchester layers: road net, rail net, urban area and green area.

which states that a high mortality rate is observed in a ward *A* that includes an urban area *B* and has a high value of Townsend index. The support (40.72%) and the high confidence (72.47%) confirm a meaningful association between a geographical factor such as living in deprived urban areas and a social factor such as the mortality rate. It is noteworthy that SPADA generates the following rule:

is_a(A,ward), crossed_by_urbanarea(A,B), is_a(B, urban_area),
⇒mortality_rate(A, high) (56.7%, 60.77%)

which has a greater support and a lower confidence. These two association rules show together an unexpected association between Townsend index and urban areas. Apparently, this means that this deprivation index is unsuitable for rural areas.

At a granularity level 2, SPADA specializes the task relevant object B by generating the following rule which preserve both support and confidence:

is_a(A, ward), crossed_by_urbanarea(A, B), is_a(B, ***urban_areaL****),*
townsendidx_rate(A,high) ⇒ mortality_rate(A, high) (40.72%, 72.47%)

This rule clarifies that the urban area B is large.

Similarly, SPADA discovers association rules involving low mortality wards. For instance, at the first granularity level, the rule:

is_a(A, ward), crossed_by_urbanarea(A, B), is_a(B, urban_area),
townsendidx_rate(A, low) ⇒ mortality_rate(A, low) (21.13%, 56. 94%)

states that a low valued Townsend index ward *A* that (partly) includes an urban area *B* presents a low mortality rate.

7. Conclusions

In this paper the discovery of spatial association rules by means of ARES in two real-world case studies, namely document image analysis and geo-referenced census data analysis, is illustrated. We also present some criteria to reduce the pattern search space and to filter extracted rules in order to discover interesting association rules according to user preferences. This is achieved by exploiting the high expressive power of rule miner SPADA 3.0, integrated in ARES, and allowing the definition of a language bias. Results show that ARES mines interesting rules at different granularity levels.

For future work we plan to investigate the improvement of ARES in order to implement a tight-coupling between SPADA and the spatial database.

Acknowledgments

We would like to thank Jim Petch, Keith Cole and Mohammed Islam (University of Manchester) for expert collection, collation, editing and delivery of the several data sets made available through Manchester Computing in the context of the IST European project SPIN (Spatial Mining for Data of Public Interest).

References

1. Appice A., Ceci M., Lanza A., Lisi F.A., Malerba D. (2003) Discovery of Spatial Association Rules in Georeferenced Census Data: A Relational Mining Approach, *Intelligent Data Analysis*.
2. Džeroski, S., Lavrac, N. (2001): Relational Data Mining, Springer-Verlag, Berlin.
3. Egenhofer, M.J, Herring, J.R.(1994): Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. In:. Egenhofer, M.J, D.M. Mark, J.R. Herring: *The 9-intersection: Formalism and its Use for Natural-language Spatial Predicates*,183-271.
4. Ester M., Frommelt A., Kriegel H.-P., Sander J. (1998) Algorithms for Characterization and Trend Detection in Spatial Databases. *Proc. 4th Int. Conf.* on *Knowledge Discovery and Data Mining*, New York City, NY, 44-50.
5. Han, J., Fu, Y. (1995): Discovery of multiple-level association rules from large databases. In: Dayal, U., P.M.D. Gray, S. Nishio (eds.): VLDB'95, Proceedings of the 21st International Conference on Very Large Data Bases, Morgan-Kaufmann 420-431.
6. Hardman L., Rutledge L., & Bulterman D.(1998): Automated generation of hypermedia presentation from pre-existing tagged media objects, *Proc. Of the 2nd. Workshop on Adaptive Hypertext and Hypermedia.*
7. Hiraki, K., Gennari, J.H., Yamamoto, Y., and Anzai, Y. (1991): Learning Spatial Relations from Images, *Machine Learning Workshop,* Chicago, pages 407—411.
8. Klösgen W., May M. (2002) Spatial Subgroup Mining Integrated in an Object-Relational Spatial Database. *Principles of Data Mining and Knowledge Discovery (PKDD), 6th European Conference*, LNAI 2431, Springer-Verlag, Berlin, 275-286.
9. Koperski, K., Adhikary, J., Han, J. (1996): Spatial Data Mining: Progress and Challenges. Proc. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal, Canada
10. Koperski K. , Han J., Stefanovic N. (1998) An Efficient Two-Step Method for Classification of Spatial Data. *Proc. Symposium on Spatial Data Handling (SDH '98)*, Vancouver, Canada, 45-54.
11. Lisi F.A., Malerba D. (2002): Efficient Discovery of Multiple-level Patterns. *Decimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati SEBD'2002*, 237-250.
12. Lisi F.A. & Malerba D. (2004). Inducing Multi-Level Association Rules from Multiple Relations. *Machine Learning*, to appear.
13. Ludl, M.-C., Widmer, G. (2000): Relative Unsupervised Discretization for Association Rule Mining. In: Zighed D.A., H.J. Komorowski, J.M. Zytkow (eds.): Principles of Data Mining and Knowledge Discovery, LNCS 1910, Springer-Verlag 148-158
14. Malerba D., Lisi F.A., Appice A. & Sblendorio F. (2002) Mining Spatial Association Rules in Census Data: A Relational Approach. In P. Brito and D. Malerba (Eds.), Notes of the ECML/PKDD 2002 Workshop on Mining Official Data, 80-93.
15. Malerba D., Ceci M., Berardi M. (2003). XML and Knowledge Technologies for Semantic-Based Indexing of Paper Documents, *Database and Expert Systems Applications*, DEXA 2003, LNCS, 2736, 256-265, Springer, Berlin, Germany.
16. Ng R., Han J. (1994) Efficient and effective clustering method for spatial data mining. Proceedings of the *International Conference VLDB*, Santiago, Chile, 124-155.
17. Sander J., Ester M., Kriegel H.P., Xu X. (1998) Density-Based Clustering in Spatial Databases: A New Algorithm and its Applications. *Data Mining and Knowledge Discovery, an International Journal*, Kluwer Academic Publishers, 2(2), 169-194.
18. Tsujimoto, S., & Asada, H. (1990): Understanding Multi-articled Documents, *in Proc. of the Tenth Int. Conf. on Pattern Recognition*, Atlantic City, N.J., 551-556.