An Application of Association Rules Discovery to Geographic Information Systems

Ansaf Salleb Christel Vrain

LIFO - Université d'Orléans - BP 6759
45067 Orléans Cedex 2 - France
email: {salleb,cv}@lifo.univ-orleans.fr

Abstract.
In this paper, we are interested in the problem of extracting spatial association rules in Geographic Information Systems (GIS). We propose an algorithm that extends existing methods to deal with spatial and non-spatial data over multiple layers. It handles hierarchical, multi-valued attributes, and produces general spatial association rules. We also present a prototype, which has been applied on a real and large geographic database in the field of mineral exploration.

1 Introduction

In this paper, we are interested in Spatial Data Mining [4, 6, 8], which leads to practical applications in many domains, such as geographic information systems (GIS). We present an application of mining association rules between geographic layers according to spatial and non-spatial properties of the objects.
The concept of association rules introduced by Agrawal [1] has been extended by Koperski and Han [7] to Spatial Data. For instance, the rule:

Is(X, largetown) $\wedge$ *Intersect (X, highway)* $\rightarrow$ *AdjacentTo(X, water)* (86%)

expresses that 86 % of the large towns intersecting highways are also adjacent to water areas (rivers, lakes, ...).
They have proposed an algorithm to discover such rules. In their works, non-spatial information is organized into hierarchies, and rules including this information are learned. Nevertheless, information belonging to different levels of hierarchies cannot appear in a same rule. We have extended their framework to introduce non-spatial predicates in rules, even when attributes are not hierarchical, and we have applied it to a real application provided by BRGM [1].
The application we are interested in is explained in Section 2 and formalized in Section 3. In Section 4, we present an algorithm for extracting spatial association rules in GIS and a prototype that has been developed and applied on geographic data provided by BRGM. We conclude with a discussion and perspectives, in Section 6.

[1] BRGM ('Bureau des Recherches Géologiques et Minières') is a French public institution, present in all regions of France and in some 40 countries abroad. Based on the Earth Sciences, its expertise relates to mineral resources, pollution, risks, and the management of geological environment (http://www.brgm.fr).

2 The application

A GIS stores data in two distinct parts: spatial data representing an object by its geometric shape (a point, a line or a polygon) and its location on the map, and non-spatial data storing properties of the geographic objects. Moreover, a GIS uses the notion of thematic layers in order to separate data into classes of same kinds. Each thematic layer is described by a relational table where each tuple is linked to a geometric object. For instance, the GIS Andes developed by BRGM [2] is a homogeneous information system of the entire Andes Cordillera. There are different layers in the system, which can be combined in any way by the user (Figure 1). Our goal is to find association rules between a given thematic layer and others layers, according to spatial relations and non-spatial attributes.

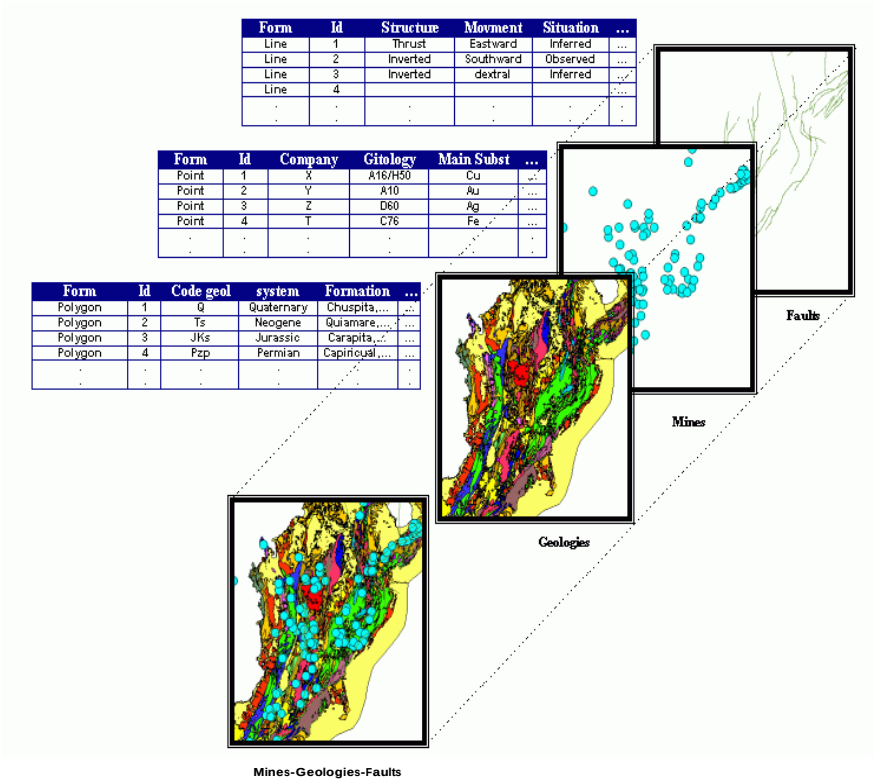

Figure 1. Three layers of an extract of the Andes

3 Problem formalization

We define two kinds of layers: **the Reference Layer** is the thematic layer which contains the spatial objects we focus on. It is unique, and in the following, the variable x is associated to the Reference Layer. **A Descriptive Layer** contains objects that have spatial relations with those of the reference layer.

In our work, a spatial association rule is a rule like:

$$P_1, P_2, ..., P_n \rightarrow Q_1, Q_2, ..., Q_m$$

where P_i, Q_j, are **predicates**. A predicate may be:
1. a definition predicate: It defines a thematic layer, its syntax is: *Definition_Predicate(var).* For instance, in the example of Section 2, *Mine(x)* is a definition predicate for the layer Mines.
2. a spatial predicate: It links the reference layer with a descriptive layer. Its general syntax is: *Spatial_Predicate*(var_i, var_j).
For instance, *Near_to(x,z)* denotes the proximity between x and z.
3. a non-spatial predicate: It describes a layer, its general syntax is: *NonSpatial_Predicate* $(var, Value)$. In our example, *Main_Substance(x, Ag)* expresses that the mine x has *silver* as main substance.
Our formalism allows to express non-spatial properties on objects. The algorithm below is able to discover association rules defined as follows:
• A *non spatial association rule* is a rule composed of the definition predicate concerning the reference layer and non-spatial predicates.
• A *spatial association rule* is a rule composed of at least two definition predicates, at least a spatial predicate and at least a non-spatial predicate.

4 The system

General Algorithm

Inputs:

- a spatial database SDB, a relational database RDB, taxonomies TDB.
- a reference Layer: Rl
- descriptive Layers: $Dl = \{L_i / i = 1, \ldots, n\}$
- a set of non-spatial attributes for Rl and for each $L_i \in Dl$
- a set of spatial predicates to compute: SP
- $Buffer$, $MinSupp, MinConf$

Outputs: solid, multi-levels, spatial and non-spatial association rules.
Begin
Step 1: Creation of link tables: $LDB = \bigcup_{i=1}^{n} LDB_i$
For each $L_i \in Dl$

- search Spatial Relations between Rl and L_i
- update LDB_i

Step 2: Computation of frequent predicate sets

- Creation of sets of predicates
- For each example E_j of LDB
  - Search the predicates sets that are true for E_j and increase their support
- Keep predicate sets that have sufficient support

Step 3: Generation of strong Spatial Association Rules
End

Details of the Algorithm

Let us notice first that each spatial object in a given layer has a unique key or

identifier. The algorithm is based on the building of *link tables*, each one relating the *reference layer* to a *descriptive layer*. The structure of a link table is composed of the following fields: the reference layer key, non-spatial attributes of the reference layer, a spatial relation linking an object of the *reference layer* and an object of the *descriptive layer*, the *descriptive layer* key, and non-spatial attributes of the descriptive layer. An *example* E_j is a set of tuples, *belonging to different link tables*, that have the same reference layer key j [3].
The support of a rule represents the percent of examples verifying all the predicates of the rule at the same time. A rule holds with confidence $c\%$, if $c\%$ of the examples of the *Reference layer* verifying the predicates given in the condition of the rule, also satisfy the predicates of the consequence of the rule.
Step 1 creates a link table LDB_i between the reference layer Rl and each descriptive layer of Dl, by searching for each object O_j of Rl, and for each layer of Dl, the objects O verifying at least a spatial predicate $sp(O_j, O)$. Then, frequent predicate sets are computed. For each link table LDB_i, a conjunction composed of a single predicate is created, as follows:
• Each non-spatial attribute of Rl (resp. L_i in the LDB_i) becomes a non-spatial binary predicate with variable x (resp. y_i).
• The set of constants that can appear in the second argument of the predicates are computed as follows:
– For each non-hierarchical attribute, we extract in the LDB_i its set of values.
– For each hierarchical attribute, values are obtained from its taxonomy.
• To each possible value of an attribute corresponds a predicate with this value as constant (second argument of the predicate, see Section 3).
• Each variable y_i must be linked to x by a spatial predicate (because the support concerns only the reference layer).
• The support of each predicate in the LDB_i is computed, and frequent predicates are kept in a list L.
• As in *apriori*, predicate sets of size k are generated by combination of frequent predicate sets of size (k-1), and only those with a sufficient support are kept.
Step 3 is a classical process for generating association rules from the set L.

Example: Let us suppose that we specify in the inputs the following parameters:
• Reference layer Rl =*Mines*, with non-spatial attributes: {*Gitology, Main_Subst*}. Note that Gitology is a hierarchical attribute (it takes its possibles values from the nodes of its taxonomy: $A_1, A_2, A_3, \ldots$).
• Descriptive layer L_1 =*Geologies*, with attributes: {*System, Code_Geol*}.
• Sp = {*Included_in, Near_to*}, Buffer = 10 km, MinSupp=5% and MinConf=40%
We aim at finding relations between *Mines* deposits, represented by points, and the nearest *Geologies* (polygons).
The link table LDB_1 is constructed, as shown in Figure 2. For instance, the example E_j is composed of two tuples of *key=2* in the link table and means that the mine number 2 is included in Geology 102 and near to Geology 3.
The support table of large predicate sets is then built. Note here that the predicate sets with frequency less than 130 (which represents 5% of 2618 mines in the Rl) are filtered out (Figure 3).

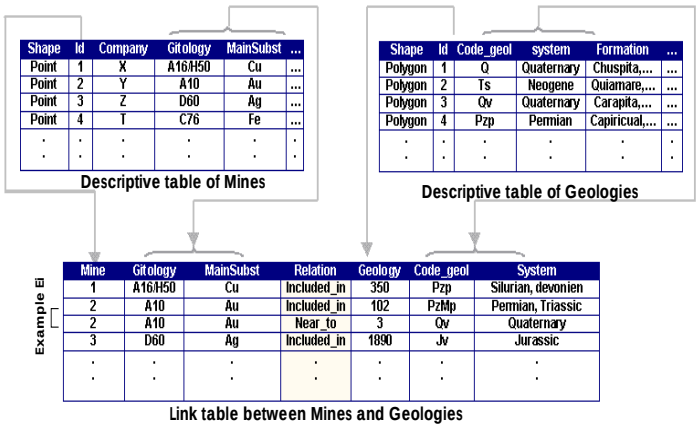

Figure 2. Link table of the example

Num	Frequent Predicate Sets	Frequency
1	Gitology (x, A)	533
2	Gitology (x, A1)	491
3	Main_substance (x, Ag)	529
4	Main_substance (x, Au)	459
5	Geologic_Code (y, Pzs), Is_inside(x,y)	384
6	System (y, Ordovician), Is_inside(x,y)	214
7	Gitology (x, A), Gitology (x, A1)	491
8	Gitology (x, A1), Main_substance (x, Ag)	180
9	Gitology (x, A1), Geologic_Code (y, Pzs), Is_inside(x,y)	177
10	Geologic_Code (y, Pzs), System (y, Ordovician), Is_inside(x,y)	209

Figure 3. An extract of the support table

Based on the support table, the algorithm generates the following rule with a support of 6.67%:

$$Mine(x) \wedge Geology(y) \wedge Code\_Geol(y,Pzs) \wedge Included\_In(x,y) \rightarrow Gitology(x,A1) \quad (6.67\%, 46.09\%)$$

which expresses that 46.09% of mines included in geologies of code *Pzs* have as gitology *A1*. We can classify generated association rules into three kinds:

- **Statistic rules**: they give the repartition of items in a hierarchy, such as:
$$Mine(x) \wedge Gitology(x, A) \rightarrow Gitology(x, A1) \quad (92.12\%)$$
- **Control rules**: experts can also verify some known correlations:
$$Mine(x) \wedge Gitology(x, H12) \rightarrow \ MainSubstance(x, AU) \quad (89.32\%)$$
- **New rules**: as for instance, the following rule with confidence 43.75%:
$$Mine(x) \wedge Fault(z) \wedge Gitology(x, C5) \wedge Near\_to(x, z) \rightarrow Structure(z, Strike\_slip)$$

We have implemented the algorithm presented in this paper, as a research prototype, named ARGIS. It has been developed in Avenue©, an object oriented language available in ArcView©[2]. In ARGIS, the user can formulate the inputs by means of an interactive graphical user interface. ARGIS handles multi-valued attributes and taxonomies, and the user can choose the levels that interest him.

[2] A GIS developed and commercialized by ESRI

We have experimented the prototype on 3 layers of GIS Andes, a database of about 150 mega bytes (15 MB for the Mines layer, 130 MB for the Geology layer and 5 MB for the Fault layer) composed of about 23 thousands of records, each time using a reference layer and a descriptive layer. The process has discovered about 70 spatial association rules, and some have been considered as very interesting by the experts.

5 Discussion and Conclusion

The system presented in this paper is an extension of a previous work of Koperski and Han [7], and an application of mining spatial association rules in GIS. First we have added non-spatial predicates to spatial ones. Second, as suggested by Koperski in [5], we can mine rules at cross-levels of taxonomies. However, to get interesting rules at low levels of the hierarchies, the support must be low. This leads to a lot of rules because we cannot give a variable support threshold according to the level in the hierarchies as done by Koperski, whose method is guided by a scroll of all the hierarchies at the same time. In order to handle a large number of layers, and in view of the phenomenal growth of data, the efficiency of the system must be improved in the following directions: pruning useless and redundant rules, interaction with statistics, dealing with numeric data, parallelization and use of multidimensional trees.

Acknowledgments: We wish to thank the BRGM, especially the Geology and Metallogeny Research group for providing us the opportunity to experiment our algorithm on their databases. **This work is supported by BRGM and by 'Région Centre'.**

References

1. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proc. 1994 Int. Conf. VLDB*, pages 487–499, 1994.
2. D. Cassard. Gis andes: A metallogenic gis of the andes cordillera. In *4th Int. Symp. on Andean Geodynamics*, pages 147–150. IRD Paris, October 1999.
3. L. Dehaspe and L. De Raedt. Mining association rules in multiple relations. In *Proc. of the 7th Int. Workshop on ILP*. Springer, 1997.
4. M. Ester, H.-P. Kriegel, and J. Sander. Spatial data mining: A database approach. *Lecture Notes in Computer Science*, 1262:47–66, 1997.
5. K. Koperski. *A progressive Refinement Approach To Spatial Data Mining*. PhD thesis, Computing Science, Simon Fraser University, 1999.
6. K. Koperski, J. Adhikary, and J. Han. Spatial data mining: Progress and challenges. In *SIGMOD'96 Workshop.DMKD'96, Canada*, June 1996.
7. K. Koperski and J. Han. Discovery of spatial association rules in geographic information databases. *Lecture Notes In Computer Science*, 951:47–66, 1995.
8. J. F. Roddick and M. Spiliopoulou. A bibliography of temporal, spatial and spatio-temporal data mining research. In *SIGKDD Explorations*, June 1999.