SPATIAL DATA MINING IMPLEMENTATION

Alternatives and performances

NADJIM CHELGHOUM, KARINE ZEITOUNI
PRiSM laboratory, University of Versailles
45, Avenue des Etats-Unis, 78 035 Versailles cedex, France

Abstract:

Spatial data mining requires the analysis of the interactions in space. These interactions can be materialized using distance tables, reducing spatial data mining to multi-table analysis. However, conventional data mining algorithms consider only one input table where each row is an observation to analyze. Simple relational joins between these tables does not resolve the problem and mislead the results because of the multiple counting of observations. We propose three alternatives of multi-table data mining in the context of spatial data mining. The first makes a hard modification in the conventional algorithm in order to consider those tables. The second is an optimization of the first approach. It pre-computes all join operations and adapts the conventional algorithm. The third re-organizes data into a unique table by completing -not joining- the target table using the existing data in the other tables, then applies any standard data mining algorithm without modification. This article presents these three alternatives. It describes their implementation for classification algorithms and compares their performances.

Key words: spatial data mining, spatial relationship, spatial database, spatial decision tree.

1. INTRODUCTION

Nowadays, spatial data mining (SDM) is a well identified domain of data mining. It can be defined as the discovery of interesting, implicit and previously unknown knowledge from large spatial data bases [Han 01], [Shashi 03], [Zeitouni 00a]. Its main characteristic is that it considers the spatial relationships- that we will call neighborhood [Egenhofer 93]. These

relationships are implicit and, to be exhibited, they require costly joins on spatial criteria. We have proposed in our previous works to materialize them by using a secondary structure called spatial join index [Zeitouni 00]. The idea is to calculate the exact spatial relationship between the locations of two collections of spatial objects and to stock it in a table with the following schema (ID1, spatial-relationship, ID2) (cf. Section 2). This allows us to palliate the spatial joins cost problem. Nevertheless, this organization cannot be analyzed directly by the conventional data mining methods because these methods consider that the input data is in a unique table and that each row of this table constitutes an observation or an individual object to analyze. So, we are confronted with the problem that we cannot exploit directly the data organized in several tables. It is possible to have one table by joining the different initials tables. However, this operation can duplicate some rows because the observations to analyze are in N-M link with the neighboring objects (cf. Figure 1). This misleads the obtained results using the conventional data mining methods because of the multiple counting of these observations.

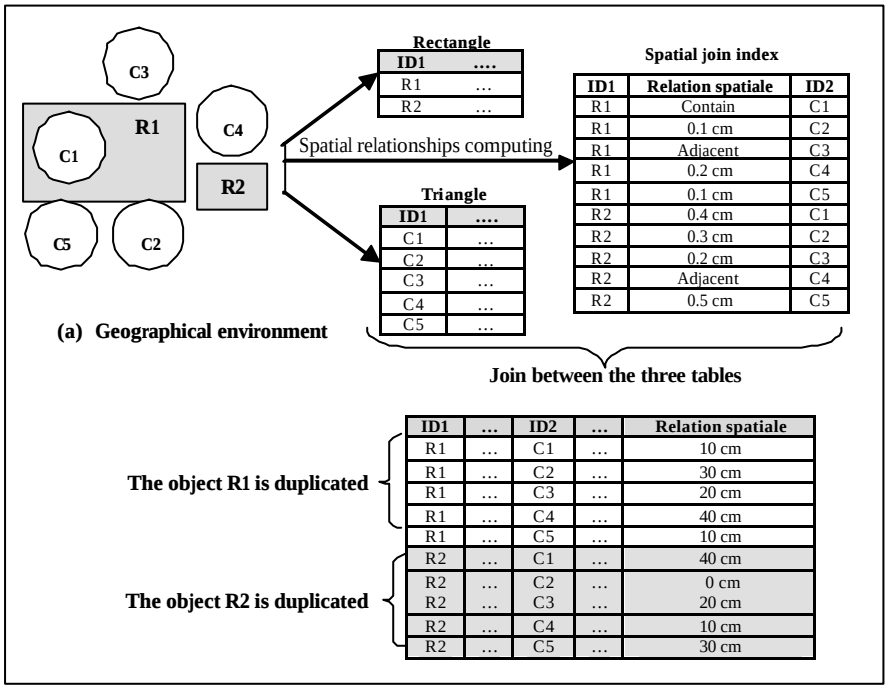

Figure 1: Problem generated by the joints between the tables

This article proposes three other alternatives of relational data mining in the context of spatial data mining. The first alternative queries on the fly the

three tables. It calls, when necessary, a multiple join operations and aggregation functions. The second alternative is an optimization of the first one. It materializes, once for all, the joins on keys between the different tables and modifies the conventional algorithms by avoiding rows' multiple counting. The third one transforms the multi-tables structure in a unique table without duplicating the observations, and then it applies a conventional data mining method. We describe these three alternatives, we present their application to the supervised classification and we compare their results.

The section 2 defines the spatial join index. Section 3 exposes the three proposed alternatives. In section 4, we present an application which allows us to illustrate these methods. The experimentations and the performance tests will be presented in the section 5, followed a discussion and a conclusion.

2. SPATIAL JOIN INDEX

Joint index has been proposed by Valduriez in [Valduriez 90] as a technique to accelerate the joins in relational database framework. Their extension to the spatial data has been proposed by Zeitouni and al. in [Zeitouni 00b]. This extension consists in adding a third attribute representing the spatial relationship between two objects (cf. Figure2). Each tuple (ID1, Spatial_Relationship, ID2) traduces the existence of a spatial relationship between the pair of spatial objects identified by ID1 and ID2. This relationship can be topological or metric. In the case of topological relationship, the Spatial_Relationship attribute will contain a negative code. Otherwise, it will store the exact distance value. For performance reason, the calculation of distances is limited to a given useful perimeter around objects.

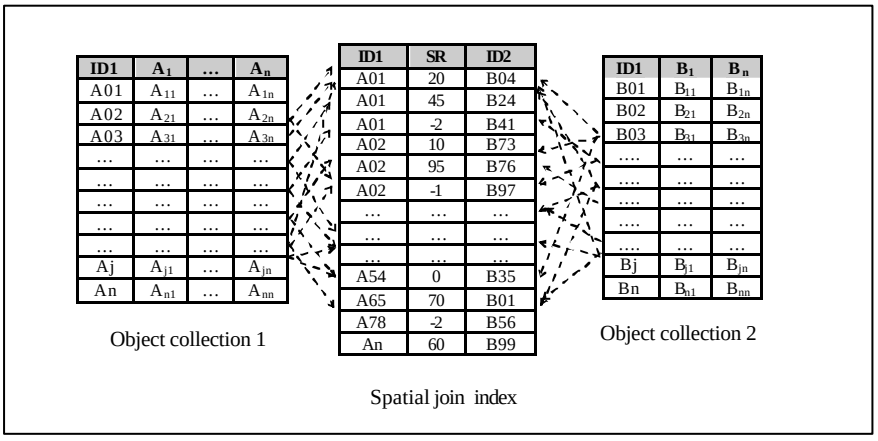

Figure 2: spatial join index

Spatial join index has two main advantages: (i) the storage of the spatial join index avoid the re-computing of the spatial relationship for every application, (ii) the same index allows optimizing the join operation according to all topological or metric spatial predicate.

3. PROPOSED SOLUTION

As emphasized above, the introduction of the spatial join index in the spatial data mining has the big advantage to reduce the spatial data mining problem to a multi-tables data mining problem. It formalizes neighborhood links within thematic layers and represents them using relational table. So, spatial data mining methods can directly use the relational schema instead of predicates set. Indeed, the methods use a target table, the join index table, and neighbor tables describing other themes. However, this new data organization is not directly exploitable by the conventional data mining methods because these last consider only one input table with one observation by row. Recently, some works in relational data mining [Dzeroski 01] have been done to work out this problem. They are based on induction logic programming (ILP). Their inconvenience is that their use requires expensive transformations of the relational data into a set of first order logic assertions. Instead, we propose three alternatives of multi-tables data mining in the context of the spatial data mining. They are detailed below.

3.1 First alternative: Querying on the fly the different tables

Unlike the conventional algorithms, the proposed method takes as input three tables: table of objects to analyze, neighborhood objects table and spatial join index table. Whenever, the attribute to analyze is a neighborhood attribute, the algorithm does two join operations between the target table, the spatial join index table and the neighbors table (It is here where the modifications on the existing algorithms got to be made). Otherwise, we apply the conventional algorithm without modification. An example of algorithm using this alternative is given in [Chelghoum 02].

The inconvenience of this solution is that the execution time deteriorates dramatically with the increase of the data volume (cf. Figure 9) because the joins queries are invoked in a loop, multiplying the table accesses. So, we propose another alternative that we present below.

3.2 Second alternative: Join materialization

This alternative materializes, once for all, the joins on keys between the three tables. This avoids the multiplication of the joins queries of the first alternative. However, these joins lead to the duplication of the analyzed objects. So, we are forced to modify the existing data mining algorithms in order to take in consideration this duplication in the different calculations (we count the observation only one time). For example, the calculation of the informational gain for observations represented on several rows should count the observation only one time. The figure 6 summarizes the modified CART method applied on the joins result, also in the context of the spatial data classification. This alternative has the advantage to be faster than the previous thanks to the joins materialization.

We note that these two alternatives (1 and 2) are not specific to the spatial data mining. They can be applied, in general, in a multi-table case. Nevertheless, they present an inconvenience: the modification of each algorithm is hard. So we cannot use the existing data mining tools.

3.3 Third alternative: Reorganizing the data

It reorganizes the data in a unique table by joining the three tables without duplicating the analyzed objects. The idea here is to complete, and not to join, the target table by the data present in others tables. We propose a new operator called COMPLETE defined below. Its principle is to generate for each attribute value of the linked table an attribute in the result table. The previous application of this operator has the advantage to avoid the duplication of the analyzed objects and to allow the use of any data mining method, without any modification.

Definition of COMPLETE operator

Let R (ID1, A_1,…, A_n), V (ID2, B_1, …, B_m) and I (ID1, ID2, W) are three tables. The keys are underlined in each table. The B_i (i=1, …, m) are the qualitative attributes and b_{ij} (j = 1 , …K_i) are their distinct values. Let F = {F_1, F_2,…, F_m} a set of aggregate functions.

COMPLETE (R, V, I, F) is a table T having the following schema:
T (ID1, A_1,… A_n, W_b_{11}, …, W_b_{1K1}, …, W_b_{m1}, …, W_b_{mKm}) where:

- ID1 is a key,
- $\forall t = (id1, a_1, a_2, \ldots, a_n, W\_b_{11}, \ldots, W\_b_{1K1}, \ldots, W\_b_{m1}, W\_b_{m2}, \ldots, W\_b_{mKm}) \in T$,
  - $(id1, a_1, a_2, \ldots, a_n) = \sigma_{(ID1 = Id1)}(R)$,

- $W\_b_{ij} = F_i\ (\sigma_{(ID1 = Id1)}\ (I) \infty \sigma_{(Bi = bij)}\ (V); W)$ if $\sigma_{(ID1 = Id1)}\ (I)$ is not empty[1], NULL value otherwise.

Examples

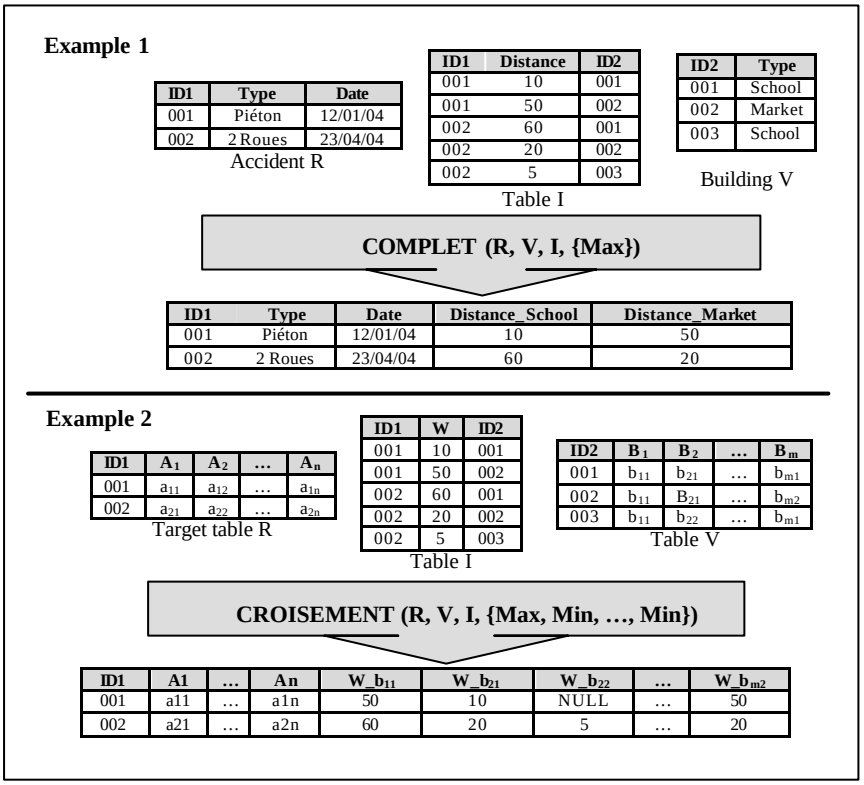

Figure 3: Application example of COMPLETE operator

Where table I is a weighted correspondence table that joins the target table R with the table V which contains other considered dimension. This operator is only recommended when the attributes Bi of V do not contain many distinct values. The COMPLETE operator main property is that the result always includes, as left part, all objects of R without duplication and that this one is completed in right part by the weights of the "dimension" coming from V and I. This case happens usually in the relational data where the correspondence table represents the links with N-M cardinalities. This operator can be seen as a mean of relational data preparation for data mining.

We notice that for a same tuple of R and a same value of b_j attribute of V, we can have several links in I with possibly different weights. The W_b_j value of the result table T must be unique. So, we introduced the aggregates

[1] ∞ and σ symbols express, respectively, the join and the selection operation.

functions in order to calculate only one value for W_bij. If there is not such link, the NULL value replaces this function (The rows not having a correspondent in I is completed by NULL values all as a left external join).

Recently, Oracle has proposed an operator called UNPIVOT to reorganize tables within ETL process in a data warehouse [Oracle9i 03]. The difference between the COMPLETE operator that we have proposed and the UNPIVOT operator is that the first one, contrary to the second, takes in input three tables and include the aggregate functions which fulfils our needs. The possible expression of the COMPLETE operator by Oracle's UNPIVOT is given by the following formula:

COMPLETE (R,I,V,F) = R ∞ UNPIVOT (F_1 (I ∞ V, ID_1 , B_1; W)) ∞ UNPIVOT (F_2 (I ∞ V, ID_1 , B_2 ; W))∞…∞ UNPIVOT (F_m (I ∞ V, ID_1 , B_m; W))

However, a direct implementation of this operator avoids the multiple joins in the case of several attributes as well as the storage of the intermediary results of UNPIVOT.

4. APPLICATION TO SPATIAL DECISION TREE

In the literature, two main works of spatial decision tree exists: Ester and al. [Ester 95] and Koperski and al. [Koperski98], but they are limited compared to our method.

Ester et al. [Ester 95] proposed an algorithm dealing with spatial databases based on ID3 [Quinlin 86]. They use the concept of neighborhood graph to represent the spatial relationships. This algorithm takes in consideration the properties of neighboring objects in addition to those of the actual object. But, each object could have many neighbors (e.g. an accident could be near a school and a bus stop). So, spatial criteria are not discriminative and the segmentation is wrong. Moreover, this method is limited to a single given relationship. Finally, it doesn't support the concept of thematic layers which is essential in geographical applications.

Koperski and al. [Koperski98] propose another classification method. The data are first generalized, then all "attribute = value" are transformed into logic predicates. Such transformation is costly. Furthermore, this method is limited to few spatial relationships.

We propose three spatial decision tree algorithms based on the three previously presented alternatives. In [Chelghoum 02], we proposed an algorithm based on the alternative 1. It is an extension of the CART method [Breiman 84] to the relational data. This extension results in the modification of the informational gain computing formula. The figure 4 summarizes this algorithm. The figure 5 describes the computing procedure of this criterion.

Another spatial decision tree algorithm has also been proposed using the alternative 2. It is also an extension of the CART method. The algorithm materializes, once for all, the joints on keys between the tables and builds then the spatial decision tree by taking in consideration objects' duplication. The figure 6 summarizes this algorithm and describes the informational gain computing formula.

Finally, the third proposed algorithm implements the third alternative. It corresponds to the application of this new operator "COMPLETE" in pretreatment, followed by the application of CART algorithm. The figure 7 describes this algorithm.

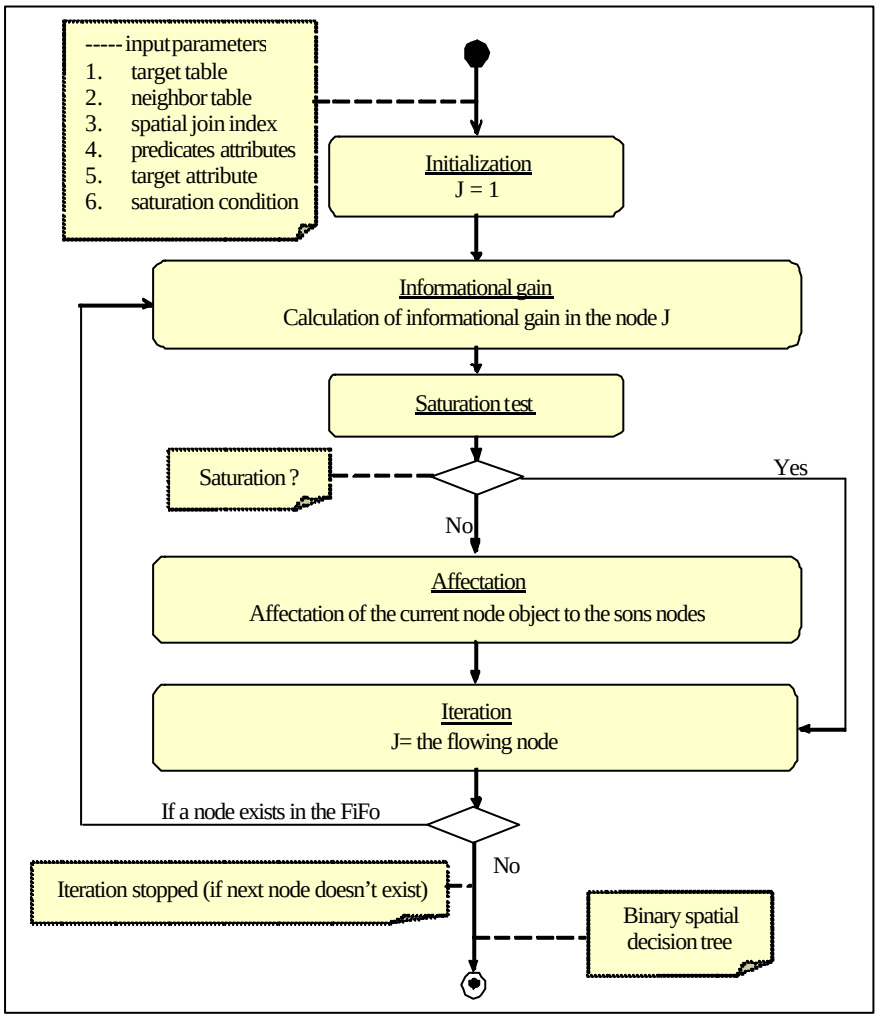

Figure 4: spatial decision tree algorithm using the alternative 1

INFORMATION GAIN PROCEDURE

- **INPUT PARAMETERS**
  1. Target_Table: The analyzed objects,
  2. Neighbor_Table: neighbor of analyzed objects,
  3. Spatial_Join_Index : Spatial join index table,
  4. Target_Attribute C: Attribute to predict (i.e. class label),
  5. Predictive_Attributes Att: Attributes from target table or neighbor table that could be used to predict the target attribute,
- **FORMULA: TWOING CRITÉRIUM FO R ONE VALUE OF ATTRIBUTE Î ATT**
  1. **If** the predictive attribute belong to the target table Þ **The formula is identical to the one of classical CART.**
  2. **Else**
     - Object number of left son NG:

```
SELECT COUNT ( DISTINCT A. identifiant ) AS NG
FROM Target_Table A, Neighbor_Table V, Spatial_Join_Index I
WHERE A. identifiant = I.ID1 AND I.ID2 = V.identifiant AND Left_Condition;
```

     - Object number of right son ND

ND = Object number of father node – NG

     - Object number for every class and in a given node NC_i

```
SELECT C, COUNT (DISTINCT A. identifiant) AS NCi
FROM Target_Table A, Neighbor_Table V, Spatial_Join_Index I
WHERE A. identifiant = I.ID1 AND I.ID2 = V.identifiant AND Node_Condition
GROUP BY C;
```

Figure 5: informational gain procedure using the alternative 1

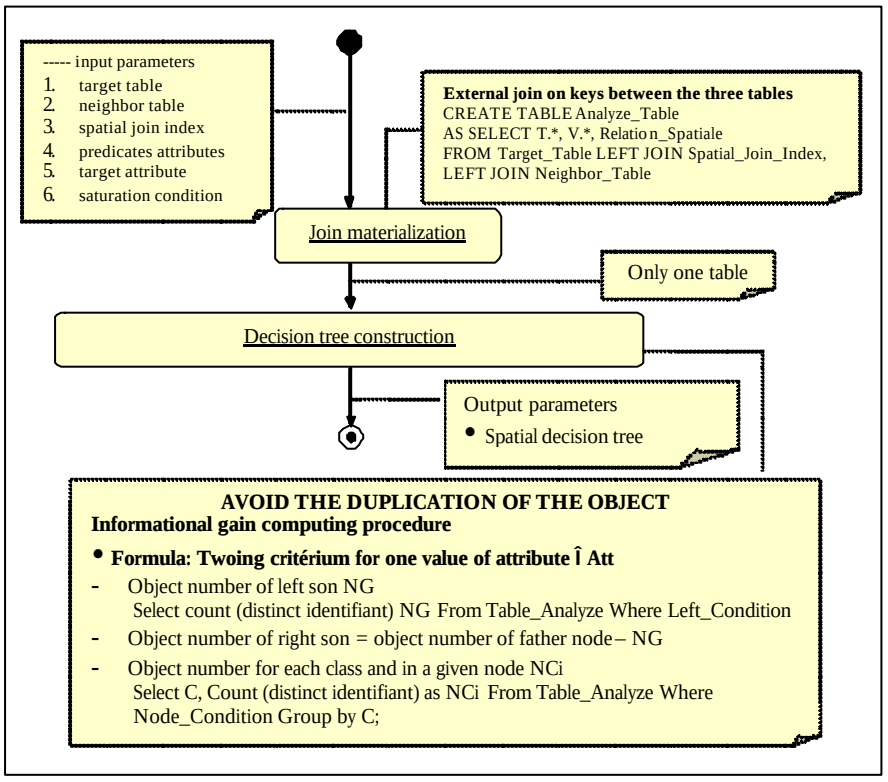

Figure 6: Spatial decision tree algorithm using the alternative 2

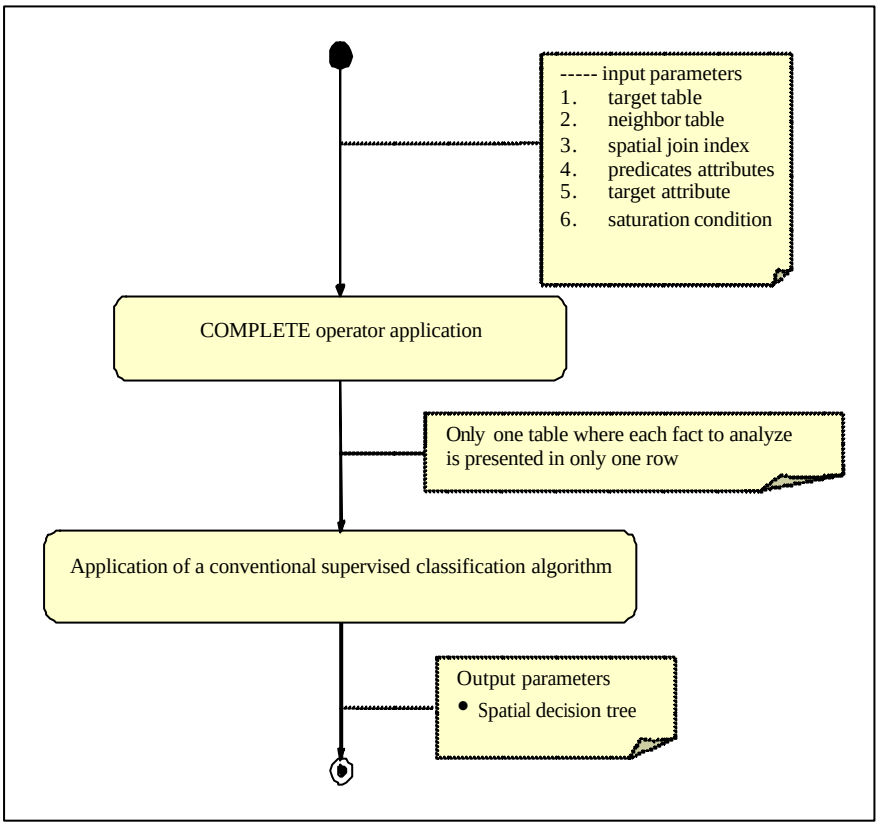

Figure 7: Spatial decision tree algorithm using the alternative 3

5. EXPERIMENTATION AND PERFORMANCE ANALYSIS

These three previous alternatives have been tested in the framework of traffic risk analysis. The analysis is done on a spatial database provided in the framework of an industrial collaboration. This base contains data on the road accidents and others on the geographical environment (buildings, roads …). The objective is to construct a predictive model by looking for some correspondences between the accidents and the other thematic layers as the road network, building… etc. It consists in applying the classification by decision tree while integrating the accidents spatial character and their interaction with the geographical environment. The experimentations

presented below will concern the application of the alternatives described in this article in the specific case of a spatial decision tree.

5.1 Experimental results

An example of result is given in the figure 8. It is obtained by using the third alternative. It classifies the accidents according to the involved categories (Pedestrian, 2 wheels- bicycles and motorcycles- or other – vehicles-). Explanatory attributes are whether linked to the road sections where the accidents are localized (P_SB: meaning the presence of stop bus), or are linked to the urban environment (School, market, administration…). Like in the first case, where there is only one road, or one crossroads, the attribute is added by simple joint. But in the second case, we can have more than one neighbor and multiple spatial relationships. Therefore, we join the problematic presented in this article.

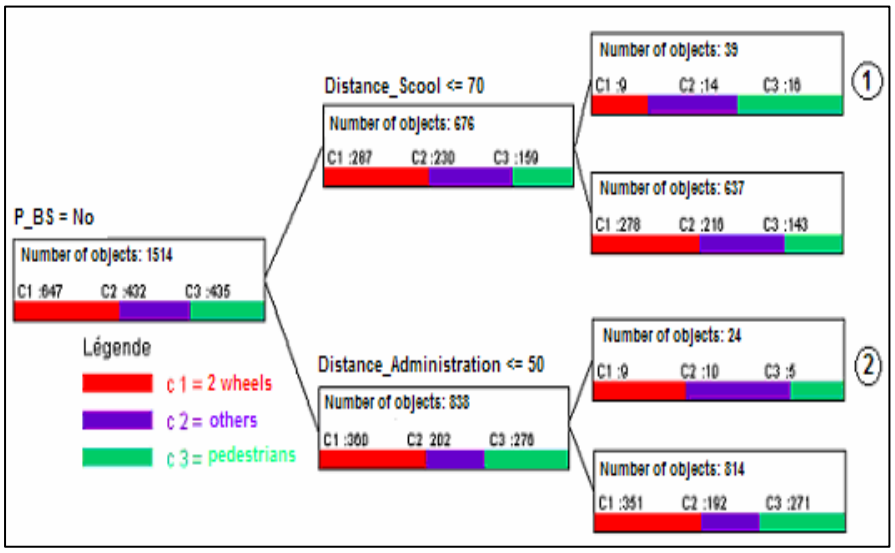

Figure 8: Spatial decision tree

As shown here, the first segmentation condition is the presence of bus stop. This variable belongs to the target table. The left son of the root corresponds to the accidents localized in road sections not having a bus stop. It is divided in two nodes. The left node contains the accidents that are near the schools where the rate of accidents involving some pedestrians is stronger. At this level, the segmentation condition is a combination of the "school" attribute value of the neighbor table and the spatial relationship "distances".

We can also underline the following rules:

- We have more accident of "Pedestrian" class when we are near the schools and there isn't a bus stop (node 1).
- We have less accident involving pedestrians near the administrations and there is a bus stop (node 2).

5.2 Performance and analyze

The table below (Table 1) summarizes the execution costs (in seconds) of each alternatives based on CART decision tree method. These experimentations were realized on a desktop PC with a 2.5 Ghz Pentium IV CPU. The implementation was realized in Oracle 9i DBMS and JAVA language. The algorithms 1, 2 and 3 correspond respectively to the alternatives 1, 2 and 3. The phase 1 corresponds to the previous data transformation step (to the join materialization for the alternative 2 or to the application of the COMPLETE operator for the alternative 3). The phase 2 represents the construction of the decision tree step.

The tests aim to compare the performances of each alternative. We kept three criteria: the size of the target table, the size of the linked table (neighbors table in SDM case) and the size of the correspondence table (spatial join index in SDM case). The figure 9 gives the execution time of the three algorithms according to the size of the target table.

			1st alternative	**2nd alternative**			**3rd alternative**		
A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
122	147	37	240	3	1	2	3	1	2
204	148	52	300	4	1	3	3	1	2
1594	6330	869	16860	7	1	6	9	5	4
3437	20180	869	24799	8	1	7	76	71	5
8668	20180	869	35205	18	2	16	157	150	7
15574	27054	869	48403	19	2	17	640	626	14
21892	53631	869	70020	56	2	54	925	882	43
29810	74302	869	88320	32	2	30	1372	1330	42

A: Size of R (objects)
B: Size of I (objects)
C: Size of V (objects)
D: Total time of the 1st alternative (seconds)
E: Total time of the 2nd alternative (seconds)
F: execution time of the first step (seconds)
G: execution time of the second step (seconds)
H: Total time of the 3rd alternative (seconds)
I: execution time of the first step (seconds)
J: execution time of the second step (seconds)

Table 1: Execution time

The analysis of these results shows that the execution time of the algorithm 1 is distinctly more important than the execution time of the algorithms 2 and 3. In fact, the repetitive and costly joins used in the

calculations penalize the algorithm 1. For example, for every class combination, attribute value and link in the algorithm 1, several joints are invoked. We also note that the algorithm 3 is a little less powerful than the algorithm 2. This is due to an increase of the data volume, compared to the join results, thanks to the COMPLETE operator. This can happen when the distinct values are numerous or when a few among them are bound with the target table. This decrease is noted in the data preparation phase (phase 1). The phase 2 is practically equivalent between the algorithms 2 and 3.

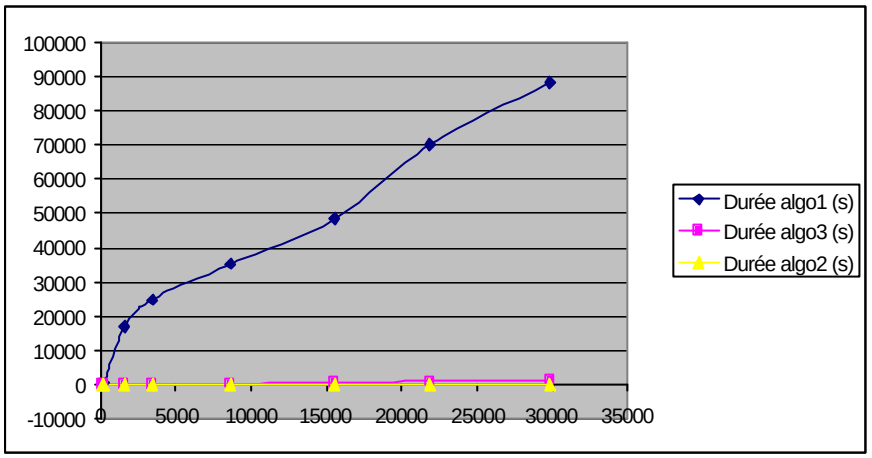

Figure 9: Execution time according to the target table size

The analysis of these results shows that the execution time of the algorithm 1 is distinctly more important than the execution time of the algorithms 2 and 3. In fact, the repetitive and costly joins used in the calculations penalize the algorithm 1. For example, for every class combination, attribute value and link in the algorithm 1, several joints are invoked. We also note that the algorithm 3 is a little less powerful than the algorithm 2. This is due to an increase of the data volume, compared to the join results, thanks to the COMPLETE operator. This can happen when the distinct values are numerous or when a few among them are bound with the target table. This decrease is noted in the data preparation phase (phase 1). The phase 2 is practically equivalent between the algorithms 2 and 3.

6. CONCLUSION

This article describes how to translate all spatial data problem to multi-tables data mining problem, and then, to reduce this multi-tables data mining

problem to mono-table data mining problem by introducing the COMPLETE operator definite in this paper. The big advantage is that it allows the use of any conventional data algorithm (clustering, association rules…).

We have proposed and analyzed three alternatives of relational data mining. Their application to the spatial decision tree method has been described. Their performances have been reported. The obtained results using the implemented prototype confirm the efficiency of our approach. Nevertheless, the tests carried out can guide the recommendation for one or more of these alternatives. Thus, when we want to have flexibility to apply any data mining tool, the previous usage of the COMPLETE operator is recommended. The algorithm 1 constitutes a naive method and it is more expensive in execution time. Consequently, it is necessary to avoid it. The join materialization is accompanied with a hard modification in the algorithms. Except this inconvenience, it appears to be the fastest.

Tracks to evolve these methods are under consideration. The optimization of conventional and spatial CART algorithm could use statistical pre-computations as proposed by Graefe and al. [Graefe 98]. It will be necessary to precisely measure the gain by considering the pretreatments cost. In addition, these previous counts could inspire the optimization of other methods of complex data mining as the association rules. Also in perspectives, the operator's extension in the context of a data warehouse schema where several tables are linked as a star. This induces to consider a multi-dimensional index instead of the join index. At the beginning, this work was guided by the spatial data mining problem, but this research can cover common problems of complex data mining due to their organization.

7. REFERENCES

[Breiman 84] Breiman L., Friedman J.H., Ol shen R.A., Stone C.J, Classification and Regression Trees. Ed: Wadsworth & Brooks. Montery, California, 1984.

[Chelghoum 02] Chelghoum N., Zeitouni K, A decision tree for multi-layered spatial data, In: Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa, Canada, July 8-12 2002.

[Dzeroski 01] Dzeroski S., Lavrac N., Relational Data Mining, Springer, 2001.

[Egenhofer 93] Egenhofer M.J., Sharma J, Topological Relations Between Regions in R2 and Z2, 5th International Symposium,SSD'93, Singapore, June1993, Springer-Verlag, pp. 316-331.

[Ester 97] Ester M., Kriegel H.P., Sander J, Spatial Data Mining: A Database Approach, In proceedings of 5th Symposium on Spatial Databases, Berlin, Germany, 1997.

[Graefe 98] Graefe G., Fayyad U., Chaudhuri S, On the efficient gathering of sufficient statistics for classification of large SQL databases, In Proceedings of the Fourth International Conference on Knowledge Discovery and Data-Mining (KDD 98), AAAI Press, New York City, August 27-31, 1998.

[Han 01] Han J., Kamber M, Data Mining. Concepts and Techniques. Morgan Kaufmann Edition. 2001.

[Knobbe 98] Knobbe A.J., Siebes A., Wallen V., Daniel M.G, Multi-relational Data Mining, Technical Report of CWI, INS-R9908, ISSN 1386-3681, http://www.cwi.nl/static/publications/reports/abs/INS-R9908.html, 1999.

[Knobbe 99] Knobbe. A.J., Siebes A., Wallen V., Daniel M.G, Multi-relational Decision Tree Induction, In Proceedings of PKDD' 99, Prague, Czech Republic, Septembre 1999.

[Koperski 98] Koperski K., Han J., Stefanovic N, An Efficient Two-Step Method for Classification of Spatial Data, In proceedings of International Symposium on Spatial Data Handling (SDH'98), p. 45-54, Vancouver, Canada, July 1998.

[Lefébure 98] Lefébure R., Venturi G, Le Data Mining, Eyrolles, 1998.

[Quinlan 86] Quinlan J.R, Induction of Decision Trees, Machine Learning (1), pp 82 - 106, 1986.

[Shashi 03] Shashi S., Sanjay C. Spatial Databases: A Tour, Prentice Hall, p237, 2003.

[Valduriez 90] Valduriez P, Join indices, ACM Transactions on Database Systems, 12 (2), pp 218-246.

[Zeitouni 00a] Zeitouni K, Fouille de données spatiales, Revue internationale de géomatique n° 4/99, Numéro spécial, Edition Hermès Sciences, Avril 2000.

[Zeitouni 00b] Zeitouni K.,Yeh L., Aufaure M-A, Join indices as a tool for spatial data mining, Int. Workshop on Temporal, Spatial and Spatio-Temporal Data Mining, Lecture Notes in Artificial Intelligence n° 2007, Springer, pp 102-114, Lyon, France, September 12-16, 2000.

[Oracle9i] Oracle9i Warehouse Builder Transformation Guide (2003), Release 2 (9.0.4) for Windows and UNIX, Part No. B10658-01, February 2003, Oracle Corporation.