Fuzzy Spatial OQL for Fuzzy Knowledge Discovery in Databases

Nara Martini Bigolin* and Christophe Marsala

LIP6, Université Pierre et Marie Curie,
4 place Jussieu, 75252 Paris cedex 05, FRANCE.
email: {Nara.Martini-Bigolin, Christophe.Marsala}@lip6.fr

Abstract. In this paper, we introduce a fuzzy spatial object query language, called FuSOQL, to select, process and mine data from Spatial Object-Oriented Databases (SOODB). Fuzzy set theory is introduced in this extension of OQL to handle spatial data. Afterwards, the knowledge discovery process is applied to the selected data. In our case, this data mining is done by means of a fuzzy decision tree based technique. An experiment on a region of France is conducted with this algorithm to discover classification rules related to houses and urban area.

Keywords: Data mining, knowledge discovery in databases, spatial object-oriented databases, fuzzy decision tree.

1 Introduction

Knowledge discovery in databases (KDD) [7] has been used to extract implicit information from vast amounts of data. Recently, this technology has attracted the interest of researchers in several fields such as databases, statistics, machine learning, data visualization and information theory.

Researchers in Geographic Information Systems (GIS) [8] have also shown interest in knowledge discovery in spatial databases, called *Spatial Data Mining*, which has been defined as *the extraction of interesting spatial patterns and features, general relationships between spatial and non-spatial data, and their general data characteristics not explicitly stored in spatial databases* [14]. This technology is becoming more and more important because a tremendous amount of spatial and non-spatial data have been collected and stored in large spatial databases using automatic data collection tools. The main challenge of Spatial Data Mining has been the relevant data selection to discover knowledge. Thus, new data selection methods have to be studied. Usually, these methods have been developed as extensions of query languages. The more significant query languages to mine spatial data have been developed to discover knowledge from relational databases [6], [10], [12]. However, the development of query languages to access Spatial Object-Oriented Databases (SOODB) is still an open research area with large potential.

* Supported by the CNPq - Brazil.

In this paper, an object query language (OQL) extension to select and process data from a SOODB is presented. Fuzzy set theory is introduced in this language to handle spatial data. Afterwards, the knowledge discovery process can be applied with the selected data. For instance, in our case, this process is a fuzzy decision tree based technique.

This paper is composed as follows: in Section 2, query languages are introduced. In Section 3, our approach based on the extension of a query language to discover knowledge from an SOODB, is presented. An application of this approach is given in Section 4. Finally, we conclude and present some directions for future works.

2 Query languages

The first standard query language to access a relational database is SQL [11]. Its basic structures consists of **select**, **from** and **where** clauses. Typical query has the form: `SELECT` A_1, A_2,...,A_n `FROM` r_1, r_2,...,r_n `WHERE` P. where each A_i represents an *attribute* of a database, each r_i is a data collection (a *relation* in this case), and P is a *selection predicate* (condition).

Severals extensions of SQL were developed to handle complex data such as object structure (Object Query Language (OQL) [4]), spatial data (Spatial SQL [5]. Others extensions of SQL integrating operations to process data have been proposed. For instance, a Fuzzy Query Language (Fsql) [2] processes fuzzy data, and a Data Mining Query Language (DMQL) [9] discovers knowledge.

2.1 Object Query Language

An OQL [4] is an extension of the standard query language SQL to query an OODB. An OODB is composed by a set of *objects*. An object is associated with an *object-identifier* and a *value*. A value possesses a type either *atomic* (string,...) or *structured*. The structure can be a *collection* (a list, a set,...) or a *tuple* (a set of typed attributes). Objects are grouped into *classes* which are organized in *hierarchies*. The object's behavior is determined by a set of methods. The instances of a class are defined as a set of *objects*. Each object is associated with a *name* that references it in the database.

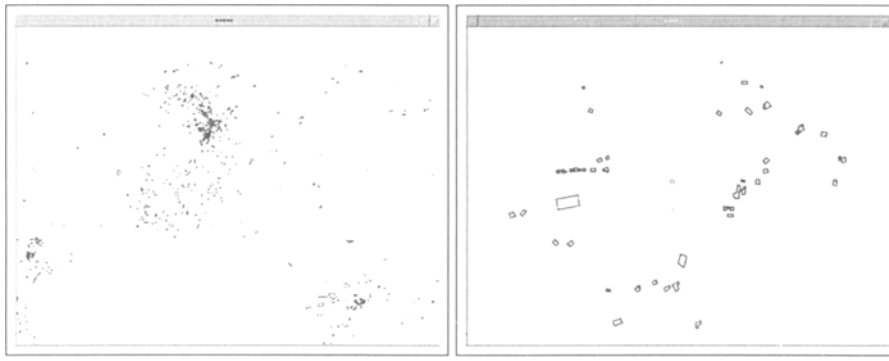
In an ODL, the statement of the `select` clause can be composed by, for instance, a set of objects, structures,... The statement of the `from` clause is composed by *expression paths* associated with a name. An expression path describes a path in the hierarchies beginning from this name. The statement of the `where` clause is a condition on the objects to select. The answer to a query is a set of objects or a set of values for these objects.

2.2 Spatial Query Language

A spatial query language [5] handles spatial data which are stored, for instance, in a geographical database. Such a database is composed of non-spatial and

spatial data. Non-spatial data are information describing objects such as: name, population of town etc... Spatial data specify the localization of the non-spatial data. It is generally represented by three spatial primitives: points, lines and area. A point represents (the geometric aspect of) an object for which only its location in space, but not its extent, is relevant.

For example, a city can be a point in a large geographic area (a large scale map). A region is the abstraction for something having an extent in 2d-space, e.g. a country, a lake, a national park or a house in small scale map (Figure 1a). A line is the basic abstraction for facilities for moving through space, or connections in space (roads, rivers, cables for phone, electricity, etc) [13].



1a. A region of France (1:125000) **1b.** Result of a query (1:31860)

Figure1. Geographical data

Non-spatial data can be handled with a classic query language, but the characteristics of spatial data require specific operations that can process and handle graphically spatial data. In a spatial query, these operations are represented by methods in the `where` clause. The `select` clause can be done by means of the graphical interface.

2.3 Fuzzy Query Language

A Fuzzy Query Language [2] is based on fuzzy set theory. In classical theory, given a set $\mathcal{X}$ and a set $U \subseteq \mathcal{X}$, each element $x \in \mathcal{X}$ either belongs to U or does not belong to U. It can be summarized as follows: given a set $\mathcal{X}$ and a set $U \subseteq \mathcal{X}$, let μ_U be the *characteristic function* such that: $\mu_U : \mathcal{X} \longrightarrow \{0, 1\}$, and $\forall x \in \mathcal{X}$, if $x \in U$ then $\mu_U(x) = 1$, otherwise $\mu_U(x) = 0$. In *fuzzy set theory* [17], the *membership degree* of an element x can vary from 0 to 1. The membership function μ_U of the *fuzzy set* U is defined as: $\mu_U : \mathcal{X} \longrightarrow [0, 1]$.

Given a numerical attribute U that takes numerical values in $\mathcal{X}$, a *fuzzy partition* can be defined on $\mathcal{X}$ by means of a set of fuzzy sets of $\mathcal{X}$. For instance, in Figure 2, the numerical attribute *distance* and its fuzzy values (*near*, *far*, *very far*) forming a fuzzy partition of its universe $\mathbb{R}^+$ are represented.

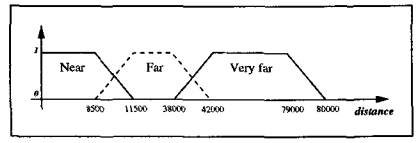


Figure2. Fuzzy values for the distance

The **where** clause of a fuzzy query is composed by fuzzy values. The answer to this kind of query is a fuzzy set of elements defined by these fuzzy values.

2.4 Data Mining Query Language

Data mining query language is used to discover knowledge in databases. This kind of languages has been introduced by [9] for relational databases. Their language is an extension of the classical SQL to mine a relational database. It is based on four major primitives: the set of data relevant to data mining, the kind of rules to discover, the background knowledge and the justification of the interestingness of the knowledge (*cf.* [9]). In addition to the **select**, **from** and **where** part, a data mining query is composed by a clause that defines the kind of rules to discover and the data mining algorithm to be used.

3 Fuzzy Spatial Object Query Language and Data Mining

To discover knowledge from an SOODB, both the object-oriented nature of the data and their spatial specificity need to be taken into account simultaneously. However, no data mining techniques that can handle these two data properties exist yet.

We propose a *Fuzzy Spatial Object Query Language* (FuSOQL) that is an object query language (OQL) extension to select and process data from a SOODB. Fuzzy set theory is introduced in this langage to handle spatial data. Afterwards, a data mining process is applied to the selected data to discover knowledge. A *Fuzzy Spatial Object Query* has the form:

DATAMINING <*Data mining algorithm*>
WITH
  SELECT A_1, A_2,...,A_n
  FROM o_1, o_2,...,o_n
  WHERE P

In the clause **DATAMINING**, the statement <*Data mining algorithm*> is the algorithm that has been chosen to mine the selected data. The other part of the query is a traditional **select** clause extended to spatial objects stored in object structure. The *condition* P of the clause is composed by: graphical methods of the graphical interfaces to visualize spatial data; spatial methods to compute informations from spatial data; and fuzzy methods to process numerical values of spatial data. These methods make use of background knowledge such as the

semantics linked to the object structure, the spatial object topology and the expert knowledge.

3.1 FuSOQL : Fuzzy Spatial Object Query Language

The data selection is made through a fuzzy spatial query $Q : SOODB \longrightarrow SO$. This query is run on the $SOODB$ to extract a subset of spatial objects SO relevant to the data mining task. Afterwards, a process is done on every spatial data stored in an object structure. This process is a function $\sigma : SO \longrightarrow TS$ which transforms a set of spatial objects SO into a training set TS. Such processing functions can be of two kinds: mathematical functions or fuzzy set theory based functions.

Mathematical functions. The transformation of a spatial object $O_S \in SO$ into a data set $D \subseteq TS$ is performed thanks to a set $\mathcal{S}$ of *compute* operations C. We have $\mathcal{S} = \{C \mid C : SO \longrightarrow \mathcal{X}_C\}$ where $\mathcal{X}_C$ is the set of values computed by the function C.

For instance, a *line* l is composed by a set of points $\{p_{l_1}, ..., p_{l_n}\}$ and the Euclidean distance is the function $C : SO \longrightarrow \mathbb{R}$ defined as $C(l) = ||p_{l_1} - p_{l_n}||$.

Fuzzy set theory based functions. For a given *fuzzy* operation f, if $\mathcal{X}_f$ is a set of continuous values (*ie.* $\mathcal{X}_f \subseteq \mathbb{R}$), a transformation $T_{\mathcal{X}_f}$ of these values into fuzzy values can be done according to a fuzzy partition of $\mathcal{X}_f$ into m fuzzy sets $U_1, ..., U_m$. Thus, we have $T_{\mathcal{X}_f} : \mathcal{X}_f \longrightarrow [0.1]^m$.

For instance, the Euclidean distance $||p_1 - p_2||$ between two points is a real value from $\mathbb{R}^+$. Given the set $\{near, far, very\ far\}$ of fuzzy sets describing the distance (see Figure 2), the real value $||p_1 - p_2||$ is converted into a set of three membership degrees to each fuzzy set: $\mu_{near}(||p_1 - p_2||)$, $\mu_{far}(||p_1 - p_2||)$ and $\mu_{very\ far}(||p_1 - p_2||)$.

3.2 Data mining: Fuzzy decision trees

The statement of the **DATAMINING** clause transforms a training set into a set of rules. Given a training set TS and a particular attribute $A_d \in \mathcal{A}$ (the decision), provided by the expert, the data mining step is a *mining* operation $\Theta : TS \times \mathcal{A} \longrightarrow RB$. This operation generates a set $RB = \{R_1, ..., R_N\}$ of rules $R_i : Pr_i \longrightarrow Co_i$. The premise Pr_i of a rule R_i is a conjunction of tests $A_k = a_{k_j}$ on values a_{k_j} of attributes $A_k \in \mathcal{A} - \{A_d\}$. The conclusion Co_i of a rule is a value d_i for A_d.

In our application, the **DATAMINING** clause is performed thanks to the *fuzzy decision trees* data mining technique [3, 15]. Each node in a decision tree is associated with a test on the values of an attribute, all edges from a node are labeled with values of the attribute belonging to a partition of its universe, and each leaf of the tree is associated with a value of the class. Edges can also be labeled by fuzzy values that lead to the generalization of decision trees into *fuzzy decision*

trees [3]. Fuzzy decision trees handle fuzzy values either during their construction or when classifying new cases. The use of fuzzy set theory enhances the understandability of decision trees when considering numerical attributes. An example of a fuzzy decision tree is given in Figure 4. A decision tree can be constructed from a set of examples (the *training set*). Each example is a case completely known, associated with a pair [description, class] where the *description* is a set of pairs [attribute, value] which is the available knowledge. Moreover, a fuzzy decision tree is equivalent to a fuzzy rule base $RB = \{R_1, ..., R_N\}$. A path of the tree is equivalent to an IF...THEN rule $R : Pr \longrightarrow Co$. Where the premise Pr is composed by tests on values of attributes, and the conclusion Co is the value of the decision that labels the leaf of the path.

4 Application

An application of our method is presented in this section. Its aim is to find a set of classification rules related to the description of houses and their localization in an *urban* or in a *non-urban* area.

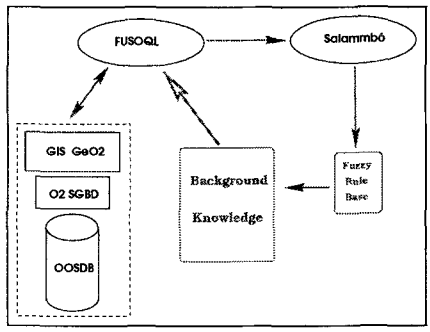


Figure3. System architecture

The system architecture used in this application is shown in Figure 3. The FuSOQL is developed on top of the O_2 DBMS [1] using the GIS GeO_2 component [16]. The queries are performed using background knowledge as detailed in Section 3. The Salammbô system of [15] is used to generate a fuzzy decision tree from the answer to the Fuzzy Spatial OQL. The SOODB and the GIS were provided by the French "Institut Géographique National" (IGN).

4.1 Query

The used fuzzy spatial object query is:

```
DATAMINING Fuzzy Decision Tree
WITH
   SELECT x.house
   FROM x in Database1
```

WHERE x.house→inArea(CoordPtMin, CoordPtMax);

The clause **DATAMINING** is based on the algorithm to build fuzzy decision trees. It extracts a set of fuzzy rules, given as a fuzzy decision tree, from the data selected by means of the clause **select**. In the *condition* of this clause, the method *inArea(CoordPtMin, CoordPtMax)* determines whether an object *house* is in the area defined by the two points *CoordPtMin* and *CoordPtMax*.

This area has been defined by the user with the graphical interface system. A point has been selected with the mouse. It is associated with the center of the area and **CoordPtMin** and **CoordPtMax** are the boundaries of this area, computed from this selected point.

4.2 Fuzzy spatial object query

First of all, the result of the **select** clause is the data set $\mathcal{H}$ of all houses pertaining to the given area (see Figure 1b). Each object from $\mathcal{H}$ is associated with additional information to make up an example of a training set. This information is computed by means of knowledge related to the application. For instance, such knowledge is represented as the mathematical function $d(p_1, p_2)$ which computes the Euclidean distance between the two points p_1 and p_2.

A particular kind of such knowledge consists in fuzzy modalities on the numerical universe of distances (Figure 2). Given a house h, the number of houses in the three fuzzy area defined by these fuzzy modalities is valued. For each house h' different from h, the distance $d(p_h, p_{h'})$ is evaluated. This distance is transformed into membership degrees $\mu_{near}(h')$, $\mu_{far}(h')$ and $\mu_{very\ far}(h')$. Thus, the number of houses in the area defined by the modality *near* is given by the fuzzy measure of cardinality: $Nr\ near = \sum_{h' \in \mathcal{H}} \mu_{near}(h')$ And so on, for the other modalities. Some examples of the obtained training set are shown Table 1.

Table1. Examples from the Training set

House	*Nr near*	*Nr far*	*Nr very far*	*Urban*
h1	0.1	4.2	5.7	No
h2	2.0	2.0	6.0	No
h3	3.3	3.8	7.9	No
h4	16.3	15.7	12.0	Yes
h5	11.0	24.2	9.7	Yes
h6	7.7	20.2	16.1	Yes

4.3 Data Mining

The statement of the **DATAMINING** clause is the algorithm to construct fuzzy decision trees implemented in the system Salammbô [15]. A fuzzy decision tree, that can be considered as a set of classification rules, is constructed from the training set obtained in the last section (*cf* Figure 4). It enables us to decide wether a house is urban or not.

As mentioned, the Salammbô system generates automatically a fuzzy decision tree and determines fuzzy modalities for the universe of values of each numerical attribute. The obtained fuzzy modalities on the number of houses in area are given in Figure 2. These fuzzy modalities will label the premises of the induced classification rules.

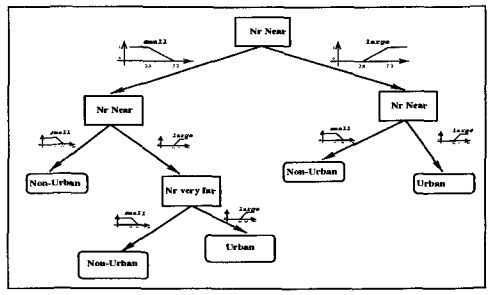


Figure4. Fuzzy decision tree

4.4 Validation of the fuzzy rule base

This set of rules, induced from houses belonging to a zone around a particular town, has been tested with other houses pertaining to a zone around another town. The selection clause and the processing step 4.2 were resumed with another center for a zone to generate a set (the *test set*) to check the fuzzy rule base obtained in step 4.3.

In our application, the studied region (Figure 1a) is composed of three towns T_1, T_2 and T_3. A training set was generated from a zone around the town T_1 (Figure 1b) and a test set was generated from a zone around the two other towns. The average error rate when classifying houses around towns T_2 and T_3 with the obtained fuzzy decision tree is 10.1%. In other words, given 413 houses from the new zone, 371 houses are perfectly classified as *urban* or *non-urban* with the induced set of fuzzy rules.

5 Conclusion

In this paper, we propose a fuzzy spatial object query language, called Fu-SOQL, which selects and processes data from Spatial Object-Oriented Databases (SOODB). This language is based on the introduction of mathematical and fuzzy set theory based functions to process and mine a SOODB. After a spatial object query and a mathematical and fuzzy preprocessing, we apply a fuzzy decision tree based technique to discover knowledge. This introduction of a preprocessing step and a fuzzy decision tree technique enables us to handle spatial data related to a geographical region. Our algorithm has been applied and validated on a region of France to discover characterization rules related to houses and urban area.

In future work, new experiments will be conducted to discover another kind of knowledge. Moreover, we plan to automate the process of building the queries used to construct the training set.

Acknowledgments

This work has been made possible thanks to the database provided by IGN (French National Geographic Institute).

The authors express their thanks to Bernadette Bouchon-Meunier and Anne Doucet for their guidance and their helpful comments.

References

1. F. Bancilhon, C. Delobel, and P. Kanellakis. *Building an Object-Oriented Databases Systems: The story of 02.* Morgan Kaufmann, 1992.
2. P. Bosc and O. Pivert. Sqlf : A relational databases language for fuzzy query. *IEEE Trans. on Fuzzy Systems*, 3:1–17, 1995.
3. B. Bouchon-Meunier, C. Marsala, and M. Ramdani. Learning from imperfect data. In H. P. D. Dubois and R. R. Yager, editors, *Fuzzy Information Engineering: a Guided Tour of Applications*, pages 139–148. John Wileys and Sons, 1997.
4. R. Cattell. Odmg-93 - le standard des bases de donnees objet. ITP France, Paris, France, 1995.
5. M. Egenhofer. Spatial sql: A query and presentation language. *IEEE Transactions on Knowledge and Data Engineering*, 6(1):86–95, 1994.
6. M. Ester, H.-P. Kriegel, and J. Sander. Spatial data mining: A database approach. *Proc. 5th Symp. on Spatial Databases, Berlin, Germany*, 1997.
7. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to discovery knowledge in databases. *AI Magazine*, 3(17):37–54, 1996.
8. A. Fotheringham and S. P. Rogerson. *Spatial analysis and GIS : applications in GIS.* London Washington, 1993.
9. J. Han, Y. Fu, W. Wang, K. Koperski, and O. Zaiane. DMQL: A data mining query language for relational databases. In *Proceedings of SIGMOD'96 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96)*, Montreal, Canada, June 1996.
10. J. Han, K. Koperski, and N. Stefanovic. Geominer: A system prototype for spatial data mining. *Proc. 1997 ACM-SIGMOD Int'l Conf. on Management of Data(SIGMOD'97), Tucson, Arizona*, May 1997.
11. ISO. Database language sql. ISO/IEC 9075, 1992.
12. K. Koperski, J. Han, and J. Adhikary. Mining knowledge in geographic data. *Comm. ACM (to appear)*, 1998.
13. R. Laurini and D. Thompson. Fundamentals of spatial information systems. Academic Press., 1992.
14. W. Lu, J. Han, and B. C. Ooi. Discovery of general knowledge in large spatial databases. *Proc. of 1993 Far East Workshop on Geographic Information Systems-(FEGIS'93), Singapore*, pages 275–289, June 1993.
15. C. Marsala. *Apprentissage inductif en présence de données imprécises : construction et utilisation d'arbres de décision flous.* Thèse de doctorat, Université Pierre et Marie Curie, Paris, France, Janvier 1998. Rapport LIP6 n° 1998/014.
16. L. Raynal and G. Schorter. Geo2. Technical report, COGIT - IGN, 1995.
17. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.