A Framework For Integrating Ontologies And Pattern-Bases

Evangelos E. Kotsifakos, Gerasimos Marketos and Yannis Theodoridis

Evangelos E. Kotsifakos
Department of Informatics,
University of Piraeus, Greece
80 Karaoli-Dimitriou St., GR-18534 Piraeus, Greece

Tel: +302104142437

ek@unipi.gr

Gerasimos Marketos
Department of Informatics,
University of Piraeus, Greece
80 Karaoli-Dimitriou St., GR-18534 Piraeus, Greece

Tel: +302104142437

marketos@unipi.gr

Yannis Theodoridis
Department of Informatics,
University of Piraeus, Greece
80 Karaoli-Dimitriou St., GR-18534 Piraeus, Greece

Tel: +302104142449

ytheod@unipi.gr

A Framework For Integrating Ontologies And Pattern-Bases

Abstract

Pattern Base Management Systems (PBMS) have been introduced as an effective way to manage the high volume of patterns available nowadays. PBMS provide pattern management functionality in the same way where a Database Management System provides data management functionality. However, not all the extracted patterns are interesting; some are trivial and insignificant because they do not make sense according to the domain knowledge. Thus, in order to automate the pattern evaluation process, we need to incorporate the domain knowledge in it. We propose the integration of PBMS and Ontologies as a solution to the need of many scientific fields for efficient extraction of useful information from large databases and the exploitation of knowledge. In this chapter, we describe the potentiality of this integration and the issues that should be considered introducing an XML-based PBMS. We use a case study of data mining over scientific (seismological) data to illustrate the proposed PBMS and ontology integrated environment.

Keywords: pattern management, ontologies, knowledge discovery

INTRODUCTION

In the *Knowledge Discovery from Data* (KDD) process, Data Mining techniques are used to find patterns from a large collection of data (see Data Mining step in Fig. 1). The role of the domain experts in this process is crucial. Their knowledge is used in early stages to prepare data (i.e. to decide for the data cleaning and preparation) and to choose the appropriate parameters for the data mining algorithms. Their contribution is also necessary for the evaluation and interpretation of the extracted patterns that lead to the generation of knowledge (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

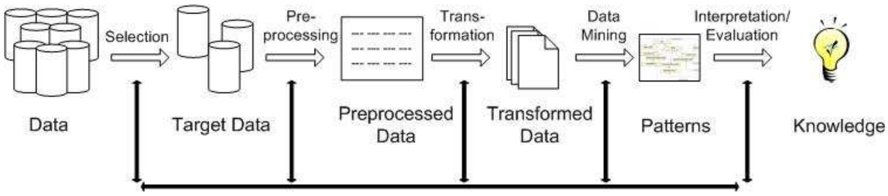

Fig. 1. The KDD process

In essence, extracted patterns are used from domain experts to explore new relations on data, evaluate theories on the field of interest, and discover unknown and hidden knowledge that will lead to new experiments and theories. However, some of the extracted patterns are considered trivial and some others insignificant, according to the domain knowledge. To evaluate extracted patterns experts have defined a lot of different, either objective or subjective interestingness measures based mostly on statistical properties of the patterns. Nevertheless, analyzing and assessing the usefulness of discovered patterns is a laborious task and is considered a hard problem (Piatetsky-Shapiro, 2000).

Two important issues raise here. The first refers to the *manipulation and management of the patterns in a unified way*, either they have been evaluated or not. Currently, the majority of the available data mining tools support the visualization of patterns, and in the best case storage in relational tables. Combined with the characterization of patterns as complex, compact and rich in semantics representation of data (Rizzi et al., 2003), this issue raises the challenge for pattern management. In this context, we propose an XML-based Pattern Base Management System (PBMS) for representing, storing, querying, indexing and updating patterns. This system is data mining engine independent, supporting interoperability and exchange of patterns between different pattern bases.

The second issue is related to the *incorporation of the existing domain knowledge in the Data Mining process*, and especially in the pattern evaluation phase. Several statistical and interestingness measures have been proposed for the evaluation of patterns (Piatetsky-Shapiro, 1991; Freitas, 1999; Silberschatz & Tuzhilin, 1996; Piatetsky-Shapiro, & Matheus, 1994). These measures are applied either before or during the data mining process. In the first case, they are used to reduce the number of patterns that will be extracted and to speed up the data

mining process, while in the evaluation phase, they are used to clean up the patterns considered insignificant.
Nevertheless, no such measure for pattern evaluation is efficient enough as the domain expertise itself. Domain experts can better evaluate the patterns and decide whether they are trivial or not. It is the user who will distinguish interesting rare occurrences of patterns from statistical noise using his/her background knowledge (Pohle, 2003). In order to automate the pattern evaluation process, we need to incorporate the domain knowledge in it. It is generally acceptable that domain knowledge can be represented efficiently using ontologies (Pohle, 2003). An *ontology* is a specification of a conceptualization, a description of the concepts and relationships that can exist for an agent or a community of agents (Gruber, 1993).
We argue that domain knowledge expressed with ontologies could function as a filter in the evaluation phase of the KDD process. Patterns extracted from data mining algorithms would be first evaluated with respect to the ontology. Patterns that contradict to knowledge widely accepted according to the ontology provided (hereafter, called "*noisy*") will be marked as possibly invalid. Whereas, acceptable patterns, will be further evaluated by the domain expert and, if recognized as useful knowledge, the ontology could be updated to incorporate these new patterns (of course, domain experts might reconsider the ontology by adding/removing relations, associations etc). In this case, priority is given to patterns considered interesting, at the same time not conflicting with well established beliefs. This approach could reduce the cost in terms of running time of the data mining algorithm and the effort of the domain expert to evaluate the discovered patterns. Note that "noisy" patterns are marked as invalid and are not being discarded unless user wishes so. Thus the danger to drop really useful knowledge is quite limited.
Towards the purpose of incorporating the domain knowledge in the evaluation phase of the Knowledge Discovery process, we propose the integration of the PBMS with ontologies that describe the field of interest. In the following sections we will analyze each issue separately and we will discuss the various challenges and problems that have to be faced considering a real case study from the seismology domain.
The outline of this chapter is as follows: The related work on pattern management and data mining using ontologies is presented in section 2. Section 3 discusses the need for pattern evaluation using domain knowledge through a case scenario and the ontology for that scenario. The proposed ontology-enhanced PBMS along with a preliminary validation study are presented in sections 4 and 5, respectively. Section 6 summarizes the conclusions and gives hints for future work.

RELATED WORK

In the following paragraphs, we review the background literature in the areas of pattern management and ontology-aware data mining.

Pattern Management

Domain experts are interested in patterns extracted from large datasets. Patterns are of great importance because unlike the original data, they are compact and they represent knowledge hidden in data. Therefore, patterns should be stored, queried, compared and combined with previously extracted patterns in an efficient and unified way. As more and more patterns are available, there is an emerging need for a pattern base management system nowadays. Few efforts have been made to store and manage patterns, including Predictive Model Markup Language (PMML, 2006), Common Warehouse Model (CWM, 2001), SQL/MM-DM (2001), JAVA Data Mining API (JDM, 2003). Most deal with pattern storage, using relational tables or XML documents. PMML is the most popular approach, as it represents patterns (data mining models) in a unified way. A review of these approaches in relation to pattern management can be found in (Catania & Maddalena, 2006). These approaches deal with common data mining patterns and do not provide pattern management functionalities.
Two research projects (funded by the European Community) have dealt with the pattern management problem in a more general way, namely Consortium on Discovering Knowledge with Inductive Queries (CINQ, 2001) and Patterns for Next-generation Database Systems (PANDA, 2001). CINQ is based on the inductive database approach and assumes that patterns and data are stored in the same database, while PANDA assumes that patterns are stored in a different database than raw data, the pattern base. Recently a prototype PBMS that was based on the PANDA pattern model, called PSYCHO, has been presented (Catania, Maddalena & Mazza, 2005). PSYCHO manages different types of patterns in a unified way and it is developed with specific tools over the object-relational Oracle DBMS.

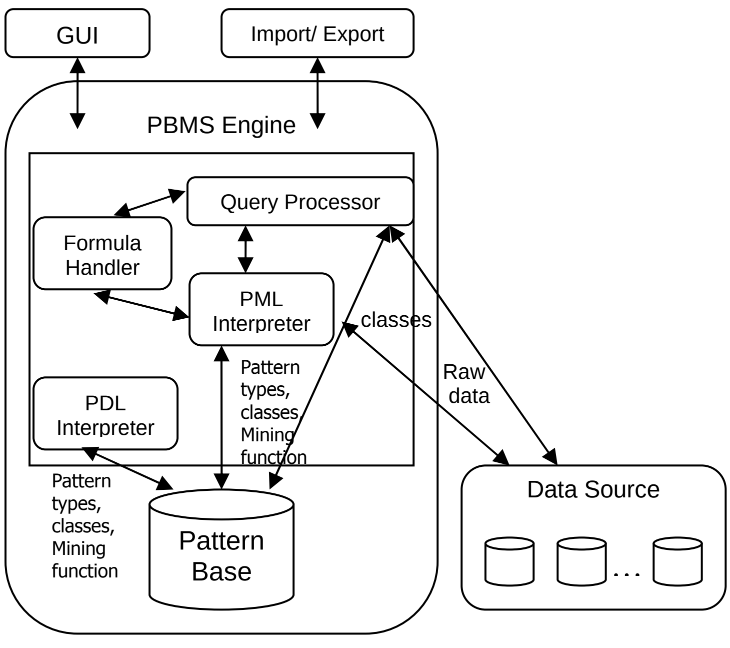

Fig. 2. PSYCHO architecture

PSYCHO architecture is shown in Fig. 2. The system is composed of three distinct layers. The physical layer contains both the *Pattern Base* that stores patterns and the *Data Source* that stores raw data from which patterns have been extracted. The middle layer, called *PBMS Engine*, supports functionalities for pattern manipulation and retrieval (pattern storage and querying). The external layer corresponds to a set of user interfaces (a shell and a GUI) from which the user can send requests to the engine and import/export data in other formats.
Our proposed ontology-enhanced PBMS, which will be presented below, is independent from the data mining engine and uses XML to store patterns in the pattern base. In addition to the pattern management operations, it provides pattern filtering functionalities using ontologies to automate the pattern evaluation step. The efficient management of patterns extracted using data mining techniques as well as user-defined patterns is very important for domain experts in evaluating patterns. Patterns can be stored and retrieved as well as compared to find similarities between them. Further analysis of the pattern base management systems is beyond the main scope of this chapter. More information about PBMSs can be found in (PBMS, 2006).

Data Mining Using Domain Knowledge

Until recently, although the importance of knowledge management was widely known, limited research has been devoted to intelligent pattern analysis and the accumulation of discovered knowledge with prior knowledge (Pohle, 2003).
Regarding the use of domain knowledge in the data mining process only a few related approaches can be found. Domain knowledge can be applied in the data mining process in three different ways. In the preprocessing step (to prepare the data to be mined), during the data mining process (data mining algorithm is using the domain knowledge to decide about the next step), or after the data mining process (to evaluate the extracted patterns).
Considering the first way, X. Chen, Zhou, Scherl, & Geller (2003) propose using an ontology as a concept hierarchy to prepare demographic data for association rule mining. In some tuples of the demographic database there are values from a lower level of the hierarchy while in other tuples, in the same column, there are values from higher levels of the hierarchy (for example the value “basketball” and the value “recreation sports” that are found at different levels in an interests hierarchy). By replacing the values of lower level with values at a higher level (raising), the authors show that the rule support is increasing and thus, more rules can be found.
Several papers can be found about how some interestingness measures (either objective or subjective) are used to evaluate extracted patterns. Objective interestingness measures are based in statistical functions. In (Piatetsky-Shapiro, 2000) basic principles of objective rule interestingness measures are defined, while in (Freitas, 1999) a comparison of objective interestingness criteria can be found. In contrast with objective interestingness measures, subjective measures try to take into account individual conditions of the human analyst. A general discussion can be found in (Silberschatz & Tuzhilin, 1996), while (Piatetsky-

Shapiro, & Matheus, 1994) and (Padmanabhan, & Tuzhilin, 1998) attempt to address this problem. All these approaches provide a way to evaluate patterns but do not make use of the domain knowledge.
There are also few attempts using domain knowledge to improve evaluation of extracted patterns. Domain knowledge in the form of concept hierarchies can be used to improve Web mining results (Pohle & Spiliopoulou, 2002), while an interestingness analysis system that requires the user to express various types of existing knowledge in terms of a proprietary specification language is presented in (Liu, Hsu, S. Chen, & Ma, 2000). These approaches do use domain knowledge, but their disadvantage is that they require the user to previously provide his/her knowledge in a specified and narrow form, according to the application each time.
In order to incorporate domain knowledge in data mining and to allow conceptual model sharing in domains, the use of ontologies is necessary (Maedche, Motik, Stojanovic, Studer, & Volz, 2003). An application of using ontologies before, during and after the data mining process is the one presented by Hotho, Maedche, Staab and Zacharias (2002), in which authors use ontologies and Information Extraction technologies to improve text mining algorithms and pattern interpretation.
Our system uses ontologies to improve the pattern evaluation step and querying the pattern base. During the evaluation step, the system based on the provided ontology and parameters that have been defined from the domain expert, filters the patterns and marks as “noisy” patterns that contradict to domain knowledge. Domain expert can then discard or further evaluate them. The system will also use the filtering mechanism to prevent a naïve user to query the pattern base for “noisy” patterns.

PROBLEM DESCRIPTION

Various examples indicating the need for integration of domain knowledge and data mining can be found, however, dealing with scientific data is more efficient mainly because domain experts in these areas know their data in intimate detail (Fayyad, Haussler, & Stolorz, 1996). In this section, we present a real case study of mining seismological data to illustrate the use of a PBMS and ontologies in an integrated environment for pattern management and evaluation.

A Case Scenario From The Seismological Domain

Let us consider a seismological database containing historical data about earthquake events (Theodoridis, Marketos, & Kalogeras, 2004). Such a database would include information about the event (magnitude, latitude / longitude coordinates, timestamp and depth), the geographical position of both the earthquake epicenter and the affected sites that partitions world in disjoint polygons), as well as details about the fault(s) related with the event. Additionally, our database includes demographical and other information about the administrative partitions of countries, details about the geological morphology of

the areas of various countries and macroseismic information (intensity, etc) (Theodoridis, Marketos, & Kalogeras, 2004).
Seismologists use the database to store the data, a data warehouse to aggregate and analyze them, a knowledge base to store documents collected by various sources, and a tool to define ontologies to represent the domain area. Furthermore, they are interested in discovering hidden knowledge. Patterns produced by the KDD process are evaluated and stored in a PBMS. Obviously, if the above “islands of information” are not integrated under a single tool then the maximum value of the stored information could not be utilized. The researcher is interested in posing a number of questions, perhaps having no idea about which tool to use to get the answers. Some query examples are:

- *Query 1*: Find the average magnitude and the max depth for the earthquakes happened in the North Adriatic Sea (or in a particular geographical area) for the decade 1994-2004.
- *Query 2*: Is there any information about the earthquake maximum recorded intensity when I know that the depth of the epicenter is over 60 km and the geology of the site is characterized as rocky?
- *Query 3*: Find similarities in shock sequences (a main shock that follows pre-shocks and is followed by intensive aftershocks) happened in Greece during 2004.

Query 1 can be easily answered by a data-warehouse using the average and the max function on the appropriate earthquake data. Query 2 can also be easily answered using a decision tree. In case such a decision tree model (pattern) has not been already stored in the PBMS then an appropriate classification algorithm can be applied on the data. Query 3 is more challenging since it requires the incorporation of more advanced domain knowledge: a) the specification of the similarity measure and b) the definition of the shock sequence by the domain expert.
It is clear that Query 3 requires a lot of pre-processing work to be done by the seismologist in collaboration with a database analyst. Hierarchies and rules about seismological concepts and data have to be defined before a data mining algorithm is applied. Furthermore, even when patterns are produced and stored in the PBMS some more post-processing work (similar to the pre-processing step) has to be done in order to extract the appropriate information. The seismologist may have already represented the required knowledge using ontologies, their integration into the PBMS could resolve the above problems.
On the other hand, other queries , such as:

- Query 4: Find any relation between earthquake magnitude and average temperature of the area around the epicenter during a related time period.
- Query 5: Find any relation between earthquake magnitude and season of the year.

can also be posed by a naïve (i.e. non-expert) user and answered applying data mining tasks while semantically unacceptable (for example, seismologists do not

recognize any relation between either earthquake magnitude and surface temperature or earthquake magnitude and season of the year). Although, the data mining engine could return results regarding these relations, a domain expert would definitely discard them.
Nevertheless, such a filtering is nowadays done manually at a post-processing step. Exactly this is the contribution of the integrated ontology-enabled PBMS we propose: to filter out “noisy” patterns efficiently (i.e. online without the need of post-processing) and effectively (i.e. with a quality guaranteed by the ontology-filter).

Domain Knowledge Using Ontologies

One of the challenges in incorporating prior knowledge in the Knowledge Discovery process is the representation of the domain knowledge. Ontologies are useful in providing the formalization of the description of a domain. They are considered as the explicit specification of a conceptualization (Guarino & Giaretta, 1995). Using ontologies, hierarchies of concepts, constraints and axioms can be defined. In other words, ontologies provide a domain vocabulary capturing a shared understanding of terms.
To represent the seismological domain, we choose the Suggested Upper Merged Ontology (IEEE Standard Upper Ontology) (Niles & Pease, 2001), the Mid-Level Ontology (Niles & Terry, 2004) and, finally, an ontology for representing geographical information all available at (SUMO). An upper ontology is limited to concepts that are meta, generic, abstract or philosophical, and hence are general enough to address (at a high level) a broad range of domain areas. Concepts specific to particular domains are not included in an upper ontology, but such an ontology does provide a structure upon which ontologies for specific domains (e.g. medicine, finance, engineering, etc.) can be constructed. A mid-level ontology is intended to act as a bridge between the high-level abstractions of the SUMO and the low-level detail of the domain ontologies which in our case is the geography ontology. The following schema is based on the above ontologies (Fig. 3).

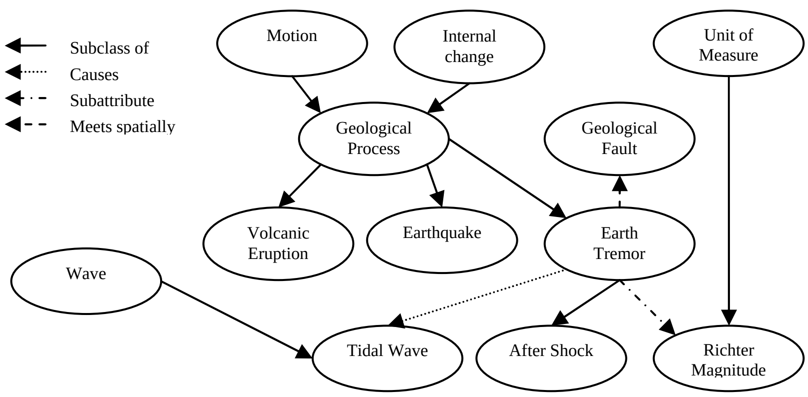

Fig. 3. A subset of the SUMO for seismology

Obviously, the above figure does not represent the "Universe of Discourse", but is a part of the geography ontology related to seismology. It is clear that using ontologies, horizontal relationships between concepts can be defined (Pohle, 2003). For instance, in the domain of seismology there is such a relationship between seismology and geology (faults). This is important as the patterns that are stored for each domain in the PBMS, can be combined offering more complete querying and visualization capabilities to the user.

INTEGRATION OF ONTOLOGIES IN THE KDD EVALUATION PHASE

The system we propose provides both naïve users and domain experts functionalities for efficient pattern management and pattern evaluation using an ontology discarding the non-useful patterns and thus improving the performance of the data mining tasks and the query answering over the pattern base. The system is able to evaluate patterns before, during and after the data mining process, as well as every time a user poses a query to the pattern base. The system architecture is depicted in Fig. 4.

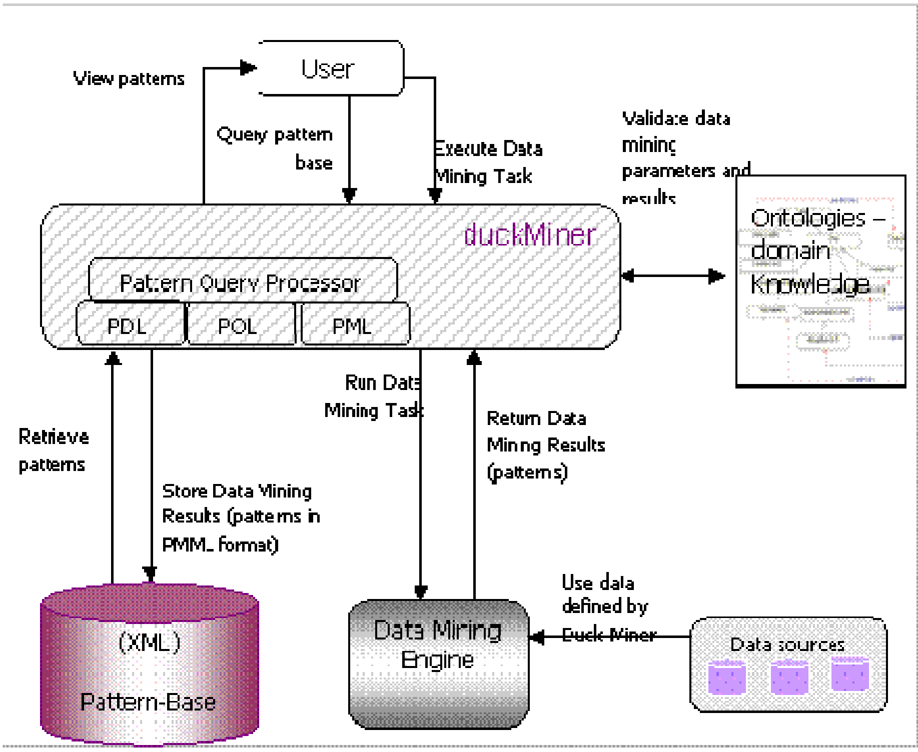

Fig. 4. The proposed ontology-enhanced PBMS architecture

Independent from data mining engine, the PBMS stores the extracted patterns in an XML pattern base. We choose XML for pattern storage as it performs better than Relational and Object-Relational models (Kotsifakos, Ntoutsi, & Theodoridis, 2005). The pattern model used is the theoretical model defined in PANDA project (Rizzi, et al., 2003) enhanced to support pattern temporal

validation and semantically related pattern classes. Our extended model defines four logical concepts. *Pattern type*, *pattern*, *class* and *superclass.*
More specifically, each *pattern type* contains metadata information about:

- the data mining algorithm applied to extract the patterns it represents and its parameters,
- the date and time of the data mining process,
- the validity period,
- the data source,
- the mapping function, and finally,
- information about the structure and the measures of the patterns it represents.

Patterns are instances of pattern types. In our XML architecture, pattern types are the XML Schema for a pattern (XML document). The pattern document contains metadata about the data mining process as well as the patterns extracted by that process. For example, an association rule pattern instance and its pattern type are shown in Fig. 5 and Fig. 6, respectively.

```
<pt_assocRule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="assocRule" pt_descr="association
rules" pt_id="1" xsi:noNamespaceSchemaLocation="pt_assocRule.xsd">
        <pt_metadata>
                <algorithm>apriori</algorithm>
        <parameters>min_support=0.1,min_conf=0.4,rules=10</parameters>
                <source>select * from earthquakes</source>
                <date>2006/04/12 13:03:34</date>
                <validity>2006/06/12 13:03:34</validity>
                <mapping_function>{{‘depth’, ‘magnitude’, ‘season’} ⊆ transaction}
 </mapping_function>        </pt_metadata>
        <patterns>
                <pattern p_id="1">
                        <structure>
                                <body>
                                        <attrib>depth</attrib>
                                        <attrib_value>0-1</attrib_value>
                                </body>
                                <head>
                                        <attrib>magnitude</attrib>
                                        <attrib_value>(3,4]</attrib_value>
                                </head>
                        </structure>
                        <measures>
                                <measure_name>support</measure_name>
                                <measure_value>0.18</measure_value>
                                <measure_name>confidence</measure_name>
                                <measure_value>0.67</measure_value>
                        </measures>
                </pattern>
                <pattern p_id="2">
                        <structure>
                                <body>
                                        <attrib>season</attrib>
                                        <attrib_value>Autumn</attrib_value>
                                </body>
                                <head>
                                        <attrib>magnitude</attrib>
                                        <attrib_value>(3-4]</attrib_value>
                                </head>
                        </structure>
                        <measures>
                                <measure_name>support</measure_name>
                                <measure_value>0.18</measure_value>
                                <measure_name>confidence</measure_name>
                                <measure_value>0.58</measure_value>
                        </measures>
                </pattern> </patterns> </pt_assocRule>
```


Fig. 5. Association rule patterns, XML example

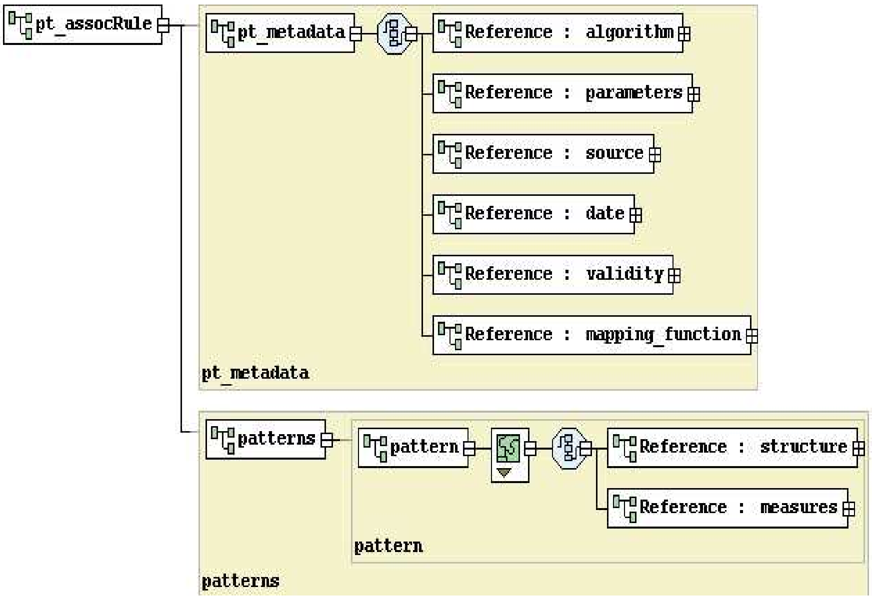

Fig. 6. Pattern Type Association Rule XSD diagram

Apart from the *pattern type* and *pattern* concepts, *class* is defined as a set of semantically related patterns of the same pattern type. A class is defined by the user to group patterns that have a common meaning and belong to a specific pattern type. Each pattern may belong to more than one different classes. For example a user could define a class containing association rules related to seismic activity in the summer of 2003. This class would contain a lot of patterns that may belong to different association rule mining result sets but it will have the same meaning for the user. Fig. 7 illustrates the pattern base logical model.

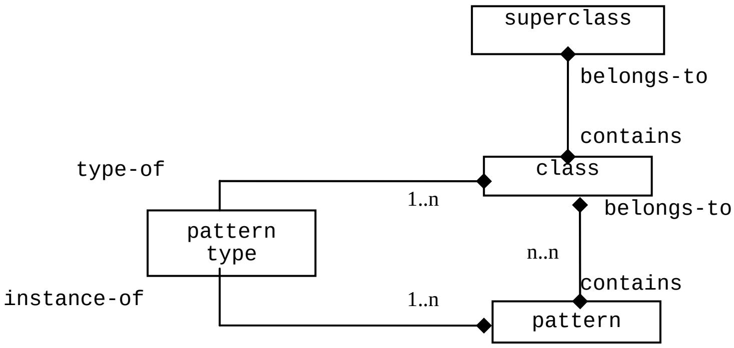

Fig. 7. Pattern Base logical model

Furthermore, the concept of *superclass* is defined, which is a set of classes, probably of different pattern types. Thus, patterns belonging to different pattern types can be grouped together. For instance, a user might want to group all association rules related to seismic activity in the summer of 2003 and the clusters of faults that gave earthquakes of magnitude M>3 during the same time period. The link between the two types would be the magnitude of earthquakes. In other words, we are interested in studying the relation between earthquakes and geological faults, thus the grouping of classes of different pattern types is necessary.

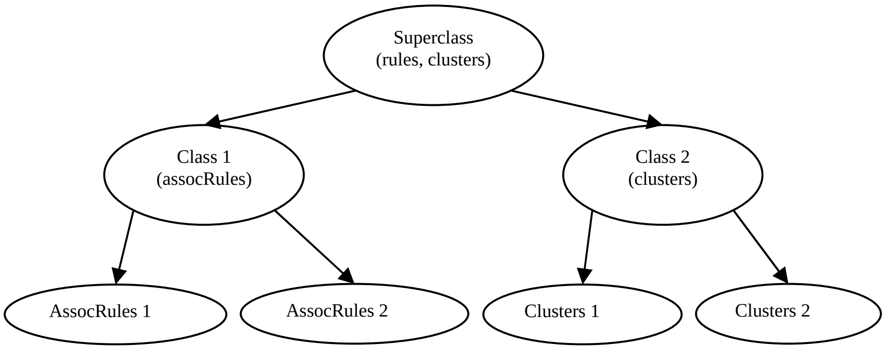

Fig. 8. Class and Superclass relation

Ontologies are stored in external files and are written in OWL (Horrocks & Patel-Schneider, 2003). Regarding association rule mining, a general rule that can be used to evaluate patterns with ontologies, is that patterns should associate attributes that belong to the same class or to subclasses of the same class. Reasonably, association of attributes belonging in different classes (in the ontology-hierarchy graph) or in classes that are several nodes away in the ontology diagram might result in false associations of irrelevant (according to domain knowledge) attributes. Edge-distance and other approaches have been already proposed for searching semantic similarity between objects in an ontology. Such measures can be used to assess the relation between two attributes. This implies that the user can select the level of relevance between the attributes, defining the maximum distance that a class can have from another in the ontology graph.

The task of defining the rules that will be used to filter the patterns to be extracted involves the study of the ontology as well as the study of the pattern type and the results that users anticipate. Ontology components are classes, attributes and relations between them. Classes have subclasses and each class may have a number of attributes. Usually classes for related concepts, belong to the same parent class while not related concepts are under different classes. The whole class and subclass diagram, define a kind of hierarchy with various levels of detail. For example, classes “VolcanicEruption” and “Earhtremor” (Fig. 3) lie at the same level, while their subclasses “volcanicGasRelease” and “AfterShock” lie at lower level.

As each pattern has different structure, filters for every pattern type have to be defined. Specifically for the Association Rule pattern type, we define the Association Rule Filter. Each part of the rule contains attributes (depth, magnitude etc) that are related in the relational model, but also related in some way in the ontology. Thus, we can define for each rule a distance metric between the main earthquake class (*earth tremor*) and the nodes of the attributes contained in the rule. The shorter this distance is, the more the attributes are semantically related.

In fact, we can define two approaches to measure this distance: in the so-called "*Risky*" approach, we consider the maximum distance between the nodes of the attributes and the main earthquake class, whereas in the "*Not Risky*", we consider the minimum distance between them. Obviously, the attributes of the earthquake class have distance=0 and thus there are not included in this calculation.

A user selects the level of semantic relevance by specifying the maximum distance of the nodes from the main earthquake class. For instance, one may be interested in finding relationships not just between the attributes on an earthquake but also between them and geological faults. Thus, the level of semantic relevance has to be increased so as to include the appropriate node.

With the above described process, a subgraph of the ontology that contains the attributes under consideration is constructed. Attributes of the produced rules are validated against this ontology subgraph. If all are included in the subgraph then the association rule that contains them is considered as semantically valid. Otherwise, if some of the attributes are not in the subgraph, the rules containing them are marked as "noisy". Note that the system does not reject "noisy" rules (although there is such an option) as they might contain previously unknown knowledge about the relations of some attributes, and thus domain expert's attention is required. Some rules can lead to new interesting relations and domain experts might reconsider the ontology.

Since the ontology represents the domain of interest, it has to be well designed. In this way, pattern evaluation can be more accurate and may give useful results to domain experts.

PRELIMINARY VALIDATION STUDY

In this section, we use the example from seismology domain and the ontology defined in section *Domain knowledge with ontologies* to describe system functionality. The system performs a validation test before the data mining process, checking if the user defined parameters make sense. For example, a user could ask the system to perform the Apriori algorithm to find associations between the "magnitude" and the "date" of an earthquake. As mentioned in section 3.2 this association is not acceptable by the seismology domain and thus the system will suggest the user to change the parameters. If the user does not specify the attributes that he/she wants to search for associations, the system will perform the data mining algorithm using all attributes but, when generating the frequent itemsets, it will discard itemsets that contain values from attributes not related in the ontology. In this way, the time consuming phase of frequent itemset generation will be improved and no irrelative association rules will be generated. Of course, this is not always desirable as some interesting rules might not be generated. In this case the user should decide for these rules. So, it is given as option to the user either to enable the system to automatically discard them or just to mark the "noisy"ones for further evaluation. In the latter case, the user decides which rules are interesting and should be stored to the pattern base.

Another case is when a user is posing a query to the pattern base to retrieve patterns for example "fetch association rule patterns that contain both "season"

and “depth” attributes and the support of the rule is greater than 0.3”. Such rules are not valid according to the domain knowledge and thus the system notifies the user that it is rather impossible to find rules like those in the pattern base.
In our first experiments, we ran the Apriori data mining algorithm implemented in WEKA (Witten & Frank, 2005) to extract some association rules using real macroseismic data collected by the Greek Institute of Geodynamics (Seismo-Surfer). Attributes such as earthquake depth, intensity, site, date and season of the year are some of the attributes of the table that contains 10336 tuples for the earthquake events during the 20th century. Table 1 lists 25 out of 70 rules extracted by Apriori confidence threshold = 30% and support threshold = 10% are listed in Table 1.

Table 1: Association rules extracted from seismological data

id	Association Rule	Conf.	Supp.
1	intensity≥5 → distance≤80	74%	19%
2	weekDay=Tuesday, 11≤depth≤20 → season=Summer	71%	10%
3	weekDay=Tuesday → season=Summer	71%	17%
4	weekDay=Monday → season=Spring	68%	10%
5	season=Summer → 11≤depth≤20	65%	21%
6	weekDay=Saturday → 21≤depth≤50	62%	12%
7	depth≥50 → season=Spring	60%	11%
8	distance≥150 → intensity≤3	59%	15%
9	weekDay=Tuesday, season=Summer → 11≤depth≤20	57%	10%
10	weekDay=Tuesday → 11≤depth≤20	57%	14%
11	11≤depth≤20 → season=Summer	57%	21%
12	season=Autumn → 11≤depth≤20	55%	14%
13	season=Summer → weekDay=Tuesday	54%	17%
14	intensity≤3 → distance≥150	54%	15%
15	distance≤80 → intensity≥5	52%	19%
16	distance≥150 → 1000<population≤4000	48%	13%
17	3<intensity≤4 → 80<distance<150	48%	15%
18	season=Summer, 11≤depth≤20 → weekDay=Tuesday	48%	10%
19	weekDay=Tuesday → 1000<population≤4000	46%	11%
20	season=Spring → 21≤depth≤50	46%	14%
21	intensity≤3 → 1000<population≤4000	46%	13%
22	21≤depth≤50 → season=Spring	45%	14%
23	500<population≤1000 → distance≤80	43%	11%
24	season=Spring → 1000<population≤4000	43%	13%
25	80<distance<150 → 1000<population≤4000	43%	15%

Out of these 25 rules, the domain expert marked only five rules (ids 1, 8, 14, 15, 17) as interesting and all others as “noisy” because they describe a correlation between attributes/classes that is meaningless in the domain of seismology. The system needs a threshold parameter to be defined in order to mark some rules as “noisy”. This threshold is the maximum path distance from the main “earth tremor” node/class. When this threshold is defined, the system retrieves the subgraph of the ontology defined by the “earth trermor” node and all the nodes with path distance less or equal to the threshold. Every rule that has attributes belonging to that subgraph, will be considered interesting while all others will be marked as “noisy”.

Trying to detect a reasonable threshold in order for the system to retrieve the rules that will match the expert's evaluation, we varied threshold value from 1 to 5 and computed the rules marked as "noisy" by the system. This is illustrated in Fig. 9.

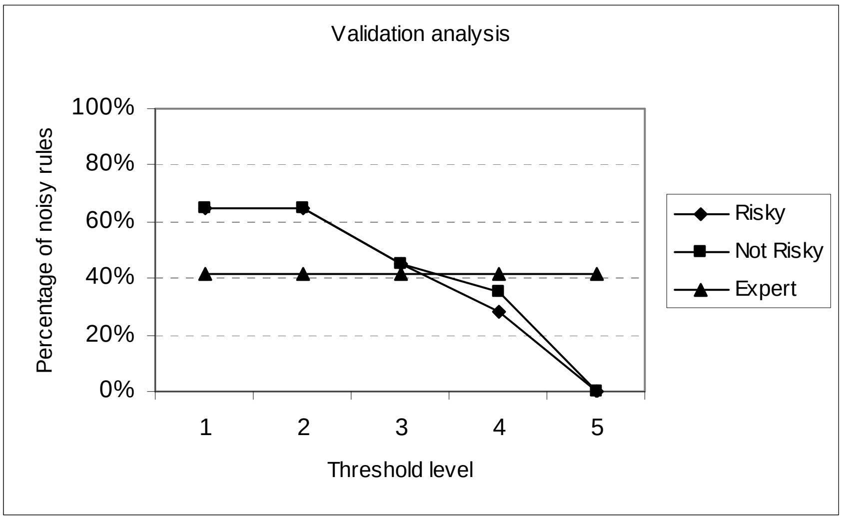

Fig. 9 Threshold and rules rejected by the system and the seismologist

According to this experiment we conclude that with threshold 3 the system matches expert choices. As such, this threshold can be used by the user for the next running of Apriori or can even be stored as meta-data for the specific dataset and KDD process for future data mining.

With the procedure described above we can measure the percentage of the rules that will be marked as "noisy" by the system and by the expert, but we do not know if these are the same rules i.e. if the rules marked by the system are the same with the rules marked by the domain expert (precision). While in our particular experiment we had a perfect match, it is not sure that we will have a perfect match each time.

6 Conclusions and future work

In this work, we proposed a framework consisting of a PBMS that uses ontologies to improve data mining tasks, to evaluate extracted patterns and to improve querying over pattern base. The PBMS interacts with the data mining engine to discover patterns and stores them in XML format according to the presented pattern model in the pattern base. Users can pose queries over the pattern base and the data sources. Ontology is used to evaluate the patterns extracted from the data mining process, and to validate the queries the user is posing.

The main idea is that the system is independent from the data mining engine and the ontology. So, according to the application domain, the pattern types and the ontology have to be defined. The PBMS has also the proper design for defining complex patterns and for comparing patterns using similarity measures based on the structure and the measure component of the patterns.

The proposed framework provides domain experts with a powerful tool that can help them to better manage and evaluate the patterns extracted from data mining algorithms. We aim in improving and enhancing the KDD process. Domain knowledge and knowledge representation techniques can help both in reducing the required time to run a data mining algorithm and in evaluating the extracted patterns. It is clear though, that due to the complexity of ontologies and the lack of standards on ontology creation, this incorporation is not an easy task. We have listed the theoretical and technical problems that should be faced.

Both PBMS and ontologies are areas of recent research and their applications could be many. Apart from geosciences, every field that has a well defined ontology can use the integrated framework to improve the KDD process. For example in the domain of B2B marketplaces, finding associations between products is more efficient when using the hierarchies defined in the product ontology. Although there is not currently universally accepted product ontology, efforts are made to integrate different product ontologies (Omelayenko, 2000) towards this end.

In order to be able to use ontologies in KDD process and to have the results available to domain experts, ontologies have to be defined in a common way. There are a lot of efforts for ontology matching (Doan, Madhavan, Domingos, & Halevy, 2003) and ontology integration (Cui, Jones, & O'Brien, 2002), (Pinto & Martins, 2001) and this illustrates the need for an ontology creation standard. In this way, exchange and comparison of ontologies describing different domains could be possible. Until now, only several domain specific ontologies and tools have been developed.

Integrating ontologies to the data mining process is not an easy task and a lot of issues have to be addressed. Things are complicated due to the fact that scientists and companies create ontologies according to their needs instead of adopting a universal ontology. There is a large number of ontology languages most of them designed for the semantic web like RDF (Beckett, 2004), SHOE (Luke & Heflin, 2000), DAML, DAML+OIL (Harmelen, Patel-Schneider, & Horrocks, 2001), OWL (McGuinness, & Harmelen, 2005). New ontologies are constructed for various fields and applications without centralized guidance and common agreement. This is getting even more complex as recent studies have indicated semantic and syntactic conflicts between these languages, especially between DAML+OIL and OWL (Horrocks & Patel-Schneider, 2003) (Patel-Schneiderand & Fensel, 2002). Therefore, building a system that uses ontologies in the data mining process requires choosing a specific ontology language to support.

Another important theoretical issue concerns the evaluation of various pattern types using ontologies. It is very hard to define general rules that apply to all pattern types. The most popular pattern types from data mining field are association rules, clusters, decision trees, neural networks and time series. We have defined filters for association rule mining but depending on the application filters for each pattern type separately have to be defined in order to build a system to support the majority of pattern types. Furthermore we investigate the precision issue regarding the patterns that system marks as "noisy".

Our framework is currently under development. Extended experiments and expert evaluation of the results on real case studies with association rule and decision trees mining are to be conducted. Early results have been positively evaluated by seismologists. Future work includes defining filters for decision trees and clusters patterns for seismological data as well as applying the framework to other domains.

ACKNOWLEDGEMENTS

Research supported by the General Secretariat for Research and Technology of the Greek Ministry of Development under a PENED'2003 grant.

REFERENCES

Bartolini, I., Ciaccia, P., Ntoutsi, I., Patella, M., & Theodoridis, Y. (2004). A Unified and Flexible Framework for Comparing Simple and Complex Patterns. *Proceedings of PKDD'04*, Pizza, Italy.

Beckett, D. (2004). RDF/XML Syntax Specification (Revised), W3C Recommendation, 2004. http://www.w3.org/TR/rdf-syntax-grammar/.

Catania B., & Maddalena A. (2006) Pattern Management: Practice and Challenges. In J. Darmont & O. Boussaid, (Eds). *Processing and Managing Complex Data for Decision Support*. Idea Group Publishing.

Catania B., Maddalena A., & Mazza M. (2005). PSYCHO: A Prototype System for Pattern Management. *In Proceedings of the International Conference on Very Large Data Bases 2005*.

Chen X., Zhou X., Scherl R., & Geller J. (2003). Using an interest ontology for improved support in rule mining. *In DaWaK 2003.* pp. 320-329.

CINQ (Consortium on Discovering Knowledge with Inductive Queries). (2001). http://www.cinq-project.org.

Cui Z., Jones D., & O'Brien P. (2002). Semantic B2B Integration: Issues in Ontology-based Approaches. *ACM SIGMOD Record archive* Vol. 31, Issue 1 (March 2002) *SPECIAL ISSUE: Data management issues in electronic commerce table of contents.* pp. 43–48

CWM (Common Warehouse Model) (2001) homepage. http://www.omg.org/cwm.

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2003). Ontology Matching: A Machine Learning Approach. In S. Staab and R. Studer (Eds), *Handbook on Ontologies in Information Systems*. Springer-Velag.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery, an overview. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, (Eds.), *Advances in Knowledge Discovery and Data Mining.* (pp. 1–30). Menlo Park, Calif. AAAI/MIT Press.

Fayyad, U., Haussler, D., & Stolorz, P. (1996). Mining scientific data. *Communications of the ACM, Vol. 39,* Issue 11, 51-57.

Freitas, A.A. (1999). On rule interestingness measures. *Knowledge-Based Systems, Vol. 12.* number 5-6. pp. 309–315, October 1999. Elsevier

Gruber, T. R (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220.

Guarino, N., & Giaretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarification. In N. Mars (Ed.), *Towards very large knowledge bases: Knowledge building and knowledge sharing* (pp.25-32). Amsterdam, IOS Press.

Harmelen, F.V., Patel-Schneider, P.F., & Horrocks, I. (2001). Reference Description of the DAML+ OIL Ontology Markup Langauge. http://www.daml.org/2001/03/daml+oil-index.html.

Horrocks, I., & Patel-Schneider, P.F. (2003). Three theses of representation in the semantic web. *In Proceedings of the Twelfth International Conference on World Wide Web*.

Hotho, A., Maedche, A., Staab, S., & Zacharias, V. (2002). On knowledgeable unsupervised text mining. *In Proceedings of the DaimlerChrysler Workshop on Text Mining*, Ulm, April 26–27 2002. Springer.

ISO SQL/MM Part 6 (2001). http://www.sql-99.org/SC32/WG4/Progression_Documents/ FCD/fcd-datamining-2001-05.pdf.

Java Data Mining API homepage (2003), http://www.jcp.org/jsr/detail/73.prt.

Kotsifakos, E., Ntoutsi, I., & Theodoridis, Y. (2005). Database Support for Data Mining Patterns. *In Proceedings of PCI'05*. pp. 14-24. Springer Verlag.

Liu, B., Hsu W., Chen S., & Ma, Y. (2000). Analyzing the subjective interestingness of association rules. IEEE Intelligent Systems, 15(5):47-55.

Luke, S., & Heflin, J. (2000). SHOE 1.01 Proposed Specification, *SHOE Project*, http://www.cs.umd.edu/projects/plus/SHOE/spec.html.

Maedche, A., Motik, B., Stojanovic, L., Studer, R., & Volz, R. (2003). Ontologies for enterprise knowledge management. *IEEE Intelligent Systems*, 18(2):26–33, March/April 2003.

McGuinness, D.L., & Harmelen, F.V. (2005). OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/ (current Feb. 2005).

Niles, I., & Pease, A. (2001). Toward a Standard Upper Ontology. *In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001).*

Niles, Ian & Terry, Allan. (2004). The MILO: A general-purpose, mid-level ontology. In 2004 International Conference on Information and Knowledge Engineering (IKE'04).

Omelayenko B. (2000). Integration of Product Ontologies for B2B Marketplaces: A Preview. *In ACM SIGecom Exchanges.* Vol. 2, issue 1, pp. 19-25.

Padmanabhan B. & Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. *In Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 94–100, August 1998.

PANDA (Patterns for Next-generation Database Systems) (2001) project homepage. http://dke.cti.gr/panda.

Patel-Schneiderand, P.F., & Fensel, D. (2002). Layering the semantic web: Problems and Directions. *In Proceedings of the 1st International Semantic Web Conference*. LNCS 2342, Springer.

PBMS (2006) homepage. http://www.pbms.org.

Piatetsky-Shapiro G. & Matheus C.J. (1994). The interestingness of deviations. *In Proceedings of KDD-94: AAAI-94 Knowledge Discovery in Databases Workshop*, pages 25–36. AAAI Press, July 1994.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro & W.J. Frawley (Eds). *Knowledge Discovery in Databases*, pp. 229–248. AAAI/MIT Press, Cambridge, MA.

Piatetsky-Shapiro, G. (2000). Knowledge discovery in databases: 10 years after. *SIGKDD Explorations, Vol. 1, no 2*. pp. 59–61, January 2000.

Pinto, H.S., & Martins, J.P. (2001) A methodology for ontology integration. *1st International conference on Knowledge Captur*. Pp 131-138.

PMML (Predictive Model Markup Language) (2006). http://www.dmg.org/pmml-v3-1.html.

Pohle, C. & Spiliopoulou, M. (2002). Building and exploiting ad hoc concept hierarchies for web log analysis. In Y. Kambayashi, W. Winiwarter, & M. Arikawa (Eds). *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, DaWaK 2002*, volume 2454 of Lecture Notes in Computer Science, pages 83–93, Aix en Provence, France, September 4–6 2002. Springer-Verlag.

Pohle, C. (2003) Integrating and updating domain knowledge with data mining. *VLDB PhD Workshop*.

Rizzi, S., Bertino, E., Catania, B., Golfarelli, M., Halkidi, M., Terrovitis, M., Vassiliadis, P., Vazirgiannis, M., & Vrahnos, E. (2003). Towards a logical model for patterns. *Proceedings of ER'03 conference*.

Seismo-Surfer, A WebGIS application for integrating, visualizing and analyzing seismic data. http://www.seismo.gr.

Silberschatz A. & Tuzhilin, A. (1996). What makes patterns interesting. In knowledge discovery systems. IEEE Transactions on Knowledge and Data Engineering, 8(6):970–974, December 1996.

SUMO, Suggested Upper Merged Ontology. http://ontology.teknowledge.com.

Theodoridis, Y., Marketos, G., & Kalogeras, I.S. (2004). Collecting and Mining Seismic Data in Greek Territory - The Seismo-Surfer Tool. *Proc. 7th Panhellenic Geographical Conference of the Hellenic Geographical Association (HGA'04),* Mytilene, Lesvos, Greece.

Witten I.H., & Frank E. (2005). *Data Mining: Practical machine learning tools and techniques*. 2nd Edition. Morgan Kaufmann, San Francisco.