Outlier detection in relational data: a case study in geographical information systems

Joris Maervoet[a,c,*], Celine Vens[b], Greet Vanden Berghe[a], Hendrik Blockeel[b], Patrick De Causmaecker[c]

[a] *CODeS, KaHo Sint-Lieven, Departement Industrieel Ingenieur, Vakgroep IT, Gebr. Desmetstraat 1, 9000 Gent, Belgium*
[b] *Katholieke Universiteit Leuven - Department of Computer Science, Celestijnenlaan 200 A, 3001 Leuven, Belgium*
[c] *CODeS, Katholieke Universiteit Leuven Campus Kortrijk, Faculty of Sciences, Department of Informatics, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium*

Abstract

Geographical information systems are commonly used for a variety of purposes. Many of them make use of a large database of geographical data, the correctness of which strongly influences the reliability of the system. In this paper, we present an approach to quality maintenance that is based on automatic discovery of non-perfect regularities in the data. The underlying idea is that exceptions to these regularities ('outliers') are considered probable errors in the data, to be investigated by a human expert. A case study shows how the tool can be used for extracting valuable knowledge about outliers in real-world geographical data, in an adaptive manner to the evolving data model supporting it. While the tool aims specifically at geographical information systems, the underlying approach is more broadly applicable for quality maintenance in data-rich intelligent systems.

Keywords: relational outlier detection, geographical information systems, quality maintenance, WARMR

*Corresponding author
Email addresses: joris.maervoet@kahosl.be (Joris Maervoet), celine.vens@cs.kuleuven.be (Celine Vens), greet.vandenberghe@kahosl.be (Greet Vanden Berghe), hendrik.blockeel@cs.kuleuven.be (Hendrik Blockeel), patrick.decausmaecker@kuleuven-kortrijk.be (Patrick De Causmaecker)

1. Introduction

Outliers in a data set are commonly defined as *individuals* that are substantially different from the rest of the data. Such irregularities can indicate an error in the data, or abnormal behaviour of the underlying system. In research areas such as machine learning and statistics, a great diversity of algorithms for outlier detection have been proposed in the last years (Breunig et al., 2000; Knorr et al., 2000; Aggarwal & Yu, 2001; Caruso & Malerba, 2007). Most of them refer to a statistical deviation of the outlier values from the rest of the data set. However, a lot of contemporary applications of outlier detection such as fraud and network intrusion detection, have a relational character. The data consist of several interrelated data types, implying that the concept of outlier detection can be seen in a broader perspective. Besides the detection of deviating values for a specific variable, it is also possible to look for deviating structures in the relational data.

In this paper, a case study of relational outlier detection on geographical data is presented. It concerns learning anomalies in the core database of the geographic content provider Tele Atlas[1]. This company possesses a large amount of geographical road data, collected from different sources. Irregularities, e.g. a wrong speed restriction, creep in due to human mistakes or inconsistencies between different sources. Therefore, a quality maintenance system has been set up by the company enabling data engineers to manually formulate rules to which the data should conform and providing infrastructure to trace violations against these rules in a brute-force manner. An example of such a rule is "A road segment adjacent to a primary school has always a speed restriction of 30". More information about the problem context is described by Maervoet et al. (2008). In the present paper, we apply a relational frequent pattern miner to discover such rules automatically from the data. Exceptions to these rules will be considered probable erroneous data, to be presented to a human expert for evaluation.

The paper is structured as follows. Section 2 presents some related work in the domains of relational outlier detection and spatial data mining. Section 3 thoroughly describes the geographic data quality problem and the relational

[1]Since 2007, the company is a wholly-owned subsidiary of automotive navigation system manufacturer TomTom.

outlier detection approach to it. The actual case study is presented in more detail in Section 4. Section 5 reports some regularity rules and corresponding outliers found by the system. Finally, we indicate some directions for future work in Section 6 and conclude in Section 7.

2. Related work

2.1. Outlier detection

Given an input data set, an **outlier** is an instance or a set of instances[2] that show(s) exceptional behaviour compared to the rest of the input data set or to a local context within the input data set. **Outlier detection** is the non-trivial process of extracting a set of previously unknown anomalies from data. It is a form of data mining.

A first dimension categorises outlier detection approaches according to the type of learning. Lazarevic et al. (2008) distinguish between:

- **Supervised outlier detection.** Both the outliers and regularities from the input data set are labelled. In this case, the problem can be reduced to classification.
- **Semi-supervised outlier detection.** Some examples of anomalies and/or regularities from the input data set are given. It can be applied in interactive learning systems. Zhu et al. (2004) e.g. predict all the outliers from the input data set based on an outlier sample set indicated by the end user.
- **Unsupervised outlier detection.** This type of outlier detection assumes unlabeled input data. It involves fitting one or more models over the input data and identifying the model deviations as outliers.

Table 1 shows a classification of unsupervised outlier detection systems. These systems can be classified into statistical and relation outlier detection and into detection by direct and indirect description.

Statistical outlier detection. This type of system looks for outliers with a statistical deviation from the rest of the data, assuming one global model

[2]In the latter case, individual set members are not anomalous.

	Indirect description	**Direct description**
Statistical outliers	Clustering (distance or density based) Probability distribution (histogram, Gaussian)	Nearest neighbour (distance or density based)
Relational outliers	Frequent pattern discovery	Anomaly pattern discovery

Table 1: Unsupervised outlier detection.

that distinguishes the outliers from the regular data. A common technique applies clustering: the data is clustered, and the elements that do not belong to any clusters are outliers. Other methods use density (Breunig et al., 2000) and/or proximity analysis (Knorr et al., 2000; Ramaswamy et al., 2000). Aggarwal & Yu (2001) introduce evolutionary algorithms for identifying outliers in data with a high number of dimensions. Frank et al. (2007) mine for spatial regional outliers. These are neighbourhoods of anomalous objects that maximise the non-spatial attribute value deviation between the object and its neighbouring objects.

Relational outlier detection. Many contemporary outlier applications are relational. In the context of network security, for example, there is a high interest in so-called anomaly detection. This is the detection of deviant behaviour in network traffic that possibly indicates an attack. Caruso & Malerba (2007) propose an adaptive model for network traffic. If a new network connection deviates substantially from the model, the system examines whether it concerns an outlier, or a legal connection, whereupon the model is adapted.

In relational outlier detection, the input data is composed of several interrelated data types and so the outliers have a relational character too. Often, it assumes multiple models that explain why an outlier differs from the rest of the data. In (Angiulli et al., 2007), a theory of normal behaviour is modelled using a formal knowledge representation language (first-order logic). Both outliers and so-called witness sets are searched for. Outliers are entities that are inconsistent with the given background knowledge. Corresponding witness sets describe the causes behind the outliers in the data.

Relational outlier detection is a form of relational data mining, a research area that has gained a lot of interest during the last years. A large amount

of the research is carried out in the context of inductive logic programming (ILP) (Lavrač & Džeroski, 1994). The data, as well as the discovered patterns and the background knowledge, are represented as logic programs. The major part of research on ILP (and on relational data mining in general) has been carried out on supervised learning. Less research has been performed on unsupervised learning, in which a set of hypotheses that describe the whole set of facts as accurately as possible are learnt. Relational clustering (Ramon, 2002), finding frequent patterns in first-order logic (Dehaspe & Toivonen, 1999) and clausal discovery (De Raedt & Dehaspe, 1997) belong to the latter category.

Discovery by direct description. This means that patterns describing exceptional situations are looked up directly. For instance, Laros (2005) looks for a substring, as short as possible, that appears exactly once in a set of strings. With regard to relational outliers, several definitions and corresponding algorithms for anomalous pattern discovery can be found: sporadic rules (Koh & Rountree, 2005; Koh et al., 2008) (rules with low support but high confidence), minimal infrequent itemsets (Haglin & Manning, 2007) and unexpected rules (Plantevit et al., 2007) (with a support between two thresholds). Exception rules (Suzuki, 2002) refer to the extension of the premise of a 'common sense rule', refuting the consequence of that rule. Anomalous association rules (Berzal et al., 2004) refer to association rules for which anomalous itemsets exist that always contradict the rule.

Discovery by indirect description. Instead of looking for patterns that describe the exceptions directly, the complementary problem can be examined as well. It involves looking for regularities, followed by the identification of data that does not comply with those regularities. K-Means clustering of network traffic data followed by the identification of traffic anomalies (Münz et al., 2007) is an example of this category. Discovery by indirect description allowed us to apply state of the art techniques from the domain of frequent pattern mining.

2.2. Spatial rule mining

Spatial rule mining is a common machine learning approach to spatial data mining (SDM), which aims at extracting useful or interesting patterns from spatial databases. Shekhar et al. (2003) indicate that the data input, statistical foundation, output patterns and computational process are dif-

ferent for SDM. Zeitouni (2002) identified several generic SDM tasks, and associated existing methods with these tasks. With regard to rule learning, we can distinguish 4 types of approaches:

- **Characteristic rules.** This type of rules describes characteristic object and neighbourhood properties of a set of spatial objects in the database (Ester et al., 1998). It is a form of summarisation.

- **Classification rules.** This form of supervised classification involves the discovery of a set of rules (often represented as decision trees) comparing the object and neighbourhood properties of a set of spatial objects of choice, called the target class, to one or more contrasting classes. For instance, Ceci & Appice (2006) learn geographic impact factors for several rent prize categories from census data. Frank et al. (2009) propose a Voronoi-based framework to integrate spatial relationships in the search process.

- **Association rules.** Spatial association rule mining is the identification of frequently occurring spatial-related patterns in a set of data items in a spatial database (Han et al., 1997). It can be categorised as spatial data dependencies mining.

- **Trend rules.** Basically, trend rules describe patterns of change of one or more non-spatial attributes of objects or objects in their neighbourhood (Ester et al., 1998).

Spatial association rule mining. Koperski & Han (1995) defined spatial association rules (SAR) as association rules with at least one spatial predicate in the antecedents or consequent. Such a spatial predicate could refer to topological relationships, orientation and ordering and contain distance information. The spatial rule mining algorithm employs refinement in a hierarchy of topological relations i.e. starting from approximate spatial computation.

Concept hierarchy refinement. Spatial multi-level association rule mining extends the approach above by refinement in a (spatial or attribute) concept hierarchy. An example of *spatial hierarchy* is the refinement of a country into one or more provinces. An example of *conceptual hierarchy of attributes* is the refinement of areas into rural and urban areas.
SPADA (Spatial Pattern Discovery Algorithm) is a system for spatial association rule mining, in which the rules have a Datalog representation. It

uses refinement through a concept hierarchy of objects. SPADA is used by Malerba et al. (2002) and Appice et al. (2003) for analysing socio-economic issues in census data in order to improve transport planning. Lisi & Malerba (2004) improved this system by the design of the hybrid language AL-log, which yields a unified treatment for both relational and structural data features.

3. Problem description

3.1. System analysis

The company Tele Atlas collects geographical information from several sources, such as satellite images and mobile mapping. It provides the geographical data for companies active in the areas of car navigation systems, geographical information systems and location-based services. From these application areas, the company is facing an ever increasing demand for geographic data quality. It manages a large central database, which is subject to continuous updates, originating from core data collection and processing using high quality standards. There are two strategies by which the quality of the geographic data in the database can be maintained and improved:

- by processing the navigation logs of and explicit update requests by the end user community
- by making quality domain knowledge explicit and verifying it against the data.

In line of the latter strategy, Tele Atlas has set up an infrastructure that allows manually building quality rules and verifying data against these rules. Passive verification implies a check of each update against a limited set of rules. Active verification is done by separate processes, checking the rest of the rules against the whole database. In this paper, we introduce a tool that automates building quality rules.

This tool is used by data engineers and extracts previously unknown relations that are present in the data. It supports the use cases below:

- First of all, the user selects a data sample and formulates a question e.g. "How do speed restrictions of a road element (i.e. elementary piece of any road) relate to adjacent points of interest (POIs)?"

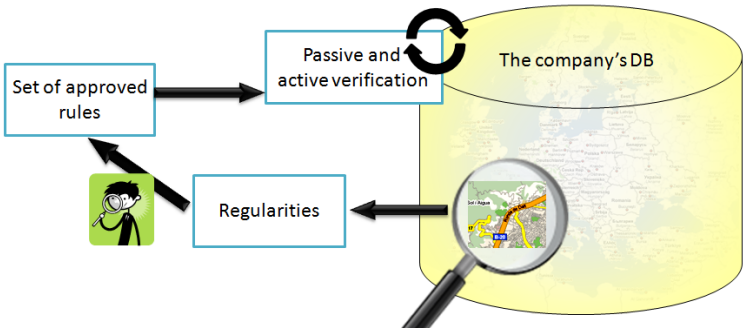

Figure 1: Identification of regularities and anomalies within the quality maintenance business process.

- Within a reasonable amount of time, the user receives direct and complete answers to the question w.r.t. the selected data sample, in the form of rules, together with their statistical relevance.
- The user is able to trace the violations (outliers) against these rules and to visualise them.
- Guided by a rule's outliers, the user decides whether to accept the rule in the quality maintenance system, by exporting it, or not. Also very similar rules without violations can be considered for acceptance. After approval, the rule can be used for active and passive verification, in order to discover outliers w.r.t. the complete database.

These functionalities clearly require a system for **outlier detection by indirect description**. Figure 1 shows its impact on the quality maintenance business process. Data sampling is required to cope with the large dimensions of the database. The data engineers should match data samples to questions. The sample size should be significantly large in order to avoid overfitting.

3.2. The dynamic data model

Besides the geographical data, the data model has a dynamic nature too. The data model changes, for example, when the engineers decide to adopt a new type of POI, or when, entering a new country, the address interpolation representation by integers does not apply any more. In order to design a rule miner tool that copes with this dynamic data model, the metamodel

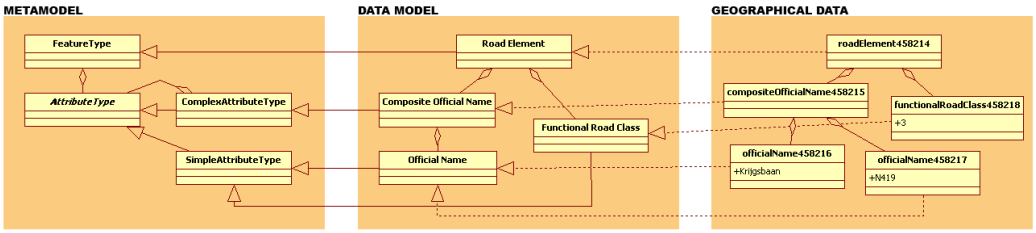

Figure 2: An excerpt of the metamodel, the data model, and the data (UML class diagram).

is constructed. This is the model of the datamodel, which does not change over time. It will be used to design a rule language that is independent from the data model that is currently in use. Figure 2 conceptually shows the relationship between the metamodel, the data model, and the data.

We explain the most important concepts in the metamodel, their implementations in the data model, and the data itself:

- **Feature type.** The company's geographical data basically consists of features of a certain type, for example restaurants, water areas, junctions or road elements.
- **Simple and complex attribute type.** Each feature type is composed of a tree of attributes types, in which the internal nodes are complex attributes and the leaves are simple attributes, i.e. containing a value. A road element has, for instance, a functional road class, which is a value indicating the road importance (highway, secondary road,...) and a composite official name, of which multiple official names contain the name strings. The most important attribute is the **geometry**, which scales down to a point, a polyline, a polygon or a combination of these. The data model also defines spatial relationships such as overlap and distance.
- **Association type** (not in figure). An association type links several feature types by specific roles, e.g. a forbidden traffic manoeuvre between two road elements or connectivity between junctions and road elements.
- **Inheritance support** (not in figure). Furthermore, the metamodel

supports association, feature and attribute type inheritance. For example, restaurants, junctions and schools inherit from the POI type. All types inherit the geometry attribute type from the base feature type.

The rules, generated by the tool, are expressed in terms of the data model and describe previously unknown relationships in the data. Note that a possible data model update requires a set of data transformations, which apply to the already discovered rules as well.

3.3. Rule and outlier type analysis

At this point, we go into more detail about the type of rules that the tool is expected to extract. We received a set of sample rules in advance, out of which 3 typical rules are listed in the second column of Table 2. The company expects their type is very similar to the type of rules the tool might discover.

Question	**Example rule**	**Anomaly description**
"How do road element speed restrictions relate to adjacent POIs?"	"A road element adjacent to a primary school has always a speed restriction of 30 km/h"	road elements adjacent to a school with a speed restriction different from 30 km/h
"How do the road element's attributes interrelate?"	"A road element with a speed restriction of 120 km/h, has always functional road class 1 (i.e. high road importance)"	a road element with speed restriction 120 and functional road class > 1
"How does a roundabout relate to the attributes of its associated features?"	"Each roundabout has at least one connection-association with a road element with a traffic flow away from the roundabout."	a roundabout without connected road elements with a traffic flow away from the roundabout

Table 2: Possible experiment scenarios for the example rule set

According to the definitions by Koperski & Han (1995), this rule set contains both spatial and non-spatial association rules. Table 2 shows a possible experiment scenario w.r.t. the system analysis functionalities for each of the example rules. The questions are examples of system inquiries by users that

definitely result in a concise set of rules, of which the example rule is an unexpected member. 'Unexpected' means that data engineers who do not know the examples, are not able to predict the rule from the question. The anomaly descriptions state which kind of outliers, for each of the rules, the user expects to highlight during visualisation.
The table shows that

- each rule can be mapped to a question aiming at feature and attribute type relations, given a fixed spatial relation or association type.
- for each rule, the expected anomaly descriptions refer to spatial objects for which the fixed relation or association type holds, but the rule fails.[3]

This representation situates the problem as a **relational outlier detection** problem and requires the use of relational association rule mining techniques to look up regularities in a first phase. Moreover, omitting the refinement of spatial operators during the search process speeds up the search, compared to spatial rule mining.

3.4. The integration of a relational datamining technique

Hypothesis language requirements. Building an operational tool that is able to discover patterns in the complete data set, independent from the data model version in use, requires a uniform representation language in terms of the metamodel, in which the data and the rules can be expressed. Besides, the hypothesis language should enable the expression of aggregate ('has-a') relationships between features and attributes.

The algorithm. WARMR (Dehaspe, 1998) is a relational datamining algorithm that induces association rules in datalog representation, which meets the above language requirements. It uses learning from interpretations (Blockeel et al., 1999), which is typically used for description. This learning setting assumes that the input data is presented in the form of interpretations. These are database partitions that represent a set of relational states. A candidate hypothesis describes certain interpretation properties. It covers an interpretation if and only if the interpretation is a model for the hypothesis. The

[3]For instance, in example 1, it would not make sense that the outlier detection comes up with pairs of primary schools and road elements with speed restriction 30 km/h that are not adjacent.

algorithm is linear in the number of interpretations.
First, WARMR executes a level-wise discovery of **frequent queries** that cover the given set of interpretations. Frequent queries are conjunctions of literals that fulfil the language bias provided by the user. The language bias consists of a set of constraints, which determine which frequent queries are searched for. The support of a frequent query is defined as the number of interpretations the query covers to the total number of interpretations. Levelwise discovery involves that, at each level, the frequent queries are specialized by extending them with each of the allowed literals, until a specified maximum number of levels (literals) is reached or until the support has decreased below a specified minimal support. Note that also background knowledge, in the form of rules, can be taken into account during interpretation coverage control.
Next, frequent queries are processed into **query extensions**. A query extension is a datalog clause of the form $h :- b_1, b_2, ..., b_m$, generated from the queries $b_1, b_2, ..., b_m$ and $b_1, b_2, ..., b_m, h$. The confidence of the query extension is defined as the support of the latter query to the support of the first. The support of the first and the latter query are said to be the bodyfrequence and the support of the query extension. The discovery of **outliers** to a query extension is trivial. Outliers are the interpretations that are covered by the first query but not by the latter.
Clare & King (2003) treat the distribution of levelwise rule discovery algorithms such as WARMR. They describe Farmer, Worker and Merger processes to distribute frequent query support counts within equal amounts of interpretations over multiple machines.
In our case study, we use the WARMR implementation of the ACE Datamining System (Blockeel et al., 2009). It implements a set of Inductive Logic Programming (ILP) algorithms, of which the efficiency has been improved by the query pack mechanism (Blockeel et al., 2002).

4. System design

4.1. Rationale

The system analysis in the previous section states that the user starts an experiment from a selected data sample, which is typically a geographic area and a question. The rule type analysis showed that rule mining with a relational approach, realised by WARMR, can formulate answers to these questions. This algorithm learns from interpretations. The user's question

consists of 3 data selection items. The design choices for each of these items are motivated below.

- **Central feature type.** In the quality maintenance system of the company, a rule starts by definition by a universal quantification for the features of a specific type, as in "For all features of type x: ...". Strictly speaking, the rules produced by WARMR have a relative quantification over the interpretations, as in "For 99% of the interpretations: ...". However, this rule is adopted as a perfect rule (and thus universally quantified) by the quality maintenance system. We prefer to build interpretations for features in a geographic area and of one specific type. As a consequence, WARMR produces rules that conform to the specifications of the system. Therefore, the user has to select a feature type.

- **Feature inclusion condition.** One approach to including spatial information (e.g. distance) in the rules, would be to define a set of spatial relations in the background knowledge. This would result in a large number of spatial calculations (not any information will be processed and cached only once) during the knowledge discovery process, resulting in poor computational performance. A common technique for performance improvement in spatial data mining is the materialisation of spatial relationships, described by Shekhar et al. (2003). It involves that all necessary spatial calculations are executed during preprocessing and that the results are integrated in the input data. Therefore, we prefer the user to select a spatial relation, e.g. overlap, and this information is incorporated in the interpretations. This is realised by the inclusion of features for which the relation holds. The same principle is applied for associations, not for performance but for uniformity reasons.

- **Attribute types.** The data engineer might not be interested in possible relations for each of the simple and complex attribute types involved. It should thus be possible to restrict the attributes types entered in the interpretations.

These data selection items are shown in Fig. 3.
The next subsection details the design of a rule language that covers the example rules in Table 2. The following subsections discuss the data preprocessing, mining and postprocessing steps of the rule miner tool in Fig. 3.

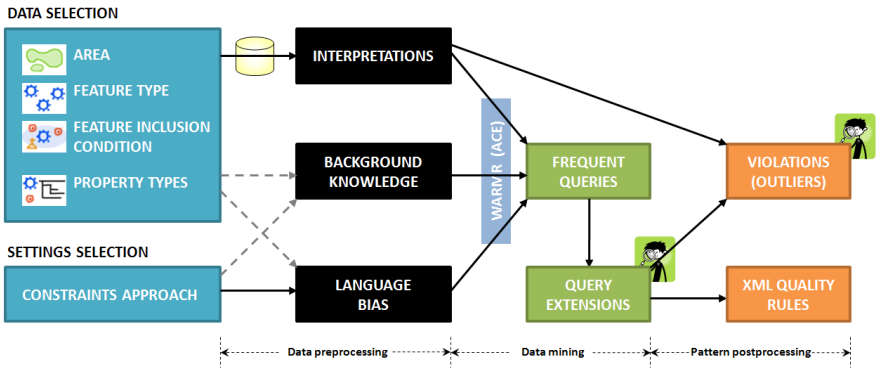

Figure 3: Design of the rule miner prototype.

4.2. Generic rule language

In this section, a rule language is defined in terms of the metamodel, in order to support data model evolution. In Table 3, we define and illustrate a set of primitive functions.

The rules that are generated will test for the existence of related features, certain attributes, or certain attribute values. The rule language consists of the components defined in Table 4. $Feat\_rel(CF, F)$ is a boolean function, which returns *true* if the relationship between the features CF and F holds. Example functions for $Feat\_rel(CF, F)$, will be defined in Subsection 4.3. Let us assume that $adjacent50(CF, F)$ is *true* when two features are less than 50 metres apart. The ground term
$feature\_exists(church53, \text{“}Church\text{”}, adjacent50, road54)$
means that feature $church53$ of type “Church” is adjacent to feature road54. The first rule in Table 2 is defined as

$$
\begin{aligned}
&foreach\_feature(A, \text{“}Road\ Element\text{”}) : \\
&\qquad feature\_exists(B, \text{“}School\text{”}, adjacent50, A), \\
&\qquad simple\_att\_exists(C, \text{“}Type\text{”}, B, \text{“}primary\text{”}), \\
&\qquad complex\_att\_exists(D, \text{“}Composite\ Speed\ Restriction\text{”}, A) \\
&\qquad \Rightarrow simple\_att\_exists(E, \text{“}Speed\ Restriction\text{”}, D, 30)
\end{aligned}
$$

Definitions	**Examples**
$type(DataElement)$ returns the specific data model type of a feature, an attribute or an association.	$type(feat4497) =$ "$Road$" $type(attr4498) =$ "$Address$" $type(ass4499) =$ "$Forbidden\ Turn$"
$value(SimpleAttribute)$ returns the assigned value of a simple attribute.	$value(attr4498) =$ "$Elm\ Park$"
$has(DataElement1, DataElement2)$ returns *true* if the first element contains the second one. According to the metamodel, only features and complex attributes can contain other attributes. Associations contain features.	$has(feat4497, attr4498) = true$ $has(ass4499, feat4497) = true$
$spat\_dist(Feature1, Feature2)$ returns the spatial distance between the "Geometry" attributes of the 2 features. Returns 0 if both geometries overlap.	$spat\_dist(feat4496, feat4497) = 20$

Table 3: Primitive function definitions

No.	**Definitions**
1	$foreach\_feature(feature\ \underline{CF}, type\ T)$ $\forall\ CF : type(CF) = T$
2	$feature\_exists(feature\ \underline{F}, type\ T, boolfunction\ Feat\_rel, feature\ \mathbf{CF})$ $\exists\ F : type(F) = T \wedge Feat\_rel(CF, F)$
3	$complex\_att\_exists(attribute\ \underline{A}, type\ T, element\ \mathbf{P})$ $\exists\ A : type(A) = T \wedge has(P, A)$
4	$simple\_att\_exists(attribute\ \underline{A}, type\ T, element\ \mathbf{P}, value\ V)$ $\exists\ A, V : type(A) = T \wedge has(P, A) \wedge value(A) = V$

Table 4: Rule language components

given the model in which speed restrictions belong to a composite attribute and primary is a value for the attribute named "Type" contained by the "School" feature type.

4.3. Data preprocessing

Interpretation generation. First, the user enters a selection of the geographical database partitions to be inspected, whereupon these partitions are loaded from the database. Next, the interpretations are generated from the loaded data. This generation consists of the following steps (present in Fig. 3):

- The user chooses a central feature type of interest, around which the interpretations are built. For each instance of this central feature type in the data, an interpretation is constructed. This step determines the T parameter in rule language component 1.

Approach	$Feat\_rel(CF, F)$	**Example rule**
Inclusion by overlap adds all features of some types of choice (in the set $ftypeset$) that overlap the central feature.	$spat\_dist(CF, F) = 0$ $\wedge\, (type(F) \in ftypeset)$	Each "Service Area" overlaps at least one "Service Point".
Inclusion by offset distance adds all features of some types of choice that are situated an offset distance d apart from the central feature.	$spat\_dist(CF, F) < d$ $\wedge\, (type(F) \in ftypeset)$	Rule 1 in Table 2.
Inclusion by association type adds all features that are associated with the central feature type for some association types of choice (in the set $atypeset$).	$\exists\ A : type(A) \in atypeset$ $\wedge\, has(A, CF) \wedge has(A, F)$	Each "Slip Road" is associated with a "Forbidden Turn".

Table 5: Inclusion condition: three approaches

- By formulating inclusion conditions, the user is able to include other features in the interpretations that are somehow related to the central feature. This step both constrains the T parameter and defines the $Feat\_rel$ function to be included in rule language component 2. Currently, 3 types of inclusion condition, shown in Table 5, are supported.

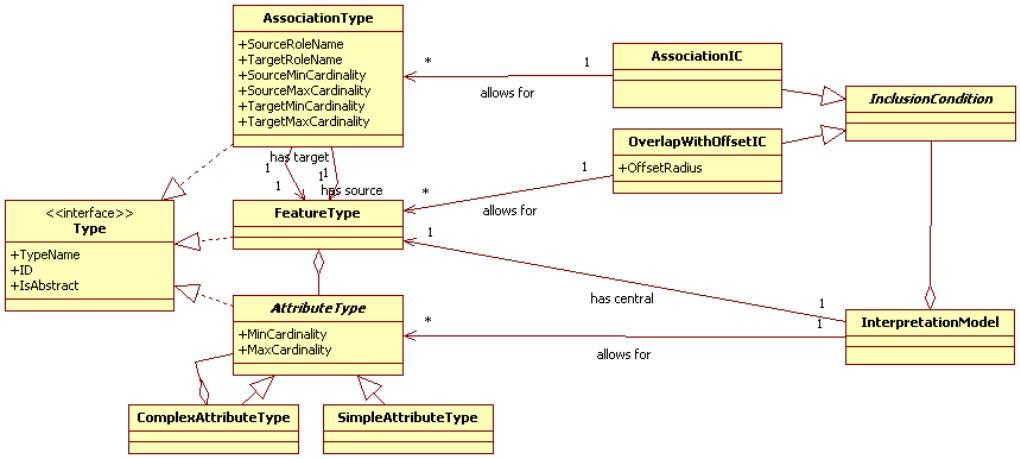

Figure 4: The geographic metamodel and its relation to interpretation construction (UML class diagram).

- The user has to indicate an attribute type subtree for each of the feature types involved. Only the information for these attribute types is recorded in the interpretations. This step constrains the T parameters in rule language components 3 and 4.

Fig. 4 shows the relationship between the geographic metamodel and the concepts of interpretation construction. The interpretations are generated into Prolog notation, based on the rule language components.

Language bias and background knowledge generation. The language bias and background knowledge are partially fixed, partially generated in a semi-automated manner. The language bias ensures that recurring variables only bind parameters of the same type, as listed in Table 4. Note that features and attributes are both elements. The same table contains annotations that indicate how variables and constants are introduced in candidate rules. By default, the underlined terms will be replaced by new variables, the normal terms by constants and the bold terms by previously introduced variables. There are some possible variations:

- The end user can select an alternative language bias and background knowledge pair, resulting in *data model mining*. An example rule is: "Each restaurant is overlapped by exactly one restaurant area."

This setting only includes information about the existence of simple attributes instead of the values (cf. underlined parameter V of rule language component 4), and allows to mine for cardinalities. In this case, the background knowledge contains the definition of an isunique-predicate, indicating whether an attribute of a given type only occurs once for each parent attribute or feature. It is included in the language bias.

- The tool supports abstract feature and attribute types in the hypothesis language. An example rule is "Each service point (this is an abstract feature type) is adjacent to a road". In this case, the background knowledge contains the necessary rules to derive whether a feature or attribute type implements an abstract type. It is necessary to enumerate all possible abstract and non-abstract types in the language bias.

Table 6 shows a sample data flow during preprocessing.

4.4. Data mining and pattern postprocessing

Data mining. The interpretations, background knowledge and language bias files are fed to the ACE Datamining System. Before mining, the user is asked a minimum support, a minimum confidence and a maximal rule length (i.e. maximal number of literals). In a level-wise manner, WARMR generates the frequent queries (above the minimum support), which are processed into query extensions (above the minimum confidence) afterwards. Some examples are given in the results section. These query extensions are presented to the end user and can be selected individually for outlier detection and rule export.

Outlier detection. Outliers for an individual rule are the interpretations for which the body of the rule holds, but the body extended by the head fails. Outlier detection involves the execution of these two queries on the logic program of each of the interpretations separately, each time extended by the background knowledge. The tool supports a generic visualisation of outlying interpretations on a geographical map. Interpretations always represent features that have a geometry, which scales down to a (set of) points, polylines or polygons.

Raw data	$type(f001) = type(f002) = type(f003) =$ "*Road Element*" $type(f004) = type(f005) =$ "*School*" $spat\_dist(f001, f004) = 8 \quad spat\_dist(f001, f005) = 66$ $spat\_dist(f002, f004) = 24 \quad spat\_dist(f002, f005) = 22$ $spat\_dist(f003, f004) = 16 \quad spat\_dist(f003, f005) = 44$ $spat\_dist(f004, f005) = 21$ $type(ass001) = type(ass002) =$ "*Connected Road Elements*" $has(ass001, f001) = has(ass001, f003) = true$ $has(ass002, f002) = has(ass002, f003) = true$ $type(a001) = type(a002) = type(a003) =$ "*Comp. Speed Restriction*" $type(a011) = type(a012) = type(a013) =$ "*Speed Restriction*" $type(a004) = type(a005) =$ "*Type*" $has(f001, a001) = has(f002, a002) = has(f003, a003) = true$ $has(a001, a011) = has(a002, a012) = has(a003, a013) = true$ $has(f004, a004) = has(f005, a005) = true$ $value(a011) = 30 \quad value(a004) =$ "*primary*" $value(a012) = 50 \quad value(a005) =$ "*university*" $value(a013) = 30$
Input settings	- Central feature type: "Road Element" - Inclusion by offset distance: all restaurants, schools and gas stations that are situated 50m apart from the central feature - Attribute types: all attribute types for the feature types involved
Interpretations	%interpretation for "Road Element" f001 ... %interpretation for "Road Element" f002 $complex\_att\_exists(f002,$ "*Composite Speed Restriction*"$, a002)$. $simple\_att\_exists(a002,$ "*Speed Restriction*"$, a012, 50)$. $feature\_exists(f004,$ "*School*"$, adjacent50, f002)$. $simple\_att\_exists(a004,$ "*Type*"$, f004,$ "*primary*"$)$. $feature\_exists(f005,$ "*School*"$, adjacent50, f002)$. $simple\_att\_exists(a005,$ "*Type*"$, f005,$ "*university*"$)$. %interpretation for "Road Element" f003 ...

Table 6: Sample data flow during preprocessing for the school/road data set

Language bias	Default settings.
Frequent queries	$foreach\_feature(A, \text{“}RoadElement\text{”})$: $-$ $complex\_att\_exists(B, \text{“}Comp.\ Speed\ Restriction\text{”}, A)$. Supp: 1 $-$ $feature\_exists(B, \text{“}School\text{”}, adjacent50, A)$. Supp: 1 $-$ ... $-$ $feature\_exists(B, \text{“}School\text{”}, adjacent50, A)$, $simple\_att\_exists(C, \text{“}Type\text{”}, B, \text{“}university\text{”})$, $complex\_att\_exists(D, \text{“}Comp.\ Speed\ Restriction\text{”}, A)$. Supp: 0.67 $-$ $feature\_exists(B, \text{“}School\text{”}, adjacent50, A)$, $simple\_att\_exists(C, \text{“}Type\text{”}, B, \text{“}primary\text{”})$, $complex\_att\_exists(D, \text{“}Comp.\ Speed\ Restriction\text{”}, A)$. Supp: 1 $-$ $feature\_exists(B, \text{“}School\text{”}, adjacent50, A)$, $simple\_att\_exists(C, \text{“}Type\text{”}, B, \text{“}primary\text{”})$, $complex\_att\_exists(D, \text{“}Comp.\ Speed\ Restriction\text{”}, A)$, $simple\_att\_exists(E, \text{“}Speed\ Restriction\text{”}, D, 30)$. Supp: 0.67
Query extensions	... %query extension 20 $foreach\_feature(A, \text{“}Road\ Element\text{”})$: $feature\_exists(B, \text{“}School\text{”}, adjacent50, A)$, $simple\_att\_exists(C, \text{“}Type\text{”}, B, \text{“}primary\text{”})$, $complex\_att\_exists(D, \text{“}Composite\ Speed\ Restriction\text{”}, A)$ $\Rightarrow simple\_att\_exists(E, \text{“}Speed\ Restriction\text{”}, D, 30)$. Conf: 0.67
Outliers	The outliers for query extension 20 are: - interpretation for “Road Element” f002 Visualisation: $g002 : type(g002) = \text{“}Geometry\text{”} \wedge has(f002, g002)$.

Table 7: Sample data flow during datamining and postprocessing for the school/road data set

Rule export. The XML rule format used by the company is a semantical superset of the rule language defined in subsection 4.2. The rule export involves syntactical conversion, conversion to primitive functions and the removal of duplicate information. For example, the feature type set constraint in $Feat\_rel(FC, F)$ can be omitted, because rule language component 2 involves a feature type declaration. The rule export module allows the end user to export an accepted rule to the quality maintenance system, which uses the rule for active or passive verification.

Table 7 presents a sample data flow during the data mining and outlier detection steps, which is subsequent to the flow in Table 6.

5. Results

In this section, we present a set of example rules found by the system. We first present the outcome of two specific experiments, focussing on the query extensions that have ***almost*** 100% confidence. These rules are of particular interest, because they directly indicate possible outliers in the data sample. For each of the rules, expert feedback is given. Next, we present a sanity check, in which experiments are reconstructed for a set of rules that have been designed from specifications manually by the data engineers.

5.1. Experiment 1: discovering inter-feature relations

In a first experiment, we try to induce relationships between associated features of junctions. Therefore, we used following input settings:

- Geographical data set: northern Barcelona (consisting of 1404 junctions)
- Central feature type: junction
- Inclusion of: all features that are associated by one of the 16 association types defined on the junction feature type
- Attribute types: 10 (official names and type IDs) from the set of all attribute types for the feature types involved
- Minimal support: 0.05
- Minimal confidence: 0.90
- Maximal rule length: 5

This results in 90 frequent queries and 79 query extensions. The outcome rule with the highest confidence below 100% is:

$foreach\_feature(A, \text{“}Junction\text{”}) :$

$feature\_exists(B, \text{“}Calculated\ Prohibited\ Manoeuvre\text{”}, assoc, A),$

$\Rightarrow feature\_exists(C \neq B, \text{“}Calculated\ Prohibited\ Manoeuvre\text{”}, assoc, A)$

Confidence: 0.9795

Support: 0.1019

Explanation. The rule means that, if a prohibited manoeuvre is defined over a junction, also another prohibited manoeuvre exists over this junction. The 'Calculated Prohibited Manoeuvre' association type defines forbidden traffic turns over a set of junctions, connected by the role type 'Via Junction'.

Feedback. Data experts identify this rule as a promising check, although 'Calculated Prohibited Manoeuvre' is an attribute generated from basic attributes that are already present in the data. This rule has 3 outliers in the data, 2 of which are located at the border of the data set. These are false-positive outliers due to incomplete information. A third one triggered further study by the engineers.

5.2. Experiment 2: discovering intra-feature relations

In a second experiment, we try to find relationships amongst the attributes of road elements.

- Geographical data set: northern Barcelona (consisting of 1851 road elements)
- Central feature type: road element
- Inclusion of: none
- Attribute types: 20 attribute types (about name, postal information, speed restriction, routing classes, etc.) belonging to the road element feature type
- Minimal support: 0.05
- Minimal confidence: 0.90
- Maximal rule length: 4

This results in 190 frequent queries and 169 query extensions. The 3 most interesting outcome rules with confidence below 100% are:

$foreach\_feature(A, \text{“}Road\ Element\text{”}) :$

$simple\_att\_exists(B, \text{“}Routing\ Class\text{”}, A, \text{“}Local\ Roads\ of\ High\ Importance\text{”}),$

$\Rightarrow simple\_att\_exists(C, \text{“}Road\ Conditions\text{”}, A, \text{“}Paved\text{”})$

Confidence: 0.9981

Support: 0.5786

$foreach\_feature(A, \text{“}Road\ Element\text{”}):$

$simple\_att\_exists(B, \text{“}Functional\ Road\ Class\text{”}, A, \text{“}Local\ Roads\text{”}),$

$\Rightarrow simple\_att\_exists(C, \text{“}Routing\ Class\text{”}, A, \text{“}Destination\ Traffic\text{”})$

Confidence: 0.9947

Support: 0.3047

$foreach\_feature(A, \text{“}Road\ Element\text{”}):$

$simple\_att\_exists(B, \text{“}Form\ Of\ Way\text{”}, A, \text{“}Road\ in\ Pedestrian\ Zone\text{”}),$

$\Rightarrow simple\_att\_exists(C, \text{“}Functional\ Road\ Class\text{”}, A,$
$\text{“}Local\ Roads\ of\ Minor\ Importance\text{”})$

Confidence: 0.9917

Support: 0.0643

Explanation. These rules show obvious correlations between a road's importance, its form and its actual condition. Their respective meanings are that each 'Road Element' that

1. has the 'Routing Class' label 'Local Roads of High Importance', has the 'Road Condition' label 'Paved'.
2. has the 'Functional Road Class' label 'Local Roads', has the 'Routing Class' label 'Destination Traffic'.
3. has the 'Form Of Way' label 'Road in Pedestrian Zone', has the 'Functional Road Class' label 'Local Roads of Minor Importance'.

Feedback. According to the data experts, the first rule reveals an interesting relationship, but is too much dependent on geography. A 'Routing Class' reflects a relative importance, whereas a 'Road Condition' describes a physical state. This means that an individual 'Routing Class' attribute is strongly related to the global attribute distribution over a country, such that the 'Routing Class' distribution for unpaved roads varies from country to country. Note that it is not unusual to include country-dependent information in the quality rules, but that including the geographical dimension in the analysis is beyond the primary scope of this tool for automated rule discovery.

The second rule shows a correlation between two road class categorisation systems. This correlation is already implied by internal road class production rules.
The third rule indicates an interesting correlation between the 'Functional Road Class' and 'Form Of Way' attribute. The first one indicates a relative importance w.r.t. functional aspects of a road, whereas the latter combines both physical and functional aspects. In this case, 'Road in Pedestrian Zone' is a purely functional determinant. The single outlier, a relative important road element in a pedestrian zone, is most probably an anomaly and the rule has been accepted for further inspection.

5.3. Rule set for experiment reconstruction

In this evaluation phase, we verify whether end users would be able to discover rules that are currently in use by the quality maintenance system. This sanity check involves experiment reconstruction for this selection of rules. It assumes unawareness by end users of these rules. The top column of Table 8 shows 4 rules that have been manually designed from specifications by data engineers. For each of the rules, we set the experiment parameters such that it has the rule amongst its results and such that data engineers are not able to predict the rule as an outcome of the experiment set-up.

Table 8 shows some detailed information about the experiments. In practice, it is often needed to lower the minimal support in order to find the target rules. The target rules could be found in experiment 1,2 and 4. For experiment 3, the targeted relation was not present in the input data set (which was checked manually). No outliers could be detected w.r.t. these rules, because they had already been adopted by the quality maintenance system. Each of the rules comes with a set of other rules, most of the time containing valuable information. Most of the targeted rules are short, so the total number of rules can be kept low by lowering the maximum rule length.

6. Future work

The sanity check in Section 5 has shown that the tool is able to discover realistic quality rules. However, the current rule language still has limitations. This section presents two language extensions that adapt the rule expressiveness to real-world standards.

Quality rule	A Road Element that is part of a Freeway Intersection, shall not be part of another Freeway Intersection (FWI).	A face shall not be part of 2 or more Postal Districts (PDs)	Road Elements having a Functional Road-class (FRC) attribute 'Motorway', 'Major Road', 'Other Major Road', 'Secondary Road' or 'Stubble' shall have a 'No Obstruction' Blocked Passage attribute.	A Junction can bound exactly 2 or 0 Road Elements with Form of Way (FOW) Roundabout.
Experiment description	Find relations between backward associated features to each road element; in this set FWI is unique	Find relations between backward associated features to each face; in this set PD is unique	Find relations between attributes of each road element	Find relations between (attributes of) road elements that overlap each junction
Geographical data set	Crisler	Crisler	Crisler + Elzie + Malta + Nilsson (reason: FRC variation)	Nilsson
Central feature type	Road Element	Face	Road Element	Junction
# interpretations	1851	674	6765	1611
Inclusion condition	Association	Association	-	Overlap
	All non-abstract backward associations	All non-abstract backward associations	-	Road Element
Attribute Types	-	-	Everything from Composite Blocked Passage + FOW and FRC from Road Element	(Composite) Official Name, FOW and FR from Road Element
Constraint approach	Datamodel mining	Datamodel mining	default	default
Minimal support	0.02 (FWI has low support)	0.05	0.01 (FRC 2 3 4 8 have low support)	0.01 (FOW 3 has low support)
Minimal confidence	0.90	0.90	0.7 (to show invalidity of target rule)	0.90
Maximal rule length	4	2	5	3
Targeted rule	$foreach\_feature(A,$ "$Road\ Element$") : $feature\_exists(B,$ "FWI", $assoc, A)$ $\Rightarrow is\_unique(B, A)$	$foreach\_feature(A,$ "$Face$") : $feature\_exists(B,$ "PD", $assoc, A)$ $\Rightarrow is\_unique(B, A)$	not found	$foreach\_feature(A,$ "$Junction$") : $feature\_exists(B,$ "$Road\ Element$", $overl, A)$, $simple\_att\_exists(C,$ "FOW", $B,$ "$Roundabout$") $\Rightarrow feature\_exists(C \neq B,$ "$Road\ Element$", $overl, A)$
Confidence and support	1.0 0.0427	1.0 1.0	-	1 0.02
Violations	0	0	-	0
Number of rules per level	4+26+105+289	7+59	1+0+9+16+28	2+2+7

Table 8: Sanity check details.

Association. Presently, association is only used as a condition to include other features in an interpretation. Full integration means that the rule language is able to capture associations (by name and by role) between features and to list properties of associations. This would enable:

- the discovery of recurring patterns in association roles and association properties. An example could be: if a junction is the first junction of a manoeuvre, it is always the last junction of another manoeuvre.
- the combinatorial application of different inclusion conditions. For example, this would enable finding that a junction's associated intersection also overlaps this junction.

Spatial functions and concepts. There is a number of functions and concepts, tailored to the domain of geographic databases, that would be very useful when integrated in the current system.

- Feature count, for example, supports the discovery of certain types of anomalies in geographical data, such as erroneous duplication of data. An example rule is: the number of hotels in a city is lower than the number of restaurants.
- Spatial distance (for feature sizes as well as distances between features) can be realised by calculation during preprocessing, and making it explicit in the rule language. This measure would enable finding that the distance between a gas station and a motorway is always between 10 and 100 metres.

7. Conclusion

We have built a tool to mine for relational regularities and corresponding outliers in geographical data. This tools assists a geographic content providing company in reasoning about the structure of the data and about the data itself. It is able to extract previously unknown knowledge in an automated way, which can be integrated in the quality maintenance process directly. It anticipates the process of manual rule formulation driven by individual reporting of anomalies in the data. Moreover, it is independent from the data model currently in use.

The WARMR algorithm is the central component of this tool. Its input

consist of interpretations, a background knowledge and a language bias, generated from the end user's data selection and mining preferences. Its output is used for relational outlier detection by indirect description i.e. first WARMR mines for rules that describe regularities and next, violations of these rules are identified as outliers.

The case studies show that relatively simple experiments yield valuable information about regularities and outliers in the sample data. Three out of 4 manually designed example rules were reconstructed using the tool. Only one rule was not found because it had very low confidence over the sample data. The validation shows that the system requirements of our tool are met.

Acknowledgements

This research is part of an R&D project funded by IWT (050730). Special thanks to Gert Vervaet, Frank Maes and Dieter Verhofstadt (Tele Atlas) for the constructive feedback on the rule miner prototype. Celine Vens is a postdoctoral fellow of the Research Foundation Flanders (FWO-Vlaanderen).

References

Aggarwal, C. C., & Yu, P. S. (2001). Outlier detection for high dimensional data. In *SIGMOD Conference*.

Angiulli, F., Greco, G., & Palopoli, L. (2007). Outlier detection by logic programming. *ACM Trans. Comput. Logic*, *9*, 7.

Appice, A., Ceci, M., Lanza, A., Lisi, F. A., & Malerba, D. (2003). Discovery of spatial association rules in geo-referenced census data: A relational mining approach. *Intelligent Data Analysis*, *7*, 541–566.

Berzal, F., Cubero, J.-C., & Marín, N. (2004). Anomalous association rules. In *IEEE ICDM Workshop Alternative Techniques for Data Mining and Knowledge Discovery*.

Blockeel, H., De Raedt, L., Jacobs, N., & Demoen, B. (1999). Scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge Discovery*, *3*, 59–93.

Blockeel, H., Dehaspe, L., Demoen, B., Janssens, G., & Vandecasteele, H. (2002). Improving the efficiency of inductive logic programming through the use of query packs. *Journal of Artificial Intelligence Research*, *16*, 135–166.

Blockeel, H., Dehaspe, L., Ramon, J., Struyfand, J., Assche, A. V., Vens, C., & Fierens, D. (2009). *The ACE Data Mining System, User's Manual*. DTAI, K.U.Leuven.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density-based local outliers. In W. Chen, J. F. Naughton, & P. A. Bernstein (Eds.), *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA* (pp. 93–104). ACM.

Caruso, C., & Malerba, D. (2007). A data mining methodology for anomaly detection in network data. In *KES '07: Knowledge-Based Intelligent Information and Engineering Systems and the XVII Italian Workshop on Neural Networks on Proceedings of the 11th International Conference* (pp. 109–116). Berlin, Heidelberg: Springer-Verlag.

Ceci, M., & Appice, A. (2006). Spatial associative classification: propositional vs structural approach. *J. Intell. Inf. Syst.*, *27*, 191–213.

Clare, A., & King, R. D. (2003). Data mining the yeast genome in a lazy functional language. In *PADL '03: Proceedings of the 5th International Symposium on Practical Aspects of Declarative Languages* (pp. 19–36). London, UK: Springer-Verlag.

De Raedt, L., & Dehaspe, L. (1997). Clausal discovery. *Mach. Learn.*, *26*, 99–146.

Dehaspe, L. (1998). *Frequent Pattern Discovery in First-Order Logic*. Ph.D. thesis Department of Computer Science, Katholieke Universiteit Leuven, Belgium.

Dehaspe, L., & Toivonen, H. (1999). Discovery of frequent datalog patterns. *Data Min. Knowl. Discov.*, *3*, 7–36.

Ester, M., Frommelt, E., peter Kriegel, H., & Sander, J. (1998). Algorithms for characterization and trend detection in spatial databases. In *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD* (pp. 44–50).

Frank, R., Ester, M., & Knobbe, A. (2009). A multi-relational approach to spatial classification. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 309–318). New York, NY, USA: ACM.

Frank, R., Jin, W., & Ester, M. (2007). Efficiently mining regional outliers in spatial data. In D. Papadias, D. Zhang, & G. Kollios (Eds.), *Advances in Spatial and Temporal Databases, 10th International Symposium, SSTD 2007, Boston, MA, USA, July 16-18, 2007, Proceedings* (pp. 112–129). Springer volume 4605 of *Lecture Notes in Computer Science*.

Haglin, D. J., & Manning, A. M. (2007). On minimal infrequent itemset mining. In R. Stahlbock, S. F. Crone, & S. Lessmann (Eds.), *Proceedings of the 2007 International Conference on Data Mining, DMIN 2007, June 25-28, 2007, Las Vegas, Nevada, USA* (pp. 141–147). CSREA Press.

Han, J., Koperski, K., & Stefanovic, N. (1997). Geominer: a system prototype for spatial data mining. *SIGMOD Rec.*, *26*, 553–556.

Knorr, E. M., Ng, R. T., & Tucakov, V. (2000). Distance-based outliers: Algorithms and applications. *VLDB Journal: Very Large Data Bases*, *8*, 237–253.

Koh, Y. S., & Rountree, N. (2005). Finding sporadic rules using apriori-inverse. In T. B. Ho, D. W.-L. Cheung, & H. Liu (Eds.), *PAKDD* (pp. 97–106). Springer volume 3518 of *Lecture Notes in Computer Science*.

Koh, Y. S., Rountree, N., & O'Keefe, R. A. (2008). Mining interesting imperfectly sporadic rules. *Knowl. Inf. Syst.*, *14*, 179–196.

Koperski, K., & Han, J. (1995). Discovery of spatial association rules in geographic information databases. In M. J. Egenhofer, & J. R. Herring (Eds.), *Proc. 4th Int. Symp. Advances in Spatial Databases, SSD* (pp. 47–66). Springer-Verlag volume 951.

Laros, J. F. J. (2005). *Unique factors in the human genome*. Master's thesis Leiden University.

Lavrač, N., & Džeroski, S. (1994). *Inductive Logic Programming: Techniques and Applications*. New York: Ellis Horwood.

Lazarevic, A., Srivastava, J., Kumar, V., Banerjee, A., & Chandola, V. (2008). Data mining for anomaly detection (tutorial). In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

Lisi, F. A., & Malerba, D. (2004). Inducing multi-level association rules from multiple relations. *Machine Learning*, *55*, 175–210.

Maervoet, J., De Causmaecker, P., Nowé, A., & Vanden Berghe, G. (2008). Feasibility study of applying descriptive ILP to large geographic databases. In *Workshop on Mining Multidimensional Data*.

Malerba, D., Esposito, F., Lisi, F., & Appice, A. (2002). Mining spatial association rules in census data. *Research in Official Statistics*, *5*, 19–44.

Münz, G., Li, S., & Carle, G. (2007). Traffic anomaly detection using k-means clustering. In *Proc. of Leistungs-, Zuverlässigkeits- und Verlässlichkeitsbewertung von Kommunikationsnetzen und Verteilten Systemen, 4. GI/ITG-Workshop MMBnet 2007*. Hamburg, Germany.

Plantevit, M., Goutier, S., Guisnel, F., Laurent, A., & Teisseire, M. (2007). Mining unexpected multidimensional rules. In I.-Y. Song, & T. B. Pedersen (Eds.), *DOLAP* (pp. 89–96).

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, *29*, 427–438.

Ramon, J. (2002). *Clustering and instance based learning in first order logic*. Ph.D. thesis K.U.Leuven.

Shekhar, S., Zhang, P., Huang, Y., & Vatsavai, R. R. (2003). Trends in spatial data mining. In H. Kargupta, A. Joshi, K. Sivakumar, & Y. Yesha (Eds.), *Data Mining: Next Generation Challenges and Future Directions*. AAAI/MIT Press.

Suzuki, E. (2002). Undirected discovery of interesting exception rules. *IJPRAI*, *16*, 1065–1086.

Zeitouni, K. (2002). A survey of spatial data mining methods databases and statistics point of views. In *Data warehousing and web engineering* (pp. 229–242). Hershey, PA, United States: IRM Press.

Zhu, C., Kitigawa, H., Papadimitriou, S., & Faloutsos, C. (2004). Outlier detection adaptive to users' intentions. In *Proceedings of the 15th IEICE Data Engineering Workshop*.