Learning Characteristic Rules Relying on Quantified Paths

Teddy Turmeaux
Ansaf Salleb
Christel Vrain
Université d’Orléans, LIFO

Daniel Cassard
BRGM

Rapport N° 2003-03
Learning Characteristic Rules Relying on Quantified Paths

Teddy Turmeaux  Ansaf Salleb  Christel Vrain

LIFO, Université d’Orléans, rue Léonard de Vinci
BP 6759, F-45067 Orléans Cedex 2, France
{Turmeaux, Salleb, Vrain}@lifo.univ-orleans.fr

Daniel Cassard

BRGM, 3, avenue Claude Guillemin, B.P. 6009
Orléans Cedex 2, France
d.cassard@brgm.fr

Abstract

We address the characterization task that aims at finding properties shared by a given set of objects. This task is interesting for some real applications since it does not require negative examples. We present a general framework for the characterization of a target set of objects by means of their own properties, but also the properties of objects linked to them. According to the kinds of objects, various links can be considered. For instance, in the case of relational databases, associations are the straightforward links between pairs of tables. We propose Cartetext, a new algorithm for mining characterization rules and we show how it can be used on multi-relational and spatial databases.

Keywords Machine Learning, Inductive Logic Programming, Data Mining, Characteristic Rules, Relational Databases, Spatial Databases.

1 Introduction

Characterization is a descriptive data mining task which aims at mining concise and compact descriptions of a subset of objects, called the target set. It consists in discovering properties that characterize that objects set, taking into account their own properties but also properties of the objects linked to them.

In comparison to classification and discrimination, characterization is interesting since it does not require negative examples. This is an important feature for some real world applications where it is difficult to collect negative examples.

Several fields have contributed to this task. On the one hand, characterization has been treated as descriptive generalization in the field of Machine Learning [10]. Characterizing a set of objects has also been considered as computing the least general generalization in Inductive Logic Programming, [13, 12], but such an approach leads to complexity problems. On the other hand, in data mining, Han et al. [7, 6] have introduced attribute oriented induction for data generalization, but in their framework, background knowledge such as taxonomies is needed for generalizing data, and objects are described in a single table, which limit the applicability of such a method.

We can also consider that characterization is close to the task of mining frequent properties on the target set. This task has already long been studied [1, 9, 5, 14], since in many systems, it is the first step for mining association rules. Nevertheless, most works suppose that data is stored in a single table, and few algorithms [3] really handle multi-relational databases. Moreover, the frequency (also called the support) is not sufficient to characterize the objects of the target set, because it is also important to determine whether a property is truly a characteristic feature by considering also the frequency of that property outside the target set.

The approach we propose handles multi-relational databases taking into account the structure of the database. It relies on the definition of a Quantified Path which is an expression that specifies
how to take into account different kinds of objects and their relationships, starting from the target objects. For instance, considering as a target set the set of films produced by a given person $Sp$ and denoted by $Movie(Sp)$, the following expression:

$$Movie(Sp) : \exists \text{Award} :: \text{Award.kind in(Oscar, GoldenPalm)}$$

is a characteristic rule which means that each movie produced by $Sp$ has received at least one Oscar award or Golden Palm award.

The expression $Movie(Sp) : \exists \text{Award}$ is a quantified path. It specifies that we are interested in the properties satisfied by at least one award received by the $Sp$'s movies. On the other hand, considering the Quantified Path $Movie(Sp) : \forall \text{Award}$ means that we are looking for properties satisfied by all the awards received by all $Sp$'s movies.

In the framework we propose, we give a definition of a characteristic rules based on the notion of quantified paths and a generality relation on characteristic rules. We propose CharacteriX, a level-wise\textsuperscript{1} algorithm for mining interesting characteristic rules. CharacteriX takes into account the properties of the target objects but also the properties of the objects linked to the target objects. It is achieved in the algorithm by starting with the most general Quantified Paths, exploring the search space, according to the notion of generality between rules. Moreover, it uses two heuristics, link-coverage and open-coverage to prune efficiently the search space. Another important feature of our approach is the form of the rules, which relies on both quantified paths defining how to 'navigate' between sets of objects, but also on the properties. As far as we now the form of rules we have introduced has not yet been used in that field.

The paper is organized as follows. Section 2 formalizes the problem of mining characteristic rules. In Section 3, we give some definitions on which our approach relies: the notion of quantified paths, properties and characteristic rules, the notion of coverage and generality orders. Section 4 is devoted to the general algorithm and Section 5 to experiments.

2 Problem Statement

We consider that we have knowledge about objects that are typed and relationships between them. We consider also a subset of objects with the same type, called in the following the target set. We aim at mining characteristic rules describing this target set, taking into account the properties of the target objects but also the properties of the objects in relations with them. More formally, let $E$ be a set of objects, $E = E_1 \cup E_2 \cdots \cup E_n$, where each $E_i$ represents a set of objects with the same type $T_i$. A set of attributes is defined for each type of objects, and objects are described by attribute-value pairs. Let $R$ be a set of binary relations. In the following, $r_{ij}$ denotes a binary relation on $E_i \times E_j$.

The characterization task we are interested in can now be formulated as follows:

- given a set $E$ of objects, $E = E_1 \cup E_2 \cdots \cup E_n$, where each $E_i$ contains objects with the same type $T_i$,
- given a set $R$ of binary relations,
- given a target set $E_{\text{target}}$, such that there exists $i$, $E_{\text{target}} \subseteq E_i$,

$\rightarrow$ find a set of characterization rules of $E_{\text{target}}$.

The size of the search space for the characterization rules depends, among others, on the number of relations in $R$ and on their cardinalities. Without restrictions on the possible forms of the rule, the search space may become so large that the learning task is intractable.

**Example 1 Application to relational databases**

Our approach is illustrated throughout this paper by a running example $Movies$\textsuperscript{2} given in figure 1. This database is stored in a relational form composed of several files. There is information on actors, casts, directors, producers, studios, etc. The main file $Movie$ is a list of movies described

\textsuperscript{1}see [9, 11] for a complete description of level-wise algorithms family.

\textsuperscript{2}inspired from http://kdd.ics.uci.edu/databases/movies/movies.html
by their category, title, year, process etc. The actors for those movies are listed with their roles in another file \textit{Casts}. More information about individual actors such as name, date of birth, gender and origin can be found in the file \textit{Actors}. The file \textit{People} gives more information about actors, directors, producers, writers, and cinematographers. \textit{Remakes} links movies to their remakes, whereas \textit{Awards} gives the different awards that can be won by a movie. Finally, \textit{Studios} provides some information about each studio, such as the location and the founder.

For instance, one would like to characterize the properties of comic movies or movies produced by a given producer, etc.

3 General Framework

3.1 Quantified Path

\textbf{Definition 1} The relationships between objects define a set of possible \textbf{Quantified Paths} (denoted in the following by QP). A QP on $X_0$ is a formula:

$$Q_1 X_1 \ldots Q_n X_n$$

where $n \geq 0$, $X_0$ is the type of the target objects, and for each $i \neq 0$, $Q_i = \forall$ or $\exists$, $X_i$ is a type of objects, and there exists a relationship in $\mathcal{R}$ between $X_{i-1}$ and $X_i$. When necessary, in order to remember the target set, it will be denoted by $X_0 : Q_1 X_1 \ldots Q_n X_n$.

Let us notice that when there exists several relationships between $X_{i-1}$ and $X_i$, the quantifier $Q_i$ may be indexed by the relation used in the QP.

A QP has a size $n$, which is the number of its quantifiers.

\textbf{Example 2} • Links between movies (M) and awards (W) give two paths denoted by $M : \forall W$ and $M : \exists W$. $M : \forall W$ means "all awards of each movie", while $M : \exists W$ stands for "for at least one award of each movie".

• $P_{name=Hitchcock} : \forall M \forall W$ is another path, where $P_{name=Hitchcock}$ is a target set of people (P). This path means that we are interested in all awards of all Hitchcock's movies.

\textbf{Definition 2} We say that two quantified paths are variants if they have the same size, if they involve the same type of objects, the same relations in the same order and if they differ by at least a quantifier.

\textbf{Example 3} If we consider people (P) as a target set and links between people and movies (M), we have the four following paths: $P : \forall M \forall W$, $P : \forall M \exists W$, $P : \exists M \exists W$, $P : \exists M \forall W$. These QPs are variants of size 2.
Definition 3 We say that a quantified path \( \delta_1 \) is more general than another quantified path \( \delta_2 \) (denoted by \( \delta_1 \succeq \delta_2 \)) iff \( \delta_1 \) and \( \delta_2 \) are variants and for \( 1 \leq i \leq \text{size}(\delta_1) (= \text{size}(\delta_2)) \), either:
- \( Q^1_i \equiv Q^2_i \), or
- \( Q^1_i = \exists \) and \( Q^2_i = \forall \).

Example 4 For instance, we have \( P : \exists M \exists W \geq P : \forall M \forall W \) and also \( P : \exists M \exists W \geq P : \forall M \forall W \) but \( P : \forall M \forall W \not\geq P : \exists M \forall W \) and \( P : \forall M \forall W \not\geq P : \forall M \exists W \).

3.2 Properties
A set of properties is associated to each type of objects. It can be defined as a feature or a characteristic description that can be given or not to an object. For a type \( T \) and a property \( p \) on \( T \), we assume that there exists a boolean function \( \forall_p \), such that for each object \( o \) of type \( T \), \( \forall_p(o) = \text{true} \) or \( \forall_p(o) = \text{false} \). It means that a property may be satisfied by an object \( o \) or not. We have many kinds of properties such as:
- attribute=value
- attribute \( \in \) \{value_1, ..., value_n\}
- attribute \( \leq \) value, attribute \( \leq \) value, ... 
- Aggregates, such as: count, min, max, ...

Definition 4 We define two basic properties True and False such that for any object \( o \):
\[ \forall_{\text{True}}(o) = \text{true} \] and \[ \forall_{\text{False}}(o) = \text{false} \].

Example 5 Let origin=USA be a property on the relation Actors, we have for example:
\[ \forall_{\text{origin=USA}}(Paul Newman) = \text{true} \].

Definition 5 We say that a property \( p_1 \) is more general than a property \( p_2 \) (denoted by \( p_1 \succeq p_2 \)) iff all objects that satisfy the property \( p_2 \) also verify the property \( p_1 \).

Example 6 The property \( W\.kind \in \{\text{Oscar, Golden Palm}\} \) where \( W \) represents the set of awards is more general than \( W\.kind \in \{\text{Golden Palm}\} \).

3.3 Characteristic Rules
Definition 6 We define a characteristic rule on a target set \( X_0 \) as the conjunction of a quantified path \( \delta \) and a property \( p \), denoted by:
\[ X_0 : \delta \Rightarrow p \]

Definition 7 We say that two characteristic rules (on the same target set \( T \)) \( r_1 (T: \delta_1:p_1) \) and \( r_2 (T: \delta_2:p_2) \) are variants if \( \delta_1 \) and \( \delta_2 \) are variants and \( p_1 \equiv p_2 \).

Example 7 \( P_{\text{name=Hitchcock}} : \forall M : M\.category = \text{Suspense} \) is a characteristic rule, where \( P_{\text{name=Hitchcock}} \) is a target set of People such that the attribute name is Hitchcock. This rule means that all Hitchcock’s movies belong to the Suspense category.

3.4 Coverage
The notion of coverage is defined for a property \( p \) relatively to a quantified path \( \delta \). It measures the number of objects that have this property. For a rule \( r = X_0 : \delta \Rightarrow p \) and an object \( o \), we define \( \forall_r(o) \) recursively as follows:
- \( \forall_{\forall X:_\delta:p}(o) = \forall_{\forall X:p}(o) \) or \( \text{false} \) if there is no object linked to \( o \)
- \( \forall_{\exists X:_\delta:p}(o) = \forall_{\exists X:p}(o) \lor \forall_{\exists X:p}(o) \) or \( \text{false} \) if there is no object linked to \( o \)
- \( \forall_{\exists^*:_\delta:p}(o) = \forall_p(o) \), that is \( \text{true} \) intuitively if \( o \) has the property \( p \), \( \text{false} \) otherwise.
Where \( o_1, ..., o_n \) are the objects of type \( X \) linked to the object \( o \), and \( \delta^0 \) is the path of size 0.
Example 8 Let us consider the rule:
\[ r = P_D : \forall \text{M} \exists W :: w \cdot \text{kind} \in \{\text{Oscar, Goldenpalm}\}. \] 
\[ P_D \] denotes the directors in the relation people.
\[ \forall \text{M} \exists W :: w \cdot \text{kind} \in \{\text{Oscar, Goldenpalm}\}(\text{Spielberg}) = \]
\[ \forall \text{M} :: w \cdot \text{kind} \in \{\text{Oscar, Goldenpalm}\}(\text{film}_1) \land \ldots \land \forall \text{M} :: w \cdot \text{kind} \in \{\text{Oscar, Goldenpalm}\}(\text{film}_m) \]
where film_1, ..., film_m denote the movies directed by Spielberg.
Coverage is given by the following:
\[ \text{coverage}(r, E_{\text{target}}) = \frac{|\{o \in E_{\text{target}} \text{ and } v_r(o) = \text{true}\}|}{|E_{\text{target}}|} \]

Example 9 Let us consider all the movies as the target set. The coverage of the rule:
\[ M : \exists A :: A \cdot \text{gender} = \text{female} \text{ is equal to } 2526, \text{ where 2526 is the number of movies with female actors and } 11404 \text{ is the total number of movies. In the same way, we can calculate } \]
\[ \text{coverage}(M : \exists A :: A \cdot \text{gender} = \text{animal, movies}) = \frac{2526}{11404}. \]
The complexity for evaluating the coverage of a rule partly depends on the form of the path of the rule. It can be done, for example, by translating them into SQL queries.

3.5 Generality order

Definition 8 We say that a characteristic rule \( r_1 (\delta_1 :: p_1) \) is more general than another rule \( r_2 (\delta_2 :: p_2) \) (denoted by \( r_1 \preceq r_2 \)) if \( \delta_1 \preceq \delta_2 \) and \( p_1 \preceq p_2 \).

Example 10 \( M(S) : \exists W :: W \cdot \text{kind in} (\text{Oscar, Golden-Palm}) \preceq M(S) : \forall W :: W \cdot \text{kind in} (\text{Oscar}) \).

Lemma 1 Coverage is monotone with respect to the generality order, that is to say:
if \( \text{coverage}(r_2, E_{\text{target}}) \geq \epsilon \) and \( r_1 \preceq r_2 \) then \( \text{coverage}(r_1, E_{\text{target}}) \geq \epsilon \)
which is equivalent to:
if \( -\text{coverage}(r_1, E_{\text{target}}) \geq \epsilon \) and \( r_1 \preceq r_2 \) then \( -\text{coverage}(r_2, E_{\text{target}}) \geq \epsilon \).

3.6 Specialization Operator

Definition 9 We define the specialization operator \( \rho \) as a binary relation on the set of characteristic rules as follows:
\[ \rho(\delta :: p) = \{\delta :: p' | \delta \text{ differ from } \delta \text{ by one } \exists \text{ quantifier set to } \forall \} \cup \{\delta :: p | p \preceq p' \text{ and there is no } p'' \text{ s.t. } p \preceq p' \preceq p''\} \]
Let us notice that for all \( r'' \in \rho(r), \) there is no \( r' \notin \rho(r) \) such that \( r \preceq r' \) and \( r' \preceq r'' \).

Example 11 Suppose that we consider only the following properties for Actors:
\{Actor.gender = male, Actor.gender = female\}, and Movies as a target set. The complete search space starting with \( \exists A :: \text{True} \) is given in the figure 2.

The definition of a specialization operator allows to define a top down, levelwise, search strategy, for mining characteristic rules.

For pruning the search space, we define two notions, open-coverage and link-coverage.

3.7 Open-Coverage

We denote by:
\[ \text{open-coverage}(\delta :: p, E_{\text{target}}) = \text{coverage}(\text{open}(\delta) :: p, E_{\text{target}}) \]
where \( \text{open}(\delta) \) is obtained by setting all the quantifiers of \( \delta \) to \( \exists \). Intuitively, open-coverage means that there is at least one object linked to the target objects that have the properties.

3.8 Link-Coverage

We denote by:
\[ \text{link-coverage}(\delta :: p, E_{\text{target}}) = \text{coverage}(\text{open}(\delta) :: \text{True}, E_{\text{target}}) \]
Intuitively, link coverage measures whether there is enough objects linked to the objects of the target set to consider properties on these objects. This can be used when there is a 0..* relation, which means that some objects can be linked to none objects by this relation. Note that a small link-coverage denotes the absence of related objects, and this can be very informative.
3.9 Interesting Characteristic Rules

For a rule $\delta \models p$, coverage measures the number of objects in the target set having the property $p$. We would like to estimate whether this property is really characteristic of $\mathcal{E}_{target}$ or not. This can be achieved by verifying if the property covers enough objects in the target set, while covering few objects outside the target set. One should find a trade-off between these two conditions and estimate the quality of rules by using some heuristics measures that can filter the discovered rules.

Furthermore, in descriptive data mining tasks, such as characterization, thousands of rules may be discovered, so making the rule filtering step as a necessary post processing step. In our framework, we define a function named Interesting that can filter the rules relying on such heuristics in order to keep only interesting ones.

In [8], Lavrač et al. analyze some rule evaluation measures used in machine learning and knowledge discovery. They propose only one measure that can be considered as a measure of novelty, precision, accuracy, negative reliability, or sensitivity. In our experiments, we used their novelty measure: the novelty of a rule $r$ of the form $H \leftarrow B$ is given by:

$$Novelty(r) = P(HB) - P(H) * P(B)$$

where $P$ designates a probability.

For a characteristic rule $r$, for each object $o \in \mathcal{E}$, we can consider the facts $o \in \mathcal{E}_{target}$ and $\forall_r(o) = true$. We are looking for a strong association between these two facts. This one can be estimated by the novelty measure. If we express the characteristic rule by $o \in \mathcal{E}_{target} \leftarrow \forall_r(o) = true$, then, the novelty of this rule can be estimated by:

$$Novelty(r) = \frac{|\{o|o \in \mathcal{E}_{target} \land \forall_r(o) = true\}|}{|\mathcal{E}|} - \frac{|\mathcal{E}_{target}|}{|\mathcal{E}|} \cdot \frac{|\{o|o \in \mathcal{E} \land \forall_r(o) = true\}|}{|\mathcal{E}|}$$

According to this definition, we have $-0.25 \leq Novelty(r) \leq 0.25$. A strongly positive value indicates a strong association between the two facts, while a strongly negative value indicates a strong association between 'being an object having the property' and 'do not belong to the target set'.

**Function Interesting** $(r, \mathcal{E}_{target})$: boolean

If Novelty$(r) \rightarrow 0.25$ then return True
else return False

We can also use other measures such as entropy, purity, or Laplace estimate. See [4] for more details. In addition to the novelty we used in our experiments the Laplace estimate given by:

$$Laplace(r) = \frac{\text{coverage}(r, \mathcal{E}_{target}) + 1}{\text{coverage}(r, \mathcal{E}_{target}) + \text{coverage}(r, \mathcal{E} - \mathcal{E}_{target}) + 2}$$

Figure 2: Search space example starting with the rule Target Movies: $\exists A:: True$
0 ≤ Laplace(r) ≤ 1, if a rule covers no examples, then Laplace will be 0.5.

4 Algorithm

We can use a variant of the levelwise algorithm [9] for mining all potentially interesting characteristic rules.

<table>
<thead>
<tr>
<th>Character Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input</strong> C_1 = {r, such that there is no r’, r’ is more general than r }</td>
</tr>
<tr>
<td>i = 1</td>
</tr>
<tr>
<td><strong>while</strong> ( C_i \neq \emptyset )</td>
</tr>
<tr>
<td>1. ( F_i = {r \in C_i</td>
</tr>
<tr>
<td>2. ( F'_i = {r \in F_i</td>
</tr>
<tr>
<td>3. ( F''_i = {r \in F'_i</td>
</tr>
<tr>
<td>4. ( C_{i+1} = (\bigcup \rho(r)</td>
</tr>
<tr>
<td>5. ( i = i + 1 )</td>
</tr>
<tr>
<td><strong>end while</strong></td>
</tr>
<tr>
<td><strong>output</strong> ( {r \in \bigcup_{j \leq i} F''_j</td>
</tr>
</tbody>
</table>

**Character** starts with \( C_1 \), the set of the most general characteristic rules given by the user, i.e. rules \( r \) such that there is no \( r' \) more general than \( r \). The algorithm then iterates alternating at each iteration, coverage tests (lines 1, 2, 3) and generation of next candidates rules (line 4), taking care to discard previously considered rules. These latter are obtained by specializing the rules of the previous step covered by enough objects. The iterations stop when it is not possible to generate further candidates rules. Pruning heuristics, link-coverage (line 1) and open-coverage (line 2) are used to reduce the number of evaluation coverage done in line 3, by constructing progressively the sets \( F_i, F'_i \) et \( F''_i \). Open-coverage and, a fortiori, link-coverage are the same for several rules. They are stored and retrieved as needed to avoid unnecessary computations. Let us notice that these pruning strategies do not exclude interesting characteristic rules but avoid testing coverage for rules that have certainly low coverages according to \( \epsilon \). The algorithm then outputs the set of all interesting rules according to a set of given measures.

**Lemma 2** Character is correct and complete.

**Proof 1** The proof relies on the following inequality:

\( \text{link-coverage}(r, \mathcal{E}_{\text{target}}) \geq \text{open-coverage}(r, \mathcal{E}_{\text{target}}) \geq \text{coverage}(r, \mathcal{E}_{\text{target}}). \)

5 Experiments

We have experimented our approach on a real geographic database. For this purpose, we have extended our framework in order to take into account the spatial dimension, mainly the topological and distance information between geographic objects. In our experiments we have used a GIS [2], which handles many layers: geographic, geologic, seismic, volcanic, mineralogic, gravimetric,… These layers store more than 70 thousands geographic objects.

We aim at finding characterization rules for characterizing mineral ore deposits using geological information, faults, volcanoes … This task can be stated as follows:

- given a set \( \mathcal{E} \) of geographic objects, \( \mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3 \cup \mathcal{E}_4 \cup \mathcal{E}_5 \), where \( \mathcal{E}_1 \) contains mineral deposits, \( \mathcal{E}_2 \) represents the geology, \( \mathcal{E}_3 \) the volcanoes, \( \mathcal{E}_4 \) the faults and \( \mathcal{E}_5 \) the seisms;

- given a set \( \mathcal{R} \) of binary relations based on spatial proximity;

- given a target set \( \mathcal{E}_{\text{target}} = \{\text{gold mines}\} \subseteq \mathcal{E}_1 \);

→ find a set of characterization rules of \{gold mines\}
To take into account the distance between objects, we introduce a new parameter $\lambda$ and $r^\lambda_{ij}$ represents a binary relation between objects in $E_i$ and objects in $E_j$ parameterized by $\lambda$. In the case of geographic objects, this parameter may denote the distance between objects. For instance $r^i_{10km}$ represents binary relations between mineral deposits and volcanoes at a distance less or equal to 100 kilometers.

As a consequence, the notion of quantified path described in section 3.1 has been extended, considering the parameter $\lambda$ used in binary relations.

$$\delta_{\lambda_1, \lambda_2} = M : \forall \lambda_1 \exists \lambda_2 V,$$

where $M$ denotes mineral deposits, $F$ faults and $V$ volcanoes. For example:

$$\delta_{10km, 5km} = M : \forall_{10km} F \forall_{5km} V$$

denotes all the volcanoes at less than 5 kilometers than faults at less than 10 kilometers than each mine. In order to handle distance information between objects, we construct growing buffers around target objects progressively, while checking for the properties satisfied by objects entering into the buffers. This notion is illustrated by Figure 3, where buffers are constructed around mineral deposits.

The Quantified path generality ordering defined in section 3.1 can be used with such parametrized quantified paths. In fact, in the case of characteristic rules with one parameter, we have $\delta \lambda \succeq \delta \lambda'$ if ($\lambda \geq \lambda'$ and $\lambda$ indexes a $\exists$) or ($\lambda \leq \lambda'$ and $\lambda$ indexes a $\forall$). We have:

$$M : \forall_{3km} F \succeq M : \forall_{5km} F \succeq M : \forall_{10km} F$$

Intuitively, this means that if a property holds for all faults at a distance less than 10km from a mine, then this property also holds for all faults at less than 5km and 3km from this mine. Vice versa, if there exists a fault at less than 3km from a mine with a given property, than there exists a fault at less than 5km and less than 10km with the same property.

Let us notice that in the case of a quantified path with one parameter, the order induced by the relation $\succeq$ is a total order, when $\lambda$ varies. When we have more than one parameter, we can induce a partial order, by taking into account the relation $\delta_{\lambda_1, \ldots, \lambda_n} \succeq \delta_{\lambda'_1, \ldots, \lambda'_n}$ if $\forall i, (\lambda_i \geq \lambda'_i$ and $\lambda_i, \lambda'_i$ indexes a $\exists$) or ($\lambda_i \leq \lambda'_i$ and $\lambda_i, \lambda'_i$ indexes a $\forall$).

5.1 Results

Our system tested hundreds of rules, table 1 gives some examples.

The following rule has been discovered (among others) and covers 60% of gold mines and rejects most of the other mines.
<table>
<thead>
<tr>
<th>Rule</th>
<th>Coverage</th>
<th>Laplace</th>
<th>Novelty</th>
</tr>
</thead>
<tbody>
<tr>
<td>M: M.Era ∈ Mesozoic, Cretaceous</td>
<td>6.59%</td>
<td>0.144</td>
<td>-0.0133</td>
</tr>
<tr>
<td>M: M.Era ∈ Mesozoic, Jurassic, Cretaceous</td>
<td>6.42%</td>
<td>0.266</td>
<td>0.0102</td>
</tr>
<tr>
<td>M: M.Lithology = sedimentary deposits</td>
<td>5.50%</td>
<td>0.070</td>
<td>-0.0413</td>
</tr>
<tr>
<td>M: M.Lithology = volcanic deposits</td>
<td>64.22%</td>
<td>0.365</td>
<td>0.0529</td>
</tr>
<tr>
<td>M: M.DistanceBenioff ∈ [170,175]</td>
<td>66.97%</td>
<td>0.259</td>
<td>0.0086</td>
</tr>
<tr>
<td>M: ∃15km G::G.Age = tertiary</td>
<td>86.24%</td>
<td>0.310</td>
<td>0.0069</td>
</tr>
<tr>
<td>M: ∃15km V::V.Age = recent</td>
<td>7.34%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Some examples of tested rules

This rule, considered as interesting by experts, expresses that for all gold mines, there exists a tertiary volcanic geology at a distance less than 10 km from this mine, and these mines are epithermal ones with a morphology of veins and are at a benioff depth between 75 and 150 m and at a slope benioff of 8° and 16°. According to geologist experts, this rule is interesting because it is related to a natural phenomenon: the plate tectonics.

Figure 4 illustrates the notion of link-coverage and represents the number of gold mines that contain at least a fault in a buffer of size A around the mine and such that the fault contains at least a volcano in a buffer of size B around this fault.

6 Conclusion

In this paper, we have presented a new general approach for mining a new kind of characteristic rules in a target set of objects. These rules handle both properties and quantified paths. These latters specify how to take into account different kinds of objects and their relationships, in other words, how to go from objects to others without flattening the tables describing these objects. We propose CharacterX, a level-wise algorithm exploring the search space looking for characteristic rules. It takes into account a generality relation between rules i.e. between quantified paths, but also between properties. Moreover, the notions of link-coverage and open-coverage are useful heuristics to prune the search space. We have experimented our approach on a geographic databases and we have submitted our rules to geologists. They considered that these rules are interesting and give a good description of a set of chosen target objects. Quantified paths give a convivial way to look for the characteristics of the target objects according to the spatially linked objects. In the future, we
aim at extending our framework on other kinds of databases, such as object oriented databases.

References


