Learning Characteristic Rules Relying on Quantified Paths

Teddy Turmeaux[1], Ansaf Salleb[1], Christel Vrain[1], and Daniel Cassard[2]

[1] LIFO, Université d'Orléans, rue Léonard de Vinci
BP 6759, F-45067 Orléans Cedex 2, France
{Turmeaux, Salleb, Vrain}@lifo.univ-orleans.fr
[2] BRGM, 3, avenue Claude Guillemin, B.P. 6009
Orléans cedex 2, France
d.cassard@brgm.fr

Abstract. In this paper, we address the *characterization* task and 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 CaracteriX, 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 set of objects, called the *target set*. It consists in discovering properties that characterize these objects, 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 [12]. Characterizing a set of objects has also been considered as computing the least general generalization (l.g.g.) in Inductive Logic Programming [14], but such an approach leads to complexity problems. An object oriented view for computing the l.g.g. called *structural matching* has been proposed in [8, 17] and applied to air traffic control in [9]. 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, 11, 5, 16] , 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:

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

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

At LIFO, we have developed $\mathfrak{Caracteri}\mathfrak{X}$, a levelwise[1] algorithm, for mining interesting characteristic rules. It starts with the most general Quantified Paths, exploring the search space, according to notion of generality between rules. Moreover, it uses two heuristics, *link-coverage* and *open-coverage*, to efficiently prune the search space. Another important feature of our approach is the form of the rules, which relies on quantified paths defining how to 'navigate' between sets of objects. As far as we know 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 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

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

- given a set of types T_i, and attributes for describing objects of type T_i,
- given a set $\mathcal{E}$ of objects, $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cdots \cup \mathcal{E}_n$, where each $\mathcal{E}_i$ contains objects with the same type T_i,

[1] see [11, 13, 15] for a description of levelwise algorithms family.

- given a set $\mathcal{R}$ of binary relations (in the following, r_{ij} denotes a binary relation on $\mathcal{E}_i \times \mathcal{E}_j$)
- given a target set $\mathcal{E}_{target}$, such that there exists i, $\mathcal{E}_{target} \subseteq \mathcal{E}_i$,

$\rightarrow$ find a set of characterization rules of $\mathcal{E}_{target}$.

The size of the search space for the characterization rules depends, among others, on the number of relations in $\mathcal{R}$ 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

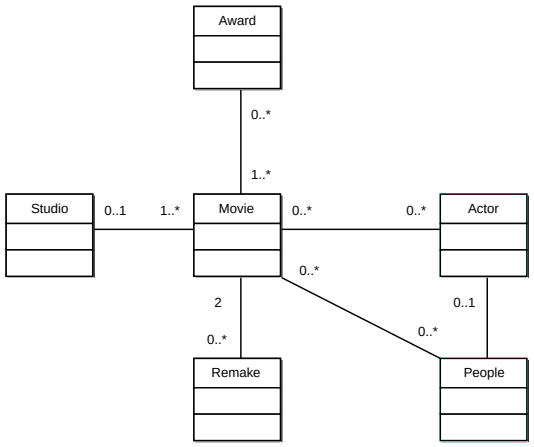

Fig. 1. Movies database

Our approach is illustrated throughout this paper by a running example *Movies*[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,... The main file *Movie* is a list of movies described by their category, title, year, process, and so on. The actors are listed with their roles in another file *Casts*. More information about individual actors such as *name, date of birth, gender* and *origin* can be found in the file *Actors*. The file *People* gives more information about actors, directors, producers, writers, and cinematographers. *Remakes* links movies to their remakes, whereas *Awards* gives the different awards that can be won by a movie. Finally, *Studios* provides some information about each studio, such as the *location* and the *founder*.

For instance, we could be interested by characterizing the properties of comic movies, or the properties of movies produced by a given producer, and so on.

[2] inspired from `http://kdd.ics.uci.edu/databases/movies/movies.html`

3 General Framework

3.1 Quantified Path

Definition 1. *A* **Quantified Path** *(denoted in the following by $\mathcal{QP}$) on X_0 is a formula:*

$$Q_1\ X_1 \ldots Q_n\ X_n$$

where $n \geq 0$, X_0 represents 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 prefixed by X_0 leading to $X_0 : Q_1\ X_1 \ldots Q_n\ X_n$.
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 $\mathcal{QP}$.
A $\mathcal{QP}$ has a size n that is the number of its quantifiers.

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

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.*

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 $\mathcal{QP}s$ are variants of size 2.

Definition 3. *We say that a quantified path δ_1 is more general than a quantified path δ_2 (denoted by $\delta_1 \succeq \delta_2$) iff δ_1 and δ_2 are variants and for $1 \leq i \leq size(\delta_1)(= size(\delta_2))$, either :*
- $Q_i^1 \equiv Q_i^2$, or
- $Q_i^1 = \exists$ and $Q_i^2 = \forall$.

Example 4. For instance, we have $P : \exists M \exists W \succeq P : \forall M \exists W \succeq P : \forall M \forall W$ and also $P : \exists M \exists W \succeq P : \exists M \forall W \succeq P : \forall M \forall W$ but $P : \forall M \exists W \not\succeq P : \exists M \forall W$ and $P : \exists M \forall W \not\succeq P : \forall M \exists W$.

3.2 Properties

A set of properties is associated to each type of objects. We consider many kinds of properties such as: *attribute=value*, *attribute* $\in \{value_1, \ldots, value_n\}$, *attribute* $\geq$ *value*, *attribute* $\leq$ *value*, and even aggregates such as: count, min, max, ... For a type T and a property p on T, we assume that there exists a boolean function $\mathcal{V}_p$, such that for each object o of type T, $\mathcal{V}_p(o) = true$ or $\mathcal{V}_p(o) = false$. It means that a property may be satisfied by an object o or not.

Definition 4. *We define two basic properties $True$ and $False$ such that for any object o, $\mathcal{V}_{True}(o) = true$ and $\mathcal{V}_{False}(o) = false$.*

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 5. The property $W.kind \in \{Oscar, GoldenPalm\}$ where W represents the set of awards is more general than $W.kind \in \{GoldenPalm\}$.

3.3 Characteristic Rules

Definition 6. *We define a characteristic rule on a target set X_0 as the conjunction of a quantified path δ and a property* p *, denoted by: $X_0 : \delta :: p$.*

Definition 7. *We say that two characteristic rules r_1 ($T : \delta_1 :: p_1$) and r_2 ($T : \delta_2 :: p_2$) are variants if δ_1 and δ_2 are variants and $p_1 \equiv p_2$.*

Example 6. $P_{name=Hit} : \forall M :: M.category = Suspense$
is a characteristic rule, where $P_{name=Hit}$ is a target set of *People* whose name is Hit. This rule means that all Hit'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 δ. It measures the number of objects that have this property. For a rule $r = X_0 : \delta :: p$ and an object $o \in X_o$, we define $\mathcal{V}_{\delta::p}(o)$ recursively as follows:
- $\mathcal{V}_{\forall X.\delta'::p}(o) = \mathcal{V}_{\delta'::p}(o_1) \wedge \cdots \wedge \mathcal{V}_{\delta'::p}(o_n)$ or $false$ if there is no object linked to o
- $\mathcal{V}_{\exists X.\delta'::p}(o) = \mathcal{V}_{\delta'::p}(o_1) \vee \cdots \vee \mathcal{V}_{\delta'::p}(o_n)$ or $false$ if there is no object linked to o
- $\mathcal{V}_{\delta^\emptyset::p}(o) = \mathcal{V}_p(o)$, that is $true$ if o has the property p, $false$ otherwise.

Where $o_1, \ldots o_n$ are the objects of type X linked to the object o, and $\delta^\emptyset$ is the empty path (size 0).

Example 7. Let us consider the rule: $P_D : \forall M \exists W :: w.kind \in$ {Oscar, Golden palm}, where P_D denotes the directors in the relation people.
$\mathcal{V}_{\forall M \exists W::w.kind \in \{Oscar, Goldenpalm\}}(Sp) =$
$\mathcal{V}_{\exists W::w.kind \in \{Oscar, Goldenpalm\}}(film_1) \wedge \ldots \wedge \mathcal{V}_{\exists W::w.kind \in \{Oscar, Goldenpalm\}}(film_m)$
where $film_1, \ldots, film_m$ denote the movies directed by Sp.

Definition 8. *Coverage is given by the following:*

$$coverage(r, \mathcal{E}_{target}) = \frac{|\{o | o \in \mathcal{E}_{target} \text{ and } v_r(o) = true\}|}{|\mathcal{E}_{target}|}$$

Example 8. Let us consider all the movies as the target set. The coverage of the rule $M : \exists A :: A.gender = female$ is equal to $\frac{2526}{11404}$, where 2526 is the number of movies with female actors and 11404 is the total number of movies. In the same way, we can calculate coverage($M : \exists A :: A.gender = animal, movies$)=$\frac{16}{11404}$.

3.5 Generality order

Definition 9. *We say that a characteristic rule r_1 (δ_1::p_1) is more general than a rule r_2 (δ_2::p_2) (denoted by $r_1 \succeq r_2$) iff $\delta_1 \succeq \delta_2$ and $p_1 \succeq p_2$. We write $r_1 \succ r_2$, when $r_1 \succeq r_2$ and $\neg(r_2 \succeq r_1)$.*

Example 9. $M : \exists W :: W.kind\ in(Oscar, Golden\text{-}Palm) \succeq M : \forall W :: W.kind\ in(Oscar)$.

Lemma 1. *Coverage is monotone with respect to the generality order, i.e., if $coverage(r_2, \mathcal{E}_{target}) \geq \epsilon$ and $r_1 \succeq r_2$ then $coverage(r_1, \mathcal{E}_{target}) \geq \epsilon$, or else if $\neg(coverage(r_1, \mathcal{E}_{target}) \geq \epsilon)$ and $r_1 \succeq r_2$ then $\neg(coverage(r_2, \mathcal{E}_{target}) \geq \epsilon)$.*

3.6 Specialization Operator

Definition 10. *We define the specialization operator ρ as a binary relation on the set of characteristic rules as follows:*

$\rho(\delta :: p) = \{\delta'$::$p | \delta'$ differs from δ by one $\exists$ quantifier set to $\forall\} \cup \{\delta$::$p' | p \succ p'$ and there is no p'' s.t. $p \succ p'' \succ p'\}$

Let us notice that for all $r'' \in \rho(r)$, there is no $r' \notin \rho(r)$ such that $r \succ r'$ and $r' \succ r''$.

Example 10. Suppose that we consider only the following properties for Actors: $\{Actor.gender = male, Actor.gender = female\}$, and Movies as the target set. The complete search space starting with $\exists A :: True$ is given in Figure 2.

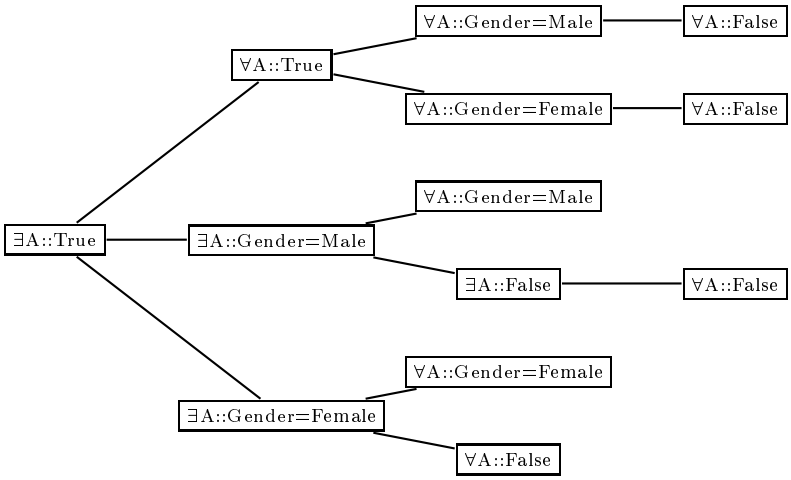

Fig. 2. Search space starting with the rule Movies: ∃ A::True

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 Link-Coverage

We define *link-coverage* $(\delta{::}p, \mathcal{E}_{target})$ =coverage(open(δ)$::True, \mathcal{E}_{target}$) Intuitively, link coverage measures the number of target objects for which there exists at least an object linked to them through δ. This can be useful when there is a $0..*$ relation, which means that some objects can be linked to none objects by this relation.

3.8 Open-Coverage

We define *open-coverage* $(\delta{::}p, \mathcal{E}_{target}) = coverage(open(\delta){::}p, \mathcal{E}_{target})$ where $open(\delta)$ is obtained by setting all the quantifiers of δ to $\exists$. Intuitively, open-coverage counts the number of target objects for which there is at least an object linked to them by δ and satisfying p.

3.9 Interesting Characteristic Rules

For a rule $\delta :: p$, 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.

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 [10], Lavrač et al. analyze some rule evaluation measures used in Machine Learning and Knowledge Discovery. They propose only a 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 $H \longleftarrow B$ is given by: (P represents a probability)

$$Novelty(H \longleftarrow B) = P(HB) - P(H) * P(B)$$

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

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

According to [10], we have $-0.25 \leq$*Novelty(r)*≤ 0.25. A strongly positive value indicates a strong association between the two facts.

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:

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

$0 \leq Laplace(r) \leq 1$. If a rule covers no examples, then Laplace is equal to 0.5.

4 Algorithm

We can use a variant of the *levelwise algorithm* [11] for mining all potentially interesting characteristic rules.

CaracteriX Algorithm
input $\mathcal{C}_1 = \{r$, such that there is no r', $r' \succ r\ \}$
$i = 1$
while $\mathcal{C}_i \neq \emptyset$
1. $\mathcal{F}_i = \{r \in \mathcal{C}_i | link\text{-}coverage(r, \mathcal{E}_{target}) \geq \epsilon\}$
2. $\mathcal{F'}_i = \{r \in \mathcal{F}_i | open\text{-}coverage(r, \mathcal{E}_{target}) \geq \epsilon\}$
3. $\mathcal{F''}_i = \{r \in \mathcal{F'}_i | coverage(r, \mathcal{E}_{target}) \geq \epsilon\}$
4. $\mathcal{C}_{i+1} = (\bigcup \rho(r) | r \in \mathcal{F''}_i) \setminus \bigcup_{j \leq i} \mathcal{C}_j$
5. $i = i + 1$

end while
output $\{r \in \bigcup_{j<i} \mathcal{F''}_j | Interesting(r, \mathcal{E}_{target})\}$

CaracteriX starts with $\mathcal{C}_1$, the set of the most general characteristic rules given by the user. The algorithm then iterates coverage tests (lines 1,2,3) and generation of next candidate rules (line 4), taking care to discard previously considered rules. The iteration stops when it is not possible to generate further candidate rules. Pruning heuristics, link-coverage (line 1) and open-coverage (line 2) are used to reduce the number of coverage evaluations done in line 3. Open-coverage and, *a fortiori*, link-coverage are the same for variant rules. They are stored and retrieved as needed to avoid unnecessary computations. Let us notice that these pruning strategies only exclude characteristic rules that do not fulfill the minimum coverage requirement ϵ. The algorithm then outputs the set of all interesting rules.

Lemma 2. CaracteriX *is correct and complete w.r.t.* $\mathcal{C}_1$.

Proof. The proof relies on the following inequality: $link\text{-}coverage(r, \mathcal{E}_{target}) \geq open\text{-}coverage(r, \mathcal{E}_{target}) \geq coverage(r, \mathcal{E}_{target})$.

5 Experiments

The model that we have proposed and the system CaracteriX have been developed by the first three authors at LIFO, and experimented on a real geographic

database provided by the BRGM[3]. The rules that have been learned have been evaluated by a geologist expert (the fourth author of the paper). 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, volcanos ... 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 $\mathcal{E}_{target} = \{gold\ mines\} \subseteq \mathcal{E}_1$;
$\rightarrow$ find a set of characterization rules of $\{gold\ mines\}$

To take into account the distance between objects, we introduce a parameter λ and r_{ij}^{λ} represents a binary relation between objects in $\mathcal{E}_i$ and objects in $\mathcal{E}_j$ parameterized by λ. In the case of geographic objects, this parameter may denote the distance between objects. For instance $r_{1,3}^{100km}$ represents a binary relation between mineral deposits and volcanoes at a distance less or equal to 100 kms. As a consequence, the notion of quantified path described in section 3.1 has been extended, considering the parameter λ used in binary relations. For instance: $M : \forall_{10km} F \forall_{5km} V$ 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 order defined in Section 3.1 can be extended to 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 λ, λ' *indexes* a $\exists$) or ($\lambda \leq \lambda'$ and λ, λ' indexes a $\forall$). We have:

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

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.

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 λ_i, λ'_i indexes a $\exists$) or ($\lambda_i \leq \lambda'_i$ and λ_i, λ'_i indexes a $\forall$).

[3] French public institution based on Earth Sciences

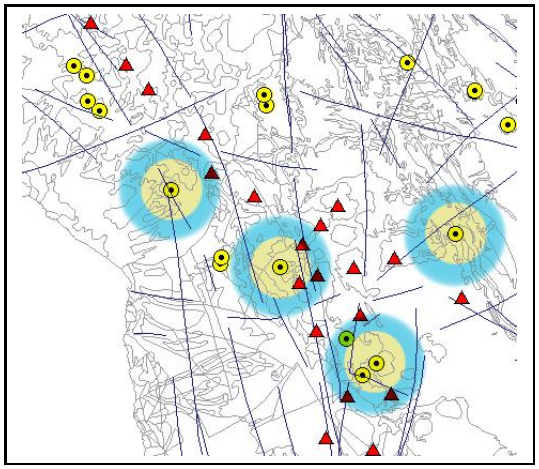

Fig. 3. Buffers around some target points in the GIS. Layers represented here are *geology, mineral deposits, fault and volcanoes*

5.1 Results

Our system tested hundreds of rules. Some examples are given Table 1.

Rule	Coverage	Laplace	Novelty
M: M.Era ∈ {Mesozoic,Cretacious}	4.59%	0,750	0,0080
M: M.Era ∈ {Mesozoic, Jurassic, Cretacious}	6,42%	0,148	-0,0133
M: M.Lithology = sedimentary deposits	5,50%	0,070	-0,0413
M: M.Lithology=volcanic deposits	64,22%	0,266	0,0102
M: M.Distance_Benioff ∈ [170..175]	66,97%	0,365	0,0529
M: $\exists_{10km}$ G::G.Age=tertiary	86,24%	0,259	0,0086
M: $\exists_{5km}$ V::V.Age=recent	7,34%	0,310	0,0030
...	...	...	...

Table 1. Some examples of tested rules

The following rule has been discovered and covers 60% of gold mines and rejects most of the other mines.

$M : \exists_{10km}$ G :: M.MainSubstance= $au\wedge$
G.CodeGeology= $TertiaryVolcanic\wedge$
M.BenioffDepth∈ [75..150]∧
M.Distance_Benioff∈ [170..275]∧
M.BenioffSlope ∈ [8°..16°]∧
G.Age= $tertiary\wedge$
M.Lithology= $volcanic\wedge$
M.Gitology= $epithermal\wedge$
M.Morphology= $veins$

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 epithemal ones with a morphology of veins and are at a benioff depth between 75 and 150 km 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 volcanoe in a buffer of size B around this fault.

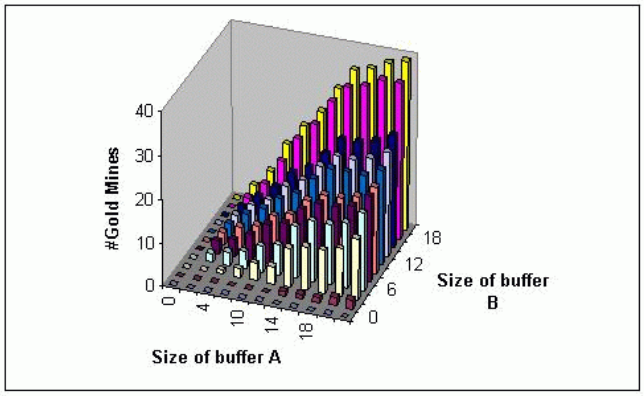

Fig. 4. Link Coverage of the rule $M : \exists_A \mathrm{F}\ \exists_B \mathrm{V}::\mathrm{True}$

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 $\mathfrak{CaracteriX}$, a levelwise algorithm exploring the search space looking for characteristic rules, taking into account a generality relation between rules. 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 database 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

1. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. *In Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 207–213, 1993.
2. D. Cassard. Gis andes: A metallogenic gis of the andes cordillera. In *4th Int. Symp. on Andean Geodynamics*, pages 147–150. IRD Paris, October 1999.
3. L. Dehaspe and L. De Raedt. Mining association rules in multiple relations. In S. Džeroski and N. Lavrač, editors, *ILP97*, volume 1297, pages 125–132. Springer-Verlag, 1997.
4. J. Furnkranz. Separate-and-conquer rule learning. Technical Report OEFAI-TR-96-25, Austrian Research Institute for Artificial Intelligence Schottengasse, 1996.
5. K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. *1st IEEE International Conference on Data Mining*, November 2001.
6. J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: An attribute-oriented approach. In Li-Yan Yuan, editor, *Proceedings of the 18th International Conference on Very Large Databases*, pages 547–559, San Francisco, U.S.A., 1992. Morgan Kaufmann Publishers.
7. J. Han and Y. Fu. Exploration of the power of attribute-oriented induction in data mining. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhr Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. AIII Press/MIT Press, 1996.
8. Y. Kodratoff and J-G. Ganascia. *Machine Learning: An Artificial Intelligence Approach*, chapter Improving the Generalization Step in Learning. Morgan Kaufmann, 1986.
9. Y. Kodratoff and C. Vrain. Acquiring first order knowledge about air traffic control. *Knowledge Acquisition Journal, B.R. Gaines & J.H. Boose, (Eds.), Academic Press Limited*, pages 353–386, 1993.
10. N. Lavrač, P. Flach, and B. Zupan. Rule evaluation measures: A unifying view. In S. Džeroski and P. Flach, editors, *ILP99*, volume 1634 of *LNAI*, pages 174–185. Springer-Verlag, 1999.
11. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
12. R. S. Michalski. A theory and methodology of inductive learning. In *Machine Learning: An Artificial Intelligence Approach*, volume 2(4), pages 83–134, 1983.
13. T.M. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence IJCAI*, pages 305–310. Cambridge MA, August 1977.
14. G.D. Plotkin. A note on inductive generalization. In *Machine Intelligence*, volume 5, pages 153–163. Edinburgh University Press, 1970.
15. L. De Raedt and S. Kramer. The level-wise version space algorithm and its application to molecular fragment finding. In Bernhard Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 853–862. Morgan Kaufmann, 2001.
16. A. Salleb, Z. Maazouzi, and C. Vrain. Mining maximal frequent itemsets by a boolean approach. In IOS Press Amsterdam F. van Harmelen, editor, *ECAI'2002*, pages 385–389, Lyon, France, 2002.
17. C. Vrain. *Machine Learning, an Artificial Intelligence Approach*, volume 3, chapter OGUST: a system which learns using domain properties expressed as theorems, pages 360–382. Morgan Kaufman publisher, 1990.