Discernibility in First Order Rule Generation

Jarosław Stepaniuk[1]

Abstract
The aim of this paper is to introduce an algorithm for finding first order logic rules. Rough set methodology is used in the process of selecting literals, which may be part of the rule. The criterion of selecting literal is as follows: only such literal is selected, which adding to the rule makes that the rule discerns the most examples from those, which were yet indiscernible.

1. Introduction

The theory of rough sets provides a powerful foundation for discovery of important regularities in data and for objects classification. In recent years numerous successful applications of rough set methods for real-life data have been developed (see e.g. [1], [2], [3]). Rough set approach has been used in a lot of applications aimed to description of concepts. In most cases only approximate descriptions of concepts can be constructed because of incomplete information about them. In learning approximations of concepts there is a need to choose a description language. This choice may limit the domains to which a given algorithm can be applied. There are at least two basic types of objects: structured and unstructured. An unstructured object is usually described by attribute-value pairs. For objects having an internal structure first order logic language is often used. Attribute-value languages have the expressive power of propositional logic. These languages sometimes do not allow for proper representation of complex structured objects and relations among objects or their components. The background knowledge that can be used in the discovery process is of a restricted form and other relations from the database cannot be used in the discovery process. Using first-order logic (or FOL for short) has some advantages over propositional logic. First order logic provides a uniform and very expressive means of representation. The background knowledge and the examples, as well as the induced patterns, can all be represented as formulas in a first order language. Unlike propositional learning systems, the first order approaches do not require that the relevant data be composed into single relation but, rather can take into account data, which is organized in several database relations with various connections existing among them. First order logic can face problems which cannot be reduced to propositional logics, such as recurrent structures. On the other hand, even if a problem can be reduced to propositional logics, the solutions found in FOL are more readable and simpler than the corresponding ones in propositional logics.
We consider application of rough set methods to discovery of interesting patterns expressed in a first order language. Rough set methodology is used in the process of selecting literals, which may be part of the rule. The criterion of selecting literal is as follows: only such literal is selected, which adding to the rule makes that the rule discerns the most examples from those, which were yet indiscernible.

2. Relational learning

Knowledge discovery is the process of discovering particular patterns over data. In this context data is typically stored in a database. Approaches using First Order Logic (FOL, for short) languages for the description of such patterns offer data mining the opportunity of discovering more complex regularities which may be out of reach for attribute-value languages. Knowledge discovery based on FOL still has other advantages. Complex background knowledge provided by experts can be encoded as first order formulas and be used in the discovery task. The expressiveness of FOL enables discovered patterns to be described in a concise way, which in most cases increases readability of the output. Multiple relations can be naturally handled without explicit (and expensive) joins.

[1] Department of Computer Science, Białystok University of Technology, Wiejska 45A, 15-351 Białystok, Poland, e-mail: jstepan@ii.pb.bialystok.pl

2.1 Learning task for relational learning systems

In this subsection we recall selected definitions of first-order language.
Before moving on to algorithm for learning set of rules, let us introduce some basic terminology from relational learning.
Relational learning (also called empirical inductive logic programming) algorithms learn classification rules for a concept. The program typically receives a large collection of positive and negative examples from real-world databases as well as background knowledge in the form of relations. Let p be a target predicate of arity m and $r_1,...,r_l$ be background predicates, where $m,l>0$ are given natural numbers. We denote the constants by $con_1,...,con_n$, where $n>0$. A term is either a variable or a constant. An atomic formula is of the form $p(t_1,...,t_m)$ or $r_i(t_1,...)$ where the t' s are terms and $i=1,...,l.$ A literal is an atomic formula or its negation. If a literal contains a negation $(\neg)$ symbol, we call it a negative literal, otherwise a positive literal.
We assume that the expressions are not permitted to contain function symbols (this reduces the complexity of the hypothesis space search).
The learning task for relational learning systems is as follows:
Given:

- a set of positive and negative training examples (expressed by literals without variables) for the target relation,
- background knowledge (or BK for short) expressed by literals without variables and not including the target predicate.
- a language of hypothesis (used for rules searching)
- a notion of satisfaction

Find:

- a set of **if** λ **then** ξ rules, where ξ is an atomic formula of the form $p\left(var_1^p,...,var_m^p\right)$ with the target predicate p and λ is a conjunction of literals over background predicates $r_1,...,r_l$, such that the set of rules satisfies positive examples relatively to background knowledge.

2.2 Sketch of the RSRL algorithm

In this subsection we introduce the RSRL algorithm:
RSRL$\left(Target_predicate, BK, X^+_{target} \cup X^-_{target}\right)$
$Pos := X^+_{target}$; $Neg := X^-_{target}$; $Learned\_rules := \varnothing$;
while $Pos \neq \varnothing$ do
begin
Learn a $NewRule$; $NewRule :=$ most general rule possible; $NewRuleNeg := Neg$;
while $NewRuleNeg \neq \varnothing$ do
begin
Add a new literal to specialize $NewRule$; $Candidate\_literals :=$ generate candidates;
$Best\_literal := \text{argmax}_{L \in Candidate\_literals} RSRL(L, NewRule)$;
Add $Best\_literal$ to $NewRule$ preconditions;
$NewRuleNeg :=$ subset of $NewRuleNeg$ that satisfies $NewRule$ preconditions;
end;
$Learned\_rules := Learned\_rules \cup \{NewRule\}$;
$Pos := Pos -$ {members of Pos covered by $NewRule$ };
end;
To generate candidate specializations of the current rules, RSRL generates a variety of new literals.

More precisely, suppose that the current rule being considered is

$Rule$: **if** L_1 **and** ... **and** L_j **then** $p(var_1^p, var_2^p, ..., var_m^p)$.

RSRL generates candidate specializations of this rule by considering new literal L that fit one of the following forms:

- $r(var_1, ..., var_s)$, where $r \in \{r_1, ..., r_l\}$ and at least one of the var_i in the created literal must already exist as a variable in the rule.
- The negation of the above form of literal.

To select the most promising literal from the candidates generated at each step, RSRL considers the performance of the rule over the training data. The evaluation function used by RSRL to estimate the utility of adding a new literal is based on the numbers of discernible positive and negative examples before and after adding the new literal.

References

[1] Pawlak, Z.: *Rough Sets. Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht 1991.

[2] Polkowski, L., Skowron, A. (Eds.): *Rough Sets in Knowledge Discovery 1 and 2*. Physica-Verlag, Heidelberg 1998.

[3] Stepaniuk, J.: Knowledge Discovery by Application of Rough Set Models. In: L. Polkowski, S. Tsumoto, T.Y. Lin, (Eds.) *Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems*, Physica–Verlag, Heidelberg, 2000, pp.137–233.