Rough Sets and Relational Learning

Jaroslaw Stepaniuk
Institute of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
email: jstepan@ii.pb.bialystok.pl

ABSTRACT: Rough set methodology Pawlak (1991) is based on concept (set) approximations constructed from available background knowledge represented in information systems. In many applications only partial knowledge about approximated concepts is given. Hence quite often first a parameterised family of concept approximations is built and next, by parameters tuning the best, in a sense, approximation is chosen. Relational learning (see e.g. Lavrac and Dzeroski (1994), Quinlan (1990)) uses an expressive first-order logic framework instead of the traditional attribute-value framework and facilitates the use of background knowledge. In this paper we discuss combination of rough set methods and relational learning methods.

KEYWORDS: rough sets, relational learning, knowledge discovery, inductive logic programming

1 INTRODUCTION

Rough set theory was developed by Pawlak (1991). It deals with the classificatory analysis of data tables. The data can be acquired from measurements or from human experts. The main goal of the rough set analysis is to synthesise approximation of concepts from the acquired data.
Knowledge discovery and data mining systems have to face several difficulties, in particular related to the huge amount of input data. This problem is especially related to relational learning (or RL for short) systems (see for example Quinlan (1990), Lavrac and Dzeroski (1994)) which employ algorithms that are computationally complex. Learning time can be reduced by feeding the RL algorithm only a well-chosen portion of the original input data. Such transformation of the input data should throw away unimportant formulas but leave ones that are potentially necessary to obtain proper results.
In this paper two approaches to data reduction problem are proposed. Both are based on rough set theory. Rough set techniques serve as data reduction tools to reduce the size of input data fed to more time-expensive (search-intensive) RL techniques. First approach transforms input formulas into decision table form, then uses reducts to select only meaningful data. Second approach introduces a special kind of approximation space. When properly used, iterated lower and upper approximations of target concept have the ability to preferably select facts that are more relevant for concept approximation, at the same time throwing out the non-relevant facts.

2 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. The prototypical example for this research is FOIL Quinlan (1990) and its various successors, but there are several other approaches like LINUS and DINUS Lavrac and Dzeroski (1994).
Let p be a target predicate of arity m and $r_1,\ldots,r_l$ be background predicates. We denote the constants by $con_1,\ldots,con_n$. A term is either a variable or a constant. An atomic formula is of the form $p(t_1,\ldots,t_m)$ or $r_i(t_1,\ldots)$ where the *t*'s are terms. A literal is an atomic formula or its negation.
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.

Find:

- a set of if l then z rules, where z is an atomic formula with the target predicate and l is a conjunction of atomic formulas over background predicates.

Let us note that the learning problems can be also formulated in terms of first-order logic. Consider the background knowledge as a relational structure over the universe of constants. Then for example the concept defined by the **if** part of the rule

If $r(\mathrm{var}_x, \mathrm{var}_1)$ **and** $r(\mathrm{var}_1, \mathrm{var}_2)$ **and** $r(\mathrm{var}_2, \mathrm{var}_y)$ **then** $p(\mathrm{var}_x, \mathrm{var}_y)$

is equivalent to the predicate defined by

$\exists \mathrm{var}_1 \exists \mathrm{var}_2 (r(\mathrm{var}_x, \mathrm{var}_1) \wedge r(\mathrm{var}_1, \mathrm{var}_2) \wedge r(\mathrm{var}_2, \mathrm{var}_y))$.

The class of concepts definable by a non-recursive rules over the background knowledge is equivalent to the class of predicates definable over the relational structure corresponding to the background knowledge, by existential formulas such that their quantifier-free part is a conjunction of atomic formulas.

We discuss this problem more precisely.

Let k, l and m be given natural numbers and let $r_1, \ldots, r_l$ be predicate symbols. Let Φ_k^m be a set of formulas of the form $\exists \mathrm{var}_{i_1} \ldots \exists \mathrm{var}_{i_j}\, j\,(\mathrm{var}_1^p, \ldots, \mathrm{var}_m^p, \mathrm{var}_{i_1}, \ldots, \mathrm{var}_{i_j})$ with m free variables $\mathrm{var}_1^p, \ldots, \mathrm{var}_m^p$ and with at most k existential quantifiers, and j is a conjunction of atomic formulas over $r_1, \ldots, r_l$ with variables $\mathrm{var}_1^p, \ldots, \mathrm{var}_m^p, \mathrm{var}_{i_1}, \ldots, \mathrm{var}_{i_j}$.

For a relational structure $M = (\{con_1, \ldots, con_n\}, r_1^M, \ldots, r_l^M)$ we consider relations definable by disjunctions of formulas from the set Φ_k^m.

Example 2.1 In this example we sketch how the language used in the standard rough set approach Pawlak (1991) can be translated into discussed language. Assume that a data table $DT = (U, A \cup \{d\})$ is given. Assume without lost of generality that a set of objects $U = \{con_1, \ldots, con_n\}$ and A is a set of condition attributes and d is a decision attribute. For every attribute-value pair (a, v), where $v \in V_a$ and V_a is a set of values for attribute $a \in A$ one can define an unary predicate symbol $r_{(a,v)}$. One can construct the background knowledge by the following rule:

$r_{(a,v)}(con_i)$ is in the background knowledge if and only if $a(con_i) = v$.

Positive and negative examples can be defined using the following equivalence:

$p_{(d,v)}(con_i)$ is a positive example if and only if $d(con_i) = v$.

The relational structure M based on a given background knowledge is defined by

$M = (\{con_1, \ldots, con_n\}, (r_{(a,v)}^M)_{a \in A, v \in V_a})$.

3 TRANSFORMING FIRST-ORDER DATA TO ATTRIBUTE-VALUE FORM

In this section we discuss the following approach:

1. The data is transformed from first-order logic into decision table format by the iterative checking whether a new attribute adds any relevant information to the decision table.
2. The reducts and rules from reducts are computed from obtained decision table.

Data represented as a set of formulas can be transformed into attribute-value form, consisting of a number of objects that have certain values for certain attributes. This form is known as the decision table.

The idea of translation was inspired by LINUS and DINUS systems Lavrac and Dzeroski (1994). We start with a decision table directly derived from the target relations positive and negative examples. Assuming we have m-ary target predicate, the set U of objects in the decision table is a subset of $\{con_1, \ldots, con_n\}^m$. Decision attribute is the target predicate with values "+" or "-". All positive and negative examples of the target predicate are now put into the decision table. Each example creates a separate row in the table. Then background knowledge is applied to the decision

table. We determine all the possible applications of the background predicates to the arguments of the target relation. Each such application introduces a new Boolean attribute.

To analyse the complexity of the obtained data table, let us consider the number of condition attributes. Let A_{r_i} be a set of attributes constructed for every predicate symbol r_i, where $i=1,\ldots,l$. The number of condition attributes in constructed data table is equal to $\sum_{i=1}^{l} card\left(A_{r_i}\right)$ resulting from the possible applications of the l background predicates on the variables of the target relation. The cardinality of A_{r_i} depends on the number of arguments of target predicate (denoted by m) and the arity of r_i. Namely $card\left(A_{r_i}\right)$ is equal to $m^{ar(r_i)}$, where $ar\left(r_i\right)$ is the arity of the predicate r_i. The number of condition attributes in obtained data table is polynomial in the arity m of the target predicate p and the number l of background knowledge predicates, but its size is usually so large that its processing will be not feasible. Therefore one can check interactively if a new attribute is relevant i.e. adds any information to the decision table and next we add to the decision table only relevant attributes.

Three conditions for testing if a new attribute is relevant are proposed Stepaniuk and Maj (1998):

1. $card\left(POS\left(AS_{B\cup\{a\}},\left\{X_+, X_-\right\}\right)\right) > card\left(POS\left(AS_B,\left\{X_+, X_-\right\}\right)\right)$, where X_+ and X_- denote decision classes corresponding to the target concept. An attribute is added to the decision table if it results in a positive region growth with respect to previously selected attributes.
2. $n_{SRI}\left(X_+ \times X_-, \{(x,y)\in X_+ \times X_- : a(x)\neq a(y)\}\right) \geq theta$, where

   $n_{SRI}(X,Y)=\begin{cases}\dfrac{card(X\cap Y)}{card(X)} & \text{if} \quad X\neq\varnothing \\ 1 & \text{if} \quad X=\varnothing\end{cases}$ is the standard rough inclusion function and $theta\in[0,1]$ is a given real number. Attribute is added to the decision table if it introduces some discernibility between objects belonging to different non-empty classes X_+ and X_-.
3. $\arg\max\left\{card\left(POS\left(AS_{B\cup\{a\}},\left\{X_+, X_-\right\}\right)-POS\left(AS_B,\left\{X_+, X_-\right\}\right)\right)\right\}$. Given several potential attributes, only the attribute with maximal positive region gain is selected to be added to the decision table.

First two conditions can be applied to a single attribute before it is introduced to the decision table. If this attribute does not meet a condition it is not included in the decision table. The third condition is applied when we have several candidate attributes and must select the one that is potentially the best.

The received data table is then analysed by a rough set based systems (for example ROSETTA, see Ohrn and others (1998)). First, reducts are computed. Next, decision rules are generated.

4 SELECTION OF RELEVANT FACTS

An approach presented in this section consists of the following steps:

1. Selection of potentially important facts from background knowledge.
2. Application of relational learning system such as FOIL to selected formulas.

The selection is based on constants occurring in positive and negative examples of a target relation. The set of all constants occurring in a fact x is denoted by $CON(x)$. CON can be treated as a set valued attribute.

A set of constants for a set of facts X is defined by $CON(X)=\bigcup_{x\in X} CON(x)$.

Training set reduction begins with determining the set of constants in all positive and negative examples for the target predicate. Such set is denoted as $CON\left(X_{target}\right)$. We consider a data table $(U,\{CON\}\cup\{d\})$, where U is the set of all facts from background knowledge, $CON: U\rightarrow P(\{con_1,\ldots,con_n\})$, where $P(\{con_1,\ldots,con_n\})$ is the set of all subsets of constants and $d: U\rightarrow\{0,1\}$. For every $x\in U$ we assume $d(x)=1$ if and only if $CON(x)\subseteq CON\left(X_{target}\right)$.

The selections can be represented as lower and upper approximations of $X_{d=1} = \{x \in U : d(x) = 1\}$ in the family of approximation spaces $AS_{CON}^{f_{CON}} = (U, I_{CON}^{f_{CON}}, n_{SRI})$, where

$$f_{CON}(CON(x), CON(x')) = w_1 \bullet \frac{card(CON(x))}{card(CON(x) \cup CON(x'))} + w_2 \bullet \frac{card(CON(x'))}{card(CON(x) \cup CON(x'))} + e$$ and

w_1, w_2 and e are parameters.

Definition 4.1 Let $AS_{CON}^{f_{CON}} = (U, I_{CON}^{f_{CON}}, n_{SRI})$ be an approximation space, where

1. U is the set of all facts from background knowledge.
2. The uncertainty function $I_{CON}^{f_{CON}}$ is defined by

   $x' \in I_{CON}^{f_{CON}}(x)$ if and only if $1 - \frac{card(CON(x) \cap CON(x'))}{card(CON(x) \cup CON(x'))} \leq f_{CON}(CON(x), CON(x'))$.
3. The standard rough inclusion function $n_{SRI} : P(U) \times P(U) \to [0,1]$ is defined by

$$n_{SRI}(X,Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)} & \text{if} \quad X \neq \emptyset \\ 1 & \text{if} \quad X = \emptyset \end{cases}.$$

The lower and the upper approximations of a set $X \subseteq U$ in $AS_{CON}^{f_{CON}}$ are defined by

$$LOW(AS_{CON}^{f_{CON}}, X) = \{x \in U : n_{SRI}(I_{CON}^{f_{CON}}(x), X) = 1\},$$
$$UPP(AS_{CON}^{f_{CON}}, X) = \{x \in U : n_{SRI}(I_{CON}^{f_{CON}}(x), X) > 0\}.$$

Any uncertainty function contributes to a different approximation space which results in different kinds of approximations that show different properties.

We then define two transformations $LOW : P(U) \to P(U)$ and $UPP : P(U) \to P(U)$ based on the lower and upper approximations in $AS_{CON}^{f_{CON}}$.

Starting with $X_{d=1}$ one can construct a sequance of approximations by constantly applying one of these transformations first on $X_{d=1}$ and then on the approximation resulting from the previous step.

Thus, the problem of selection is reduced to constantly applying upper (lower) approximation in the same approximation space to the upper (lower) approximation set obtained in the previous step.

The input data reduction problem is then defined as taking into account facts that are included in $LOW(AS_{CON}^{f_{CON}}, X_{d=1})$. If this approximation appears to be too restrictive, which results in bad quality of discovered knowledge, we then consider $UPP(AS_{CON}^{f_{CON}}, X_{d=1})$. If it also does not meet our expectations, we proceed to consider the following approximations: $UPP(AS_{CON}^{f_{CON}}, UPP(AS_{CON}^{f_{CON}}, X_{d=1}))$ and so on. We can stop when the approximation is sufficient to learn up to satisfactory definition of the target concept.

Since $X_{target} = X_{target}^{+} \cup X_{target}^{-}$ (the union of positive and negative examples of the target relation) we may also consider separate approximations of sets corresponding to X_{target}^{+} and X_{target}^{-} which are added after the approximation process. This approach results in a more restrictive approximation.

5 ILLUSTRATIVE EXAMPLES

In this section we apply the proposed approaches to two examples.

Example 5.1 The daughter problem can be used to demonstrate the translation from first-order data to attribute-value form. For simplicity, the names of the persons are 1, 2, ... instead of "Mary", "Ann", ... In order to make example more readable, only the first letters of the corresponding predicate names were used. Suppose that there are the following positive and negative examples of target predicate *daughter* (d):

- positive examples: d(1,2),d(3,4),d(5,6),d(7,6),
- negative examples: not d(4,2), not d(3,2), not d(7,5), not d(2,4).

Consider the background knowledge about family relations, *parent* (p) and *female* (f)
p(2,1),p(2,4),p(4,3),p(6,5),p(6,7),f(2),f(1),f(3),f(5),f(7).
We then transform the data into attribute-value form (decision table).
Using conditions introduced in Section 3 some attributes will not be included in the decision table. For example the second condition with $theta = 0.1$ would not permit the following attributes into the decision table: $p(var_1, var_1)$ and $p(var_2, var_2)$ (see Table 1).

(var_1,var_2)	$f(var_1)$	$f(var_2)$	$p(var_1,var_2)$	$p(var_2,var_1)$	$d(var_1,var_2)$
(1,2)	true	true	false	true	+
(3,4)	true	false	false	true	+
(5,6)	true	false	false	true	+
(7,6)	true	false	false	true	+
(4,2)	false	true	false	true	-
(3,2)	true	true	false	false	-
(7,5)	true	true	false	false	-
(2,4)	true	false	true	false	-

Table 1

Then we compute reducts. We obtain one reduct: $\{f(\mathrm{var}_1), p(\mathrm{var}_2, \mathrm{var}_1)\}$. We then generate rules for $d(\mathrm{var}_1, \mathrm{var}_2)$ based on this reduct obtaining:

if $f(\mathrm{var}_1) = true$ **and** $p(\mathrm{var}_2, \mathrm{var}_1) = true$ **then** $d(\mathrm{var}_1, \mathrm{var}_2) = +$.

Example 5.2 We consider a more complicated problem Stepaniuk and Maj (1998). The experimental data set is related to document understanding and has been an object of previous studies, see for example Esposito and others (1993), Martienne and Quafafou (1998). The learning task involves identifying the purposes served by components of single-page letters. Predicate data describes thirty single page documents containing 364 components in all. Fifty seven background predicates describe properties of components such as their width and height, and relationships such as horizontal and vertical alignment with other components. Target predicates describe whether a block is one of the five predetermined types: sender, receiver, logo, reference and date.

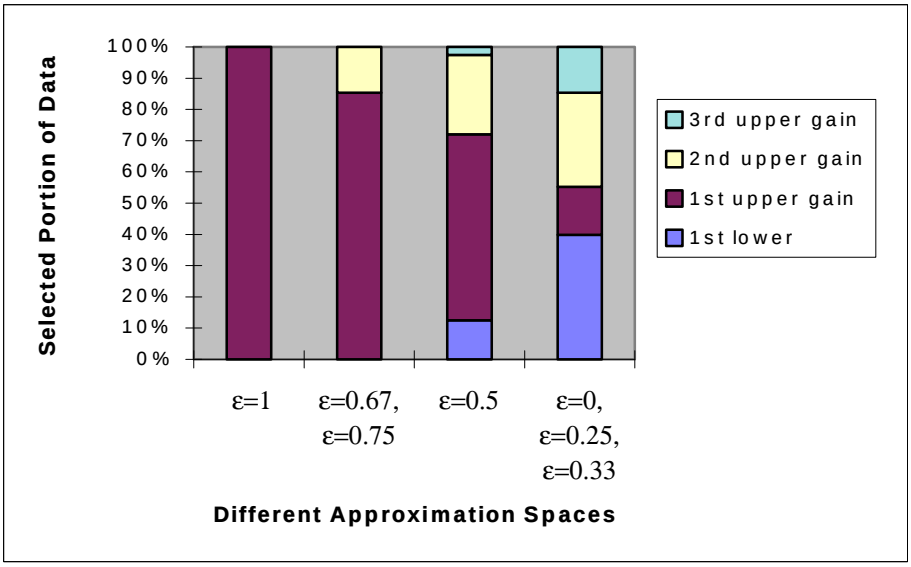

Figure 1

We consider an approximation space $AS_{CON}^{f_{CON}} = (U, I_{CON}^{f_{CON}}, n_{SRI})$ such that $w_1 = w_2 = 0$, thus $f_{CON}(CON(x), CON(x')) = e$, where $e \in [0,1]$ is a parameter. The lower approximation of order one and upper approximations of order one, two and three have been calculated. By applying approximations in different approximation spaces (with respect to e), several levels of data reduction were obtained. In this data set approximation spaces were divided into four groups, each displaying different data reduction levels. Overall there were eight data levels, ranging from the empty set to the full input data set. Figure 1 shows the results for different approximation space

groups and eight possible reduction levels resulting from four previously mentioned approximations. Bars with different patterns represent the gain in input data resulting from applying the next approximation. Experiments with FOIL system show that any non-empty approximation is sufficient to obtain satisfactory definitions of the target predicates (accuracy above 90%).

CONCLUSIONS

In this paper we discuss a combination of rough set methods and relational learning methods. Two approaches are presented. First approach, based on translation of first-order data to data table can be applied to a certain class of problems that can be transformed into attribute-value form without the loss of significant data. Second approach uses the parameterised approximation spaces. By employing a new kind of approximation space we are able to select formulas that are more relevant to the problem. If the selection appears to be too restrictive approximation can be used in multiple passes, each of them expanding the set of formulas in a way that includes only the most relevant facts from the ones that were previously thrown out.

ACKOWLEDGEMENTS

The author would like to thank Andrzej Skowron and Marcin Maj for valuable discussions. This research was supported by the grants No. 8 T11C 023 15 and 8 T11C 010 11 from the State Committee for Scientific Research, the Bialystok University of Technology Rector's Grant W/II/3/98 and Research Program of the European Union - ESPRIT-CRIT 2 No. 20288.

REFERENCES

Esposito F., Malerba D., Semeraro G., Pazzani M., 1993, A Machine Learning Approach to Document Understanding, Proceedings of the Second International Workshop on Multistrategy Learning, West Virginia, pp. 276-292.
Lavrac N., Dzeroski S., 1994, Inductive Logic Programming, Ellis Horwood, Chichester, UK.
Martienne E., Quafafou M., 1998, Learning Logical Descriptions for Document Understanding: A Rough Set Based Approach, Lecture Notes in Artificial Intelligence 1424, Springer-Verlag, pp. 202-209.
Ohrn A., Komorowski J., Skowron A., Synak P., 1998, The Design and Implementation of a Knowledge Discovery Toolkit Based on Rough Sets – The ROSETTA System, (eds.) L. Polkowski, A. Skowron, Rough Sets in Knowledge Discovery 1. Methodology and Applications, Physica-Verlag, Heidelberg, 1998, pp. 376-399.
Pawlak Z., 1991, Rough Sets. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht.
Quinlan J.R., 1990, Learning Logical Definitions from Relations, Machine Learning, 5, pp. 239-266.
Stepaniuk J., 1998, Rough Relations and Logics, (eds.) L. Polkowski, A. Skowron, Rough Sets in Knowledge Discovery 1. Methodology and Applications, Physica Verlag, Heidelberg, 1998, pp. 248-260.
Stepaniuk J., 1998, Approximation Spaces, Reducts and Representatives, (eds.) L. Polkowski, A. Skowron, Rough Sets in Knowledge Discovery 2. Applications, Case Studies and Software Systems, Physica-Verlag, Heidelberg, 1998, pp. 109-126.
Stepaniuk J., Maj M., 1998, Data Transformation and Rough Sets, Proceedings of PKDD’98, Lecture Notes in Artificial Intelligence 1510, Springer-Verlag, pp. 441-449.