Data Transformation and Rough Sets

Jaroslaw Stepaniuk Marcin Maj

Institute of Computer Science
Bialystok University of Technology Wiejska 45A,
15-351 Bialystok, Poland
e-mail: {jstepan, mmaj}@ii.pb.bialystok.pl

Abstract. 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 inductive logic programming systems, which employ algorithms that are computationally complex. Learning time can be reduced by feeding the ILP algorithm only a well-chosen portion of the original input data. Such transformation of the input data should throw away unimportant clauses 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) ILP techniques. First approach transforms input clauses 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 to the problem, at the same time throwing out the facts that are totally unimportant.

1 Introduction

Knowledge discovery in databases (KDD) is concerned with identifying interesting patterns and describing them in a concise and meaningful manner. Rough set methodology for knowledge discovery was introduced by Pawlak [8]. It provides a powerful tool for knowledge discovery from incomplete data. A number of algorithms and systems have been developed based on rough set theory which may induce a set of decision rules from a given decision table, and may use induced decision rules to classify future examples. Most of them are attempting to find and select the best minimal set of decision rules that use only a minimal subset of attributes (called reduct) from the given data table.

Rough set based systems, such as KDD-R [14], PRIMEROSE [13] and ROSETTA [7] have been applied to KDD problems. The patterns discovered by the above systems are expressed in attribute-value languages which 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 clausal logic has some advantages over propositional logic. Clausal 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 clausal 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.

In this paper we consider two directions in applications of rough set methods to discovery of interesting patterns expressed in a clausal language.

The first direction is based on translation of data represented in clausal language to decision table [8] format and next processing using rough set methods based on the notion of reduct. Our approach is based on the iterative checking whether a new attribute adds to the information.

The second direction concerns reduction of the size of the data in clausal language and is related to results described in [4, 5]. The discovery process is performed only on well-chosen portions of data which correspond to approximations in the rough set theory. Our approach is based on iteration of approximation operators.

2 Approximation Spaces and Rough Sets

In this section we recall general definition of approximation space [10, 11, 12] which can be used for example for the tolerance based rough set model.

An approximation space is a system $AS=(U,I,v)$, where U is a non-empty set of objects, $I\colon U\to P(U)$ is an uncertainty function ($P(U)$ denotes the set of all subsets of U) and $v\colon P(U)\times P(U)\to[0,1]$ is a rough inclusion function. An uncertainty function defines for every object $x\in U$ objects related to x. The rough inclusion function defines the value of inclusion between two subsets of U. Definitions of the lower and the upper approximations can be written as follows:

$LOW(AS,X)=\{x\in U: v(I(x),X)=1\}$ and $UPP(AS,X)=\{x\in U: v(I(x),X)>0\}$.

We recall some notions of the rough set theory in the case of generalized approximation spaces [12].

Let $AS=(U,I,v)$ be an approximation space and let $\{X_1,\ldots,X_r\}$ be a classification of objects (i.e. $X_1,\ldots,X_r\subseteq U$, $\bigcup_{i=1}^{r}X_i=U$ and $X_i\cap X_j=\varnothing$ for $i\neq j$, where $i,j=1,\ldots,r$).

The positive region of the classification $\{X_1,\ldots,X_r\}$ with respect to approximation space AS is defined as

$$POS(AS,\{X_1,\ldots,X_r\})=\bigcup_{i=1}^{r}LOW(AS,X_i).$$

The quality of approximation of the classification $\{X_1,\ldots,X_r\}$ in the approximation space AS is defined as follows:

$$\gamma(AS,\{X_1,\ldots,X_r\}) = \frac{card(POS(AS,\{X_1,\ldots,X_r\}))}{card(U)}.$$

This coefficient expresses the ratio of the number of all AS-correctly classified objects to the number of all objects in the data table.

It is always desirable to reduce the amount of information required to predict an outcome. A reduced number of attributes results in a large number of objects in class of objects similar to a given object, making the results more meaningful. If we can remove some of the condition attributes without affecting the degree of dependency between the subset of condition attributes and the decision, the remaining attributes will be termed a reduct [8].

To explain in more detail the notion of reduct, let $(U, A \cup \{d\})$ be a decision table with condition attributes A and a decision attribute d. Let for every subset $B \subseteq A$ approximation space AS_B is defined.

A subset $B \subseteq A$ is a relative reduct for $(AS_A, \{d\})$ if and only if

1. $POS(AS_B, \{d\}) = POS(AS_A, \{d\})$.
2. For every proper subset $B' \subseteq B$ the first condition is not true.

Approximation spaces and relative reducts are used in next section.

3 Input Data Transformation Problem

In this section we discuss problem of adequate data transformation for knowledge discovery systems. General scheme of our approach is represented on Figure 1.

3.1 Reduct Approach

In this subsection we discuss the following approach:

1. The data is transformed from clausal logic to decision table format by the iterative checking whether a new attribute adds any information to the decision table.
2. The reducts are computed from obtained decision table.
3. Rules from reducts are generated.

Data represented as a set of clauses 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. When certain conditions are not met, the transformation is imperfect, because the expressive power of attribute-value language is insufficient to properly represent some concepts. In cases like that the transformation only leaves a limited knowledge about the problem, usually not enough to discover a satisfactory definition.

The idea of translation was inspired by LINUS system [2, 1]. We start with a decision table directly derived from the target relations positive and negative examples. Assuming we have *n*-ary target predicate, the first *n* attributes in the decision table are variables of the same type as their respective target predicate arguments. Last attribute is the target predicate value - true or false. All positive and negative examples of the target predicate are now put into the decision table. Each example is put in 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 on the arguments of the target relation - the first *n* attributes in the table being constructed, taking into account argument types. Each such application introduces a new Boolean attribute.

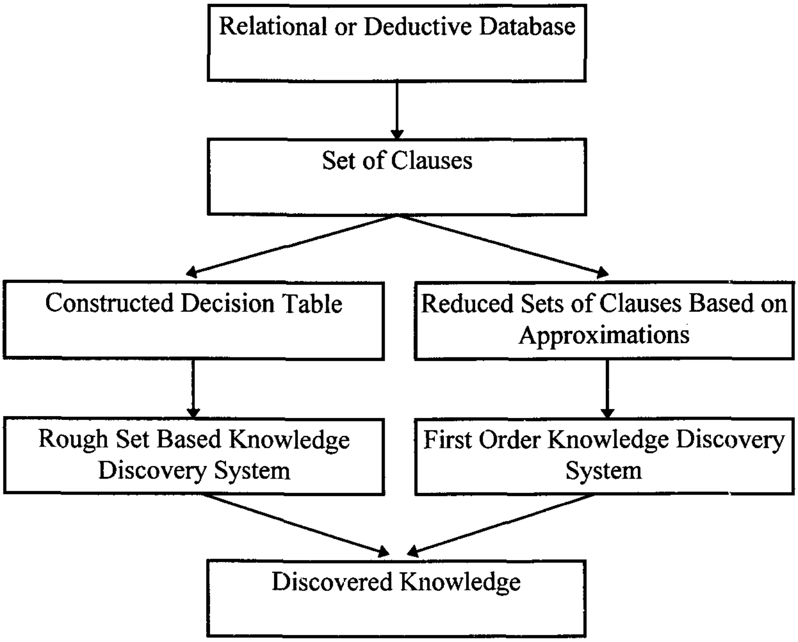
Relational or Deductive Database
Set of Clauses
Constructed Decision Table
Reduced Sets of Clauses Based on Approximations
Rough Set Based Knowledge Discovery System
First Order Knowledge Discovery System
Discovered Knowledge

Fig. 1. General Scheme

One can check if a new attribute adds any information to the decision table. Three conditions for adding a new attribute are proposed:

1. $POS\left(AS_{B\cup\{a\}},\{X_+,X_-\}\right)\supset POS\left(AS_B,\{X_+,X_-\}\right)$. Attribute is added to the decision table if it results in a positive region growth with respect to previously selected attributes.
2. $\frac{card\left(\{(x,y)\in X_+\times X_-:a(x)\neq a(y)\}\right)}{card\left(X_+\times X_-\right)}\geq\theta$, where $\theta\in[0,1]$ is a given real number. Attribute is added to the decision table if it introduces some distinction between objects that belong to different non-empty classes X_+ and X_-.
3. $\arg\max\left\{card\left(POS\left(AS_{B\cup\{a\}},\{X_+,X_-\}\right)-POS\left(AS_B,\{X_+,X_-\}\right)\right)\right\}$. Given several

potential attributes, only the attribute with maximal positive region gain is selected into 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 best.

In the end we discard the first n nominal attributes as they do not contribute to the problem - they are only used as identifiers and cannot be used in the learning process.

The transformed problem is then analyzed by a rough set based system. First, reducts are computed. Next, decision rules are generated. Although expressed in propositional logic language, the rules are easily converted to first-order logic language.

This approach is not universal and only applies to problems that can be transformed to attribute-value form without the loss of significant data. Counter-examples include problems that employ recursive rules and problems that introduce new variables into their rules, besides the ones that appear in the target predicate. It is important to note that by using a more complicated algorithm to convert these problems to decision table form we may minimize the loss of significant data. This however requires us to introduce into the decision table more than n argument attributes (variables). This greatly increases the number of possible applications of background knowledge on these arguments. Furthermore, we can consider positions of variables in the predicate argument list - this will also generate a lot of new argument attributes. The combined result will be a huge and difficult to comprehend decision table. The effects of applying a rough-set based system on such a table are still being investigated.

3.2 Approximation Space Approach

The approach presented in this subsection consists of the following steps:

1. Selection of potentially important facts from background knowledge.
2. Application of inductive logic programming system such as FOIL [9] or PROGOL [6] to selected clauses.

Such selection is based on the concept of „nominal information", first associated with input data reduction problem in [4, 5]. Nominal information of a fact L is the set of its nominal terms (nominal parameter values). It is denoted as $Nom(L)$.

Nominal information of a set of instances (or a concept) X is the sum of all instances - positive and negative - it consists of:

$$Nom(X) = \bigcup_{L \in X} Nom(L)$$

Selection of representatives (training set reduction) begins with determining the set of instances of target predicates (definitions of which we seek). Such set is denoted as X_{target}.

The selections can be represented as lower and upper approximations of $X_{target} \subseteq U$ in the family of approximation spaces $AS_{\#} = (U, I_{\#}, v)$, where $\# \in OP$ and $OP = \{=, \cap, \varepsilon, \subseteq, \supseteq\}$ is the set of operators.

Definition 3.1 Let $AS_{\#} = (U, I_{\#}, v)$ be an approximation space, where U is the set of clauses with non-empty nominal information.

1. For every $L \in U$ we define $I_{\#}(L)$, where $\# \in OP$ as

   $I_{=}(L) = \{L' \in U : Nom(L) = Nom(L')\}$,

   $I_{\cap}(L) = \{L' \in U : Nom(L) \cap Nom(L') \neq \varnothing\}$,

   $I_{\varepsilon}(L) = \left\{ L' \in U : \dfrac{card(Nom(L) \cap Nom(L'))}{card(Nom(L) \cup Nom(L'))} \geq \varepsilon \right\}$, where $\varepsilon \in [0,1]$ is a parameter,

   $I_{\subseteq}(L) = \{L' \in U : Nom(L) \subseteq Nom(L')\}$,

   $I_{\supseteq}(L) = \{L' \in U : Nom(L) \supseteq Nom(L')\}$.

2. The rough inclusion function is defined as:

   $v(X,Y) = \dfrac{card(Nom(X) \cap Nom(Y))}{card(Nom(X))}$.

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

Proposition 3.2 For every uncertainty function $I_{\#}$ exists a corresponding relation $\tau_{\#}$ defined as:

$(L, L') \in \tau_{\#}$ if and only if $L' \in I_{\#}(L)$, where $\# \in OP$.

It can be observed that:

1. $\tau_{=}$ is an equivalence relation.
2. $\tau_{\cap}$ and τ_{ε} are tolerance relations (i.e. reflexive and symmetric relations).
3. $\tau_{\subseteq}$ and $\tau_{\supseteq}$ are reflexive and transitive relations.

We then define two transformations

$LOW : \{AS_{\#} : \# \in OP\} \times P(U) \to P(U)$ and $UPP : \{AS_{\#} : \# \in OP\} \times P(U) \to P(U)$

based on the lower and upper approximations in $AS_{\#}$.

Starting with X_{target} we can construct infinite number of approximations by constantly applying one of these transformations first on X_{target} 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.

It is worth mentioning that under certain conditions it is possible that $X \subset LOW(AS_{\#}, X)$, which means that in this approximation space lower approximation has the ability to expand beyond the set it approximates. This may look surprising in comparison to the traditional understanding of approximation spaces and rough sets [8].

The input data reduction problem is then defined as taking into account clauses that are included in $LOW\left(AS_{\#}, X_{target}\right)$. If this approximation appears to be too restrictive, which results in bad quality of discovered knowledge, we then consider $UPP\left(AS_{\#}, X_{target}\right)$. If it also does not meet our expectations, we proceed to consider following approximations: $UPP\left(AS_{\#}, UPP\left(AS_{\#}, X_{target}\right)\right)$ and so on. We can stop when the approximation is sufficient to learn a satisfactory definition of the target concept. Learning is performed with any kind of ILP system.

This approach may be modified by alternating randomly or by a set pattern between the two transformations and obtaining a different kind of sequence.

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 X^{+}_{target} and X^{-}_{target} which are added after the approximation process. This approach results in a more restrictive approximation (the sets of selected representatives resulting from this approach are subsets of their respective approximations obtained from the whole set X_{target}.

We sketch the algorithm for calculating upper and lower approximations of X_{target}.

```
LOW:=∅; UPP:=∅;
Nominal:=Nom(X_target);
for every L in $U - X_{target}$ do
begin
  Class:=I#(L);
  NC:=Nom(Class);
  RoughInclusion:=v(NC,Nominal);
  if (RoughInclusion=1) then LOW:=LOW∪{L};
  if (RoughInclusion>0) then UPP:=UPP∪{L};
end;
```

Unlike standard rough set approximation calculation this algorithm's time complexity is $O\left(n^2 m \log m\right)$ where $n = card\left(U - X_{target}\right)$ is the number of clauses and $m = card(Nom(U))$ is the number of nominal terms. However, in special case of uncertainty function $I_{=}$ the time can be reduced to $O(nm \log m)$ since we do not need to calculate uncertainty class at all and $NC := Nom(L)$. Other uncertainty functions require us to calculate set intersections or perform other set theoretical operations which are quite time consuming.

Example 3.3 The experimental data set is related to document understanding and has been an object of previous studies, see for example [4]. Predicate data describes 30 single page documents. Background predicates express type, position and alignment of document blocks. Target predicates describe whether a block is one of the five predetermined types: *sender, receiver, logo, reference, date.* First lower approximation and first, second and third upper approximations were considered. By applying approximations in different approximation spaces, 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 an empty set to a full input data set. Figure 2 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%).

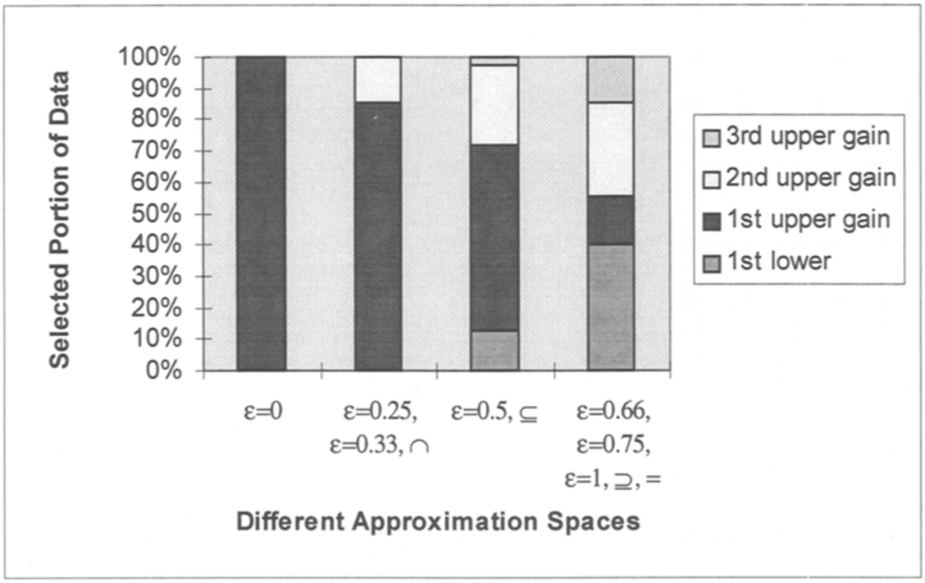

Fig. 2. Four Approximation Space Groups and Eight Approximation Levels

Conclusions

This paper has presented two approaches which aim at overcoming the difficulty met by knowledge discovery systems namely the huge amount of data. Both approaches aim at throwing away facts that are unimportant to the target concept and leaving facts that are potentially necessary. Such process can also be described as selection of representatives. First approach, based on the rough set theory concept of reducts can only be applied to a certain class of problems that can be transformed into attribute-value form without the loss of significant data. The results are quite promising and new ways to transform clauses into attribute values are still being investigated. Second approach uses another rough set theory concept, namely the approximation spaces. By employing a new kind of approximation space we are able to select clauses 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 clause set in a way that includes only the most relevant facts from the ones that were previously thrown out. The facts that are totally irrelevant to the problem are never considered.

Acknowledgments. This work has been supported by the grant no 8 T11C 023 15 from State Committee for Scientific Research (KBN).

References

1. Dzeroski S.: Inductive Logic Programming and Knowledge Discovery in Databases, (eds.) U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, Advances in Knowledge Discovery & Data Mining, The MIT Press, 1996, pp. 117-152.
2. Lavrac N., Dzeroski S., Grobelnik M.: Learning Non-Recursive Definitions of Relations with LINUS, Proceedings of Fifth European Working Session on Learning, 1991, pp. 265-281.
3. Lavrac N., Gamberger D., Turney P.: A Relevancy Filter for Constructive Induction, IEEE Intelligent Systems and Their Applications, 13(2), March/April 1998, pp. 50-56.
4. Martienne E., Quafafou M.: Learning Logical Descriptions for Document Understanding: a Rough Sets-based Approach, Proceedings of the International Conference on Rough Sets and Current Trends in Computing, Warsaw, Poland, June 22-26, 1998, Lecture Notes in Artificial Intelligence 1424, Springer Verlag, pp. 202-209.
5. Martienne E., Quafafou M.: Vagueness and Data Reduction in Concept Learning, Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98), Brighton, UK, August 23-28, 1998.
6. Muggleton S.: Inverse Entailment and Progol, New Generation Computing, 13, 1995, pp. 245-286.
7. Ohrn A., Komorowski J., Skowron A., Synak P.: 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, Physica-Verlag, Heidelberg 1998.
8. Pawlak Z.: Rough Sets. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, 1991.
9. Quinlan J.R.: Learning Logical Definitions from Relations, Machine Learning, 5, 1990, pp. 239-266.
10. Skowron A., Stepaniuk J.: Generalized Approximation Spaces, Proceedings of the Third International Workshop on Rough Sets and Soft Computing, San Jose, November 10-12, 1994, pp. 156-163.
11. Skowron A., Stepaniuk J.: Tolerance Approximation Spaces, Fundamenta Informaticae, 27, 1996, pp. 245-253.
12. Stepaniuk J.: Approximation Spaces, Reducts and Representatives, (eds.) L. Polkowski, A. Skowron, Rough Sets in Knowledge Discovery, Physica-Verlag, Heidelberg 1998.
13. Tsumoto S.: Extraction of Experts' Decision Process from Clinical Databases Using Rough Set Model, PKDD'97, Trondheim, Norway, June 1997, Lecture Notes in Artificial Intelligence 1263, Springer Verlag, pp. 58-67.
14. Ziarko W., Shan N.: KDD-R: A Comprehensive System for Knowledge Discovery in Databases Using Rough Sets, Proceedings of the Third International Workshop on Rough Sets and Soft Computing, San Jose, November 10-12, 1994, pp. 164-173.