Rough Set Theory with Applications to Data Mining

Jerzy W. Grzymala-Busse
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS 66045, USA
and
Institute of Computer Science, Polish Academy of Sciences,
01-237 Warsaw, Poland
Jerzy@ku.edu

Abstract. This paper is an introduction to rough set theory with an emphasis on applications to data mining. First, consistent data are discussed, including blocks of attribute-value pairs, reducts of information tables, indiscernibility relation, decision tables, and global and local coverings. Rule induction algorithms LEM1 and LEM2 are presented. Then the rough set approach to inconsistent data is introduced, with lower and upper approximations and certain and possible rule sets. The last topic is a rough set approach to incomplete data. How to define modified blocks of attribute-value pairs, characteristic sets, and characteristic relation are explained. Additionally, two definitions of definability and three definitions of approximations are presented. Finally, some remarks about applications of the LERS data mining system are included.

1 Introduction

Rough set theory was introduced in 1982 by Z. Pawlak, see (Pawlak 1982). As it is shown in (Grzymala-Busse 1988), rough set theory represents an objective approach to imperfections in data, all computations are performed directly on data sets, i.e., no feedback from additional experts is necessary. Thus, there is no need for any additional information about data such as, for example, a probability distribution function in statistics, a grade of membership from fuzzy set theory, etc. (Grzymala-Busse,1988).

This paper presents basic ideas of rough set theory and shows how these ideas may be utilized for data mining. In general, rough set theory may be applied to consistent data (without conflicting cases) to study relations between attributes.

For example, this way we may eliminate redundant attributes. Another typical task is to find a minimal subset of the attribute set that may be used to identify all concepts. Yet another task is to compute a family of sets of attribute-value pairs for the same reason: to identify all concepts.

Inconsistent data sets are handled by rough set theory using lower and upper approximations for every concept. These approximations are definable using existing attributes. Furthermore, from concept lower and upper approximations certain and possible rule sets are induced.

Another important area of rough set theory applications are incomplete data, i.e., data with attribute missing values. Here rough set theory may be used, again, for computing generalized lower and upper approximations for all concepts.

In this paper we will restrict our attention to only one technique of data mining: rule induction.

2 Consistent Data

In this paper we will study data sets in two forms: as information tables and as decision tables. In both cases variables are presented in columns and cases in rows. In information tables all variables are called attributes while in decision tables one of the variables is called a decision, while the remaining are attributes.

2.1 Information Tables

An example of an information table is presented in Table 1. Four attributes: *Temperature*, *Headache*, *Nausea* and *Cough* characterize six cases.

Table 1. An information table

Case	Attributes			
	Temperature	Headache	Nausea	Cough
1	high	yes	no	yes
2	very_high	yes	yes	no
3	high	no	no	no
4	high	yes	yes	yes
5	normal	yes	no	no
6	normal	no	yes	yes

Let U denote the set of all cases, A the set of all attributes, and V the set of all attribute values. Such a table defines an information function $\rho: U \times A \rightarrow V$. For example, $\rho(1, \text{Temperature}) = \text{high}$.

Let $a \in A$, $v \in V$, and $t = (a, v)$ be an attribute-value pair. A *block* of t, denoted by $[t]$, is a set of all cases from U for which attribute a has value v. For the information table from Table 1,

[(Temperature, high)] = {1, 3, 4},
[(Temperature, very_high)] = {2},
[(Temperature, normal)] = {5, 6},
[(Headache, yes)] = {1, 2, 4, 5},
[(Headache, no)] = {3, 6},
[(Nausea, no)] = {1, 3, 5},
[(Nausea, yes)] = {2, 4, 6},
[(Cough, yes)] = (1, 4, 6}, and
[(Cough, no)] = {2, 3, 5}.

Let $x \in U$ and $B \subseteq A$. An elementary set of B containing x, denoted by $[x]_B$, is the following set:

$$\bigcap \{[(a, v)] \mid a \in B, \rho(x, a) = v\}$$

Elementary sets are subset of U consisting all cases from U that are indistinguishable from x while using all attributes from B. In *soft computing* terminology elementary sets are called *information granules*. When subset B is restricted to a single attribute, elementary sets are blocks of attribute-value pairs defined by that specific attribute. Thus,

$[1]_{\{Temperature\}} = [3]_{\{Temperature\}} = [4]_{\{Temperature\}}$ = [(Temperature, high)] = {1, 3, 4},
$[2]_{\{Temperature\}}$ = [(Temperature, very_high)] = {2},
$[5]_{\{Temperature\}} = [6]_{\{Temperature\}}$ = [(Temperature, normal)] = {5, 6}.

Additionally, if B = {Temperature, Headache},

$[1]_B = [4]_B$ = [(Temperature, high)] ∩ [(Headache, yes)} = {1, 4},
$[2]_B$ = [(Temperature, very_high)] ∩ [(Headache, yes)} = {2},
$[3]_B$ = [(Temperature, high)] ∩ [(Headache, no)} = {3},
$[5]_B$ = [(Temperature, normal)] ∩ [(Headache, yes)} = {5}, and
$[6]_B$ = [(Temperature, normal)] ∩ [(Headache, no)} = {6}.

Elementary sets may be defined in another way, through the notion of an indiscernibility relation. Again, let B be a nonempty subset of the set A of all attributes. The *indiscernibility relation* IND(B) is a binary relation on U defined for x, $y \in U$ as follows

$$(x, y) \in \text{IND}(B) \text{ if and only if } \rho(x, a) = \rho(y, a) \text{ for all } a \in B.$$

Obviously, IND(B) is an equivalence relation. A convenient way to present equivalence relations is through partitions. A *partition* of U is a family of mutually disjoint nonempty subsets of U, called *blocks*, such that the union of all blocks is U. The partition induced by IND(B) will be denoted by B^*. Blocks of B^* are also called *elementary sets* associated with B. For example,

{Temperature}* = {{1, 3, 4}, {2}, {5, 6}},
{Temperature, Headache} * = {{1, 4}, {2}, {3}, {5}, {6}}.

On the other hand,

IND({Temperature}) = {(1, 1), (1, 3), (1, 4), (2, 2), (3, 1), (3, 3), (3, 4), (4, 1), (4, 3), (4, 4), (5, 5), (5, 6), (6, 5), (6, 6)}, and

IND({Temperature, Headache}) = {(1, 1), (1, 4), (2, 2), (3, 3), (4, 1), (4, 4), (5, 5), (6, 6)}.

There are important subsets of attributes called reducts. A subset B of the set A is called a *reduct* if and only if

1. $B^* = A^*$ and

2. B is minimal with this property, i.e., $(B - \{a\})^* \neq A^*$ for all $a \in B$.

For example, {Temperature} is not a reduct since

{Temperature}* = {{1, 3, 4}, {2}, {5, 6}} ≠ A^* = {{1}, {2}, {3}, {4}, {5}, {6}}.

Similarly, {Temperature, Headache} is not a reduct since

{Temperature, Headache}* = {{1, 4}, {2}, {3}, {5}, {6}} ≠ A^* = {{1}, {2}, {3}, {4}, {5}, {6}}.

On the other hand, {Temperature, Headache, Nausea} is a reduct. The systematic way to compute all reducts is based on first checking all single attributes whether $\{a\}^* = A^*$, if so, the corresponding $\{a\}$ is a reduct. The next step is to check all subsets B of A with $|B| = 2$ such B is not a superset of any existing reduct, where $|X|$ is the cardinality of the set X. We check all of such subsets B whether $B^* = A^*$, if so, B is a reduct. The next step is to check all subsets B of set A of the cardinality equal to three such that B is not a superset of a reduct, etc. For the information table presented in Table 1,

{Temperature}* ≠ A^*,
{Headache}* ≠ A^*,

{Nausea}* $\neq A^*$,
{Cough}* $\neq A^*$,

{Temperature, Headache}* $\neq A^*$,
{Temperature, Nausea}* $\neq A^*$,
{Temperature, Cough}* $\neq A^*$,
{Headache, Nausea}* $\neq A^*$,
{Headache, Cough}* $\neq A^*$,
{Nausea, Cough}* $\neq A^*$,

{Temperature, Headache, Nausea}* $= A^*$,
{Temperature, Headache, Cough}* $\neq A^*$,
{Temperature, Nausea, Cough}* $= A^*$,
{Headache, Nausea, Cough}* $= A^*$.

Therefore, reducts are:

{Temperature, Headache, Nausea}, {Temperature, Nausea, Cough}, and {Headache, Nausea, Cough}.

The above method of computing all reducts is of exponential worst time complexity with respect to the number of attributes. Therefore, in practice, we restrict our attention to compute a single reduct, using a heuristic algorithm. The first step of this algorithm is an attempt to eliminate the leftmost attribute a_1 of the set $\{a_1, a_2,..., a_n\} = A$ of all attributes. If

$$\{a_2,..., a_n\}^* = A^*$$

then a_1 can be eliminated for good, if not, we will put it back to the set. In the next step, we try to eliminate a_2, and so on, in the last step, an attempt is to eliminate a_n. In our example, the first step is based on an attempt to eliminate *Temperature*, so we are checking whether

{Headache, Nausea, Cough} $= A^*$.

This attempt is successful. However,

{Nausea, Cough} $\neq A^*$,
{Headache, Cough} $\neq A^*$, and
{Headache, Nausea} $\neq A^*$,

so {Headache, Nausea, Cough} is a reduct.

2.2 Decision Tables

In a decision table variables, presented as columns, belong to either of two categories: attributes and decisions. Usually a decision table has only one decision. Rows, like in information tables, are labeled by case names. An example of a decision table is presented in Table 2. Attributes are: *Temperature*, *Headache*, *Nausea* and *Cough*, a decision is *Flu*.

In decision table from Table 2 there are two elementary sets of {*Flu*}: {1, 2, 4} (for these cases *Flu* has value *yes*) and {3, 5, 6} (for these cases *Flu* has value *no*). Elementary sets of *decision* are called *concepts*.

Decision tables present cases that are classified or diagnosed by experts. For example, attributes may be interpreted as medical tests, and decision may correspond to a disease. Decision tables are crucial for data mining (or knowledge acquisition, or machine learning).

Table 2. A decision table

Case	Attributes				Decision
	Temperature	Headache	Nausea	Cough	Flu
1	high	yes	no	yes	yes
2	very_high	yes	yes	no	yes
3	high	no	no	no	no
4	high	yes	yes	yes	yes
5	normal	yes	no	no	no
6	normal	no	yes	yes	no

There are two main approaches to data mining from complete data sets based on rough set theory. In both approaches decision tables are used. The first approach is *global*: the entire attributes are used for analysis. The second possibility is *local*, blocks of attribute-value pairs are used.

2.2.1 Global Coverings

Rule sets may be induced using global coverings (Grzymala-Busse 1991a), also called relative reducts (Pawlak 1991). We will start from the definition of a partition being *finer* than another partition. Let α and β be partitions of U. We say that α is *finer* than β, denoted $\alpha \leq \beta$, if and only if for each block X of α there exists a block Y of β such that $X \subseteq Y$.

Let d be a decision. A subset B of the attribute set A is a *global covering* if and only if

1. $B^* \leq \{d\}^*$ and

2. B is minimal with this property, i.e., $(B - \{a\})^* \leq \{d\}^*$ is false for any $a \in B$.

For example, {Temperature} is not a global covering since {Temperature}* = {{1, 3, 4}, {2}, {5, 6}} is not finer than {Flu}* = {{1, 2, 4}, {3, 5, 6}}. However, {Temperature, Headache} is a global covering since

{Temperature, Headache}* = {{1, 4}, {2}, {3}, {5}, {6}} ≤ {{Flu}* =
{{1, 2, 4}, {3, 5, 6}}.

Algorithms for computing all global coverings and a single global covering, presented in (Grzymala-Busse 1991a), are similar to algorithms for computing all reducts and a single reduct. However, first we need to check whether

$$A^* \leq \{d\}^*.$$

If this condition is not satisfied, there is no one single global covering. In our example from Table 2,

$$A^* = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\} \leq \{\text{Flu}\}^* = \{\{1, 2, 4\}, \{3, 5, 6\}\},$$

hence we may start the main algorithm to compute all global coverings. We start from checking all single attributes

{Temperature}* ≤ {Flu}* is false,
{Headache}* ≤ {Flu}* is false,
{Nausea}* ≤ {Flu}* is false,
{Cough}* ≤ {Flu}* is false,

therefore, there is no one global covering of size one. Then we are checking all subsets of A with the cardinality equal to two

{Temperature, Headache}* ≤ {Flu}* is true,
{Temperature, Nausea}* ≤ {Flu}* is false,
{Temperature, Cough}* ≤ {Flu}* is true,
{Headache, Nausea}* ≤ {Flu}* is false,
{Headache, Cough}* ≤ {Flu}* is false,
{Nausea, Cough}* ≤ {Flu}* is false,

so there are two global coverings of size two: {Temperature, Headache} and {Temperature, Cough}. There is only one subset of A of the cardinality equal to three that is not a superset of existing global coverings, and that is {Headache, Nausea, Cough}. This set is tested in the next step of the algorithm

{Headache, Nausea, Cough}* ≤ {Flu}* is true,

so there is one global covering of size three. Obviously, the worst time complexity of the algorithm for computing all global coverings is the same as the algorithm for computing all reducts, i.e., exponential. Thus we should restrict our attention to computing a single global covering. The corresponding algorithm is an extension of the algorithm to compute a single reduct. First we need to test whether

$$A^* \leq \{d\}^*.$$

Then we will try to eliminate the first attribute, then the second one, etc. Since {Headache, Nausea, Cough}* ≤ {Flu}*, we eliminate *Temperature*. However,

{Nausea, Cough}* ≤ {Flu}* is false,
{Headache, Cough}* ≤ {Flu}* is false,
{Headache, Nausea}* ≤ {Flu}* is false,

so this algorithm will return the following single global covering: {Headache, Nausea, Cough}. A single global covering may be used for rule induction. We restrict our attention to attributes from the global covering and then, for every case, we will check whether rule conditions can be dropped. Thus, for the first case our original rule, induced from Table 2, is

(Headache, yes) & (Nausea, no) & (Cough, yes) → (Flu, yes).

We may drop the leftmost condition (Headache, yes) since

(Nausea, no) & (Cough, yes) → (Flu, yes).

is consistent with the Table 2. However, this rule cannot be further simplified since

(Cough, yes) → (Flu, yes).

is not consistent with Table 2 (due to case 6 that will be misclassified by this rule), also

(Nausea, no) → (Flu, yes).

is not consistent with Table 2 (due to cases 3 and 5). This rule covers only case 1. The next uncovered case is 2, the induced rule is

(Headache, yes) & (Nausea, yes) & (Cough, no) → (Flu, yes).

The first condition can be dropped, the remaining two conditions cannot be dropped. The new rule, covering only case 2, is

$$(\text{Nausea, yes}) \ \& \ (\text{Cough, no}) \rightarrow (\text{Flu, yes}).$$

The case 3 implies the following rule

$$(\text{Headache, no}) \ \& \ (\text{Nausea, no}) \ \& \ (\text{Cough, no}) \rightarrow (\text{Flu, no}).$$

Again, the first condition may be dropped, the remaining two conditions cannot be dropped. The rule

$$(\text{Nausea, no}) \ \& \ (\text{Cough, no}) \rightarrow (\text{Flu, no}).$$

covers two cases, 3 and 5. Case 4 implies the rule

$$(\text{Headache, yes}) \ \& \ (\text{Nausea, yes}) \ \& \ (\text{Cough, yes}) \rightarrow (\text{Flu, yes}).$$

Here the first condition cannot be dropped, but we may drop the second condition

$$(\text{Headache, yes}) \ \& \ (\text{Cough, yes}) \rightarrow (\text{Flu, yes}).$$

Note that this rule covers two cases, 1 and 4. The last uncovered case is 6, the implied rule is

$$(\text{Headache, no}) \ \& \ (\text{Nausea, yes}) \ \& \ (\text{Cough, yes}) \rightarrow (\text{Flu, no}).$$

The first condition cannot be dropped, but we may drop the second condition to obtain the following rule

$$(\text{Headache, no}) \ \& \ (\text{Cough, yes}) \rightarrow (\text{Flu, no}).$$

The rightmost condition may be dropped as well, our final rule is

$$(\text{Headache, no}) \rightarrow (\text{Flu, no}).$$

This rule cannot be further simplified. It covers two cases, 3 and 6. At the end we should eliminate the redundant rule

$$(\text{Nausea, no}) \ \& \ (\text{Cough, yes}) \rightarrow (\text{Flu, yes}).$$

since it covers only one case (1), while the rule

$$(\text{Headache, yes}) \ \& \ (\text{Cough, yes}) \rightarrow (\text{Flu, yes}).$$

covers two cases: 1 and 4. Thus, the final rule set is

2, 1, 1
(Nausea, yes) & (Cough, no) -> (Flu, yes).

2, 2, 2
(Headache, yes) & (Cough, yes) -> (Flu, yes),

2, 2, 2
(Nausea, no) & (Cough, no) -> (Flu, no),

1, 2, 2
(Headache, no) -> (Flu, no).

Rules are presented in the LERS format (every rule is equipped with three numbers, the total number of attribute-value pairs on the left-hand side of the rule, the total number of cases correctly classified by the rule during training, and the total number of training cases matching the left-hand side of the rule).

The above algorithm is a basis for the algorithm LEM1 (Learning from Examples, Module 1) of the data mining system LERS (Learning from Examples based on Rough Sets), see (Grzymala-Busse 1992).

2.2.2 Local Coverings

First we will introduce the idea of a minimal complex, which corresponds to a single rule. Let X be a concept. Let t be an attribute-value pair (a, v), and let T be a set of attribute-value pairs. Then the *block* of t, denoted $[t]$, is the set of examples for which attribute a has value v. Set X depends on a set T of attribute-value pairs, if and only if

$$\emptyset \neq \cap \{[t] \mid t \in T\} \subseteq X$$

A set T is a *minimal complex* of X if and only if X depends on T and no proper subset T' of T exists such that X depends on T'.

A minimal complex for {1, 2, 4} (i.e., for [(Flu, yes)]) is {(Temperature, high), Headache, yes)}.

Now we may introduce a local covering, which corresponds to a rule set describing a concept. Similarly to global coverings, local coverings are also useful for rule induction. Let $\mathbb{T}$ be a non-empty collection of non-empty sets of attribute value pairs. Then $\mathbb{T}$ is a *local covering* of X if and only if the following conditions are satisfied:

1. Each member T of $\mathbb{T}$ is a minimal complex of X,

2. $\cup\{T \mid T \in \mathbb{T}\} = X$, and

3. $\mathbb{T}$ is minimal, i.e., $\mathbb{T}$ has the smallest possible number of members.

An example of a local covering for {1, 2, 4} (i.e., for [(Flu, yes)]) is {{(Temperature, high), Headache, yes)}, {Temperature, very_high}}.

The algorithm LEM2 (Learning from Examples Module, version 2) for rule induction is based on computing a single local covering for each concept from a decision table (Chan and Grzymala-Busse 1994, Grzymala-Busse 1992). The user may select an option of LEM2 with or without taking into account attribute priorities. When LEM2 does not to take attribute priorities into account, the first criterion is ignored of the following procedure. The procedure LEM2 with attribute priority is presented below.

Procedure LEM2
  (**input:** a set X;
  output: a single local covering $\mathbb{T}$ of set X);
  begin
    $G := X$;
    $\mathbb{T} := \emptyset$;
    while $G \neq \emptyset$ **do**
      begin
      $T := \emptyset$
      $T(G) := \{t \mid [t] \cap G \neq \emptyset\}$;
      while $T = \emptyset$ **or not** ($[T] \subseteq X$)
        begin
          select a pair $t \in T(G)$ with the highest attribute priority, if a tie occurs, select a pair $t \in T(G)$ such that $|[t] \cap G|$ is maximum; if another tie occurs, select a pair $t \in T(G)$ with the smallest cardinality of $[t]$; if a further tie occurs, select first pair;
          $T := T \cup \{t\}$;
          $G := [t] \cap G$;
          $T(G) := \{t \mid [t] \cap G \neq \emptyset\}$;
          $T(G) := T(G) - T$;
        end; {while}
      for each t in T **do**
        if $[T - \{t\}] \subseteq X$ **then** $T := T - \{t\}$;
      $\mathbb{T} := \mathbb{T} \cup \{T\}$;
      $G := X - \bigcup_{T \in \mathbb{T}} [T]$;
    end {while};
    for each $T \in \mathbb{T}$ **do**
      if $\bigcup_{S \in \mathbb{T} - \{T\}} [S] = X$ **then** $\mathbb{T} := \mathbb{T} - \{T\}$;
  end {procedure}.

The algorithm LEM2 induced the following rule set, presented in the LERS format, from Table 2

2, 2, 2
(Headache, yes) & (Temperature, high) -> (Flu, yes)

1, 1, 1
(Temperature, very_high) -> (Flu, yes)

1, 2, 2
(Headache, no) -> (Flu, no)

1, 2, 2
(Temperature, normal) -> (Flu, no)

3 Inconsistent Data

In many cases data mining is performed on inconsistent data, i.e., on data in which there exist cases that are characterized by the same attribute values yet they were classified as members of different concepts. Such cases are called conflicting. Usually it means that in the data set some attributes are missing. Rough set theory is a perfect tool to handle inconsistent data (Pawlak 1991; Pawlak *et al*. 1995). Using rough set theory, conflicting cases are not removed from the data set. Instead, concepts are approximated by new sets called lower and upper approximations. An example of the inconsistent data set, taken from (Grzymala-Busse 2004c), is presented in Table 3.

Table 3. An inconsistent decision table

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	high	no	no	no
4	high	yes	yes	yes
5	high	yes	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	normal	yes	no	yes

A decision table is inconsistent if and only if

$$A^* \leq \{d\}^* \text{ is false.}$$

In the example from Table 3,

$$\{\text{Temperature, Headache, Nausea}\}^* = \{\{1\}, \{2\}, \{3\}, \{4, 5\}, \{6, 8\}, \{7\}\},$$

$$\{\text{Flu}\}^* = \{(1, 2, 4, 8\}, \{3, 5, 6, 7\}\},$$

so

$$\{\text{Temperature, Headache, Nausea}\}^* \leq \{\text{Flu}\}^* \text{ is false.}$$

The decision table from Table 3 is inconsistent since there exist conflicting cases 4 and 5 (and, independently, conflicting cases 6 and 8).

An obvious question is what subsets of the set U of all cases can be uniquely identified using attributes. Let B be a subset of the set A of attributes. Any union of elementary sets of B is called a *B-definable* set. An example of an A-definable set is {2, 3, 4, 5}. However, neither of the two concepts: {1, 2, 4, 8} and {3, 5, 6, 7} is A-definable. In rough set theory, for every concept X, two subsets of the set U of all cases are defined: lower and upper approximations. There exist two definitions of these approximations. For data sets in which all values are specified (i.e., no attribute value is missing), as in Table 3, these two definitions result in the same sets (Pawlak, 1991). According to the first definition, a *B-lower approximation* of X, denoted by $\underline{B}X$, is equal to the following set

$$\{x \in U \mid [x]_B \subseteq X\},$$

and a *B-upper approximation* of X, denoted by $\overline{B}X$, is equal to the following set

$$\{x \in U \mid [x]_B \cap X \neq \emptyset\}.$$

The second definition of lower and upper approximations provides the following formulas

$$\underline{B}X = \cup\{[x]_B \mid x \in U, [x]_B \subseteq X\},$$

and

$$\overline{B}X = \cup\{[x]_B \mid x \in U, [x]_B \cap X \neq \emptyset\}.$$

Thus, for both concepts from Table 3, {1, 2, 4, 8} and {3, 5, 6, 7}, lower and upper approximations are

$$\underline{A}\{1, 2, 4, 8\} = \{1, 2\},$$

$$\underline{A}\{3, 5, 6, 7\} = \{3, 7\},$$

$$\overline{A}\,\{1, 2, 4, 8\} = \{1, 2, 4, 5, 6, 8\}, \text{ and}$$

$$\overline{A}\,\{3, 5, 6, 7\} = \{3, 4, 5, 6, 7, 8\}.$$

Let Y be any subset of the set U of all cases and let B be a subset of the set A of all attributes. The lower approximation $\underline{B}\,Y$ is the largest definable set contained in Y. On the other hand, the upper approximation $\overline{B}\,Y$ is the smallest definable set containing Y. Moreover,

$$\underline{B}\,Y \subseteq Y \subseteq \overline{B}\,Y$$

hence any case from $\underline{B}\,Y$ is *certainly* a member of Y, while any member of $\overline{B}\,Y$ is *possibly* (or *plausibly*) a member of Y.

The above idea is utilized for data mining. For example, the LERS data mining system handles computing lower and upper approximation for every concept, and then inducing certain rule sets from lower approximations and possible rule sets from upper approximations. For details of LERS see, e.g., (Grzymala-Busse 1992, 1997). The performance of LERS is fully comparable with performance of AQ15 (Michalski et al. 1986) and C4.5 (Quinlan 1993), see Table 4.

Table 4. Performance of AQ15, C4.5 and LERS

Data set	Error rate		
	AQ15	C4.5	LERS
Lymphography	18–20%	23%	19%
Breast cancer	32–34%	33%	30%
Primary tumor	59–71%	60%	67%

Recently (Grzymala-Busse 2002) LEM2 algorithm was enhanced by a new version of LEM2, called MLEM2 (Modified Learning from Examples Module, version 2). MLEM2 induces rule sets directly from data with symbolic and numerical attributes, while LEM2 needs pre-discretized data. Additionally, MLEM2 induces rule sets from data with missing attribute values, see the next section.

The algorithm LEM2 induced the following rule set from the decision table presented in Table 3

Certain rule set:

1, 1, 1
(Temperature, very_high) -> (Flu, yes),

3, 1, 1
(Temperature, high) & (Nausea, no) & (Headache, yes) -> (Flu, yes),

1, 2, 2
(Headache, no) -> (Flu, no),

and possible rule set:

1, 4, 6
(Headache, yes) -> (Flu, yes),

1, 2, 3
(Temperature, normal) -> (Flu, no),

2, 1, 2
(Temperature, high) & (Nausea, yes)-> (Flu, no),

1, 2, 2
(Headache, no) -> (Flu, no).

4 Incomplete Data

Input data sets are frequently incomplete, i.e, some attribute values are missing. In other words, corresponding decision tables are incomplete. In general, in data mining two approaches are used to handle incomplete data:

- pre-processing of input data and then main processing of data mining such as rule induction. Typically, preprocessing means replacing missing attribute values by the most common value, ignoring cases with missing attribute values, etc (Grzymala-Busse and Hu 2001).
- knowledge is acquired directly from incomplete data sets taking into account that some attribute values are mising. Typical systems using this approach are C4.5 and MLEM2.

In this paper we will apply the latter approach, i.e., rule induction is performed directly from incomplete data. Furthermore, we will assume that there are two reasons for data to be incomplete. The first reason is that an attribute value, for a specific case, is lost. This may happen when the original value was erased or mistakenly not included into the data set. The second reason for incompleteness is based on the lack of relevance, e.g., given case was diagnosed on the basis of some attribute values while other attribute values were irrelevant. For example, it was possible to diagnose a patient using only selected results of tests (attributes), while other test results were redundant. Such missing attribute values will be called "do not care" conditions. We will assume that in the same decision table some attribute values are lost and some are "do not care" conditions. Such decision tables were studied in (Grzymala-Busse 2003).

Incomplete data with lost attribute values, from the viewpoint of rough set theory, were studied for the first time in (Grzymala-Busse and Wang 1997). In this paper two algorithms for rule induction, modified to handle lost attribute values, were presented. This approach was studied later in (Stefanowski 2001; Stefanowski and Tsoukias 1999, 2001), where the indiscernibility relation was generalized to describe such incomplete data.

On the other hand, incomplete data in which all missing attribute values are "do not care" conditions, again from the view point of rough set theory, were studied for the first time in (Grzymala-Busse 1991b), where a method for rule induction was introduced in which each missing attribute value was replaced by all values from the domain of the attribute. Originally such values were replaced by all values from the entire domain of the attribute, later by attribute values restricted to the same concept to which a case with a missing attribute value belongs. Such incomplete decision tables, with all missing attribute values being "do not care conditions", were extensively studied in (Kryszkiewicz 1995, 1999), including extending the idea of the indiscernibility relation to describe such incomplete decision tables.

In general, incomplete decision tables are described by characteristic relations, in a similar way as complete decision tables are described by indiscernibility relations (Grzymala-Busse 2003, 2004a, 2004b; Grzymala-Busse and Siddhaye 2004). Also, elementary sets are replaced by characteristic sets. For complete decision tables, once the indiscernibility relation is fixed and the concept (a set of cases) is given, the lower and upper approximations are unique. For incomplete decision tables, for a given characteristic relation and the concept, there are three different possible ways to define lower and upper approximations, called singleton, subset, and concept approximations (Grzymala-Busse 2003). The singleton lower and upper approximations were studied in (Kryszkiewicz 1995, 1999; Stefanowski 2001; Stefanowski and Tsoukias 1999, 2001). Similar ideas were studied in (Greco *et al*. 2000; Lin 2001; Slowinski and Vanderpooten 2000; Yao 1996, 1998, 2003). As it was observed in (Grzymala-Busse 2003), singleton lower and upper approximations are not applicable in data mining.

4.1 Blocks of Attribute-Value Pairs, Characteristic Sets, and Characteristic Relation

In the sequel we will assume that all decision values are specified, i.e., they are not missing. Also, we will assume that all missing attribute values are denoted by "?" and by"*", lost values will be denoted by "?" and "do not care" conditions will be denoted by "*". Additionally, we will assume that for each case at least one attribute value is specified. An example of an incomplete table, taken from (Grzymala-Busse 2004c), is presented in Table 5.

Table 5. An incomplete decision table

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	?	no	yes
2	very_high	yes	yes	yes
3	?	no	no	no
4	high	yes	yes	yes
5	high	?	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	*	yes	*	yes

For incomplete decision tables the definition of a block of an attribute-value pair must be modified.

- If an attribute a there exists a case x such that $\rho(x, a) = ?$, i.e., the corresponding value is lost, then the case x should not be included in any block $[(a, v)]$ for all values v of attribute a.
- If for an attribute a there exists a case x such that the corresponding value is a "do not care" condition, i.e., $\rho(x, a) = *$, then the corresponding case x should be included in blocks $[(a, v)]$ for all specified values v of attribute a.

For the example of an incomplete data set from Table 5,

[(Temperature, high)] = {1, 4, 5, 8},
[(Temperature, very_high)] = {2, 8},
[(Temperature, normal)] = {6, 7, 8},
[(Headache, yes)] = {2, 4, 6, 8},
[(Headache, no)] = {3, 7},
[(Nausea, no)] = {1, 3, 6, 8},
[(Nausea, yes)] = {2, 4, 5, 7, 8}.

The *characteristic set* $K_B(x)$ is the intersection of blocks of attribute-value pairs (a, v) for all attributes a from B for which $\rho(x, a)$ is specified and $\rho(x, a) = v$. For Table 2 and $B = A$,

$K_A(1) = \{1, 4, 5, 8\} \cap \{1, 3, 6, 8\} = \{1, 8\}$,
$K_A(2) = \{2, 8\} \cap \{2, 4, 6, 8\} \cap \{2, 4, 5, 7, 8\} = \{2, 8\}$,
$K_A(3) = \{3, 7\} \cap \{1, 3, 6, 8\} = \{3\}$,
$K_A(4) = \{1, 4, 5, 8\} \cap \{2, 4, 6, 8\} \cap \{2, 4, 5, 7, 8\} = \{4, 8\}$,
$K_A(5) = \{1, 4, 5, 8\} \cap \{2, 4, 5, 7, 8\} = \{4, 5, 8\}$,
$K_A(6) = \{6, 7, 8\} \cap \{2, 4, 6, 8\} \cap \{1, 3, 6, 8\} = \{6, 8\}$,
$K_A(7) = \{6, 7, 8\} \cap \{3, 7\} \cap \{2, 4, 5, 7, 8\} = \{7\}$, and
$K_A(8) = \{2, 4, 6, 8\}$.

Characteristic set $K_B(x)$ may be interpreted as the smallest set of cases that are indistinguishable from x using all attributes from B and using a given interpretation of missing attribute values. Thus, $K_A(x)$ is the set of all cases that cannot be distinguished from x using all attributes.

The *characteristic relation* $R(B)$ is a relation on U defined for $x, y \in U$ as follows

$$(x, y) \in R(B) \text{ if and only if } y \in K_B(x).$$

The characteristic relation $R(B)$ is reflexive but—in general—does not need to be symmetric or transitive. Also, the characteristic relation $R(B)$ is known if we know characteristic sets $K_B(x)$ for all $x \in U$. In our example,

$R(A)$ = {(1, 1), (1, 8), (2, 2), (2, 8), (3, 3), (4, 4), (4, 8), (5, 4), (5, 5), (5, 8), (6, 6), (6, 8), (7, 7), (8, 2), (8, 4), (8, 6), (8, 8)}.

4.2 Approximations

The definition of definability of completely specified data should be modified to cover incomplete data. There exist two different ways to define definable sets. According to the first definition, a union of characteristic sets, associated with B, will be called a *B-globally-definable* set. The second definition of definability is different: a union of intersections of blocks of attribute-value pairs will be called a *B-locally-definable* set. In the example of Table 5, the set {7, 8} is *A*-locally-definable since it is equal to the intersection of [(Temperature, normal)] and [(Nausea, yes)]. Nevertheless, {7, 8} is not *A*-globally-definable. Obviously, any *B*-globally definable set is a *B*-locally definable set, but the converse is false. *A*-locally definable sets have the following important property: they can be described using rules.

Let X be a concept. In general, X is not *B*-globally-definable. As for completely specified decision tables, set X may be approximated by two *B*-globally-definable sets, a *B-lower approximation of X*, denoted by $\underline{B}X$ and a *B-upper approximation of X*, denoted by $\overline{B}X$.

However, for incompletely specified decision tables lower and upper approximations may be defined in a few different ways. Following (Grzymala-Busse 2003), we are going to define three different approximations. Our first definition uses a similar idea as in the previous articles on incompletely specified decision tables (Kryszkiewicz 1995, 1999; Stefanowski 2001; Stefanowski and Tsoukias 1999, 2001), i.e., lower and upper approximations are sets of singletons from the universe U satisfying some properties. Thus we are defining lower and upper approximations by analogy with the first definition, by constructing both sets from singletons. We will call these definitions *singleton*. A singleton *B*-lower approximation of X is defined as follows:

$$\underline{B}X = \{x \in U \mid K_B(x) \subseteq X\}.$$

A singleton B-upper approximation of X is

$$\overline{B}X = \{x \in U \mid K_B(x) \cap X \neq \emptyset\}.$$

In our example presented in Table 2 let us say that $B = A$. Then the singleton A-lower and A-upper approximations of the two concepts: {1, 2, 4, 8} and {3, 5, 6, 7} are:

$\underline{A}\{1, 2, 4, 8\} = \{1, 2, 4\}$,
$\underline{A}\{3, 5, 6, 7\} = \{3, 7\}$,
$\overline{A}\{1, 2, 4, 8\} = \{1, 2, 4, 5, 6, 8\}$,
$\overline{A}\{3, 5, 6, 7\} = \{3, 5, 6, 7, 8\}$.

Note that $\underline{A}\{1, 2, 4, 8\} = \{1, 2, 4\}$. But the set {1, 2, 4} is not A-globally-definable. Furthermore, the set {1, 2, 4} is not even A-locally-definable, so no set of rules can cover precisely this set. Similarly, $\overline{A}\{3, 5, 6, 7\} = \{3, 5, 6, 7, 8\}$, and the set {3, 5, 6, 7, 8} is also not A-locally-definable. Therefore, in general, singleton approximations should not be used for data mining.

The second method of defining lower and upper approximations for complete decision tables uses another idea: lower and upper approximations are unions of elementary sets, subsets of U (Grzymala-Busse 2003). Therefore we may define lower and upper approximations for incomplete decision tables by analogy with the second definition of approximations for completely specified data, using characteristic sets instead of elementary sets. There are two ways to do this. Using the first way, a *subset* B-lower approximation of X is defined as follows:

$$\underline{B}X = \cup\{K_B(x) \mid x \in U, K_B(x) \subseteq X\}.$$

A *subset* B-upper approximation of X is

$$\overline{B}X = \cup\{K_B(x) \mid x \in U, K_B(x) \cap X \neq \emptyset\}.$$

Since any characteristic relation $R(B)$ is reflexive, for any concept X, singleton B-lower and B-upper approximations of X are subsets of subset B-lower and B-upper approximations of X, respectively. For the same the decision presented in Table 2, the subset A-lower and A-upper approximations are:

$\underline{A}\{1, 2, 4, 8\} = \{1, 2, 4, 8\}$,
$\underline{A}\{3, 5, 6, 7\} = \{3, 7\}$,
$\overline{A}\{1, 2, 4, 8\} = \{1, 2, 4, 5, 6, 8\}$,
$\overline{A}\{3, 5, 6, 7\} = \{2, 3, 4, 5, 6, 7, 8\}$.

The second possibility is to modify the subset definition of lower and upper approximation by replacing the universe *U* from the subset definition by a concept *X*. A *concept B*-lower approximation of the concept *X* is defined as follows:

$$\underline{B}X = \cup\{K_B(x) \mid x \in X, K_B(x) \subseteq X\}.$$

Obviously, the subset *B*-lower approximation of *X* is the same set as the concept *B*-lower approximation of *X*. A *concept B*-upper approximation of the concept *X* is defined as follows:

$$\overline{B}X = \cup\{K_B(x) \mid x \in X, K_B(x) \cap X \neq \emptyset\} = \cup\{K_B(x) \mid x \in X\}.$$

The concept *B*-upper approximation of *X* are subsets of the subset *B*-upper approximations of *X*. For the decision presented in Table 2, the concept *A*-lower and *A*-upper approximations are:

$\underline{A}\{1, 2, 4, 8\} = \{1, 2, 4, 8\}$,
$\underline{A}\{3, 5, 6, 7\} = \{3, 7\}$,
$\overline{A}\{1, 2, 4, 8\} = \{1, 2, 4, 6, 8\}$,
$\overline{A}\{3, 5, 6, 7\} = \{3, 4, 5, 6, 7, 8\}$.

For complete decision tables, all three definitions of lower approximations, singleton, subset and concept, coalesce to the same definition. Also, for complete decision tables, all three definitions of upper approximations coalesce to the same definition. This is not true for incomplete decision tables, as our example shows.

Singleton *B*-lower and *B*-upper approximations of the set *X* are subsets of the subset *B*-lower and *B*-upper approximations of *X*, respectively. The subset *B*-lower approximation of *X* is the same set as the concept *B*-lower approximation of *X*. The concept *B*-upper approximation of *X* is a subset of the subset *B*-upper approximation of *X*.

Rules in LERS format induced from Table 5 using concept approximations are:

the certain rule set:

2, 2, 2
(Temperature, high) & (Nausea, no) -> (Flu, yes)

2, 3, 3
(Headache, yes) & (Nausea, yes) -> (Flu, yes)

1, 2, 2
(Headache, no) -> (Flu, no)

and the possible rule set:

2, 2, 2

(Temperature, high) & (Nausea, no) -> (Flu, yes)

1, 3, 4
(Headache, yes) -> (Flu, yes)

2, 1, 3
(Temperature, high) & (Nausea, yes) -> (Flu, no)

1, 2, 3
(Temperature, normal) -> (Flu, no)

1, 2, 2
(Headache, no) -> (Flu, no)

5 Final Remarks

A successful example of rough set theory application to data mining is the LERS data mining system. The machine learning/ data mining system LERS has proven its applicability having been used for years by NASA Johnson Space Center (Automation and Robotics Division), as a tool to develop expert systems of the type most likely to be used in medical decision - making on board the International Space Station. LERS was also used to enhance facility compliance under Sections 311, 312, and 313 of Title III, the Emergency Planning and Community Right to Know. The project was funded by the U. S. Environmental Protection Agency. LERS was used in other areas as well, e.g., in the medical field to assess preterm labor risk for pregnant women and to compare the effects of warming devices for postoperative patients. Currently used traditional methods to assess preterm labor risk have positive predictive value (the ratio of all true positives to the sum of all true positives and false positives) between 17 and 38%, while the expert systems with the rule sets induced by LERS have positive predictive value between 59 and 93%. Moreover, LERS was successfully applied to diagnosis of melanoma, to prediction of behavior under mental retardation, analysis of animal models for prediction of self-injurious behavior, global warming, natural language and data transmission.

References

Chan C-C, Grzymala-Busse JW (1994) On the two local inductive algorithms: PRISM and LEM2. *Foundations of Computing and Decision Sciences* 19: 185–203.

Greco S, Matarazzo B, Slowinski R (2000) Dealing with missing data in rough set analysis of multi-attribute and multi-criteria decision problems. In Zanakis SH, Doukidis G, Zopounidis Z (eds.) *Decision Making: Recent developments and Worldwide Applications*, Kluwer Academic Publishers, Boston Dordrecht, London, 295–316.

Grzymala-Busse JW (1988). Knowledge acquisition under uncertainty—A rough set approach. *Journal of Intelligent & Robotic Systems* 1: 3–16.

Grzymala-Busse JW (1991a) Managing Uncertainty in Expert Systems, Kluwer Acad. Publ., Boston/ Dordrecht London Vol. 143 of the Kluwer International Series in Engineering and Computer Science.

Grzymala-Busse JW (1991b) On the unknown attribute values in learning from examples. Proc. of the ISMIS-91, 6th International Symposium on Methodologies for Intelligent Systems, Charlotte, North Carolina, October 16–19, 1991, 368–377, *Lecture Notes in Artificial Intelligence*, vol. 542, Springer-Verlag, Berlin, Heidelberg, New York.

Grzymala-Busse JW (1992) LERS—A system for learning from examples based on rough sets. In Slowinski R. (ed.) *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer, 3–18.

Grzymala-Busse JW (1997) A new version of the rule induction system LERS. *Fundamenta Informaticae* 31: 27–39.

Grzymala-Busse JW (2002) MLEM2: A new algorithm for rule induction from imperfect data. Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, Annecy, France, 243–250.

Grzymala-Busse JW (2003). Rough set strategies to data with missing attribute values. Proceedings of the Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining, Melbourne, FL, 56–63.

Grzymala-Busse JW (2004a) Characteristic relations for incomplete data: A generalization of the indiscernibility relation. Proceedings of the RSCTC'2004, the Fourth International Conference on Rough Sets and Current Trends in Computing, Uppsala, Sweden,. Lecture Notes in Artificial Intelligence 3066, Springer-Verlag 2004, 244–253.

Grzymala-Busse JW (2004b) Rough set approach to incomplete data. Proceedings of the ICAISC'2004, the Seventh International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland. Lecture Notes in Artificial Intelligence 3070, Springer-Verlag 2004, 50–55.

Grzymala-Busse JW (2004c) Data with missing attribute values: Generalization of idiscernibility relation and rule induction. *Transactions on Rough Sets*, Lecture Notes in Computer Science Journal Subline, Springer-Verlag, 1: 78–95.

Grzymala-Busse JW, Hu M (2001) A comparison of several approaches to missing attribute values in data mining. Proceedings of the Second International Conference, RSCTC'2000, Banff, Canada, Revised Papers. Lecture Notes in Artificial Intelligence, 2005, Subseries of Lecture Notes in Computer Science, Springer Verlag, 2001, 378–385.

Grzymala-Busse JW, Siddhaye S (2004). Rough set approaches to rule induction from incomplete data. Proceedings of the IPMU'2004, the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Perugia, Italy, vol. 2, 923–930.

Grzymala-Busse JW, A. Y. Wang AY (1997) Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. Proc. of the Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97) at the Third Joint Conference on Information Sciences (JCIS'97), Research Triangle Park, NC, 69–72.

Kryszkiewicz M (1995) Rough set approach to incomplete information systems. Proceedings of the Second Annual Joint Conference on Information Sciences,Wrightsville Beach, NC, 194–197.

Kryszkiewicz M (1999) Rules in incomplete information systems. *Information Sciences* 113: 271–292.

Lin TY (1989) Neighborhood systems and approximation in database and knowledge base systems. Fourth International Symposium on Methodologies of Intelligent Systems (Poster Sessions), Charlotte, North Carolina, 75–86.

Michalski RS, Mozetic I, Hong J, Lavrac N (1986) The multi-purpose incremental learning system AQ15 and its testing appliaction to three medical domains. Proc. of the Nat. Conf. on AI, 1041–1045.

Pawlak Z (1982) Rough Sets. *Int. J. of Computer and Information Sciences* 11: 341–356.

Pawlak Z. (1991) *Rough Sets. Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Boston Dordrecht, London.

Pawlak Z, Grzymala-Busse JW, Slowinski R, Ziarko W (1995) Rough Sets. *Communications of the ACM* 38: 89–95.

Quinlan JR (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

Slowinski R, Vanderpooten D (2000) A generalized definition of rough approximations based on similarity. *IEEE Transactions on Knowledge and Data Engineering* 12: 331–336.

Stefanowski J (2001) *Algorithms of Decision Rule Induction in Data Mining*. Poznan University of Technology Press, Poznan, Poland.

Stefanowski J Tsoukias A (1999) On the extension of rough sets under incomplete information. Proceedings of the 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, RSFDGrC'1999, Yamaguchi, Japan, 73–81.

Stefanowski J Tsoukias A (2001) Incomplete information tables and rough classification. *Computational Intelligence* 17: 545–566.

Yao YY (1996) Two views of the theory of rough sets in finite universes. *International J. of Approximate Reasoning* 15: 291–317.

Yao YY (1998) Relational interpretations of neighborhood operators and rough set approximation operators. *Information Sciences* 111: 239–259.

Yao YY (2003) On the generalizing rough set theory. Proc. of the 9th Int. Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'2003), Chongqing, China, 44–51.