Chapter 1

Rule Induction from Rough Approximations

Rule induction is an important technique of data mining or machine learning. Knowledge is frequently expressed by rules in many areas of AI, including rule-based expert systems. In this chapter we discuss only *supervised learning* in which all cases of the input data set are pre-classified by an expert.

1.1 Complete and Consistent Data

Our basic assumption is that the data sets are presented as decision tables. An example of the decision table is presented in Table 1.1. Rows of the decision table represent *cases*, columns represent *variables*. The set of all cases is denoted by U. For Table 1.1, $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Some variables are called *attributes* while one selected variable is called a *decision* and is denoted by d. The set of all attributes will be denoted by A. In Table 1.1, $A = \{$*Wind*, *Humidity*, *Temperature*$\}$ and $d =$ *Trip*. For an attribute a and case x, $a(x)$ denotes the value of the attribute a for case x. For example, *Wind* $(1) =$ *low*.

Let B be a subset of the set A of all attributes. Complete data sets are characterized by the indiscernibility relation $IND(B)$ [1, 2] defined as follows: for any $x, y \in U$,

$$(x, y) \in IND(B) \text{ if and only if } a(x) = a(y) \text{ for any } a \in B \tag{1.1}$$

Obviously, $IND(B)$ is an equivalence relation. The equivalence class of $IND(B)$ containing $x \in U$ will be denoted by $[x]_B$ and called *B-elementary* set. A-elementary sets will be called *elementary*. Any union of B-elementary sets will be called a *B-definable* set. By analogy, A-definable set will be called definable. The elementary sets of the partition $\{d\}^*$ are called *concepts*. In

Table 1.1: A complete and consistent decision table

	Attributes			Decision
Case	Wind	Humidity	Temperature	Trip
1	low	low	medium	yes
2	low	low	low	yes
3	low	medium	medium	yes
4	low	medium	high	maybe
5	medium	low	medium	maybe
6	medium	high	low	no
7	high	high	high	no
8	medium	high	high	no

Table 1.1, concepts are {1, 2, 3}, {4, 5} and {6, 7, 8}. The set of all equivalence classes $[x]_B$, where $x \in U$, is a partition on U denoted by B^*. For Table 1.1, A^* = {{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}}. All members of A^* are elementary sets.

We will quote some definitions from [3]. A rule r is an expression of the following form

$$(a_1, v_1)\&(a_2, v_2)\& \ldots \&(a_k, v_k) \rightarrow (d, w), \tag{1.2}$$

where a_1, a_2,..., a_k are distinct attributes, d is a decision, v_1, v_2,..., v_k are respective attribute values, and w is a decision value.

A case x is *covered* by a rule r if and only if any attribute-value pair of r is satisfied by the corresponding value of x. For example, case 1 from Table 1.1 is covered by the following rule r:

$$(Wind, low)\ \&\ (Humidity, low)\ \rightarrow\ (Trip, yes).$$

The concept C defined by rule r is *indicated* by r. The above rule r indicates concept {1, 2, 3}.

A rule r is *consistent* with the data set if and only if for any case x covered by r, x is a member of the concept indicated by r. The above rule is consistent with the data set represented by Table 1.1. A rule set R is consistent with the data set if and only if for any $r \in R$, r is consistent with the data set. The rule set containing the above rule is consistent with the data set represented by Table 1.1.

We say that a concept C is *completely* covered by a rule set R if and only if for every case x from C there exists a rule r from R such that r covers x. For example, the single rule

$$(Wind, low) \rightarrow (Trip, yes)$$

completely covers the concept {1, 2, 3}. On the other hand, this rule is not consistent with the data set represented by Table 1.1.

A rule set R is *complete* for a data set if and only if every concept from the data set is completely covered by R.

In this chapter we will discuss how to induce rule sets that are complete and consistent with the data set.

1.1.1 Global Coverings

The simplest approach to rule induction is based on finding the smallest subset B of the set A of all attributes that is sufficient to be used in a rule set. Such reducing of the attribute set is one of the main and frequently used techniques in rough set theory [1, 2, 4]. This approach is also called a *feature selection*.

In Table 1.1 the attribute *Humidity* is redundant (irrelevant). Remaining two attributes (*Wind* and *Temperature*) distinguish all eight cases. Let us make it more precise using fundamental definitions of rough set theory [1, 2, 4].

For a decision d we say that $\{d\}$ depends on B if and only if $B^* \leq \{d\}^*$, i.e., for any elementary set X in B there exists a concept C from $\{d\}^*$ such that $X \subseteq C$. Note that for partitions π and τ on U, if for any $X \in \pi$ there exists $Y \in \tau$ such that $X \subseteq Y$ then we say that π is smaller than or equal to τ and denote it by $\pi \leq \tau$. A *global covering* (or *relative reduct*) of $\{d\}$ is a subset B of A such that $\{d\}$ depends on B and B is minimal in A. The algorithm to compute a single global covering is presented below.

Algorithm to compute a single global covering
(**input**: the set A of all attributes,
partition $\{d\}^*$ on U;
output: a single global covering R);
begin
compute partition A^*;
$P := A$;
$R := \emptyset$;
  if $A^* \leq \{d\}^*$
    then
      begin
      for each attribute a in A **do**
        begin
        $Q := P - \{a\}$;
        compute partition Q^*;
        if $Q^* \leq \{d\}^*$
          then $P := Q$
      end {for}
      $R := P$

end {then}
end {algorithm}.

Let us use this algorithm for Table 1.1.
First, $A^* = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\} \leq \{Trip\}^*$. Initially, $P = A$, and $Q = P - Wind$,

$Q = \{Humidity, Temperature\}$ and then we compute Q^*, where

$Q^* = \{\{1, 5\}, \{2\}, \{3\}, \{4\}, \{6\}, \{7, 8\}\}$. We find that $Q^* \not\leq \{Trip\}^*$. Thus, $P = A$. Next, we try to delete *Humidity* from P. We obtain

$Q = \{Wind, Temperature\}$ and then we compute Q^*, where $Q^* = \{\{1, 3\}, \{2\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\}$. This time $Q^* \leq \{Trip\}^*$, so $P = \{Wind, Temperature\}$.

It remains to check $Q = P - \{Temperature\}$, $Q = \{Wind\}$, $Q^* = \{\{1, 2, 3, 4\}, \{5, 6, 8\}, \{7\}\}$, and $Q^* \not\leq \{Trip\}^*$. Thus $R = \{Wind, Temperature\}$ is a global covering.

For a given global covering rules are induced by examining cases of the data set. Initially, such a rule contains all attributes from the global covering with the corresponding attribute values, then a *dropping conditions* technique is used: we are trying to drop one condition (attribute-value pair) at a time, starting from the leftmost condition, checking if the rule is still consistent with the data set, then we are trying to drop the next condition and so on. For example,

$$(Wind, low) \ \& \ (Temperature, medium) \rightarrow (Trip, yes)$$

is our first candidate for a rule. If we are going to drop the first condition, the above rule will be reduced to

$$(Temperature, medium) \rightarrow (Trip, yes),$$

However, this rule covers the case 5, so it is not consistent with the data set represented by Table 1.1. By dropping the second condition from the initial rule we obtain

$$(Wind, low) \rightarrow (Trip, yes),$$

but that rule is also not consistent with the data represented by Table 1.1 since it covers the case 4, so we conclude that the initial rule is the simplest possible. This rule covers two cases: 1 and 3.

It is not difficult to check that the rule

$$(Wind, low) \ \& \ (Temperature, low) \rightarrow (Trip, yes)$$

is as simple as possible and that it covers only case 2. Thus, the above two rules consistently and completely cover the concept {1, 2, 3}.

The above algorithm is implemented as LEM1 (Learning from Examples Module version 1). It is a component of the data mining system LERS (Learning from Examples using Rough Sets). A similar system was described in [5].

1.1.2 Local Coverings

The LEM1 algorithm is based on calculus on partitions on the entire universe U. Another approach to rule induction, based on attribute-value pairs, is presented in the LEM2 algorithm (Learning from Examples Module, version 2), another component of LERS. We will quote a few definitions from [6, 7].

For an attribute-value pair $(a,\ v) = t$, a *block* of t, denoted by $[t]$, is a set of all cases from U such that for attribute a have value v, i.e.,

$$[(a, v)] = \{x \mid a(x) = v\}. \tag{1.3}$$

Let T be a set of attribute-value pairs. The block of T, denoted by $[T]$, is the following set

$$\bigcap_{t \in T} [t]. \tag{1.4}$$

Let B be a subset of U. Set B *depends* on a set T of attribute-value pairs $t = (a, v)$ if and only if $[T]$ is nonempty and

$$[T] \subseteq B. \tag{1.5}$$

Set T is a *minimal complex* of B if and only if B depends on T and no proper subset T' of T exists such that B depends on T'. Let $\mathcal{T}$ be a nonempty collection of nonempty sets of attribute-value pairs. Then $\mathcal{T}$ is a *local covering* of B if and only if the following conditions are satisfied:

(1) each member T of $\mathcal{T}$ is a minimal complex of B,

(2) $\bigcup_{t \in \mathcal{T}} [T] = B$, and

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

An algorithm for finding a single local covering, called LEM2, is presented below. For a set X, $|X|$ denotes the cardinality of X.

LEM2
(**input**: a set B,
output: a single local covering $\mathcal{T}$ of set B);
begin
    $G := B$;
    $\mathcal{T} := \emptyset$;
    while $G \neq \emptyset$
        begin
        $T := \emptyset$;
        $T(G) := \{t | [t] \cap G \neq \emptyset\}$;

```
        while T = ∅ or [T] ⊈ B
            begin
                select a pair t ∈ T(G)
                such that |[t] ∩ G| is
                maximum; if a tie
                occurs, select a pair
                t ∈ T(G) with the
                smallest cardinality of [t];
                if another tie occurs,
                select first pair;
                T := T ∪ {t} ;
                G := [t] ∩ G ;
                T(G) := {t|[t] ∩ G ≠ ∅};
                T(G) := T(G) − T ;
            end {while}
        for each t ∈ T do
            if [T − {t}] ⊆ B
                then T := T − {t};
        T := T ∪ {T};
        G := B − ∪_{T∈T}[T];
    end {while};
    for each T ∈ T do
        if ⋃_{S∈T−{T}}[S] = B
            then T := T − {T};
end {procedure}.
```

We will trace the LEM2 algorithm applied to the following input set {1, 2, 3} = [(*Trip*, *yes*)]. The tracing of LEM2 is presented in the Tables 1.2 and 1.3. The corresponding comments are:

1. The set $G = \{1, 2, 3\}$. The best attribute-value pair t, with the largest cardinality of the intersection of $[t]$ and G (presented in the third column of Table 1.2) is (*Wind*, *low*). The corresponding entry in the third column of Table 1.2 is bulleted. However, $[(\textit{Wind}, \textit{low})] = \{1, 2, 3, 4\} \not\subseteq \{1, 2, 3\} = B$, hence we need to look for the next t,

2. the set G is the same, $G = \{1, 2, 3\}$. There are three attribute-value pairs with $|[t \cap G| = 2$. Two of them have the same cardinality of $[t]$, so we select the first (top) pair, (*Humidity*, *low*). This time $\{1, 2, 3, 4\} \cap \{1, 2, 5\} = \{1, 2\} \subseteq \{1, 2, 3\}$, so {(*Wind*, *low*), (*Humidity*, *low*)} is the first element T of $\mathcal{T}$,

3. the new set $G = B - [T] = \{1, 2, 3\} - \{1, 2\} = \{3\}$. The pair [(*Humidity*, *medium*) has the smallest cardinality of $[t]$, so it is the best choice. However, [(*Humidity*, *medium*)] ={3, 4} $\not\subseteq$ {1, 2, 3}, hence we need to look for the next t,

4. the pair (*Temperature*, *medium*) is the best choice, and {3, 4} ∩ {1, 3,

Table 1.2: Computing a local covering for the concept [(*Trip*, *yes*)], part I

$(a, v) = t$	$[(a, v)]$	{1, 2, 3}	{1, 2, 3}
(*Wind*, *low*)	{1, 2, 3, 4}	{1, 2, 3} •	–
(*Wind*, *medium*)	{5, 6, 8}	–	–
(*Wind*, *high*)	{7}	–	–
(*Humidity*, *low*)	{1, 2, 5}	{1, 2}	{1, 2} •
(*Humidity*, *medium*)	{3, 4}	{3}	{3}
(*Humidity*, *high*)	{6, 7, 8}	–	–
(*Temperature*, *low*)	{2, 6}	{2}	{1, 3}
(*Temperature*, *medium*)	{1, 3, 5}	{1, 3}	{1, 3}
(*Temperature*, *high*)	{4, 7, 8}	–	–
Comments		1	2

Table 1.3: Computing a local covering for the concept [(*Trip*, *yes*)], part II

$(a, v) = t$	$[(a, v)]$	{3}	{3}
(*Wind*, *low*)	{1, 2, 3, 4}	{3}	{3}
(*Wind*, *medium*)	{5, 6, 8}	–	–
(*Wind*, *high*)	{7}	–	–
(*Humidity*, *low*)	{1, 2, 5}	–	–
(*Humidity*, *medium*)	{3, 4}	{3} •	–
(*Humidity*, *high*)	{6, 7, 8}	–	–
(*Temperature*, *low*)	{2, 6}	–	–
(*Temperature*, *medium*)	{1, 3, 5}	{3}	{3} •
(*Temperature*, *high*)	{4, 7, 8}	–	–
Comments		3	4

5} = {3} ⊆ {1, 2, 3}, so {(*Humidity*, *medium*), (*Temperature*, *medium*)} is the second element T of $\mathcal{T}$.

Thus, $\mathcal{T}$ = { {(*Wind*, *low*), (*Humidity*, *low*)}, {(*Humidity*, *medium*), (*Temperature*, *medium*)}}. Therefore, the LEM2 algorithm induces the following rule set

$$(Wind, low)\ \&(Humidity, low) \rightarrow (Trip, yes)$$

$$(Humidity, medium)\ \& (Temperature, medium) \rightarrow (Trip, yes)$$

Rules induced from local coverings differ from rules induced from global coverings. In many cases the former are simpler than the latter. For example, for Table 1.1 and the concept [(*Trip*, *no*)], the LEM2 algorithm would induce just one rule that covers all three cases

$$(Humidity, high) \rightarrow (Trip, no).$$

On the other hand, the attribute *Humidity* is not included in the global covering. The rules induced from the global covering are

$$(Temperature, high) \rightarrow (Trip, no).$$

$$(Wind, medium)\ \&\ (Temperature, low) \rightarrow (Trip, no).$$

1.1.3 Classification

Rule sets, induced from data sets, are used most frequently to classify new, unseen cases. A *classification system* has two inputs: a rule set and a data set containing new cases and it classifies every case as being member of some concept. A classification system used in LERS is a modification of the well-known bucket brigade algorithm [7, 8, 9].

The decision to which concept a case belongs is made on the basis of three factors: *strength*, *specificity*, and *support*. These factors are defined as follows: *strength* is the total number of cases correctly classified by the rule during training. *Specificity* is the total number of attribute-value pairs on the left-hand side of the rule. The matching rules with a larger number of attribute-value pairs are considered more specific. The third factor, *support*, is defined as the sum of products of strength and specificity for all matching rules indicating the same concept. The concept C for which the support, i.e., the following expression

$$\sum_{matching\ rules\ r\ describing\ C} Strength(r) * Specificity(r) \tag{1.6}$$

is the largest is the winner and the case is classified as being a member of C.

In the classification system of LERS, if complete matching is impossible, all partially matching rules are identified. These are rules with at least one attribute-value pair matching the corresponding attribute-value pair of a case. For any partially matching rule r, the additional factor, called *Matching_factor* (r), is computed. Matching_factor (r) is defined as the ratio of the number of matched attribute-value pairs of r with a case to the total number of attribute-value pairs of r. In partial matching, the concept C for which the following expression

$$\sum_{\substack{partially\ matching \\ rules\ r\ describing\ C}} Matching\_factor(r) * Strength(r) * Specificity(r) \tag{1.7}$$

is the largest is the winner and the case is classified as being a member of C.

Since the classification system is a part of the LERS data mining system, rules induced by any component of LERS, such as LEM1 or LEM2, are presented in the LERS format, in which every rule is associated with three numbers: the total number of attribute-value pairs on the left-hand side of the rule (i.e., specificity), the total number of cases correctly classified by the rule during training (i.e., strength), and the total number of training cases matching the left-hand side of the rule, i.e., the rule domain size.

1.2 Inconsistent Data

Frequently data sets contain conflicting cases, i.e., cases with the same attribute values but from different concepts. An example of such a data set is presented in Table 1.4. Cases 4 and 5 have the same values for all three attributes yet their decision values are different (they belong to different concepts). Similarly, cases 7 and 8 are also conflicting. Rough set theory handles inconsistent data by introducing lower and upper approximations for every concept [1, 2].

There exists a very simple test for consistency: $A^* \leq \{d\}^*$. If this condition is false, the corresponding data set is not consistent. For Table 1.4, $A^* = \{\{1\}, \{2\}, \{3\}, \{4, 5\}, \{6, 7, 8\}, \{9\}, \{10\}\}$, and $\{d\}^* = \{\{1, 2, 3, 4\}, \{5, 6, 7\}, \{8, 9, 10\}\}$, so $A^* \not\leq \{d\}^*$.

Let B be a subset of the set A of all attributes. For inconsistent data sets, in general, a concept X is not a definable set. However, set X may be approximated

Table 1.4: An inconsistent decision table

	Attributes			Decision
Case	Wind	Humidity	Temperature	Trip
1	low	low	medium	yes
2	low	low	low	yes
3	low	medium	medium	yes
4	low	medium	high	yes
5	low	medium	high	maybe
6	medium	low	medium	maybe
7	medium	low	medium	maybe
8	medium	low	medium	no
9	high	high	high	no
10	medium	high	high	no

by two B-definable sets, the first one is called a *B-lower approximation* of X, denoted by $\underline{B}X$ and defined as follows

$$\{x \in U | [x]_B \subseteq X\}. \tag{1.8}$$

The second set is called a *B-upper approximation* of X, denoted by $\overline{B}X$ and defined as follows

$$\{x \in U | [x]_B \cap X \neq \emptyset\}. \tag{1.9}$$

In Equations 1.8 and 1.9 lower and upper approximations are constructed from singletons x, we say that we are using so called the *first method*. The B-lower approximation of X is the largest B-definable set, contained in X. The B-upper approximation of X is the smallest B-definable set containing X.

As it was observed in [2], for complete decision tables we may use a *second method* to define the B-lower approximation of X, by the following formula

$$\cup\{[x]_B | x \in U, [x]_B \subseteq X\}, \tag{1.10}$$

while the B-upper approximation of x may be defined, using the second method, by

$$\cup\{[x]_B | x \in U, [x]_B \cap X \neq \emptyset\}. \tag{1.11}$$

Obviously, both Equations, 1.8 and 1.10, define the same set. Similarly, Equtions 1.9 and 1.11 also define the same set. For Table 1.4,

Table 1.5: A new data set for inducing certain rules for the concept [(*Trip*, *yes*)]

	Attributes			Decision
Case	Wind	Humidity	Temperature	Trip
1	low	low	medium	yes
2	low	low	low	yes
3	low	medium	medium	yes
4	low	medium	high	SPECIAL
5	low	medium	high	SPECIAL
6	medium	low	medium	SPECIAL
7	medium	low	medium	SPECIAL
8	medium	low	medium	SPECIAL
9	high	high	high	SPECIAL
10	medium	high	high	SPECIAL

$$\underline{A}\{1,2,3,4\} = \{1,2,3\}$$

and

$$\overline{A}\{1,2,3,4\} = \{1,2,3,4,5\}.$$

It is well known that for any $B \subseteq A$ and $X \subseteq U$,

$$\underline{B}X \subseteq X \subseteq \overline{B}X, \tag{1.12}$$

hence any case x from $\underline{B}X$ is *certainly* a member of X, while any member x of $\overline{B}X$ is *possibly* a member of X. This observation is used in the LERS data mining system. If an input data set is inconsistent, LERS computes lower and upper approximations for any concept and then induces *certain* rules from the lower approximation and *possible* rules from the upper approximation. For example, if we want to induce certain and possible rule sets for the concept [(*Trip*, *yes*)] from Table 1.4, we need to consider the following two data sets, presented in Tables 1.5 and 1.6.

Table 1.5 was obtained from Table 1.4 by assigning the value *yes* of the decision *Trip* to all cases from the lower approximation of [(*Trip*, *yes*)] and by replacing all remaining values of *Trip* by a special value, say *SPECIAL*. Similarly, Table 1.6 was obtained from Table 1.4 by assigning the value *yes* of the decision *Trip* to all cases from the upper approximation of [(*Trip*, *yes*)] and by replacing all remaining values of *Trip* by the value *SPECIAL*. Obviously,

Table 1.6: A new data set for inducing possible rules for the concept [(*Trip*, *yes*)]

	Attributes			Decision
Case	Wind	Humidity	Temperature	Trip
1	low	low	medium	yes
2	low	low	low	yes
3	low	medium	medium	yes
4	low	medium	high	yes
5	low	medium	high	yes
6	medium	low	medium	SPECIAL
7	medium	low	medium	SPECIAL
8	medium	low	medium	SPECIAL
9	high	high	high	SPECIAL
10	medium	high	high	SPECIAL

both tables, 1.5 and 1.6, are consistent. Therefore, we may use the LEM1 or LEM2 algorithms to induce rules from Tables 1.5 and 1.6. The rule set induced by the LEM2 algorithm from Table 1.5 is

2, 2, 2

$$(Wind, low)\ \&\ (Humidity, low) \rightarrow (Trip, yes),$$

2, 1, 1

$$(Humidity, medium)\ \&$$
$$(Temperature, medium) \rightarrow (Trip, yes)$$

1, 4, 4

$$(Temperature, high) \rightarrow (Trip, SPECIAL),$$

1, 4, 4

$$(Wind, medium) \rightarrow (Trip, SPECIAL),$$

where all rules are presented in the LERS format, see Subsection 1.1.3.

Obviously, only rules with $(Trip, yes)$ on the right hand side are informative, remaining rules, with $(Trip, SPECIAL)$ on the right hand side should be

Table 1.7: A data set with numerical attributes

	Attributes			Decision
Case	Wind	Humidity	Temperature	Trip
1	4	low	medium	yes
2	8	low	low	yes
3	4	medium	medium	yes
4	8	medium	high	maybe
5	12	low	medium	maybe
6	16	high	low	no
7	30	high	high	no
8	12	high	high	no

ignored. These two rules are *certain*. The only informative rule induced by the LEM2 algorithm from Table 1.6 is

1, 4, 5

$$(Wind, low) \rightarrow (Trip, yes).$$

This rule is *possible*.

1.3 Decision Table with Numerical Attributes

An example of a data set with numerical attributes is presented in Table 1.7.

In rule induction from numerical data usually a preliminary step called *discretization* [10, 11, 12] is conducted. During discretization a domain of the numerical attribute is divided into intervals defined by cutpopints (left and right delimiters of intervals). Such an interval, delimited by two cutpoints: c and d, will be denoted by $c..d$. In this chapter we will discuss how to do both processes: rule induction and discretization concurrently. First we need to check whether our data set is consistent. Note that numerical data are, in general, consistent, but inconsistent numerical data are possible. For inconsistent numerical data we need to compute lower and upper approximations and the induce certain and possible rule sets. In the data set from Table 1.7, $A^* = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\}$, $\{d\}^* = \{\{1, 2, 3\}, \{4, 5\}, \{6, 7, 8\}\}$, so $A^* \leq \{d\}^*$ and the data set is consistent.

A modified LEM2 algorithm for rule induction, called MLEM2 [13], does not need any preliminary discretization of numerical attributes. The domain of any numerical attribute is sorted first. Then potential cutpoints are selected as averages of any two consecutive values of the sorted list. For each cutpoint c the

Table 1.8: Computing a local covering for the concept [(*Trip*, *yes*)], part I

$(a, v) = t$	$[(a, v)]$	{1, 2, 3}	{1, 2, 3}
(*Wind*, 4..6)	{1, 3}	{1, 3}	{1, 3} •
(*Wind*, 6..30)	{2, 4, 5, 6, 7, 8}	{2}	{2}
(*Wind*, 4..10)	{1, 2, 3, 4}	{1, 2, 3} •	–
(*Wind*, 10..30)	{5, 6, 7, 8}	–	–
(*Wind*, 4..14)	{1, 2, 3, 4, 5, 8}	{1, 2, 3}	–
(*Wind*, 14..30)	{6, 7}	–	–
(*Wind*, 4..23)	{1, 2, 3, 4, 5, 6, 8}	{1, 2, 3}	–
(*Wind*, 23..30)	{7}	–	–
(*Humidity*, *low*)	{1, 2, 5}	{1, 2}	{1, 2}
(*Humidity*, *medium*)	{3, 4}	{3}	{3}
(*Humidity*, *high*)	{6, 7, 8}	–	–
(*Temperature*, *low*)	{2, 6}	{2}	{1, 3}
(*Temperature*, *medium*)	{1, 3, 5}	{1, 3}	{1, 3}
(*Temperature*, *high*)	{4, 7, 8}	–	–
Comments		1	2

MLEM2 algorithm creates two blocks, the first block contain all cases for which values of the numerical attribute are smaller than c, the second block contains remaining cases (with values of the numerical attribute larger than c). Once all such blocks are computed, rule induction in MLEM2 is conducted the same way as in LEM2.

We will illustrate rule induction from Table 1.7 using the MLEM2 rule induction algorithm. The MLEM2 algorithm is traced on Tables 1.8 and 1.9. The corresponding comments are:

1. The set $G = \{1, 2, 3\}$. The best attribute-value pair t, with the largest cardinality of the intersection of $[t]$ and G (presented in the third column of Table 1.8) is (*Wind*, 4..10). The corresponding entry in the third column of Table 1.8 is bulleted. However, $[(\textit{Wind}, 4..10)] = \{1, 2, 3, 4\} \not\subseteq \{1, 2, 3\} = B$, hence we need to look for the next t,

2. the set G is the same, $G = \{1, 2, 3\}$. There are dashes for rows (*Wind*, 4..14) and (*Wind*, 4..23) since the corresponding intervals contain 4..10. There are four attribute-value pairs with $|[t \cap G| = 2$. The best attribute-value pair, with the smallest cardinality of $[t]$ is (*Wind*, 4..6). This time $\{1, 2, 3, 4\} \cap \{1,$

Table 1.9: Computing a local covering for the concept [(*Trip*, *yes*)], part II

$(a, v) = t$	$[(a, v)]$	{2}	{2}
(*Wind*, 4..6)	{1, 3}	–	–
(*Wind*, 6..30)	{2, 4, 5, 6, 7, 8}	{2}	{2}
(*Wind*, 4..10)	{1, 2, 3, 4}	{2}	{2}
(*Wind*, 10..30)	{5, 6, 7, 8}	–	–
(*Wind*, 4..14)	{1, 2, 3, 4, 5, 8}	{2}	{2}
(*Wind*, 14..30)	{6, 7}	–	–
(*Wind*, 4..23)	{1, 2, 3, 4, 5, 6, 8}	{2}	{2}
(*Wind*, 23..30)	{7}	–	–
(*Humidity*, *low*)	{1, 2, 5}	{2}	{2} •
(*Humidity*, *medium*)	{3, 4}	–	–
(*Humidity*, *high*)	{6, 7, 8}	–	–
(*Temperature*, *low*)	{2, 6}	{2} •	–
(*Temperature*, *medium*)	{1, 3, 5}	–	–
(*Temperature*, *high*)	{4, 7, 8}	–	–
Comments		3	4

3} = {1, 3} $\subseteq$ {1, 2, 3}. Obviously, the common part of both intervals is 4..6, so {(*Wind*, 4..6) } is the first element T of $\mathcal{T}$,

3. the new set $G = B - [T] = \{1, 2, 3\} - \{1, 3\} = \{2\}$. The pair [(*Temperature*, *low*) has the smallest cardinality of $[t]$, so it is the best choice. However, [(*Temperature*, *low*)] ={2, 6} $\not\subseteq$ {1, 2, 3}, hence we need to look for the next t,

4. the pair (*Humidity*, *low*) is the best choice, and {3, 4} $\cap$ {1, 3, 5} = {3} $\subseteq$ {1, 2, 3}, so {[(*Temperature*, *low*), (*Humidity*, *low*)} is the second element T of $\mathcal{T}$.

As a result, $\mathcal{T}$ = { {(*Wind*, 4..6) }, { (*Temperature*, *low*), (*Humidity*, *low*)}}. In different words, the MLEM2 algorithm induces the following rule set for Table 1.7

1, 2, 2

$$(Wind, 4..6) \rightarrow (Trip, yes),$$

2, 1, 1

$$(Temperature, low) \ \& \ (Humidity, low) \rightarrow (Trip, yes)$$

1.4 Incomplete Data

Real-life data are frequently incomplete. In this section we will consider incompleteness in the form of missing attribute values. We will distinguish three types of missing attribute values:

- *lost values*, denoted by ?, where the original values existed, but currently are unavailable, since these values were, for example, erased or the operator forgot to input them. In rule induction we will induce rules form existing, specified attribute values,
- *"do not care" conditions*, denoted by *, where the original values are mysterious. For example, data were collected in a form of the interview, some questions were considered to be irrelevant or were embarrassing. Let us say that in an interview associated with a diagnosis of a disease a question is about eye color. For some people such question is irrelevant. In rule induction we are assuming that the attribute value is any value from the attribute domain.
- *attribute-concept value*, denoted by $-$. This interpretation is a special case of the "do not care" condition: it is restricted to attribute values typical for the concept to which the case belongs. For example, typical values

Table 1.10: An incomplete decision table

	Attributes			Decision
Case	Wind	Humidity	Temperature	Trip
1	low	low	medium	yes
2	?	low	*	yes
3	*	medium	medium	yes
4	low	?	high	maybe
5	medium	–	medium	maybe
6	*	high	low	no
7	–	high	*	no
8	medium	high	high	no

of temperature for patients sick with flu are: high and very-high, for a patient the temperature value is missing, but we know that this patient is sick with flu, if using the attribute-concept interpretation, we will assume that possible temperature values are: high and very-high.

We will assume that for any case at least one attribute value is specified (i.e., is not missing) and that all decision values are specified.

An example of a decision table with missing attribute values is presented in Table 1.10.

Definition of consistent data from Section 1.2 cannot be applied to data with missing attribute values since for such data the standard definition of the indiscernibility relation must be extended. Moreover, it is well-known that the standard definitions of lower and upper approximations are not applicable to data with missing attribute values. In Subsection 1.4.1 we will discuss three generalizations of the standard approximations: singleton, subset and concept.

1.4.1 Singleton, Subset and Concept Approximations

For incomplete data the definition of a block of an attribute-value pair is modified [14].

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

- If for an attribute a there exists a case x such that the corresponding value is an attribute-concept value, i.e., $a(x) = -$, then the corresponding case x should be included in blocks $[(a, v)]$ for all specified values $v \in V(x, a)$ of attribute a, where

$$V(x, a) = \{a(y) \mid a(y) \text{ is specified}, \\ y \in U,\ d(y) = d(x)\}. \tag{1.13}$$

For Table 1.10, $V(5, Humidity) = \emptyset$ and $V(7, Wind) = \{medium\}$, so the blocks of attribute-value pairs are:

[(Wind, low)] = {1, 3, 4, 6},
[(Wind, medium)] = {3, 5, 6, 7, 8},
[(Humidity, low)] = {1, 2},
[(Humidity, medium)] = {3},
[(Humidity, high)] = {6, 7, 8},
[(Temperature, low)] = {2, 6, 7},
[(Temperature, medium)] = {1, 2, 3, 5, 7},
[(Temperature, high)] = {2, 4, 7, 8},

For a case $x \in U$, the *characteristic set* $K_B(x)$ is defined as the intersection of the sets $K(x, a)$, for all $a \in B$, where the set $K(x, a)$ is defined in the following way:

- If $a(x)$ is specified, then $K(x, a)$ is the block $[(a, a(x)]$ of attribute a and its value $a(x)$,
- If $a(x) =?$ or $a(x) = *$ then the set $K(x, a) = U$,
- If $a(x) = -$, then the corresponding set $K(x, a)$ is equal to the union of all blocks of attribute-value pairs (a, v), where $v \in V(x, a)$ if $V(x, a)$ is nonempty. If $V(x, a)$ is empty, $K(x, a) = U$.

For Table 1.10 and $B = A$,

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

Characteristic set $K_B(x)$ may be interpreted as the set of cases that are indistinguishable from x using all attributes from B and using a given interpretation of missing attribute values. For completely specified data sets (I.e., data sets with no missing attribute values), characteristic sets are reduced to

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

$$(x, y) \in R(B) \text{ if and only if } y \in K_B(x). \tag{1.14}$$

The characteristic relation $R(B)$ is reflexive but—in general—does not need to be symmetric or transitive. Obviously, the characteristic relation $R(B)$ is known if we know characteristic sets $K_B(x)$ for all $x \in U$ and vice versa. In our example, $R(A) = \{(1, 1), (2, 1), (2, 2), (3, 3), (4, 4), (5, 3), (5, 5), (6, 6), (6, 7), (7, 6), (7, 7), (7, 8), (8, 7), (8, 8)\}$.

For a complete decision table, the characteristic relation $R(B)$ is reduced to the indiscernibility relation [2].

Definability for completely specified decision tables should be modified to fit into incomplete decision tables. For incomplete decision tables, a union of some intersections of attribute-value pair blocks, where such attributes are members of B and are distinct, will be called B-*locally definable* sets. A union of characteristic sets $K_B(x)$, where $x \in X \subseteq U$ will be called a B-*globally definable* set. Any set X that is B-globally definable is B-locally definable, the converse is not true.

For example, the set $\{2\}$ is A-locally definable since $\{2\} = [(\textit{Humidity}, \textit{low})] \cap [(\textit{Temperature}, \textit{high})]$. However, the set $\{2\}$ is not A-globally definable. On the other hand, the set $\{5\} =$ is not even locally definable since in all blocks of attribute-value pairs containing the case 5 contain also the case 7 as well.

Obviously, if a set is not B-locally definable then it cannot be expressed by rule sets using attributes from B. Thus we should induce rules from sets that are at least A-locally definable.

For incomplete decision tables lower and upper approximations may be defined in a few different ways. We suggest three different definitions of lower and upper approximations for incomplete decision tables, following [15, 14, 16]. Let X be a concept, a subset of U, let B be a subset of the set A of all attributes, and let $R(B)$ be the characteristic relation of the incomplete decision. Our first definition uses a similar idea as in the first method of Section 1.2, and is based on constructing both approximations from single elements of the set U. We will call these approximations *singleton*. A singleton B-lower approximation of X is defined as follows:

$$\underline{B}X = \{x \in U \mid K_B(x) \subseteq X\}. \tag{1.15}$$

A singleton B-upper approximation of X is

$$\overline{B}X = \{x \in U \mid K_B(x) \cap X \neq \emptyset\}. \tag{1.16}$$

In our example of the decision table presented in Table 1.10 the singleton A-lower and A-upper approximations of the concept: $\{1, 2, 3\}$ are:

$$\underline{A}\{1, 2, 3\} = \{1, 2, 3\}, \tag{1.17}$$

$$\overline{A}\{1,2,3\} = \{1,2,3,5\}. \tag{1.18}$$

We may easily observe that the set $\{1, 2, 3, 5\} = \overline{A}\{1,2,3\}$ is not A-locally definable since in all blocks of attribute-value pairs cases 5 and 7 are inseparable. Thus, as it was observed in, e.g., [15, 14, 16], singleton approximations should not be used, theoretically, for rule induction.

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. Therefore, we may define lower and upper approximations for incomplete decision tables by analogy with the second method from Section 1.2, 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:

$$\underline{B}X = \cup\{K_B(x) \mid x \in U, K_B(x) \subseteq X\}. \tag{1.19}$$

A *subset* B-upper approximation of X is

$$\overline{B}X = \cup\{K_B(x) \mid x \in U, K_B(x) \cap X \neq \emptyset\}. \tag{1.20}$$

For any concept X, singleton B-lower and B-upper approximations of X are subsets of the subset B-lower and B-upper approximations of X, respectively [16], because the characteristic relation $R(B)$ is reflexive. For the decision table presented in Table 1.10, the subset A-lower and A-upper approximations are

$$\underline{A}\{1,2,3\} = \{1,2,3\},$$
$$\overline{A}\{1,2,3\} = \{1,2,3,5,7\}$$

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:

$$\underline{B}X = \cup\{K_B(x) \mid x \in X, K_B(x) \subseteq X\}. \tag{1.21}$$

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:

$$\begin{aligned} \overline{B}X &= \cup \{K_B(x) \mid x \in X, K_B(x) \cap X \neq \emptyset\} \\ &= \cup \{K_B(x) \mid x \in X\}. \end{aligned} \tag{1.22}$$

The concept upper approximations were defined in [17] and [18] as well. The concept B-upper approximation of X is a subset of the subset B-upper approximation of X [16]. For the decision table presented in Table 1.10, the concept A-upper approximations is

Table 1.11: Computing a rule set for the concept [(*Trip*, *yes*)], Table 1.10

$(a, v) = t$	$[(a, v)]$	{1, 2, 3}	{1, 2, 3}	{3}
(*Wind*, *low*)	{1, 3, 4, 6}	{1, 3}	{1, 3}	{3}
(*Wind*, *medium*)	{3, 5, 6, 7, 8}	{3}	{3}	{3}
(*Humidity*, *low*)	{1, 2}	{1, 2}	{1, 2} •	–
(*Humidity*, *medium*)	{3}	{3}	{3}	{3} •
(*Humidity*, *high*)	{6, 7, 8}	–	–	–
(*Temperature*, *low*)	{2, 6, 7}	{2}	–	–
(*Temperature*, *medium*)	{1, 2, 3, 5, 7}	{1, 2, 3}•	–	{3}
(*Temperature*, *high*)	{2, 4, 7, 8}	{2}	–	–
Comments		1	2	3

$$\overline{A}\{1, 2, 3\} = \{1, 2, 3\},$$

Note that for complete decision tables, all three definitions of lower and upper approximations, singleton, subset and concept, are reduced to the same standard definition of lower and upper approximations, respectively.

1.4.2 Modified LEM2 algorithm

The same MLEM2 rule induction from Section 1.3 may be used for rule induction from incomplete data, the only difference is different definition of block of attribute-value pairs. Let us apply the MLEM2 algorithm to the data set from Table 1.10. First, we need to make a decision what kind of approximations we are going to use: singleton, subset or concept. In our example let us use concept approximation. For Table 1.10,

$$\underline{A}\{1, 2, 3\} = \overline{A}\{1, 2, 3\} = \{1, 2, 3\},$$

We will trace the MLEM2 algorithm applied to the set {1, 2, 3}, this way our certain rule set, for the concept [(*Trip*, *yes*)], is at the same time certain and possible. The tracing of LEM2 is presented in the Tables 1.11.

The corresponding comments are:

1. The set $G = \{1, 2, 3\}$. The best attribute-value pair t, with the largest cardinality of the intersection of $[t]$ and G (presented in the third column of Table 1.11) is (*Temperature*, *medium*). The corresponding entry in the third column of Table 1.11 is bulleted. However, [(*Temperature*, *medium*)] = {1, 2, 3,

5, 7} $\not\subseteq$ {1, 2, 3} = B, hence we need to look for the next t,

2. the set G is the same, G = {1, 2, 3}. There are two attribute-value pairs with $|[t \cap G| = 2$. One of them, (*Humidity*, *low*) has the smallest cardinality of $[t]$, so we select it. This time {1, 2, 3, 5, 7} $\cap$ {1, 2} = {1, 2} $\subseteq$ {1, 2, 3}. However, (*Temperature*, *medium*) is redundant, since [(*Humidity*, *low*)] $\subseteq$ {1, 2, 3}, hence {(*Humidity*, *low*)} is the first element T of $\mathcal{T}$,

3. the new set $G = B -$ [T] = {1, 2, 3} $-$ {1, 2} = {3}. The pair [(*Humidity*, *medium*) has the smallest cardinality of $[t]$, so it is the best choice. Additionally, [(*Humidity*, *medium*)] ={3} $\subseteq$ {1, 2, 3}, hence we are done, the set T = {(*Humidity*, *medium*)}.

Therefore, $\mathcal{T}$ = { {(*Humidity*, *low*)}, { (*Humidity*, *medium*)}}. The MLEM2 algorithm induces the following rule set for Table 1.10

1, 2, 2

$$(Humidity, low) \rightarrow (Trip, yes).$$

1, 1, 1

$$(Humidity, medium) \rightarrow (Trip, yes).$$

1.4.3 Probabilistic Approximations

In this section we are going to generalize singleton, subset and concept approximations from Subsection 1.4.1 to corresponding approximations that are defined using an additional parameter (or threshold), denoted by α, and interpreted as a probability. A generalization of standard approximations, called *probabilistic approximations*, were studied in many papers, see, e.g., [19, 20, 21, 22, 23, 24, 25, 26].

Let B be a subset of the attribute set A and X be a subset of U.

A B-singleton probabilistic approximation of X with the threshold α, $0 < \alpha \leq 1$, denoted by $appr_{\alpha,B}^{singleton}(X)$, is defined as follows

$$\{x \mid x \in U,\ Pr(X \mid K_B(x)) \geq \alpha\},$$

where $Pr(X \mid K_B(x)) = \frac{|X \cap K_B(x)|}{|K_B(x)|}$ is the conditional probability of X given $K_B(x)$ and $|Y|$ denotes the cardinality of set Y.

A B-subset probabilistic approximation of the set X with the threshold α, $0 < \alpha \leq 1$, denoted by $appr_{\alpha,B}^{subset}(X)$, is defined as follows

$$\cup\{K_B(x) \mid x \in U,\ Pr(X \mid K_B(x)) \geq \alpha\}.$$

A B-concept probabilistic approximation of the set X with the threshold α, $0 < \alpha \leq 1$, denoted by $appr_{\alpha,B}^{concept}(X)$, is defined as follows

Table 1.12: An incomplete decision table

	Attributes			Decision
Case	Wind	Humidity	Temperature	Trip
1	low	low	*	yes
2	?	low	low	yes
3	low	low	?	yes
4	high	high	high	yes
5	low	*	low	no
6	high	high	*	no
7	high	?	high	no
8	high	high	high	no

$$\cup\{K_B(x) \mid x \in X,\ Pr(X \mid K_B(x)) \geq \alpha\}.$$

For simplicity, if $B = A$, an A-singleton, B-subset and B-concept probabilistic approximations will be called singleton, subset and concept probabilistic approximations and will be denoted by $appr_\alpha^{singleton}(X)$, $appr_\alpha^{subset}(X)$ and $appr_\alpha^{concept}(X)$, respectively.

Obviously, for the concept X, the probabilistic approximation of a given type (singleton, subset or concept) of X computed for the threshold equal to the smallest positive conditional probability $Pr(X \mid [x])$ is equal to the standard upper approximation of X of the same type. Additionally, the probabilistic approximation of a given type of X computed for the threshold equal to 1 is equal to the standard lower approximation of X of the same type.

For the data set from Table 1.12, the set of blocks of attribute-value pairs is
$[Wind, low)] = \{1, 3, 5\}$,
$[(Wind, high)] = \{4, 6, 7, 8\}$,
$[(Humidity, low)] = \{1, 2, 3, 5\}$,
$[(Humidity, high)] = \{1, 4, 6, 7, 8\}$,
$[(Temperature, low)] = \{1, 2, 5, 6\}$,
$[(Temperature, high)] = \{1, 4, 6, 7, 8\}$.

The corresponding characteristic sets are
$K_A(1) = K_A(3) = \{1, 3, 5\}$,
$K_A(2) = \{1, 2, 5\}$,
$K_A(4) = \{4, 6, 8\}$,
$K_A(5) = \{1, 5\}$,
$K_A(6) = K_A(8) = \{4, 6, 8\}$,

Table 1.13: Conditional probabilities

$K_A(x)$	{1, 2, 5}	{1, 3, 5}	{1, 5}	{4, 6, 8}	{4, 6, 7, 8}
$Pr(\{1,2,4,6\} \mid K_A(x))$	0.667	0.667	0.5	0.333	0.25

$K_A(7) = \{4, 6, 7, 8\}$.

Conditional probabilities of the concept {1, 2, 3, 4} given a characteristic set $K_A(x)$ are presented in Table 1.13.

For Table 1.12, all probabilistic approximations (singleton, subset and concept) are

$appr_{0.25}^{singleton}(\{1,2,3,4\}) = U$,

$appr_{0.333}^{singleton}(\{1,2,3,4\}) = \{1,2,3,4,5,6,8\}$,

$appr_{0.5}^{singleton}(\{1,2,3,4\}) = \{1,2,3,5\}$,

$appr_{0.667}^{singleton}(\{1,2,3,4\}) = \{1,2,3\}$,

$appr_{1}^{singleton}(\{1,2,3,4\}) = \emptyset$,

$appr_{0.25}^{subset}(\{1,2,3,4\}) = U$,

$appr_{0.333}^{subset}(\{1,2,3,4\}) = \{1,2,3,4,5,6,8\}$,

$appr_{0.5}^{subset}(\{1,2,3,3\}) = \{1,2,3,5\}$,

$appr_{0.667}^{subset}(\{1,2,3,4\}) = \{1,2,3,5\}$,

$appr_{1}^{subset}(\{1,2,3,4\}) = \emptyset$,

$appr_{0.25}^{concept}(\{1,2,3,4\}) = \{1,2,3,4,5,6,8\}$,

$appr_{0.333}^{concept}(\{1,2,3,4\}) = \{1,2,3,4,5,6,8\}$,

$appr_{0.5}^{concept}(\{1,2,3,4\}) = \{1,2,3,5\}$,

Table 1.14: A modified decision table

	Attributes			Decision
Case	Wind	Humidity	Temperature	Trip
1	low	low	*	yes
2	?	low	low	yes
3	low	low	?	yes
4	high	high	high	SPECIAL
5	low	*	low	no
6	high	high	*	SPECIAL
7	high	?	high	SPECIAL
8	high	high	high	SPECIAL

$$appr_{0.667}^{concept}(\{1,2,3,4\}) = \{1,2,3,5\},$$

$$appr_{1}^{concept}(\{1,2,3,4\}) = \emptyset.$$

For rule induction from probabilistic approximations of the given concept a similar technique as in Section 1.2 may be used. For any concept and the probabilistic approximation of the concept we will create a new decision table.

Let us illustrate this idea with inducing rule set for the concept [(*Trip*, *yes*)] from Table 1.12 using concept probabilistic approximation with $\alpha = 0.5$. The corresponding modified decision table is presented in Table 1.14.

In the data set, presented in Table 1.14, all values of *Trip* are copied from Table 1.12 for all cases from

$$appr_{0.5}^{concept}(\{1,2,3,4\}) = \{1,2,3,5\},$$

while for all remaining cases values of *Trip* are replaced by the SPECIAL value. The MLEM2 rule induction algorithm, using concept upper approximation should be used with the corresponding type of upper approximation (singleton, subset and concept). In our example the MLEM2 rule induction algorithm, using concept upper approximation, induces from Table 1.14 the following rule set

1, 3, 4

$$(Humidity, low) \rightarrow (Trip, yes),$$

1, 4, 4

$$(Wind, high) \rightarrow (Trip, SPECIAL),$$

2, 1, 2

$$(Wind, low)\&(Temperature, low) \rightarrow (Trip, no).$$

The only rules that are useful should have ($Trip$, yes) on the right hand side. Thus, the only rule that survives is

1, 3, 4

$$(Humidity, low) \rightarrow (Trip, yes).$$

1.5 Conclusions

Investigation of rule induction methods is subject to intensive research activity. New versions of rule induction algorithms based on probabilistic approximations were explored, see, e.g., [27, 28]. Novel rule induction algorithms in which computation of probabilistic approximations is done in parallel with rule induction are recently developed and experimentally tested, see, e.g., [29]. The LEM2 algorithm was implemented in a bagged version [30], using ideas of ensemble learning.

Bibliography

[1] Z. Pawlak: Rough sets, International Journal of Computer and Information Sciences **11**, 341 – 356 (1982)

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

[3] J. W. Grzymala-Busse: Rule induction. In: *Data Mining and Knowledge Discovery Handbook, Second Edition*, 2 edn., ed. by O. Maimon, L. Rokach (Springer-Verlag, Berlin, Heidelberg 2010) pp. 249 – 265

[4] Z. Pawlak, J. W. Grzymala-Busse, R. Slowinski, W. Ziarko: Rough sets, Communications of the ACM **38**, 89 – 95 (1995)

[5] J. G. Bazan, M. S. Szczuka, A. Wojna, M. Wojnarski: On the evolution of rough set exploration system, Proceedings of the Fourth International Conference on Rough Sets and Current Trends in Computing, Uppsala, Sweden 2004, ed. by S. Tsumoto, R. Slowinski, J. Grzymala-Busse (Springer Verlag, Berlin, Heidelberg, New York 2004) 592 – 601

[6] J. W. Grzymala-Busse: LERS—A system for learning from examples based on rough sets. In: *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory*, ed. by R. Slowinski (Kluwer Academic Publishers, Dordrecht, Boston, London 1992) pp. 3 – 18

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

[8] L. B. Booker, D. E. Goldberg, Holland J. F.: Classifier systems and genetic algorithms. In: *Machine Learning. Paradigms and Methods*, ed. by J. G. Carbonell (MIT Press, Boston 1990) pp. 235 – 282

[9] J. H. Holland, K. J. Holyoak, R. E. Nisbett: *Induction. Processes of Inference, Learning, and Discovery* (MIT Press, Boston 1986)

[10] M. R. Chmielewski, J. W. Grzymala-Busse: Global discretization of continuous attributes as preprocessing for machine learning, International Journal of Approximate Reasoning **15**(4), 319 – 331 (1996)

[11] J. W. Grzymala-Busse: Discretization of numerical attributes. In: *Handbook of Data Mining and Knowledge Discovery*, ed. by W. Kloesgen, J. Zytkow (Oxford University Press, New York 2002) pp. 218 – 225

[12] J. W. Grzymala-Busse: Mining numerical data—A rough set approach, Transactions on Rough Sets **11**, 1 – 13 (2010)

[13] J. W. Grzymala-Busse: 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, Annecy, France 2002, ed. by B. Bouchon-Meunier, L. Foulloy, R. R. Yager (Universite de Savoie, Savoie 2002) 243 – 250

[14] J. W. Grzymala-Busse: Data with missing attribute values: Generalization of indiscernibility relation and rule induction, Transactions on Rough Sets **1**, 78 – 95 (2004)

[15] J. W. Grzymala-Busse: Rough set strategies to data with missing attribute values, Workshop Notes, Foundations and New Directions of Data Mining, in conjunction with the 3-rd International Conference on Data Mining, Melbourne, Florida 2003, ed. by T. Y. Lin, X. Hu, S. Ohsuga (IEEE Computer Society, Los Alamitos, CA 2003) 56 – 63

[16] J. W. Grzymala-Busse: Characteristic relations for incomplete data: A generalization of the indiscernibility relation, Proceedings of the Fourth International Conference on Rough Sets and Current Trends in Computing, Uppsala, Sweden 2004, ed. by S. Tsumoto, R. Slowinski, J. Grzymala-Busse (Springer Verlag, Berlin, Heidelberg, New York 2004) 244 – 253

[17] T. Y. Lin: Topological and fuzzy rough sets. In: *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*, ed. by R. Slowinski (Kluwer Academic Publishers, Dordrecht, Boston, London 1992) pp. 287 – 304

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

[19] J. W. Grzymala-Busse, W. Ziarko: Data mining based on rough sets. In: *Data Mining: Opportunities and Challenges*, 2 edn., ed. by J. Wang (Idea Group Publ., Hershey, PA 203) pp. 142 – 173

[20] J. W. Grzymala-Busse, Y. Yao: Probabilistic rule induction with the LERS data mining system, International Journal of Intelligent Systems **26**, 518 – 539 (2011)

[21] Z. Pawlak, A. Skowron: Rough sets: Some extensions, Information Sciences **177**, 28 – 40 (2007)

[22] Z. Pawlak, S. K. M. Wong, W. Ziarko: Rough sets: probabilistic versus deterministic approach, International Journal of Man-Machine Studies **29**, 81 – 95 (1988)

[23] Y. Y. Yao: Probabilistic rough set approximations, International Journal of Approximate Reasoning **49**, 255 – 271 (2008)

[24] Y. Y. Yao, S. K. M. Wong: A decision theoretic framework for approximate concepts, International Journal of Man-Machine Studies **37**, 793 – 809 (1992)

[25] W. Ziarko: Variable precision rough set model, Journal of Computer and System Sciences **46**(1), 39 – 59 (1993)

[26] W. Ziarko: Probabilistic approach to rough sets, International Journal of Approximate Reasoning **49**, 272 – 284 (2008)

[27] P. G. Clark, J. W. Grzymala-Busse: Experiments on probabilistic approximations, Proceedings of the 2011 IEEE International Conference on Granular Computing, Kaohsiung, Taiwan 2011, ed. by T. P. Hong, M. Kudo, Z. Ras (IEEE Computer Society, Washington, D. C. 2011) 144 – 149

[28] P. G. Clark, J. W. Grzymala-Busse, M. Kuehnhausen: Local probabilistic approximations for incomplete data, Proceedings of the ISMIS 2012, the 20-th International Symposium on Methodologies for Intelligent Systems, Macau 2012, ed. by L. Chen (Springer Verlag, Berlin, Heidelberg, New York 2012) 93 – 98

[29] J. W. Grzymala-Busse, W. Rzasa: A local version of the MLEM2 algorithm for rule induction, Fundamenta Informaticae **100**, 99 – 116 (2010)

[30] C. Cohagan, J. W. Grzymala-Busse, Z. S. Hippe: Experiments on mining inconsistent data with bagging and the MLEM2 rule induction algorithm, International Journal of Granular Computing, Rough Sets and Intelligent Systems **2**, 257 – 271 (2012)