Rough Set Theory in Decision Support Systems

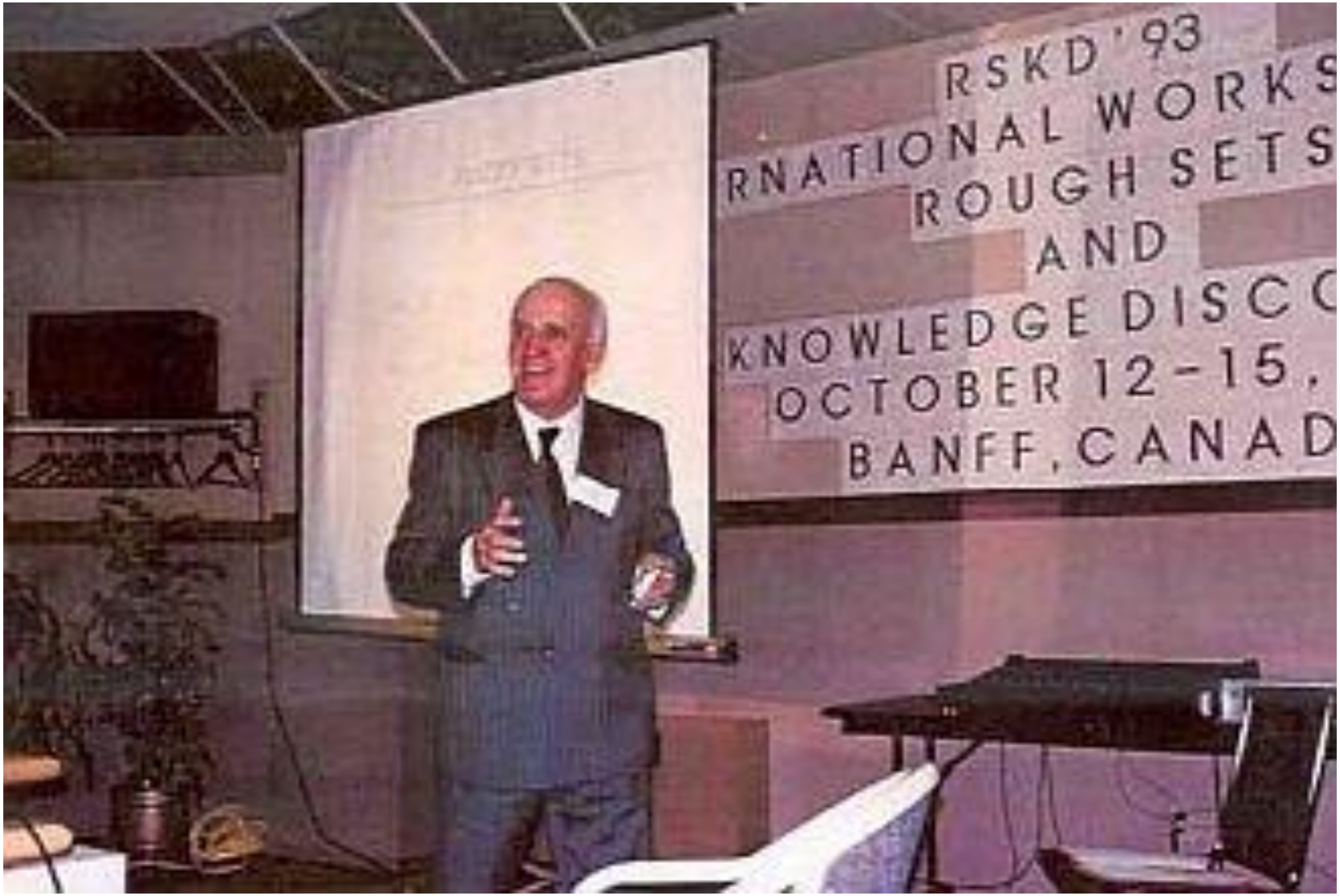
Agnieszka Nowak - Brzezinska

Advantages of Rough Set approach

- It does not need any preliminary or additional information about data − like probability in statistics, grade of membership in the fuzzy set theory.
- It provides efficient methods, algorithms and tools for finding hidden patterns in data.
- It allows to reduce original data, i.e. to find minimal sets of data with the same knowledge as in the original data.
- It allows to evaluate the significance of data.
- It allows to generate in automatic way the sets of decision rules from data.
- It is easy to understand.
- It offers straightforward interpretation of obtained results.
- It is suited for concurrent (parallel/distributed) processing.
- It is easy internet access to the rich literature about the rough set theory, its extensions as well as interesting applications, e.g. http://www.rsds.wsiz.rzeszow.pl

What is known about RS…

- In computer science, a **rough set**, first described by a Polish computer scientist Zdzisław Pawlak, is a formal approximation of a crisp set (i.e., conventional set) in terms of a pair of sets which give the *lower* and the *upper* approximation of the original set.
- In the standard version of rough set theory (Pawlak 1991), the lower- and upper-approximation sets are crisp sets, but in other variations, the approximating sets may be fuzzy sets.


RSKD'93
RNATIONAL WORKS
ROUGH SETS
AND
KNOWLEDGE DISCO
OCTOBER 12-15,
BANFF, CANAD


Click to LOOK INSIDE!
Studies in Fuzziness and Soft Computing
Rough Set Theory and Granular Computing
JingTao Yao Pawan Lingras
Wei-Zhi Wu Marcin Szczuka
Nick J. Cercone Dominik Ślęzak (Eds.)
LNAI 4481
Rough Sets and Knowledge Technology
Second International Conference, RSKT 2007
Toronto, Canada, May 2007
Proceedings
Springer
Journal Subline
LNCS 5390
Transactions on
Rough Sets IX
James F. Peters · Andrzej Skowron
Editors-in-Chief
Journal Subline
LNCS 3400
Transactions on
Rough Sets III
James F. Peters · Andrzej Skowron
Editors-in-Chief
Springer
Piero Pagliani
Mihir Chakraborty
Trends in Logic 27
A Geometry of Approximation
Rough Set Theory: Logic, Algebra and Topology of Conceptual Patterns
Springer
Studies in Computational Inte
Ajith Abraham
Rafael Falcón
Rafael Bello (Eds.)
Rough Set Theory: A True Landmark in Data Analysis
Springer
Primers Series, Vol. 2
Ivo Düntsch & Günther Gediga
Rough set data analysis
A road to non-invasive knowledge discovery
Methoδos
Advances in Soft Computing
Rough Sets
Aijun An Jerzy Stefanowski
Sheela Ramanna Cory J. Butz
Witold Pedrycz Guoyin Wang (Eds.)
LNAI 4482
Rough Sets, Fuzzy Sets, Data Mining and Granular Computing
11th International Conference, RSFDGrC 2007
Toronto, Canada, May 2007
Proceedings
Springer
Transactions on
Rough Sets I
dies in Computational Intelligence 163
Pawel Delimata
Mikhail Ju. Moshkov
Andrzej Skowron
Zbigniew Suraj
Inhibitory Rules in Data Analysis
A Rough Set Approach
Springer
Journal Subline
Transactions on
Rough Sets II
Rough Sets and Fuzzy Sets
James F. Peters · Andrzej Skowron
Springer

Rough set theory

- **was developed by Zdzislaw Pawlak in the early 1980’s.**

Pioneering Publications:

1. Z. Pawlak, “Rough Sets”, *International Journal of Computer and Information Sciences, Vol.11, 341-356 (1982).*
2. Z. Pawlak, *Rough Sets -Theoretical Aspect of Reasoning about Data, KluwerAcademic Pubilishers(1991).*

The power of the RS theory

- The main goal of the rough set analysis is induction of (learning) approximations of concepts.
- Rough sets constitutes a sound basis for KDD. It offers mathematical tools to discover patterns hidden in data.
- It can be used for feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction (templates, association rules) etc.
- identifies partial or total dependencies in data, eliminates redundant data, gives approach to null values, missing data, dynamic data and others.

Basic Concepts of Rough Sets

- Information/Decision Systems (Tables)
- Indiscernibility
- Set Approximation
- Reducts and Core
- Rough Membership
- Dependency of Attributes

Information system

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

the information about the real world is given in the form of an *information table (sometimes called a decision table).*

Thus, the information table represents input data, gathered from any domain, such as medicine, finance, or the military.

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

Rows of a table, labeled e1, e2, e3, e4, e5, and e6 are called examples (objects, entities).

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

Rows of a table, labeled e1, e2, e3, e4, e5, and e6 are called examples (objects, entities).

Conditional attributes= {Headache, muscle_pain, temperature}

Decisional attribute= {Flu}

Introduction

- Rough set theory proposes a new mathematical approach to imperfect knowledge, i.e. to **vagueness** (or **imprecision**). In this approach, vagueness is expressed by a **boundary region** of a set.
- Rough set concept can be defined by means of topological operations, *interior* and *closure*, called ***approximations.***

Information Systems/Tables

	Attributes		
	Headache	Muscle_pain	Temperature
e1	yes	yes	normal
e2	yes	yes	high
e3	yes	yes	very_high
e4	no	yes	normal
e5	no	no	high
e6	no	yes	very_high

- IS is a pair (*U, A*)
- *U* is a non-empty finite set of objects.
- *A* is a non-empty finite set of attributes such that for every $a \in A.$ $a : U \to V_a$
- V_a is called the value set of *a*.

Decision Systems/Tables

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

- DS: $T = (U, A \cup \{d\})$
- $d \notin A$ is the *decision* attribute (instead of one we can consider more decision attributes).
- The elements of *A* are called the *condition* attributes.

Indiscernibility

- The equivalence relation

  A binary relation $R \subseteq X \times X$ which is

  – reflexive (*xRx* for any object *x*) ,

  – symmetric (if *xRy* then *yRx*), and

  – transitive (if *xRy* and *yRz* then *xRz*).

- The equivalence class $[x]_R$ of an element $x \in X$ consists of all objects $y \in X$ such that *xRy*.

Indiscernibility (2)

- Let *IS = (U, A)* be an information system, then with any $B \subseteq A$ there is an associated equivalence relation:

$$IND_{IS}(B) = \{(x, x') \in U^2 \mid \forall a \in B, a(x) = a(x')\}$$

where $IND_{IS}(B)$ is called the *B-indiscernibility relation.*

- If $(x, x') \in IND_{IS}(B)$, then objects *x* and *x'* are indiscernible from each other by attributes from *B.*
- The equivalence classes of the *B-indiscernibility relation* are denoted by $[x]_B$.

The set consisting of attributes ***Headache*** *and* ***Muscle_pain****:*

IND({Headache,Muscle_pain})={{e1,e2,e3},{e4,e6},{e5}}

Examples **e1** and **e2** are characterized by the same values of both attributes: for the attribute Headache the value is yes for e1 and e2 and for the attribute Muscle_pain the value is yes for both e1 and e2.

Moreover, example **e3** **is indiscernible** from **e1** and **e2**.

Examples **e4** and **e6** are also **indiscernible** from each other.

Obviously, the indiscernibility relation is an equivalence relation.

Sets that are indiscernible are called elementary sets.

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

Thus, the set of attributes *Headache and Muscle_pain* defines the following elementary sets: {e1, e2, e3}, {e4, e6}, and {e5}.

Any finite union of elementary sets is called a *definable set.*

In our case, set {e1, e2, e3, e5} is definable by the attributes *Headache and Muscle_pain*, since we may define this set by saying that any member of it is characterized by the attribute *Headache* equal to yes and the attribute *Muscle_pain equal to yes* or by the attribute *Headache equal to no and the* attribute *Muscle_pain equal to no.*

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

An Example of Indiscernibility

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

- The non-empty subsets of the condition attributes are {Hedeache}, {Muscle_pain}, {Temperature} and {Hedeache,Muscle_pain}, {Hedeache,Temperature}, {Muscle_pain,Temperature} , {Hedeache,Temperature ,Muscle_pain}.
- IND({Hedeache}) = {{e1,e2,e3}, {e4,e5,e6}}
- IND({Muscle_pain}) = {{e1,e2,e3,e4,e6},{e5}}
- IND({Temperature}) ={{e1,e4},{e2,e5},{e3,e6}}
- IND({Hedeache, Muscle_pain}) = {{e1,e2,e3}, {e4,e6},{e5}}
- IND({Muscle_pain, Temperature}) = {{e1,e4},{e2},{e3,e6},{e5}}
- IND({Hedeache,Temperature}) ={{e1}{e2},{e3},{e4},{e5},{e6}}
- IND({Hedeache,Temperature, Muscle_pain}) ={{e1}{e2},{e3},{e4},{e5},{e6}}

Observations

- An equivalence relation induces a partitioning of the universe.
- Subsets that are most often of interest have the same value of the decision attribute.

Set Approximation

- Let *T = (U, A)* and let $B \subseteq A$ and $X \subseteq U.$ We can approximate *X* using only the information contained in *B* by constructing the *B-lower* and *B-upper* approximations of *X,* denoted $\underline{B}X$ and $\overline{B}X$ respectively, where

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

$$\overline{B}X = \{x \mid [x]_B \cap X \neq \phi\}.$$

Set Approximation (2)

- *B-boundary region* of *X*, $BN_B(X) = \overline{B}X - \underline{B}X,$ consists of those objects that we cannot decisively classify into *X* in *B.*
- *B-outside region* of *X*, $U - \overline{B}X,$ consists of those objects that can be with certainty classified as not belonging to *X*.
- A set is said to be *rough* if its boundary region is non-empty, otherwise the set is crisp.

An Example of Set Approximation

	Age	*LEMS*	*Walk*
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

- Let W = {x | Walk(x) = yes}.

$$\underline{A}W = \{x1, x6\},$$

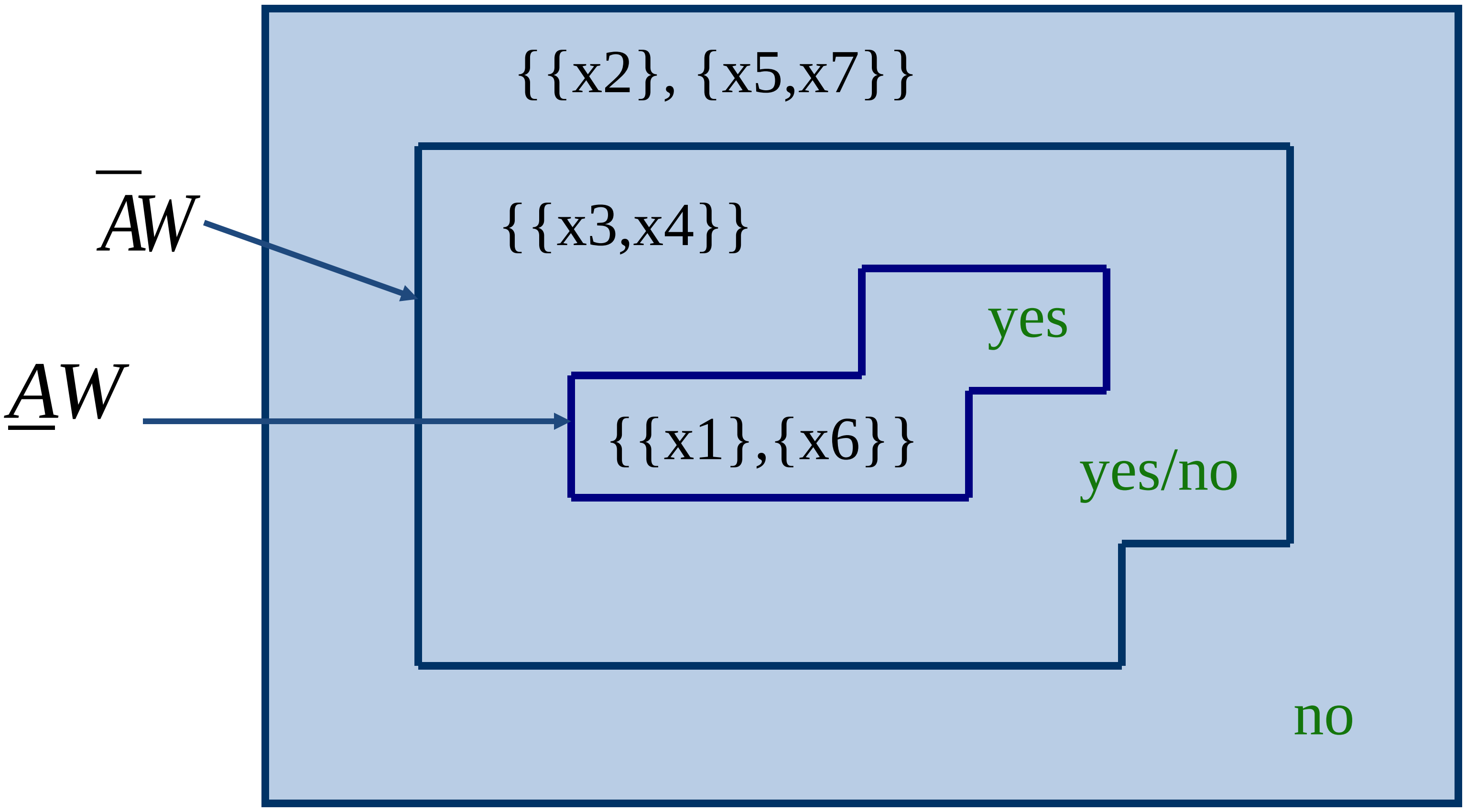
$$\overline{A}W = \{x1, x3, x4, x6\},$$

$$BN_A(W) = \{x3, x4\},$$

$$U - \overline{A}W = \{x2, x5, x7\}.$$

- The decision class, *Walk,* is rough since the boundary region is not empty.

An Example of Set Approximation (2)



Upper Approximation:

$$\overline{R}X = \bigcup \{Y \in U/R : Y \cap X \neq \phi\}$$

Lower Approximation:

$$\underline{R}X = \bigcup \{Y \in U/R : Y \subseteq X\}$$

Rough Membership Function

- Rough sets can be also defined by using, instead of approximations, a rough **membership Function.**
- In classical set theory, either an element belongs to a set or it does not.
- The corresponding membership function is the characteristic function for the set, i.e. the function takes values 1 and 0, respectively. In the case of rough sets, the notion of membership is different.

- The *rough membership function* quantifies the degree of relative overlap between the set X and the equivalence class $R(x)$ to which x belongs. It is defined as follows.

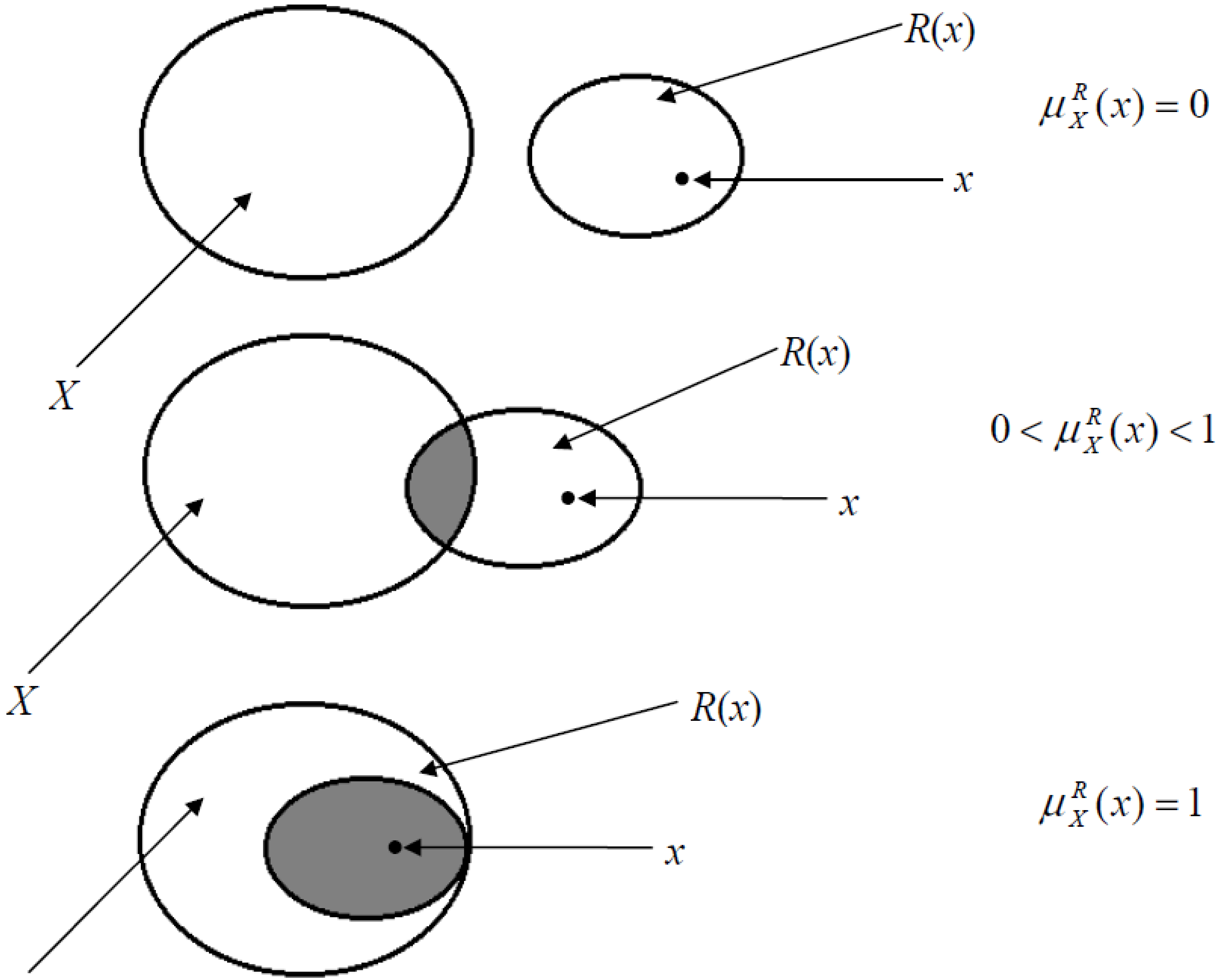
$$\mu_X^R : U \to <0,1>$$

where

$$\mu_X^R(x) = \frac{|X \cap R(x)|}{|R(x)|}$$

and $|X|$ denotes the cardinality of X.

The meaning of rough membership function



rough membership function

- The rough membership function expresses conditional probability that x belongs to X given R and can be interpreted as a degree that x belongs to X in view of information about x expressed by R.

- The rough membership function can be used to define approximations and the boundary region of a set:

$$R_*(X) = \{ x \in U : \mu_X^R(x) = 1 \},$$

$$R^*(X) = \{ x \in U : \mu_X^R(x) > 0 \},$$

$$RN_R(X) = \{ x \in U : 0 < \mu_X^R(x) < 1 \}.$$

Problems with real life decision systems

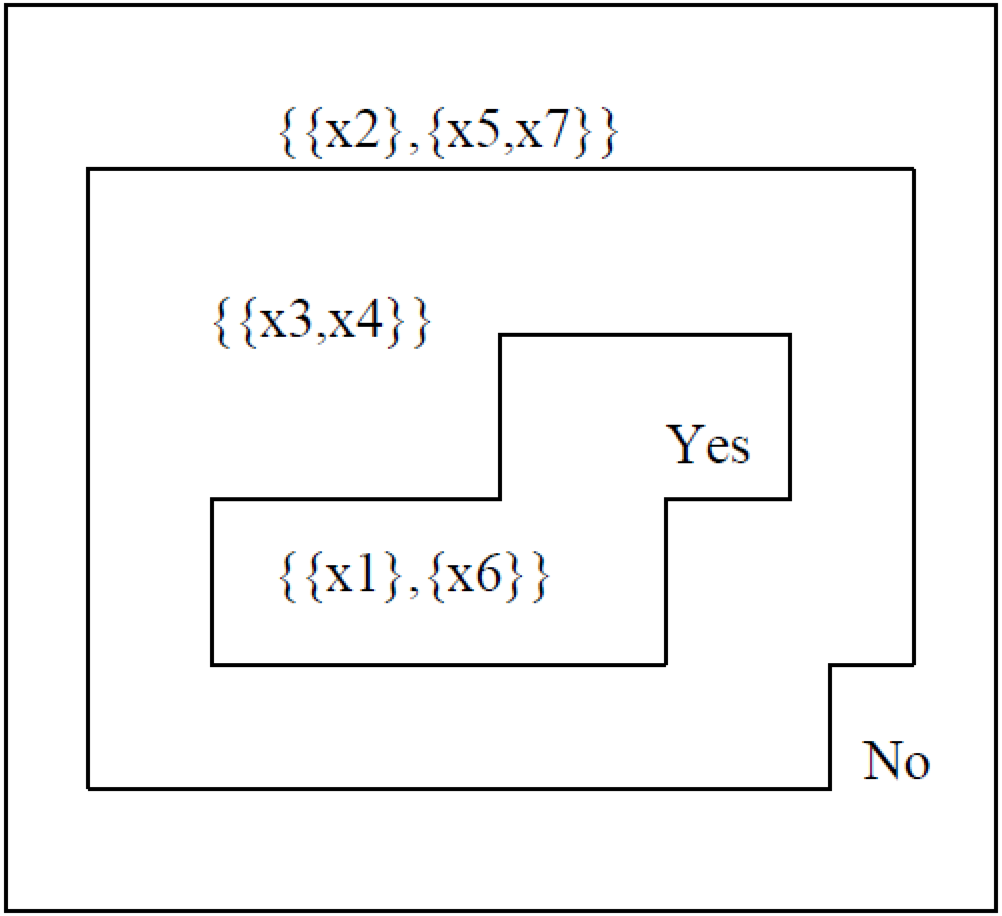
- A decision system expresses all the knowledge about the model.
- This table may be unnecessarily large in part because it is redundant in at least two ways.
- The same or indiscernible objects may be represented several times, or some of the attributes may be superfluous.

	Age	*LEMS*	*Walk*
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

$$IND(\{Age\}) = \{\{x_1, x_2, x_6\}, \{x_3, x_4\}, \{x_5, x_7\}\},$$

$$IND(\{LEMS\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_6, x_7\}\},$$

$$IND(\{Age, LEMS\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_7\}, \{x_6\}\}.$$



$$IND(\{Age\}) = \{\{x_1, x_2, x_6\}, \{x_3, x_4\}, \{x_5, x_7\}\},$$

$$IND(\{LEMS\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_6, x_7\}\},$$

$$IND(\{Age, LEMS\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_7\}, \{x_6\}\}.$$

An example

	Age	*LEMS*	*Walk*
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

- Let $X = \{x : Walk(x) = Yes\}$
- In fact, the set X consists of three objects: x1,x4,x6.
- Now, we want to describe this set in terms of the set of conditional attributes $A = \{Age, LEMS\}$.

Using the above definitions, we obtain the following approximations:
the A-lower approximation $\underline{AX}$ = {x1,x6}
the A-upper approximation $\overline{AX}$ = {x1, x3, x4, x6}
the A-boundary region $BN_S(X)$={x3, x4},
and the A-outside region $U - \overline{AX}$ = {x2, x5, x7}

It is easy to see that the set X is rough since the boundary region is not empty.

Indiscernibility relation

If we take into consideration the set $\{LEMS\}$ then objects x_3 and x_4 belong to the same equivalence class; they are indiscernible. From the same reason, x_5, x_6 and x_7 belong to another indiscernibility class. The relation IND defines three partitions of the universe.

$$IND(\{Age\}) = \{\{x_1, x_2, x_6\}, \{x_3, x_4\}, \{x_5, x_7\}\},$$

$$IND(\{LEMS\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_6, x_7\}\},$$

$$IND(\{Age, LEMS\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_7\}, \{x_6\}\}.$$

An **indiscernibility relation** is the main concept of rough set theory, normally associated with a set of attributes.

Accuracy of Approximation

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|}$$

where |X| denotes the cardinality of $X \neq \phi.$

Obviously $0 \leq \alpha_B \leq 1.$

If $\alpha_B(X) = 1,$ *X* is *crisp* with respect to *B*.

If $\alpha_B(X) < 1,$ *X* is *rough* with respect to *B.*

Issues in the Decision Table

- The same or indiscernible objects may be represented several times.
- ***Some of the attributes may be superfluous (redundant).***

  That is, their removal cannot worsen the classification.

Reducts

- Keep only those attributes that preserve the indiscernibility relation and, consequently, set approximation.
- There are usually several such subsets of attributes and those which are minimal are called *reducts.*

Dispensable & Indispensable Attributes

Let $c \in C$.

Attribute *c* is dispensable in *T*

if $POS_C(D) = POS_{(C-\{c\})}(D)$, otherwise

attribute *c* is indispensable in *T*.

The C-positive region of D:

$$POS_C(D) = \bigcup_{X \in U/D} \underline{C}X$$

Independent

- *T = (U, C, D)* is independent
  if all $c \in C$ are indispensable in *T*.

Reduct & Core

- The set of attributes $R \subseteq C$ is called a *reduct* of *C*, if *T' = (U, R, D)* is independent and

  $$POS_R(D) = POS_C(D).$$

- The set of all the condition attributes indispensable in *T* is denoted by *CORE(C).*

  $$CORE(C) = \cap RED(C)$$

  where *RED(C)* is the set of all *reducts* of *C.*

Due to the concept of indiscernibility relation, it is very simple to define **redundant (or dispensable) attributes**.

If a set of attributes and its superset define the same indiscernibility relation (i.e., if elementary sets of both relations are identical), then any attribute that belongs to the superset and not to the set is redundant.

In the example let the set of attributes be the set *{Headache, Temperature}* *and its* superset be the set of all three attributes, i.e., the set *{Headache, Muscle_pain, Temperature}*.

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

Elementary sets of the indiscernibility relation defined by the set {*Headache, Temperature*} *are* singletons, i.e., sets {e1}, {e2}, {e3}, {e4}, {e5}, and {e6}, and so are elementary sets of the indiscernibility relation defined by the set of all three attributes.

Thus, the attribute *Muscle_pain is* redundant. On the other hand, the set {*Headache, Temperature*} *does not contain* any redundant attribute, since elementary sets for attribute sets {*Headache*} *and* {*Temperature*} *are not* singletons. Such a set of attributes, with no redundant attribute, is called *minimal (or independent).*

The set P* of attributes is the *reduct (or covering)* of another set *Q of attributes if P is* minimal and the indiscernibility relations, defined by *P and Q, are the same (the last condition says that elementary* sets, determined by indiscernibility relations defined by *P and Q, are identical).

	Attributes			Decision
	Headache	Muscle_pain	Temperature	Flu
e1	yes	yes	normal	no
e2	yes	yes	high	yes
e3	yes	yes	very_high	yes
e4	no	yes	normal	no
e5	no	no	high	no
e6	no	yes	very_high	yes

In our example, the set *{Headache, Temperature} is a* **reduct** of the original set of attributes *{Headache, Muscle_ pain, Temperature}.*

Table 2 presents a new information table based on this reduct.

Table 2. Reduced Information Table.

	Attributes		Decision
	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very_high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very_high	yes

What about the decision attribute ?

We can define elementary sets associated with the decision as subsets of the set of all examples with the same value of the decision.

Such subsets will be called **concepts**.

For Tables 1 and 2, the concepts are {e1, e4, e5} and {e2, e3, e6}.

The first concept {e1, e4, e5} corresponds to the set of all patients free from flu, the second one {e2, e3, e6} to the set of all patients sick with flu.

Q: Whether we may tell who is free from flu and who is sick with flu on the basis of the values of attributes in Table 2 ?

A: We may observe that in terms of rough set theory, decision *Flu depends on attributes Headache and Temperature, since all elementary sets of indiscernibility* relation associated with *{Headache, Temperature} are* subsets of some concepts.

Table 2. Reduced Information Table.

	Attributes		Decision
	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very_high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very_high	yes

Generate the rules from data…

As a matter of fact, one may induce the following rules from Table 2:

- (Temperature, normal) -> (Flu, no),
- (Headache, no) and (Temperature, high) -> (Flu,no),
- (Headache, yes) and (Temperature, high) ->(Flu, yes),
- (Temperature, very_high) -> (Flu, yes).

Suppose we have two additional examples, e7 and e8, as presented in Table 3.
Elementary sets of indiscernibility relation defined by attributes Headache and Temperature are {e1}, {e2}, {e3}, {e4}, {e5, e7}, and {e6, e8}, while concepts defined by decision Flu are {e1, e4, e5, e8} and {e2, e3, e6, e7}.

Table 2. Reduced Information Table.

	Attributes		Decision
	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very_high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very_high	yes

Obviously, the decision *Flu does not* depend on attributes *Headache and Temperature since* neither {e5, e7} nor {e6, e8} are subsets of any concept.

In other words, neither concept is definable by the attribute set *{Headache, Temperature}.*

We say that Table 3 is **inconsistent** *because examples e5 and e7 are conflicting* (or are inconsistent)—for both examples the value of any attribute is the same, yet the decision value is different. (Examples e6 and e8 are also conflicting.)

Table 3. Inconsistent Information Table.

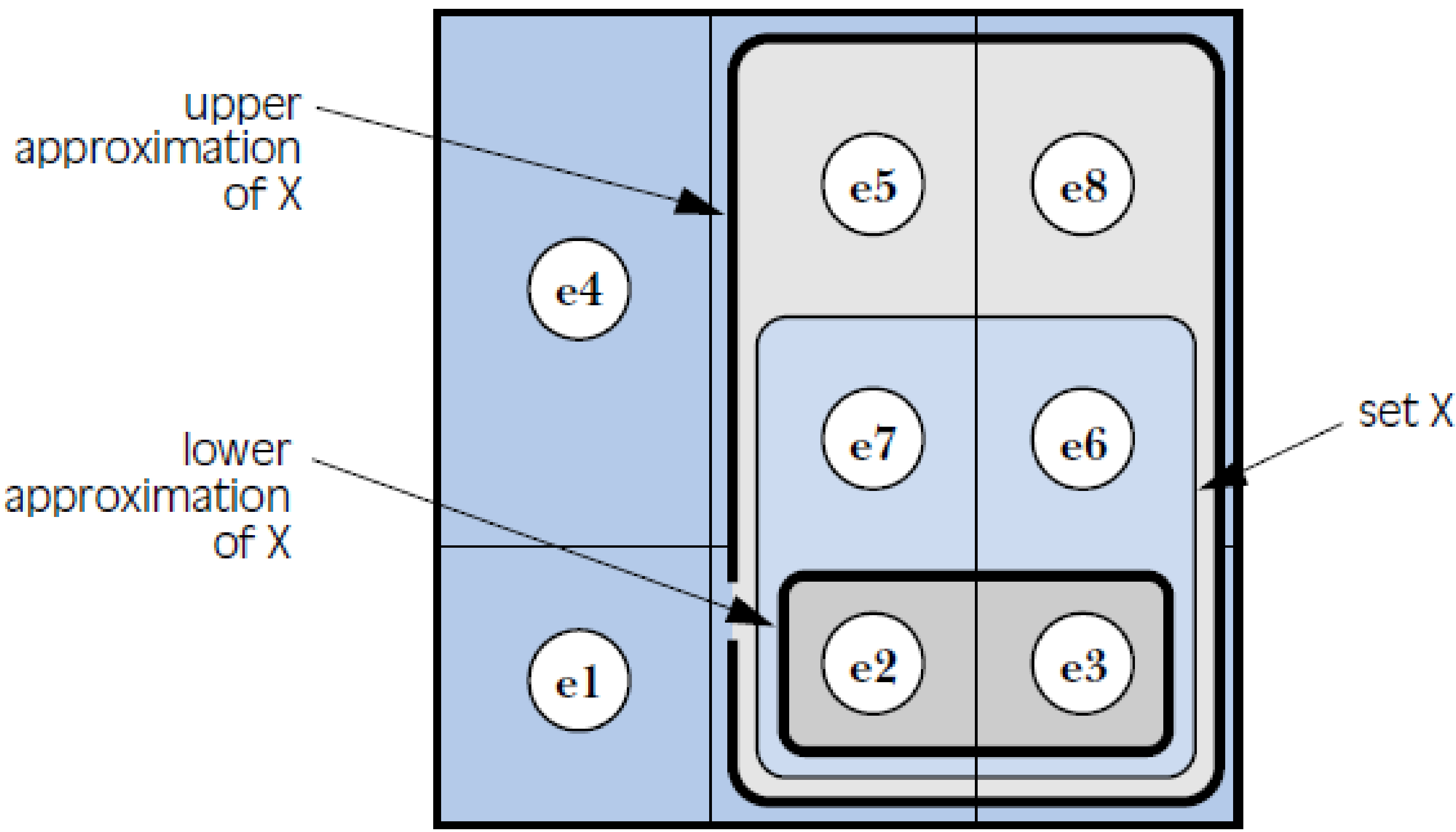
	Attributes		Decision
	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very_high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very_high	yes
e7	no	high	yes
e8	no	very_high	no

- In this situation, **rough set theory** offers a tool to deal with inconsistencies.
- The idea is very simple—for each concept *X the greatest definable set contained* in *X and the least definable set containing X* are computed.
- The former set is called a **lower approximation of X**; *the latter is called an* **upper approximation of X**.
- *In the case of Table 3, for the concept* {e2, e3, e6, e7}, describing people sick with flu, the lower approximation is equal to the set {e2, e3}, and the upper approximation is equal to the set {e2, e3, e5, e6, e7, e8}.

Table 3. Inconsistent Information Table.

	Attributes		Decision
	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very_high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very_high	yes
e7	no	high	yes
e8	no	very_high	no

Lower and upper approximations of set X



Rules generated from lower and upper approximation

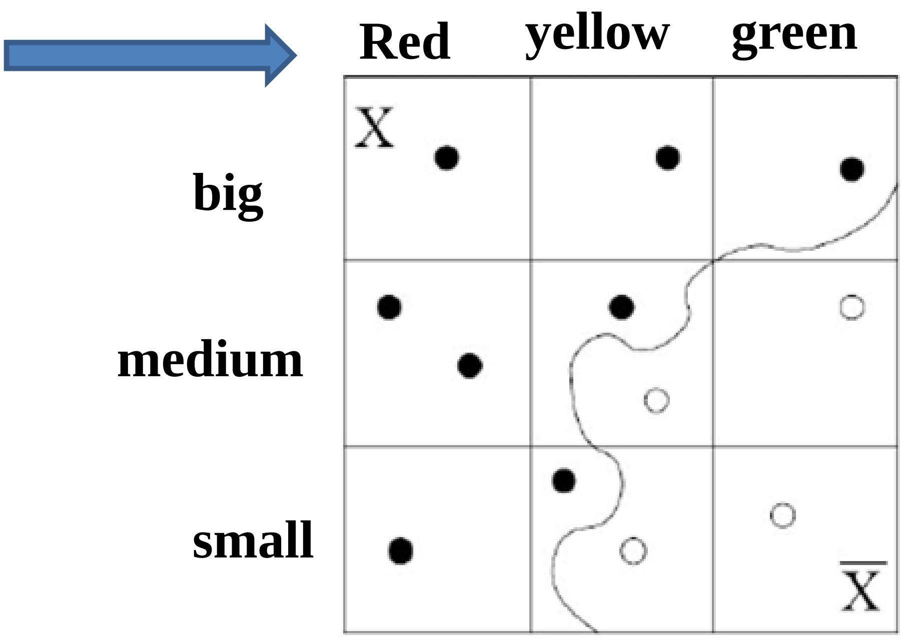
For any concept, rules induced from its lower approximation are certainly valid (hence such rules are called *certain). Rules induced from the* upper approximation of the concept are possibly valid (and are called *possible). For Table 3, certain* rules are:

(Temperature, normal) -> (Flu, no),
(Headache, yes) and (Temperature, high) -> (Flu, yes),
(Headache, yes) and (Temperature, very_high) -> (Flu, yes);

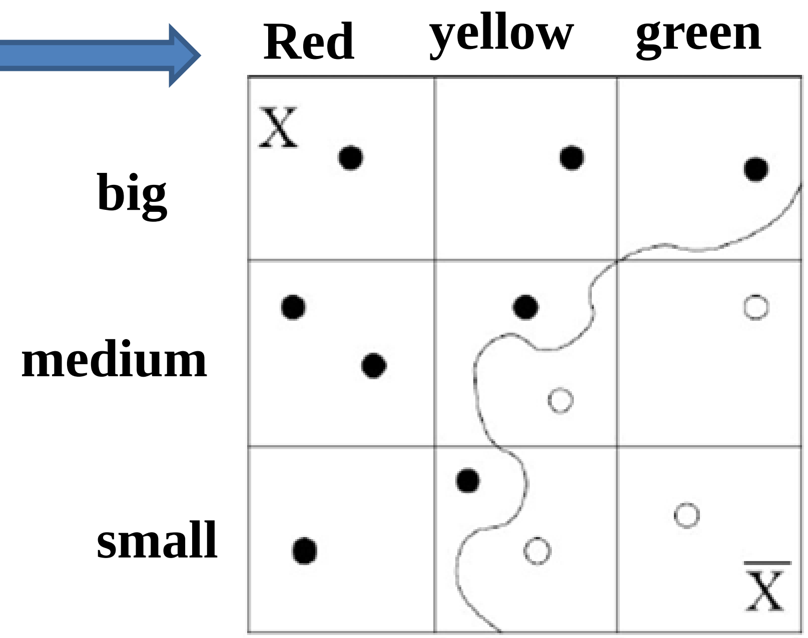
and possible rules are:

(Headache, no) -> (Flu, no),
(Temperature, normal) -> (Flu, no),
(Temperature, high) -> (Flu, yes),
(Temperature, very_high) -> (Flu, yes).

Id	colour	size	Ripe apple ?
X1	Red	Big	Yes
X2	Yellow	Medium	Yes
X3	Green	Small	No
X4	Green	Big	Yes
X5	Yellow	Medium	No
X6	Red	Medium	Yes
X7	Yellow	Big	Yes
X8	Red	Medium	Yes
X9	Yellow	Small	No
X10	Yellow	Small	Yes
X11	Red	Small	Yes
x12	green	medium	No



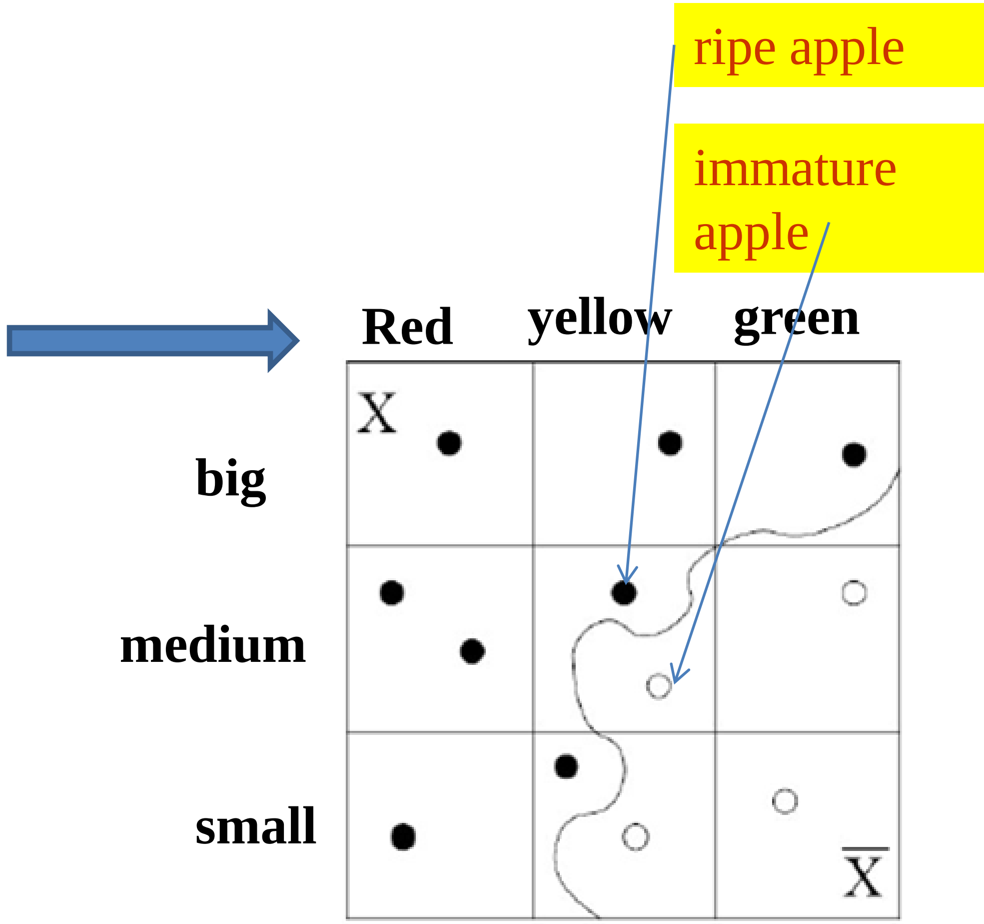
Id	colour	size	Ripe apple ?
X1	Red	Big	Yes
X2	Yellow	Medium	Yes
X3	Green	Small	No
X4	Green	Big	Yes
X5	Yellow	Medium	No
X6	Red	Medium	Yes
X7	Yellow	Big	Yes
X8	Red	Medium	Yes
X9	Yellow	Small	No
X10	Yellow	Small	Yes
X11	Red	Small	Yes
x12	green	medium	No



X – ripe apple

$\overline{X}$ – NOT ripe apple

Id	colour	size	Ripe apple ?
X1	Red	Big	Yes
X2	Yellow	Medium	Yes
X3	Green	Small	No
X4	Green	Big	Yes
X5	Yellow	Medium	No
X6	Red	Medium	Yes
X7	Yellow	Big	Yes
X8	Red	Medium	Yes
X9	Yellow	Small	No
X10	Yellow	Small	Yes
X11	Red	Small	Yes
x12	green	medium	No



X – ripe apple

$\overline{X}$ – NOT ripe apple

lower approximation

- $\underline{BX} = \cup \{Y \in IND(B): Y \subseteq X\}$

- So there are the objects that belong to the IND (B), which all are included in the set of X.
- About the objects belonging to the lower approximation, we say that they **SURELY** belong to a given concept (decision class).

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
p1	*no*	*yes*	*high*	*yes*
p2	*yes*	*no*	*high*	*yes*
p3	*yes*	*yes*	*very high*	*yes*
p4	*no*	*yes*	*normal*	*no*
p5	*yes*	*no*	*high*	*no*
p6	*no*	*yes*	*very high*	*yes*

$$\underline{\mathbf{B}}\mathbf{X} = \cup \{\mathbf{Y} \in \mathbf{IND(B)}: \mathbf{Y} \subseteq \mathbf{X}\}$$

Which of the Y belonged to IND(B) are all included in X ?

$IND(B) = \{\{1\},\{2,5\},\{3\},\{4\},\{6\}\}$

$X = X_{yes} + X_{no}$

$X_{yes} = \{1,2,3,6\}$
$X_{no} = \{4,5\}$
$\underline{\mathbf{B}}\mathbf{X}_{yes} = \{1,3,6\}$
$\underline{\mathbf{B}}\mathbf{X}_{No} = \{4\}$

Objects {1,3,6} surely have a flu !
Object {4} does not have a flue for SURE!

Upper aproximation

- $\overline{BX} = \cup\{Y \in IND(B): Y \cap X \neq \phi\}$
- So they are those objects that belong to the IND (B), which have a common part with set X.
- About the objects belonging to the upper approximation, we say that they **perhaps** belong to a given concept (decision class).

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
p1	*no*	*yes*	*high*	*yes*
p2	*yes*	*no*	*high*	*yes*
p3	*yes*	*yes*	*very high*	*yes*
p4	*no*	*yes*	*normal*	*no*
p5	*yes*	*no*	*high*	*no*
p6	*no*	*yes*	*very high*	*yes*

$$\overline{B}X = \cup\{Y \in IND(B): Y \cap X \neq \phi\}$$

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

$X = X_{yes} + X_{no}$

Which of Y that belong to IND(B) have a common part with X ?

$X_{yes} = \{1,2,3,6\}$

$\underline{X}_{no} = \{4,5\}$

$\underline{B}X_{yes} = \{1,2,3,6,5\}$

$\overline{B}X_{No} = \{2,5,4\}$

Objects {1,2,3,6,5} maybe have a flu !

Objects {2,5,4} maybe doesn't have a flu !

Patient	Headache	Muscle-pain	Temperature	Flu
p1	no	yes	high	yes
p2	yes	no	high	yes
p3	yes	yes	very high	yes
p4	no	yes	normal	no
p5	yes	no	high	no
p6	no	yes	very high	yes

1. if (h=no) and (m=yes) and (t=high) then (f=yes)
2. ….

….

Deterministic rules

$$\forall_{\substack{x,y\in U\\ x\neq y}} \left(\forall_{c\in C}(f(x,c)=f(y,c)) \Rightarrow \forall_{d\in D}(f(x,d)=f(y,d))\right)$$

NON -Deterministic rules

$$\forall_{\substack{x,y\in U\\ x\neq y}} \left(\forall_{c\in C}(f(x,c)=f(y,c)) \wedge \exists_{d\in D}(f(x,d)\neq f(y,d))\right)$$

Inconsistency

- Decisions may be inconsistent because of limited clear discrimination between criteria and because of hesitation on the part of the decision maker.
- These inconsistencies cannot be considered as a simple error or as noise. They can convey important information that should be taken into account in the construction of the decision makers preference model.
- The rough set approach is intended to deal with inconsistency and this is a major argument to support its application to multiple-criteria decision analysis.

Inconsistent decision table

The Table is *inconsistent because:*

- *examples e5 and e7 are conflicting* (or are inconsistent)—for both examples the value of any attribute is the same, yet the decision value is different.
- Examples e6 and e8 are also conflicting.

	Attributes		Decision
	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very_high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very_high	yes
e7	no	high	yes
e8	no	very_high	no

How to deal with the inconsistency

The idea is very simple—for each concept X the greatest definable set contained in X and the least definable set containing X are computed.

The former set is called a lower approximation of X;

the latter is called an upper approximation of X.

for the concept {e2, e3, e6, e7}, describing people sick with flu, the lower approximation is equal to the set {e2, e3}, and the upper approximation is equal to the set {e2, e3, e5, e6, e7, e8}.

	Attributes		**Decision**
	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very_high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very_high	yes
e7	no	high	yes
e8	no	very_high	no

- Either of these two concepts is an example of a *rough set, a set that is* undefinable by given attributes.
- The set {e5, e6, e7, e8}, containing elements from the upper approximation of *X that are not members of the lower approximation* of *X, is called a boundary region.*
- *Elements of* the boundary region cannot be classified as members of the set *X.*

- For any concept, rules induced from its lower approximation are certainly valid (hence such rules are called *certain).*
- *Rules induced from the* upper approximation of the concept are possibly valid (and are called *possible)*

	Attributes		Decision
	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very_high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very_high	yes
e7	no	high	yes
e8	no	very_high	no

Certain rules

- (Temperature, normal) -> (Flu, no),
- (Headache, yes) and (Temperature, high) -> (Flu, yes),
- (Headache, yes) and (Temperature, very_high) -> (Flu, yes);

Possible rules

- (Headache, no) -> (Flu, no),
- (Temperature, normal) -> (Flu, no),
- (Temperature, high) -> (Flu, yes),
- (Temperature, very_high) -> (Flu, yes).

How to deal with inconsistency?

1. Ask the expert „what to do ?”
2. Separate the conflicting examples to different tables
3. Remove the examples with less support
4. Quality methods – based on lower or upper approximations
5. The method of generating new (general) decision attribute

4th method

- 1 step: calculate the accuracy of the lower or upper approximation

$$\gamma_B(X) = \frac{|\underline{B}X|}{|U|} \qquad \gamma_B(X) = \frac{|\overline{B}X|}{|U|}$$

- 2 step: we remove the examples with the less value of the accuracy

- IND(B) = {{1}{2,5}{3}{4}{6}}
- X_{yes} = {1,2,3,6}
- X_{no} = {5,4}
- $\underline{BX}_{yes}$ = {1,3,6}
- $\underline{BX}_{No}$ = {4}
- γ_{yes} =3/6
- γ_{No} =1/6

Remove the object(s) with less value of acuracy of the approximation

Remove the object that make a conflict here and are in the set of decision„No”

- $IND(B) = \{\{1\}\{2,\cancel{5}\}\{3\}\{4\}\{6\}\}$
- $X_{yes} = \{1,2,3,6\}$
- $X_{no} = \{\cancel{5},4\}$
- $\underline{BX}_{yes} = \{1,3,6\}$
- $\underline{BX}_{No} = \{4\}$
- $\gamma_{yes} = 3/6$
- $\gamma_{\ No} = 1/6$

Remove the object(s) with less value of acuracy of the approximation

Remove the object that make a conflict here and are in the set of decision„No"

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
p1	*no*	*yes*	*high*	*yes*
p2	*yes*	*no*	*high*	*yes*
p3	*yes*	*yes*	*very high*	*yes*
p4	*no*	*yes*	*normal*	*no*
p6	*no*	*yes*	*very high*	*yes*

5th method

Conditional attributes

$$\underbrace{\text{a} \quad \text{b} \quad \text{c} \quad \text{e}}_{\text{Conditional attributes}}$$

U	a	b	c	e	d
1	0	0	1	0	1
2	1	1	1	0	1
3	0	1	0	0	2
4	1	1	1	0	2
5	1	0	0	0	2
6	1	1	0	0	3
7	1	1	1	1	2
8	1	1	1	0	1

U	a	b	c	e	δ_A
1	0	0	1	0	1
2	1	1	1	0	{1,2}
3	0	1	0	0	2
4	1	1	1	0	{1,2}
5	1	0	0	0	2
6	1	1	0	0	3
7	1	1	1	1	2
8	1	1	1	0	{1,2}

rule generation algorithm

Authors: Andrzej Skowron, 1993

The algorithm which find in the set of examples the set of minimal certain decision rules

student	a	b	c	d
x_1	a_1	b_1	c_1	T
x_2	a_1	b_1	c_2	T
x_3	a_2	b_1	c_3	T
x_4	a_2	b_1	c_4	N
x_5	a_1	b_2	c_1	N
x_6	a_1	b_2	c_2	T
x_7	a_2	b_2	c_3	T
x_8	a_2	b_2	c_4	N

- r1: $a = a_1 \wedge b = b_1 \rightarrow d = T$
- r2: $b = b_1 \wedge c = c_1 \rightarrow d = T$
- r3: $b = b_1 \wedge c = c_2 \rightarrow d = T$
- r4: $c = c_3 \rightarrow d = T$
- r5: $c = c_4 \rightarrow d = N$
- r6: $b = b_2 \wedge c = c_1 \rightarrow d = N$
- r7: $c = c_2 \rightarrow d = T$

Discernibility matrix

Let $S=(U,A)$ be an information system with n objects. The discernibility matrix of S is a symmetric $n \times n$ matrix with entries c_{ij} as given below.

$$c_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \text{ for } i,j=1,\ldots,n$$

Each entry thus consists of the set of attributes upon which objects x_i and x_j differ. Since discernibility matrix is symmetric and $c_{ii} = \emptyset$ (the empty set) for $i=1,\ldots,n$. Thus, this matrix can be represented using only elements in its lower triangular part, i.e. for $1 \leq j < i \leq n$.

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
*p*1	*no*	*yes*	*high*	*yes*
*p*2	*yes*	*no*	*high*	*yes*
*p*3	*yes*	*yes*	*very high*	*yes*
*p*4	*no*	*yes*	*normal*	*no*
*p*5	*yes*	*no*	*high*	*no*
*p*6	*no*	*yes*	*very high*	*yes*

	*p*1	*p*2	*p*3	*p*4	*p*5	*p*6
*p*1						
*p*2	*H,M*					
*p*3	*H,T*	*M,T*				
*p*4	*T,F*	*H,M,T,F*	*H,T,F*			
*p*5	*H,M,F*	*F*	*M,T,F*	*H,M,T*		
*p*6	*T*	*H,M,T*	*H*	*T,F*	*H,M,T,F*	

Discernibility Matrices and Functions

In order to compute easily reducts and the core we can use discernibility matrix, which is defined below.

	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$
$p1$						
$p2$	H,M					
$p3$	H,T	M,T				
$p4$	T,F	H,M,T,F	H,T,F			
$p5$	H,M,F	F	M,T,F	H,M,T		
$p6$	T	H,M,T	H	T,F	H,M,T,F	

Discernibility function

With every discernibility matrix one can uniquely associate a discernibility function defined below.

A *discernibility function* f_S for an information system S is a Boolean function of m Boolean variables $a_1^*, \ldots, a_m^*$ (corresponding to the attribute $a_1, \ldots, a_m$) defined as follows.

$$f_S\left(a_1^*, \ldots, a_m^*\right) = \forall\left\{\exists c_{ij}^* \mid 1 \le j \le i \le n, c_{ij} \ne \phi\right\}$$

where $c_{ij}^* = \left\{a^* \mid a \in c_{ij}\right\}$. The set of all prime implicants of f_S determines the set of all reducts of A.

A prime implicant is a minimal implicant (with respect to the number of its literals). Here we are interested in implicants of monotone Boolean functions only, i.e. functions constructed without negation.

Boolean Algebra Laws Summary

Boolean Theorems

1) $X \ 0 = 0$

2) $X \ 1 = X$

3) $X \ X = X$

4) $X \ \overline{X} = 0$

5) $X + 0 = X$

6) $X + 1 = 1$

7) $X + X = X$

8) $X + \overline{X} = 1$

9) $\overline{\overline{X}} = X$

10A) $X \ Y = Y \ X$

10B) $X + Y = Y + X$

Commutative Law

11A) $X(YZ) = (XY)Z$

11B) $X + (Y + Z) = (X + Y) + Z$

Associative Law

12A) $X(Y + Z) = XY + XZ$

12B) $(X + Y)(W + Z) = XW + XZ + YW + YZ$

Distributive Law

13A) $X + \overline{X}Y = X + Y$

13B) $\overline{X} + XY = \overline{X} + Y$

13C) $X + \overline{X}\overline{Y} = X + \overline{Y}$

13D) $\overline{X} + X\overline{Y} = \overline{X} + \overline{Y}$

Consensus Theorem

In this table H,M,T,F denote Headache, Muscle-pain, Temperature and Flu, respectively.

	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$
$p1$						
$p2$	H,M					
$p3$	H,T	M,T				
$p4$	T,F	H,M,T,F	H,T,F			
$p5$	H,M,F	F	M,T,F	H,M,T		
$p6$	T	H,M,T	H	T,F	H,M,T,F	

The discernibility function for this table is

$$f_S(H,M,T,F) = (H \vee M)(H \vee T)(T \vee F)(H \vee M \vee F)T$$
$$(M \vee T)(H \vee M \vee T \vee F)F(H \vee M \vee T)$$
$$(H \vee T \vee F)(M \vee T \vee F)H$$
$$(H \vee M \vee T)(T \vee F)$$
$$(H \vee M \vee T \vee F)$$

- where V denotes the disjunction and the conjunction is omitted in the formula.

DM -> reduct

- After simplification, the discernibility function using laws of Boolean algebra we obtain the following expression **HTF*,*** which says that there is only one reduct *{H,T,F} in the data table and it is the core.*
- Relative reducts and core can be computed also using the discernibility matrix, which needs slight modification.

Deterministic DT

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
p1	*no*	*yes*	*high*	*yes*
p2	*yes*	*no*	*high*	*yes*
p3	*yes*	*yes*	*very high*	*yes*
p4	*no*	*yes*	*normal*	*no*
p6	*no*	*yes*	*very high*	*yes*

Dissimilarity functions

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
p1	*no*	*yes*	*high*	*yes*
p2	*yes*	*no*	*high*	*yes*
p3	*yes*	*yes*	*very high*	*yes*
p4	*no*	*yes*	*normal*	*no*
p6	*no*	*yes*	*very high*	*yes*

	p1	**p2**	**p3**	**p4**	**P6**
P1					
P2	H,M				
P3	H,T	M,T			
P4	T	H,M,T	H,T		
p6	T	H,M,T	H	T	

Which examples have different decision ?

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
$p1$	*no*	*yes*	*high*	*yes*
$p2$	*yes*	*no*	*high*	*yes*
$p3$	*yes*	*yes*	*very high*	*yes*
$p4$	*no*	*yes*	*normal*	*no*
$p6$	*no*	*yes*	*very high*	*yes*

example	Different decision
P1	P4
P2	P4
P3	P4
P4	P1,p2,p3,p4
p6	P4

Dissimilarity function for p1

example	Different decision
P1	P4
P2	P4
P3	P4
P4	P1,p2,p3,p4
p6	P4

	p1	p2	p3	p4	P6
P1					
P2	H,M				
P3	H,T	M,T			
P4	T	H,M,T	H,T		
p6	T	H,M,T	H	T	

$$f(p1,\{yes\}) = t$$

Dissimilarity function for p2

example	Different decision
P1	P4
P2	P4
P3	P4
P4	P1,p2,p3,p4
p6	P4

	p1	p2	p3	p4	P6
P1					
P2	H,M				
P3	H,T	M,T			
P4	T	H,M,T	H,T		
p6	T	H,M,T	H	T	

$$f(p2,\{yes\}) = h \vee m \vee t$$

Dissimilarity function for p3

example	Different decision
P1	P4
P2	P4
P3	P4
P4	P1,p2,p3,p4
p6	P4

	p1	p2	p3	p4	P6
P1					
P2	H,M				
P3	H,T	M,T			
P4	T	H,M,T	H,T		
p6	T	H,M,T	H	T	

$$f(p3,\{yes\}) = h \vee t$$

Dissimilarity function for p6

example	Different decision
P1	P4
P2	P4
P3	P4
P4	P1,p2,p3,p4
p6	P4

	p1	p2	p3	p4	P6
P1					
P2	H,M				
P3	H,T	M,T			
P4	T	H,M,T	H,T		
p6	T	H,M,T	H	T	

$$f(p6,\{yes\}) = t$$

Dissimilarity function for p4

example	Different decision
P1	P4
P2	P4
P3	P4
P4	P1,p2,p3,p4
p6	P4

	p1	p2	p3	p4	P6
P1					
P2	H,M				
P3	H,T	M,T			
P4	T	H,M,T	H,T		
p6	T	H,M,T	H	T	

$$f(p4,\{no\}) = t \wedge (h \vee m \vee t) \wedge (h \vee t) \wedge t = t$$

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
$p1$	*no*	*yes*	*high*	*yes*
$p2$	*yes*	*no*	*high*	*yes*
$p3$	*yes*	*yes*	*very high*	*yes*
$p4$	*no*	*yes*	*normal*	*no*
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$p6$	*no*	*yes*	*very high*	*yes*

$$f(p1, \{yes\}) = t$$

Temperature = high → flu = yes

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
$p1$	*no*	*yes*	*high*	*yes*
$p2$	*yes*	*no*	*high*	*yes*
$p3$	*yes*	*yes*	*very high*	*yes*
$p4$	*no*	*yes*	*normal*	*no*
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$p6$	*no*	*yes*	*very high*	*yes*

$$f(p2,\{yes\}) = h \vee m \vee t$$

headache = yes → flu = yes

Muscle-pain = no → flu = yes

Temperature = hight → flu = yes

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
p1	*no*	*yes*	*high*	*yes*
p2	*yes*	*no*	*high*	*yes*
p3	*yes*	*yes*	*very high*	*yes*
p4	*no*	*yes*	*normal*	*no*
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
p6	*no*	*yes*	*very high*	*yes*

$$f(p3,\{yes\}) = h \vee t$$

headache = yes → flu = yes

Temperature = very hight → flu = yes

$$f(p6,\{yes\}) = t$$

Patient	Headache	Muscle-pain	Temperature	Flu
$p1$	no	yes	high	yes
$p2$	yes	no	high	yes
$p3$	yes	yes	very high	yes
$p4$	no	yes	normal	no
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$p6$	no	yes	very high	yes

Temperature = hight → flu = yes

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
*p*1	*no*	*yes*	*high*	*yes*
*p*2	*yes*	*no*	*high*	*yes*
*p*3	*yes*	*yes*	*very high*	*yes*
*p*4	*no*	*yes*	*normal*	*no*
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
*p*6	*no*	*yes*	*very high*	*yes*

$$f(p4,\{no\}) = t \wedge (h \vee m \vee t) \wedge (h \vee t) \wedge t = t$$

Temperature = normal → flu = no

Temperature = high → flu = yes

Temperature = high → flu = yes

headache = yes → flu = yes
Muscle-pain = no → flu = yes
Temperature = hight → flu = yes

Temperature = high → flu = yes

headache = yes → flu = yes
Temperature = very hight → flu = yes

Temperature = normal → flu = no

Temperature = high → flu = yes

$\boxed{\text{Temperature = high → flu = yes}}$

headache = yes → flu = yes

Muscle-pain = no → flu = yes

$\boxed{\text{Temperature = hight → flu = yes}}$

$\boxed{\text{Temperature = high → flu = yes}}$

headache = yes → flu = yes

Temperature = very hight → flu = yes

Temperature = normal → flu = no

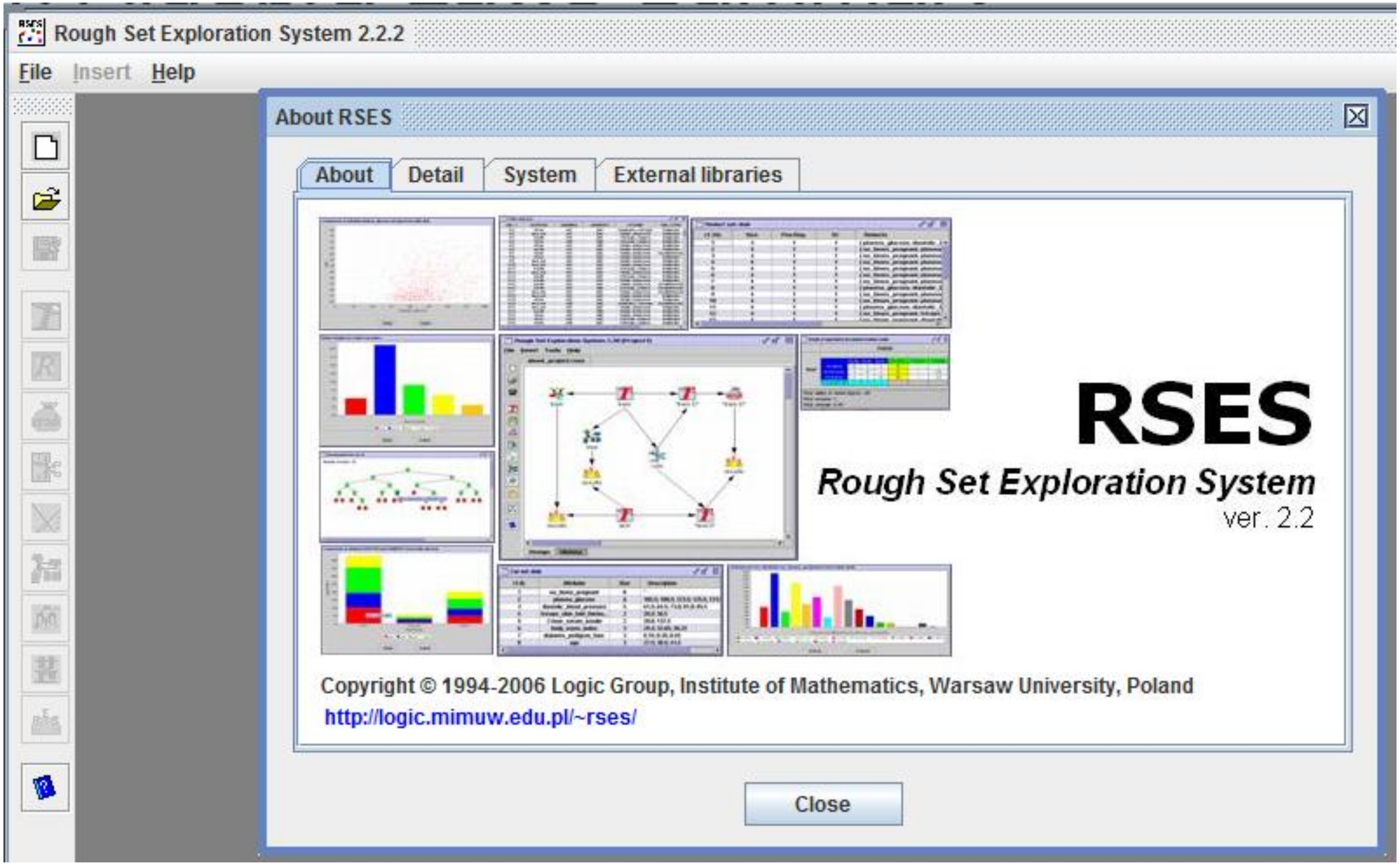
Final set of rules

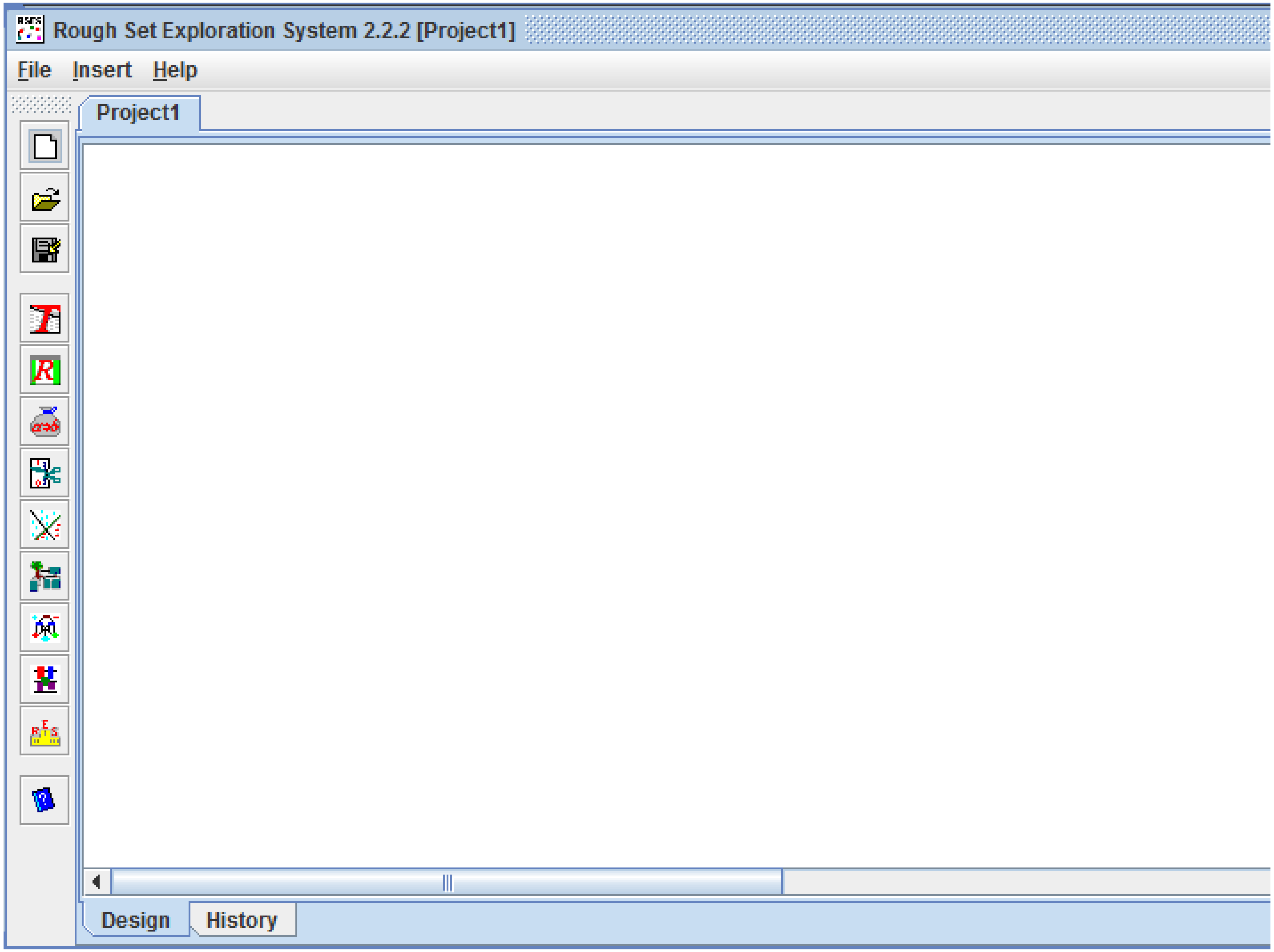
Temperature = high → flu = yes
headache = yes → flu = yes
Muscle-pain = no → flu = yes

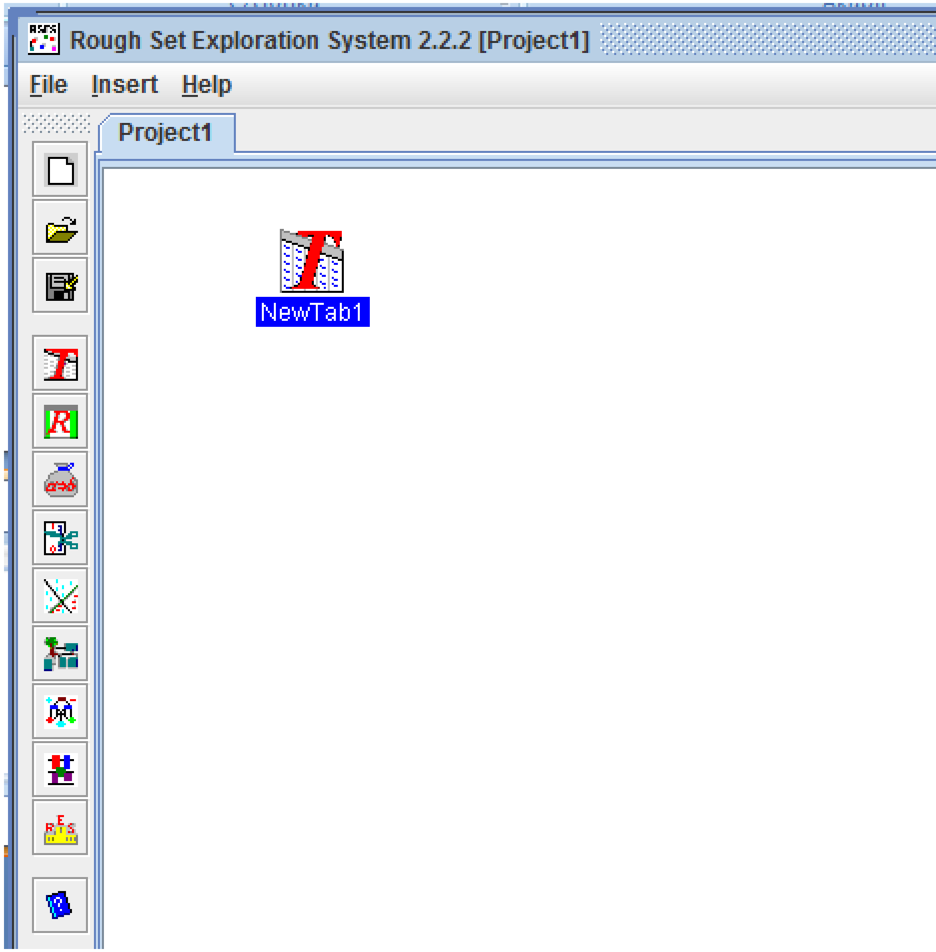
headache = yes → flu = yes
Temperature = very hight → flu = yes

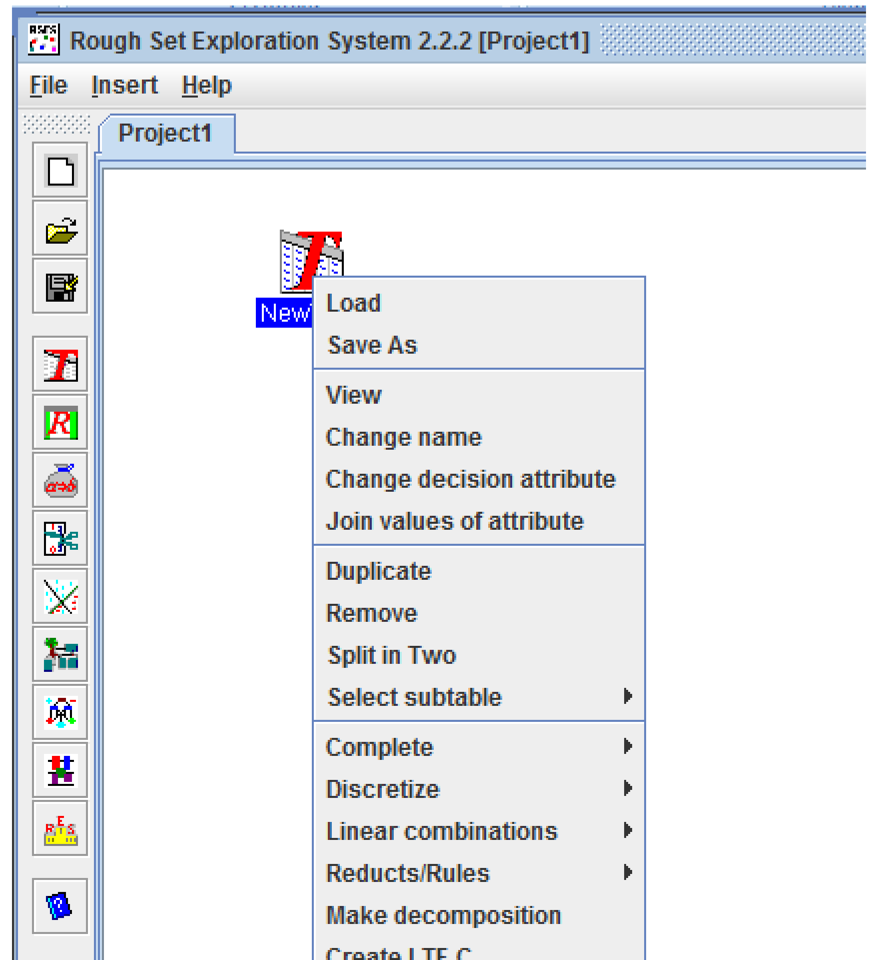
Temperature = normal → flu = no

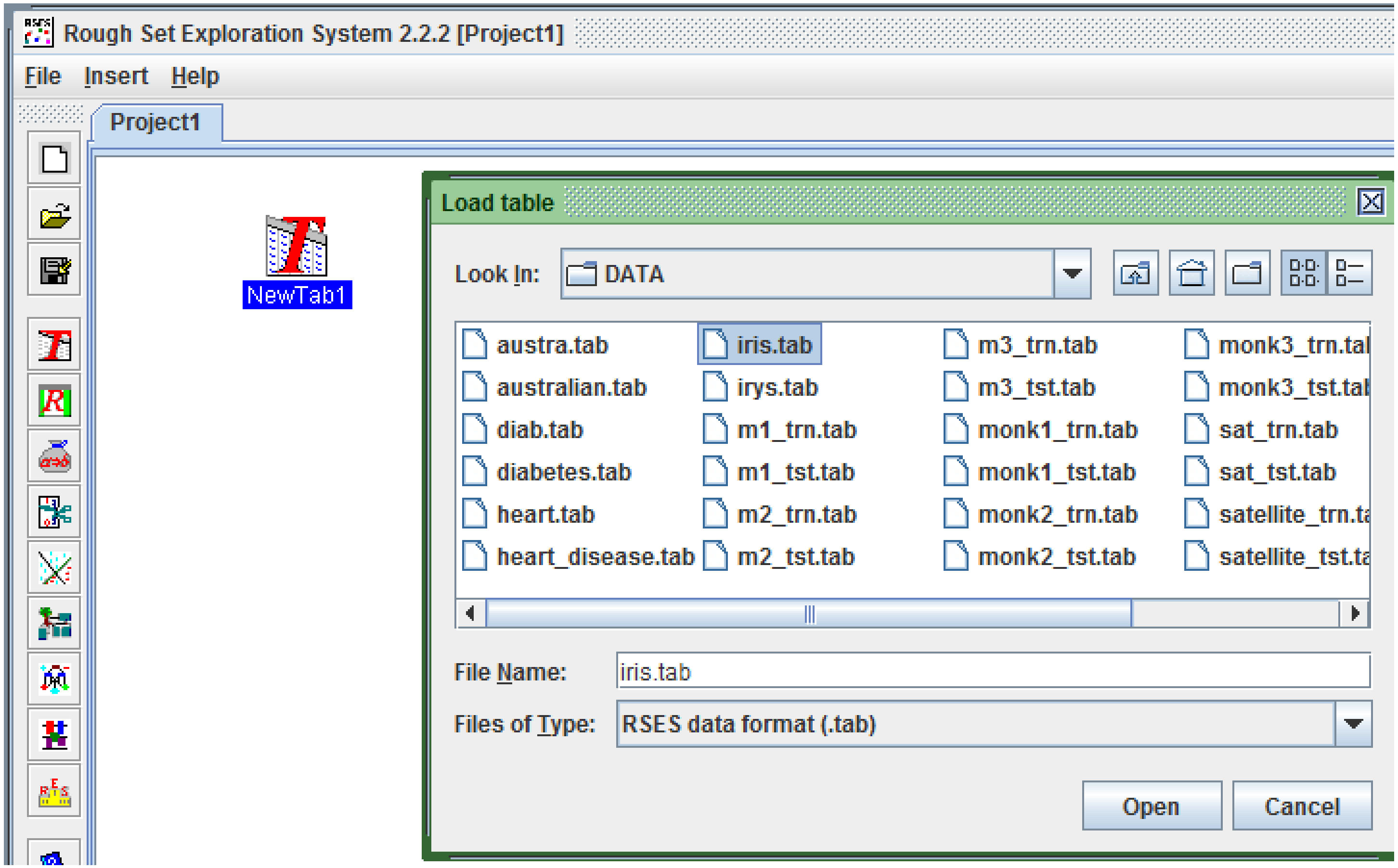
- Rough set theory offers effective methods that are applicable in many branches of AI.
- One of the advantages is that programs implementing its methods may easily run on parallel computers.

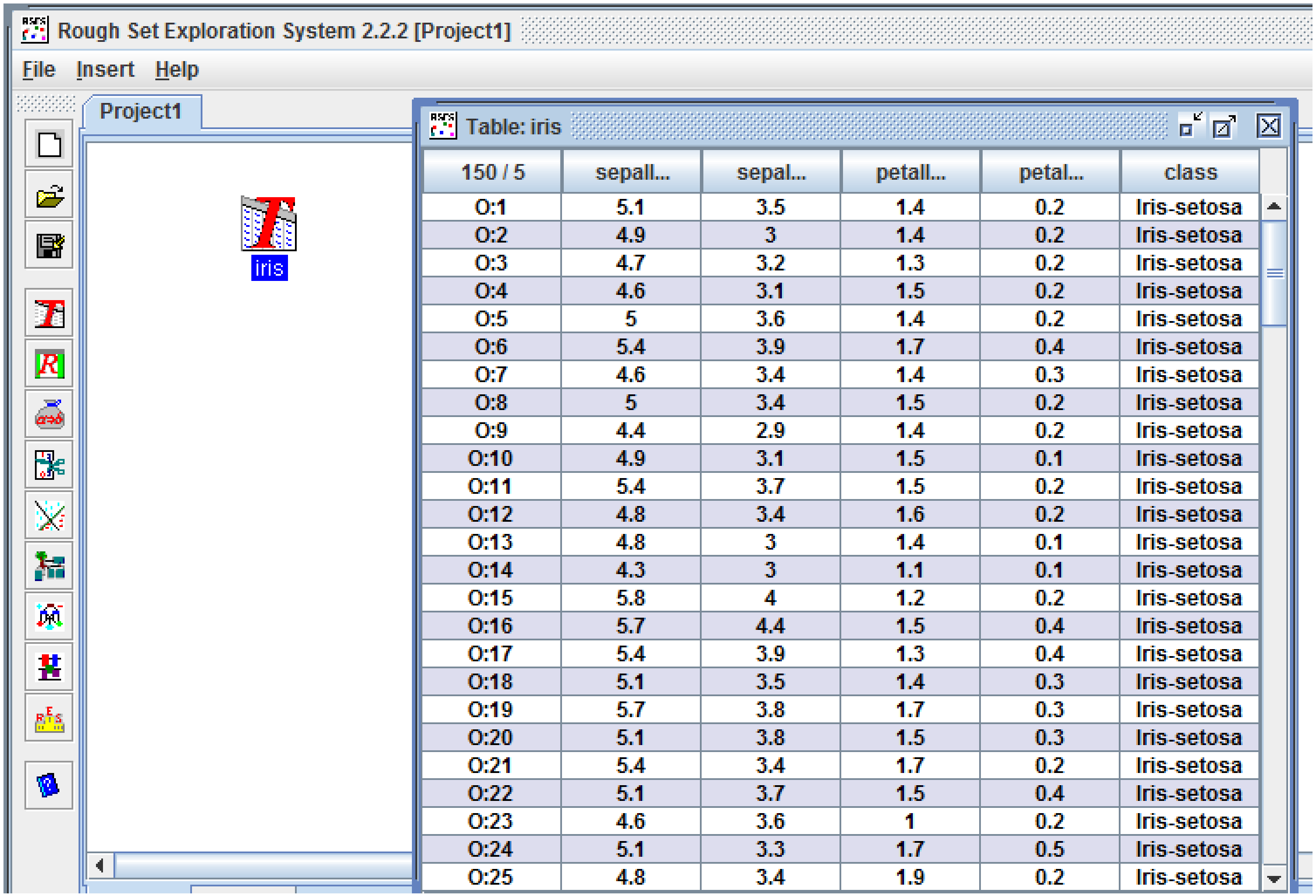

Rough Set Exploration System 2.2.2
File
Insert
Help
About RSES
About
Detail
System
External libraries
RSES
Rough Set Exploration System
ver. 2.2
Copyright © 1994-2006 Logic Group, Institute of Mathematics, Warsaw University, Poland
http://logic.mimuw.edu.pl/~rses/
Close


Rough Set Exploration System 2.2.2 [Project1]
File
Insert
Help
Project1
Design
History

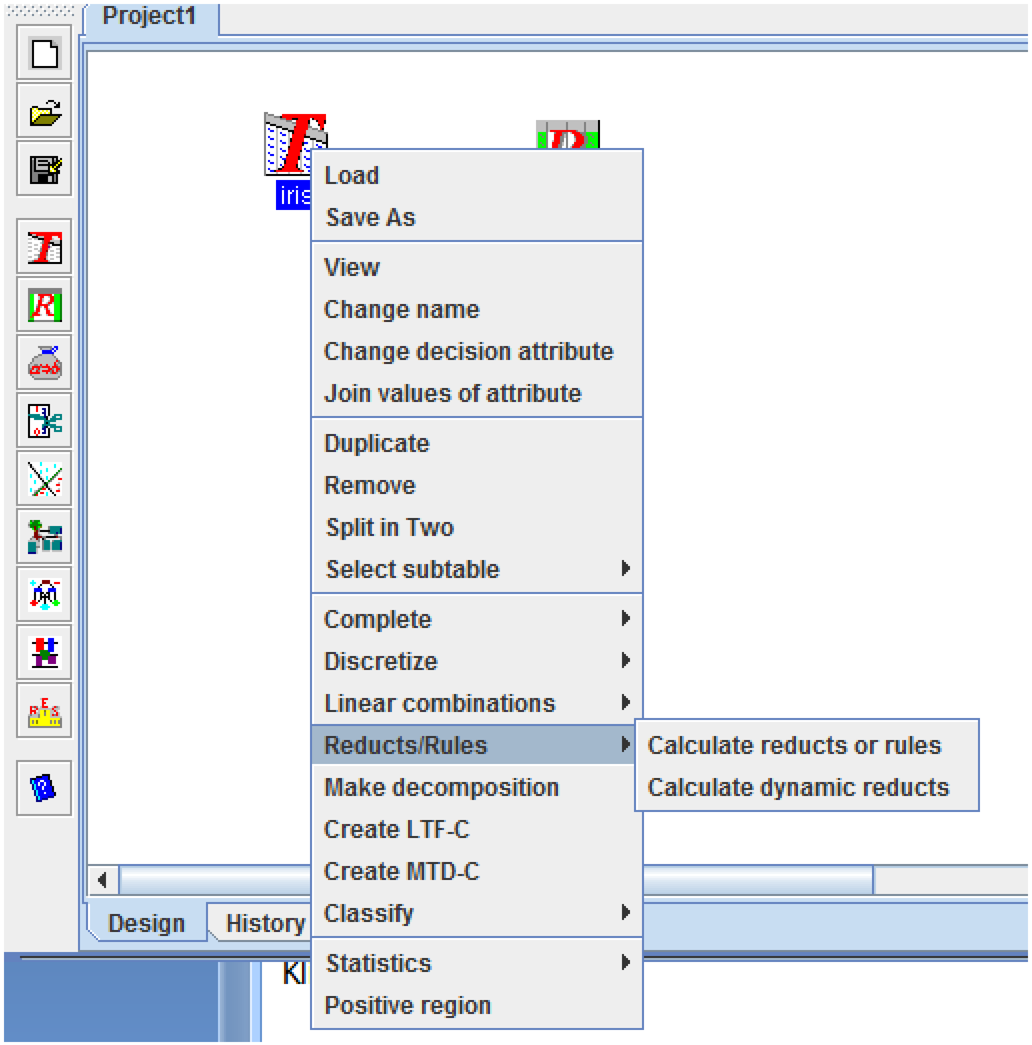

Rough Set Exploration System 2.2.2 [Project1]
File Insert Help
Project1
NewTab1

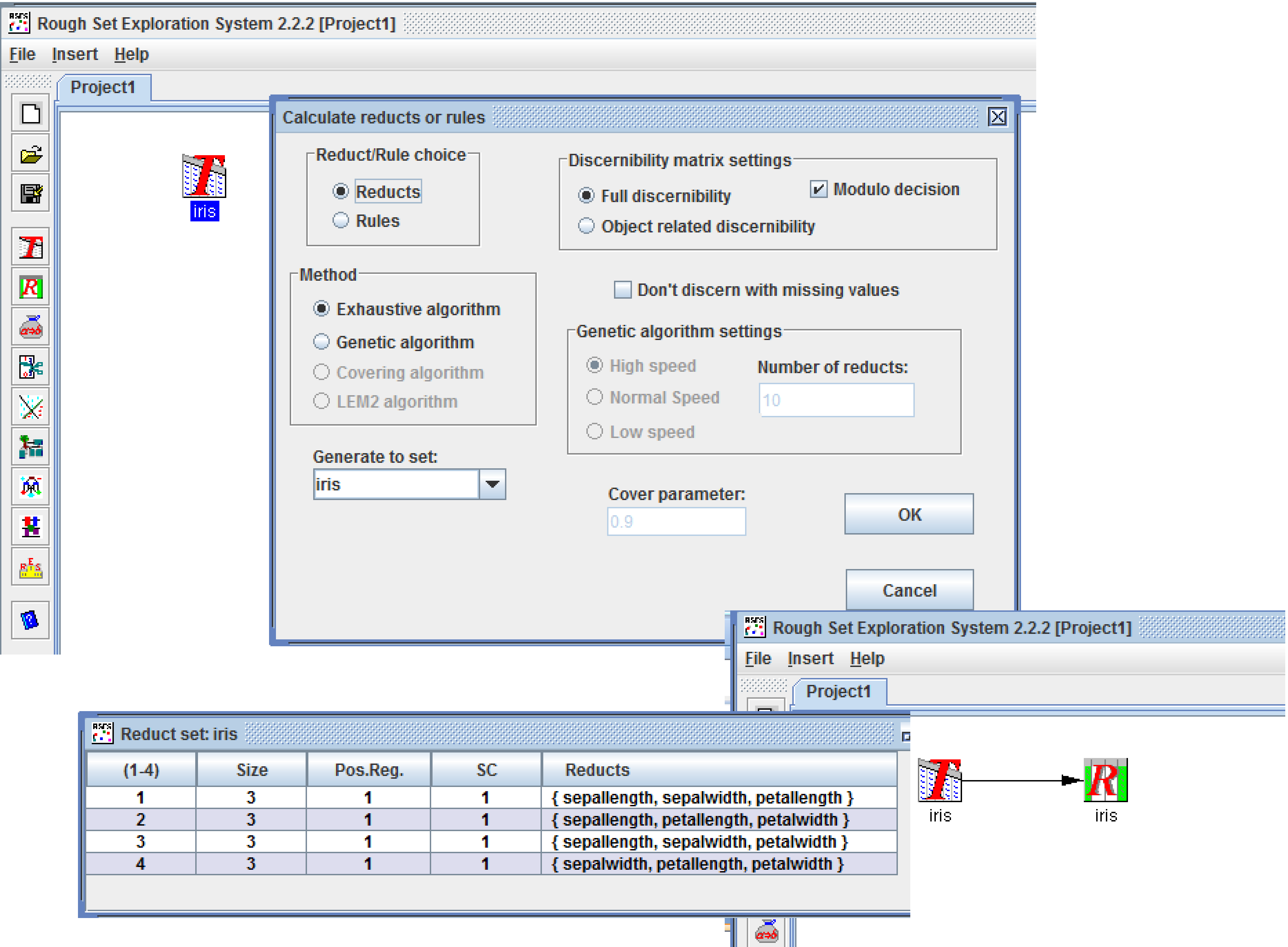

Rough Set Exploration System 2.2.2 [Project1]
File Insert Help
Project1
New
Load
Save As
View
Change name
Change decision attribute
Join values of attribute
Duplicate
Remove
Split in Two
Select subtable
Complete
Discretize
Linear combinations
Reducts/Rules
Make decomposition

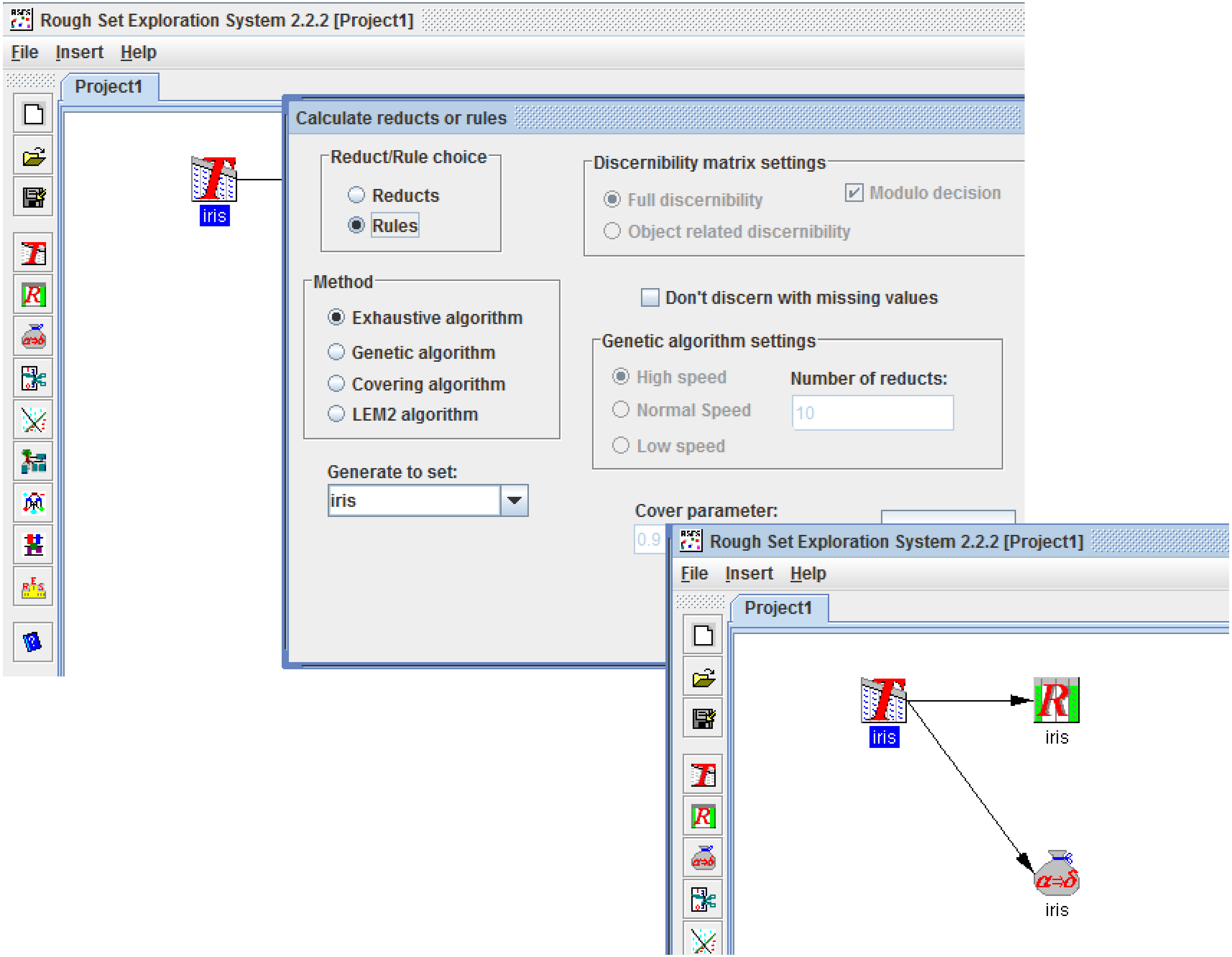

Rough Set Exploration System 2.2.2 [Project1]
File
Insert
Help
Project1
NewTab1
Load table
Look In:
DATA
austra.tab
australian.tab
diab.tab
diabetes.tab
heart.tab
heart_disease.tab
iris.tab
irys.tab
m1_trn.tab
m1_tst.tab
m2_trn.tab
m2_tst.tab
m3_trn.tab
m3_tst.tab
monk1_trn.tab
monk1_tst.tab
monk2_trn.tab
monk2_tst.tab
monk3_trn.tal
monk3_tst.tal
sat_trn.tab
sat_tst.tab
satellite_trn.ta
satellite_tst.ta
File Name:
iris.tab
Files of Type:
RSES data format (.tab)
Open
Cancel



150 / 5	sepall...	sepal...	petall...	petal...	class
O:1	5.1	3.5	1.4	0.2	Iris-setosa
O:2	4.9	3	1.4	0.2	Iris-setosa
O:3	4.7	3.2	1.3	0.2	Iris-setosa
O:4	4.6	3.1	1.5	0.2	Iris-setosa
O:5	5	3.6	1.4	0.2	Iris-setosa
O:6	5.4	3.9	1.7	0.4	Iris-setosa
O:7	4.6	3.4	1.4	0.3	Iris-setosa
O:8	5	3.4	1.5	0.2	Iris-setosa
O:9	4.4	2.9	1.4	0.2	Iris-setosa
O:10	4.9	3.1	1.5	0.1	Iris-setosa
O:11	5.4	3.7	1.5	0.2	Iris-setosa
O:12	4.8	3.4	1.6	0.2	Iris-setosa
O:13	4.8	3	1.4	0.1	Iris-setosa
O:14	4.3	3	1.1	0.1	Iris-setosa
O:15	5.8	4	1.2	0.2	Iris-setosa
O:16	5.7	4.4	1.5	0.4	Iris-setosa
O:17	5.4	3.9	1.3	0.4	Iris-setosa
O:18	5.1	3.5	1.4	0.3	Iris-setosa
O:19	5.7	3.8	1.7	0.3	Iris-setosa
O:20	5.1	3.8	1.5	0.3	Iris-setosa
O:21	5.4	3.4	1.7	0.2	Iris-setosa
O:22	5.1	3.7	1.5	0.4	Iris-setosa
O:23	4.6	3.6	1	0.2	Iris-setosa
O:24	5.1	3.3	1.7	0.5	Iris-setosa
O:25	4.8	3.4	1.9	0.2	Iris-setosa


Project1
iris
Load
Save As
View
Change name
Change decision attribute
Join values of attribute
Duplicate
Remove
Split in Two
Select subtable
Complete
Discretize
Linear combinations
Reducts/Rules
Make decomposition
Create LTF-C
Create MTD-C
Classify
Statistics
Positive region
Calculate reducts or rules
Calculate dynamic reducts
Design
History


Rough Set Exploration System 2.2.2 [Project1]
File
Insert
Help
Project1
iris
Calculate reducts or rules
Reduct/Rule choice
Reducts
Rules
Discernibility matrix settings
Full discernibility
Modulo decision
Object related discernibility
Method
Exhaustive algorithm
Genetic algorithm
Covering algorithm
LEM2 algorithm
Don't discern with missing values
Genetic algorithm settings
High speed
Normal Speed
Low speed
Number of reducts:
10
Generate to set:
iris
Cover parameter:
0.9
OK
Cancel
Rough Set Exploration System 2.2.2 [Project1]
File
Insert
Help
Project1
Reduct set: iris
(1-4)
Size
Pos.Reg.
SC
Reducts
1 3 1 1 { sepallength, sepalwidth, petallength }
2 3 1 1 { sepallength, petallength, petalwidth }
3 3 1 1 { sepallength, sepalwidth, petalwidth }
4 3 1 1 { sepalwidth, petallength, petalwidth }
iris
iris


Rough Set Exploration System 2.2.2 [Project1]
File Insert Help
Project1
iris
Calculate reducts or rules
Reduct/Rule choice
Reducts
Rules
Method
Exhaustive algorithm
Genetic algorithm
Covering algorithm
LEM2 algorithm
Generate to set:
iris
Discernibility matrix settings
Full discernibility
Modulo decision
Object related discernibility
Don't discern with missing values
Genetic algorithm settings
High speed
Normal Speed
Low speed
Number of reducts:
10
Cover parameter:
0.9
Rough Set Exploration System 2.2.2 [Project1]
File Insert Help
Project1
iris
iris
iris

Rule set: iris

(1-246)	Match	Decision rules
1	28	(petalwidth=0.2)=>(class={Iris-setosa[28]})
2	14	(petallength=1.5)=>(class={Iris-setosa[14]})
3	13	(petalwidth=1.3)=>(class={Iris-versicolor[13]})
4	12	(petallength=1.4)=>(class={Iris-setosa[12]})
5	8	(petalwidth=2.3)=>(class={Iris-virginica[8]})
6	7	(petallength=1.3)=>(class={Iris-setosa[7]})
7	7	(petalwidth=0.4)=>(class={Iris-setosa[7]})
8	7	(petalwidth=0.3)=>(class={Iris-setosa[7]})
9	7	(petallength=1.6)=>(class={Iris-setosa[7]})
10	7	(petalwidth=1)=>(class={Iris-versicolor[7]})
11	6	(sepalwidth=3.5)=>(class={Iris-setosa[6]})
12	6	(petalwidth=0.1)=>(class={Iris-setosa[6]})
13	6	(petalwidth=2.1)=>(class={Iris-virginica[6]})
14	6	(petallength=5.6)=>(class={Iris-virginica[6]})
15	6	(petalwidth=2)=>(class={Iris-virginica[6]})
16	5	(sepallength=4.8)=>(class={Iris-setosa[5]})
17	5	(petallength=4.7)=>(class={Iris-versicolor[5]})
18	5	(petallength=4.5)&(petalwidth=1.5)=>(class={Iris-versicolor
19	5	(petallength=4)=>(class={Iris-versicolor[5]})
20	5	(petalwidth=1.2)=>(class={Iris-versicolor[5]})
21	5	(petalwidth=1.9)=>(class={Iris-virginica[5]})
22	4	(sepallength=4.6)=>(class={Iris-setosa[4]})
23	4	(petallength=1.7)=>(class={Iris-setosa[4]})
24	4	(petallength=4.2)=>(class={Iris-versicolor[4]})

Advantages of Rough Set approach

- It provides efficient methods, algorithms and tools for finding hidden patterns in data.
- It allows to reduce original data, i.e. to find minimal sets of data with the same knowledge as in the original data.
- It allows to generate in automatic way the sets of decision rules from data.
- It is easy to understand.