Rough Sets Tutorial

Contents

Introduction

- **Rough set theory** was developed by Zdzislaw Pawlak in the early 1980's.
- Representative Publications:
  - Z. Pawlak, "Rough Sets", *International Journal of Computer and Information Sciences*, Vol.11, 341-356 (1982).
  - Z. Pawlak, *Rough Sets - Theoretical Aspect of Reasoning about Data*, Kluwer Academic Pubilishers (1991).

Introduction (2)

- The main goal of the rough set analysis is induction of 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.

Introduction (3)

- Recent extensions of rough set theory (**rough mereology**) have developed new methods for decomposition of large data sets, data mining in distributed and multi-agent systems, and granular computing.

This presentation shows how several aspects of the above problems are solved by the (classic) rough set approach, discusses some advanced topics, and gives further research directions.

Basic Concepts of Rough Sets

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

Information Systems/Tables

	Age	LEMS
x 1	16-30	50
x2	16-30	0
x3	31-45	1-25
x4	31-45	1-25
x5	46-60	26-49
x6	16-30	26-49
x7	46-60	26-49

- 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 $a: U \rightarrow V_a$ for every $a \in A$.
- V_a is called the value set of a.

Decision Systems/Tables

	Age	LEMS	Walk
x 1	16-30	50	yes
x2	16-30	0	no
x3	31-45	1-25	no
x4	31-45	1-25	yes
x5	46-60	26-49	no
x6	16-30	26-49	yes
x7	46-60	26-49	no

- 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.

Issues in the Decision Table

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

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 $[x]_B$.

An Example of Indiscernibility

	Age	LEMS	Walk
x 1	16-30	50	yes
x2	16-30	0	no
x3	31-45	1-25	no
x4	31-45	1-25	yes
x5	46-60	26-49	no
x6	16-30	26-49	yes
x7	46-60	26-49	no

- The non-empty subsets of the condition attributes are *{Age}*, *{LEMS}*, and *{Age, LEMS}*.
- *IND({Age}) = {{x1,x2,x6}, {x3,x4}, {x5,x7}}*
- *IND({LEMS}) = {{x1}, {x2}, {x3,x4}, {x5,x6,x7}}*
- *IND({Age,LEMS}) = {{x1}, {x2}, {x3,x4}, {x5,x7}, {x6}}*.

Observations

- An equivalence relation induces a partitioning of the universe.
- The partitions can be used to build new subsets of the universe.
- Subsets that are most often of interest have the same value of the decision attribute.

It may happen, however, that a concept such as "*Walk*" cannot be defined in a crisp manner.

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
x1	16-30	50	yes
x2	16-30	0	no
x3	31-45	1-25	no
x4	31-45	1-25	yes
x5	46-60	26-49	no
x6	16-30	26-49	yes
x7	46-60	26-49	no

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

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

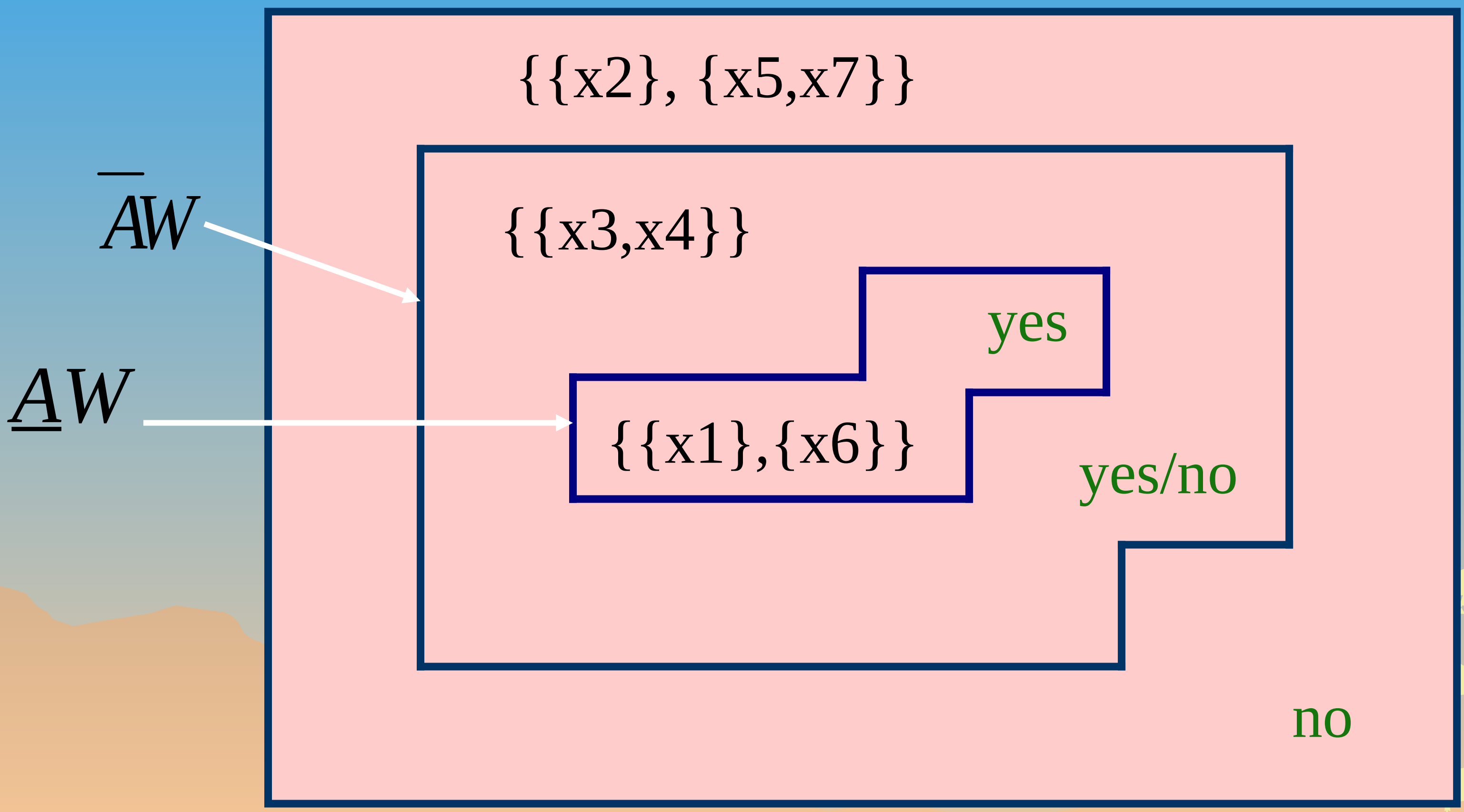
$$\overline{A}W = \{x1, x3, x4, x6\},$$

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

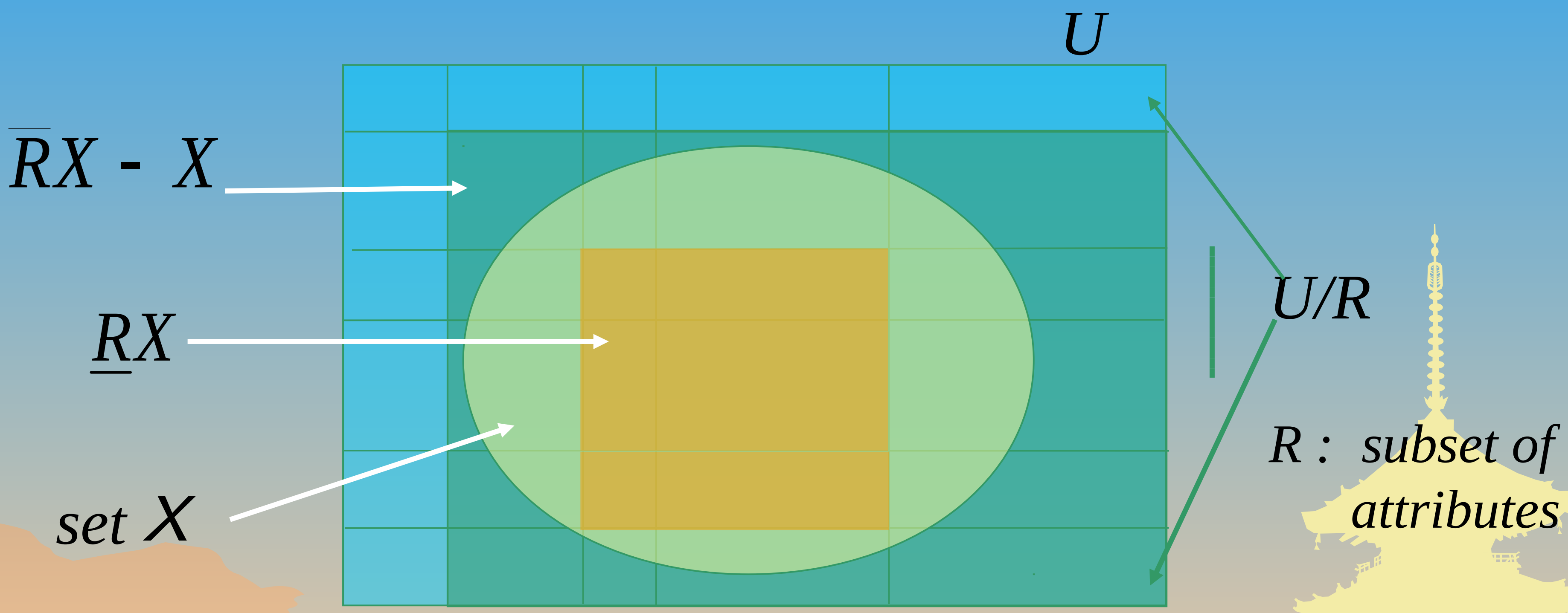
$$U - \overline{A}W = \{x2, x5, x7\}.$$

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

An Example of Set Approximation (2)



Lower & Upper Approximations



Lower & Upper Approximations (2)

Upper Approximation:

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

Lower Approximation:

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

Lower & Upper Approximations (3)

U	*Headache*	*Temp.*	*Flu*
U1	Yes	Normal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	No	Normal	No
U5	*No*	*High*	*No*
U6	***No***	***Very-high***	***Yes***
U7	*No*	*High*	*Yes*
U8	***No***	***Very-high***	***No***

The indiscernibility classes defined by $R = \{Headache, Temp.\}$ are $\{u1\}, \{u2\}, \{u3\}, \{u4\}, \{u5, u7\}, \{u6, u8\}$.

$X1 = \{u \mid \mathrm{Flu}(u) = \mathrm{yes}\}$

$= \{u2, u3, u6, u7\}$

$\underline{R}X1 = \{u2, u3\}$

$\overline{R}X1 = \{u2, u3, u6, u7, \mathbf{u8, u5}\}$

$X2 = \{u \mid \mathrm{Flu}(u) = \mathrm{no}\}$

$= \{u1, u4, u5, u8\}$

$\underline{R}X2 = \{u1, u4\}$

$\overline{R}X2 = \{u1, u4, u5, u8, \mathbf{u7, u6}\}$

Lower & Upper Approximations (4)

R = *{Headache, Temp.}*
U/R = { *{u1}*, *{u2}*, *{u3}*, *{u4}*, *{u5, u7}*, *{u6, u8}*}

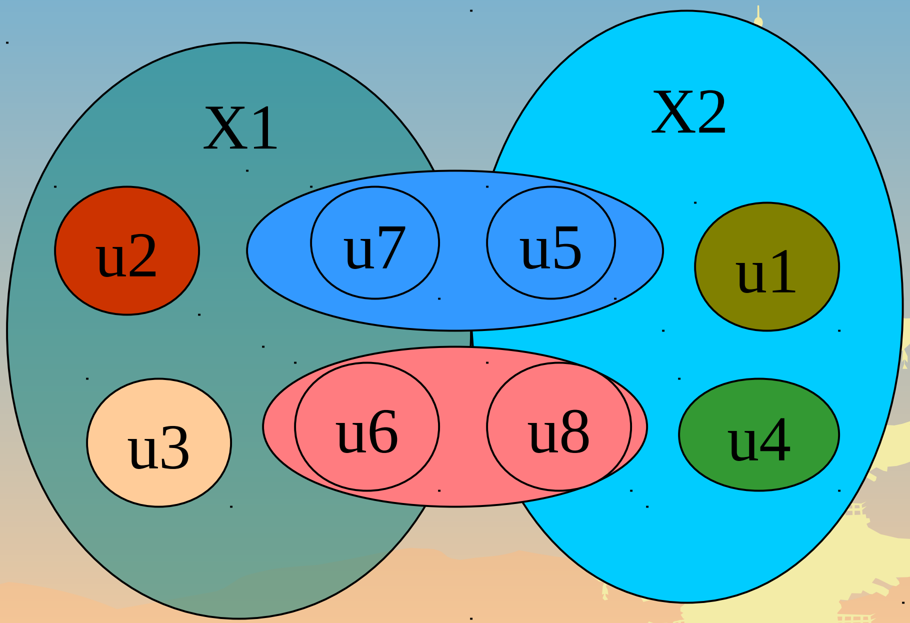
X1 = {u | Flu(u) = yes} = {*u2,u3,u6,u7*}
X2 = {u | Flu(u) = no} = {*u1,u4,u5,u8*}

$\underline{R}X1 = \{u2, u3\}$

$\overline{R}X1 = \{u2, u3, u6, u7, \mathbf{u8, u5}\}$

$\underline{R}X2 = \{u1, u4\}$

$\overline{R}X2 = \{u1, u4, u5, u8, \mathbf{u7, u6}\}$



Properties of Approximations

$$\underline{B}(X) \subseteq X \subseteq \overline{B}X$$

$$\underline{B}(\phi) = \overline{B}(\phi) = \phi, \quad \underline{B}(U) = \overline{B}(U) = U$$

$$\overline{B}(X \cup Y) = \overline{B}(X) \cup \overline{B}(Y)$$

$$\underline{B}(X \cap Y) = \underline{B}(X) \cap \underline{B}(Y)$$

$X \subseteq Y$ implies $\underline{B}(X) \subseteq \underline{B}(Y)$ and $\overline{B}(X) \subseteq \overline{B}(Y)$

Properties of Approximations (2)

$$\underline{B}(X \cup Y) \supseteq \underline{B}(X) \cup \underline{B}(Y)$$

$$\overline{B}(X \cap Y) \subseteq \overline{B}(X) \cap \overline{B}(Y)$$

$$\underline{B}(-X) = -\overline{B}(X)$$

$$\overline{B}(-X) = -\underline{B}(X)$$

$$\underline{B}(\underline{B}(X)) = \overline{B}(\underline{B}(X)) = \underline{B}(X)$$

$$\overline{B}(\overline{B}(X)) = \underline{B}(\overline{B}(X)) = \overline{B}(X)$$

where $-X$ denotes $U - X$.

Four Basic Classes of Rough Sets

- X is *roughly B-definable*, iff $\underline{B}(X) \neq \phi$ and $\overline{B}(X) \neq U$,
- X is *internally B-undefinable*, iff $\underline{B}(X) = \phi$ and $\overline{B}(X) \neq U$,
- X is *externally B-undefinable*, iff $\underline{B}(X) \neq \phi$ and $\overline{B}(X) = U$,
- X is *totally B-undefinable*, iff $\underline{B}(X) = \phi$ and $\overline{B}(X) = U$.

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*.

An Example of Reducts & Core

U	*Headache*	*Muscle pain*	*Temp.*	*Flu*
U1	Yes	Yes	Normal	No
U2	Yes	Yes	High	Yes
U3	Yes	Yes	Very-high	Yes
U4	No	Yes	Normal	No
U5	No	No	High	No
U6	No	Yes	Very-high	Yes

Reduct1 **= {Muscle-pain,Temp.}**

U	*Muscle pain*	*Temp.*	*Flu*
U1,U4	Yes	Normal	No
U2	Yes	High	Yes
U3,U6	Yes	Very-high	Yes
U5	No	High	No

Reduct2 **= {Headache, Temp.}**

U	*Headache*	*Temp.*	*Flu*
U1	Yes	Norlmal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	No	Normal	No
U5	No	High	No
U6	No	Very-high	Yes

CORE **= {Headache,Temp}** $\cap$ **{MusclePain, Temp} = {Temp}**

Discernibility Matrix (relative to positive region)

- Let *T = (U, C, D)* be a decision table, with $U = \{u_1, u_2, \ldots, u_n\}$.

By a discernibility matrix of *T*, denoted *M(T)*, we will mean $n \times n$ matrix defined as:

$$m_{ij} = \begin{cases} \{c \in C : c(u_i) \neq c(u_j)\} & \text{if } \exists d \in D\,[d(u_i) \neq d(u_j)] \\ \lambda & \text{if } \forall d \in D\,[d(u_i) = d(u_j)] \end{cases}$$

for i, j = 1,2,…,n such that u_i or u_j belongs to the *C*-positive region of *D*.

- m_{ij} is the set of all the condition attributes that classify objects *ui* and *uj* into different classes.

Discernibility Matrix (relative to positive region) (2)

- The equation is similar but conjunction is taken over all non-empty entries of *M(T)* corresponding to the indices *i*, *j* such that

  u_i or u_j belongs to the *C*-positive region of *D*.
- $m_{ij} = \lambda$ denotes that this case does not need to be considered. Hence it is interpreted as logic truth.
- All disjuncts of minimal disjunctive form of this function define the reducts of *T* (relative to the positive region).

Discernibility Function (relative to objects)

- For any $u_i \in U$,

$$f_T(u_i) = \wedge_j \{\vee m_{ij} : j \neq i, j \in \{1,2,...,n\}\}$$

where (1) $\vee m_{ij}$ is the disjunction of all variables a such that $a \in m_{ij}$, if $m_{ij} \neq \phi$.

(2) $\vee m_{ij} = \perp(false)$, if $m_{ij} = \phi$.

(3) $\vee m_{ij} = t(true)$, if $m_{ij} = \lambda$.

Each logical product in the minimal disjunctive normal form (DNF) defines a reduct of instance u_i.

Examples of Discernibility Matrix

No	a	b	c	d
u1	*a0*	*b1*	*c1*	*y*
u2	*a1*	*b1*	*c0*	*n*
u3	*a0*	*b2*	*c1*	*n*
u4	*a1*	*b1*	*c1*	*y*

In order to discern equivalence classes of the decision attribute *d*, to preserve conditions described by the discernibility matrix for this table

$C = \{a, b, c\}$

$D = \{d\}$

$$(a \vee c) \wedge b \wedge c \wedge (a \vee b)$$

$$= b \wedge c$$

Reduct = $\{b, c\}$

	u1	u2	u3
u2	a,c		
u3	b	λ	
u4	λ	c	a,b

Examples of Discernibility Matrix (2)

	a	b	c	d	E
$u1$	1	0	2	1	1
$u2$	1	0	2	0	1
$u3$	1	2	0	0	2
$u4$	1	2	2	1	0
$u5$	2	1	0	0	2
$u6$	2	1	1	0	2
$u7$	2	1	2	1	1

	u1	u2	u3	u4	u5	u6
u2	λ					
u3	b,c,d	b,c				
u4	b	b,d	c,d			
u5	a,b,c,d	a,b,c	λ	a,b,c,d		
u6	a,b,c,d	a,b,c	λ	a,b,c,d	λ	
u7	λ	λ	a,b,c,d	a,b	c,d	c,d

Core = {b}

Reduct1 = {b,c}

Reduct2 = {b,d}

Rough Membership

- The rough membership function quantifies the degree of relative overlap between the set *X* and the equivalence class $[x]_B$ to which *x* belongs.

$$\mu_X^B : U \to [0,1] \qquad \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}$$

- The rough membership function can be interpreted as a frequency-based estimate of $P(x \in X \mid u)$, where *u* is the equivalence class of *IND(B)*.

Rough Membership (2)

- The formulae for the lower and upper approximations can be generalized to some arbitrary level of precision $\pi \in (0.5, 1]$ by means of the rough membership function

$$\underline{B}_{\pi} X = \{x \mid \mu_X^B(x) \geq \pi\}$$

$$\overline{B}_{\pi} X = \{x \mid \mu_X^B(x) > 1 - \pi\}.$$

- Note: the lower and upper approximations as originally formulated are obtained as a special case with

$$\pi = 1.$$

Dependency of Attributes

- Discovering dependencies between attributes is an important issue in KDD.
- Set of attribute D depends totally on a set of attributes C, denoted $C \Rightarrow D$, if all values of attributes from D are uniquely determined by values of attributes from C.

Dependency of Attributes (2)

- Let D and C be subsets of A. We will say that D depends on C in a degree k $(0 \leq k \leq 1)$,

denoted by $C \Rightarrow_k D$, if

$$k = \gamma(C, D) = \frac{|POS_C(D)|}{|U|}$$

where $POS_C(D) = \bigcup_{X \in U/D} \underline{C}(X),$ called C-positive region of D.

Dependency of Attributes (3)

- Obviously

$$k = \gamma(C, D) = \sum_{X \in U/D} \frac{|\underline{C}(X)|}{|U|}.$$

- If $k = 1$ we say that D depends totally on C.
- If $k < 1$ we say that D depends partially (in a degree k) on C.

A Rough Set Based KDD Process

- Discretization based on RS and Boolean Reasoning (RSBR).
- Attribute selection based RS with Heuristics (RSH).
- Rule discovery by GDT-RS.

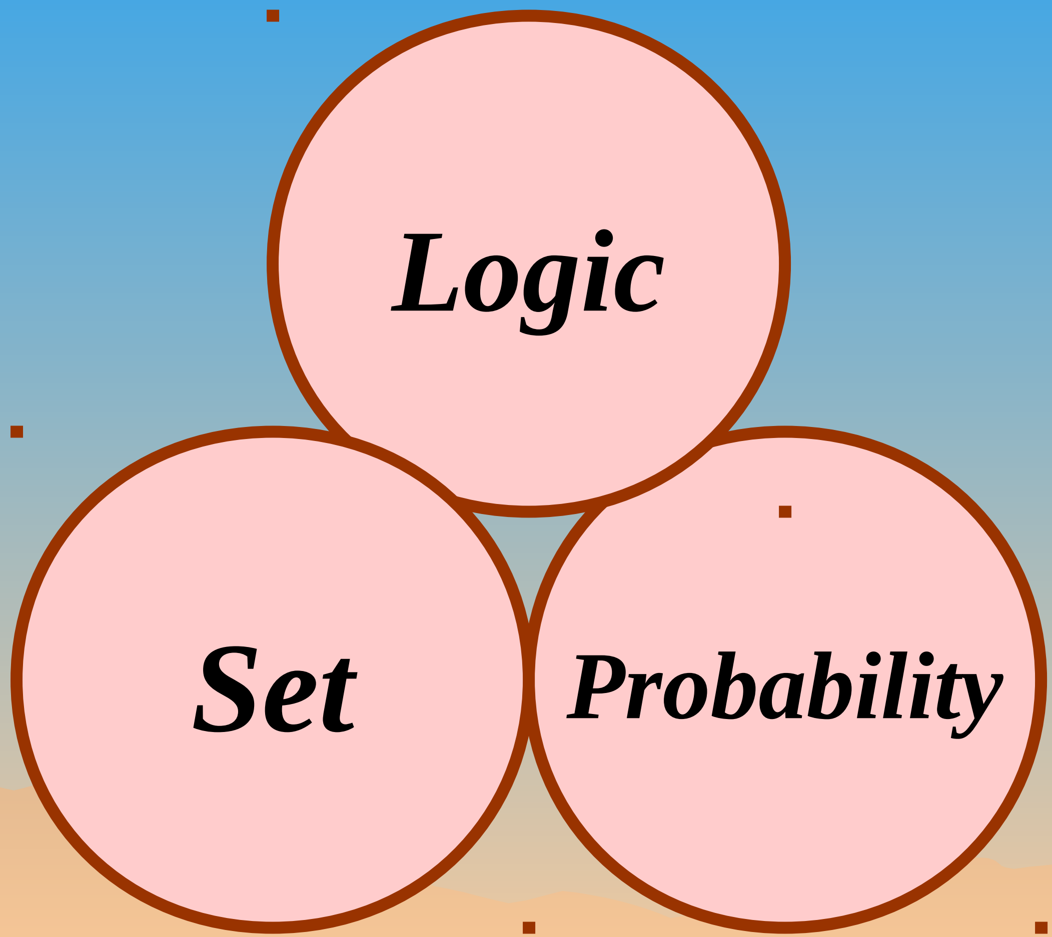
What Are Issues of Real World ?

- Very large data sets
- Mixed types of data (continuous valued, symbolic data)
- Uncertainty (noisy data)
- Incompleteness (missing, incomplete data)
- Data change
- Use of background knowledge

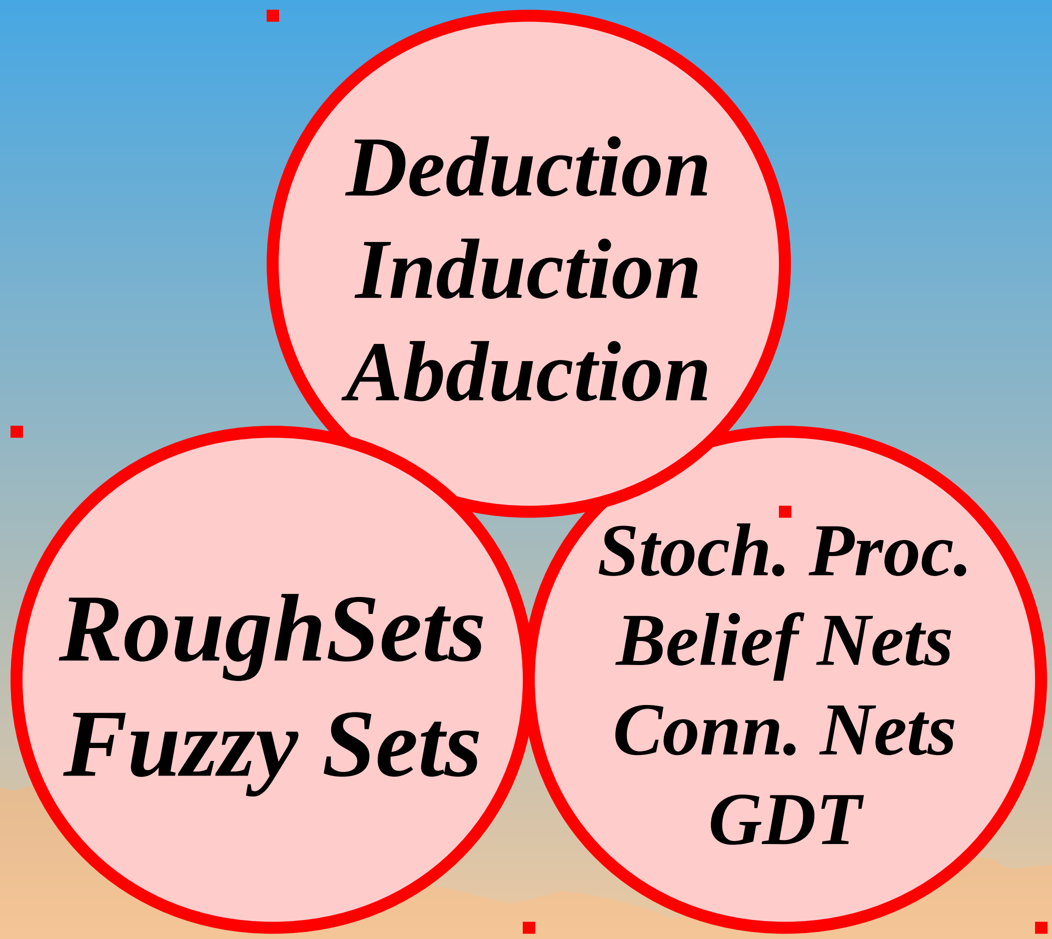
Real world issues \ Methods	ID3 (C4.5)	Prism	Version Space	BP	Dblearn
very large data set	▲	▲			▲
mixed types of data	●			▲	●
noisy data	●			●	
incomplete instances					
data change			●	●	
use of background knowledge			▲		●

● Okay ▲ possible

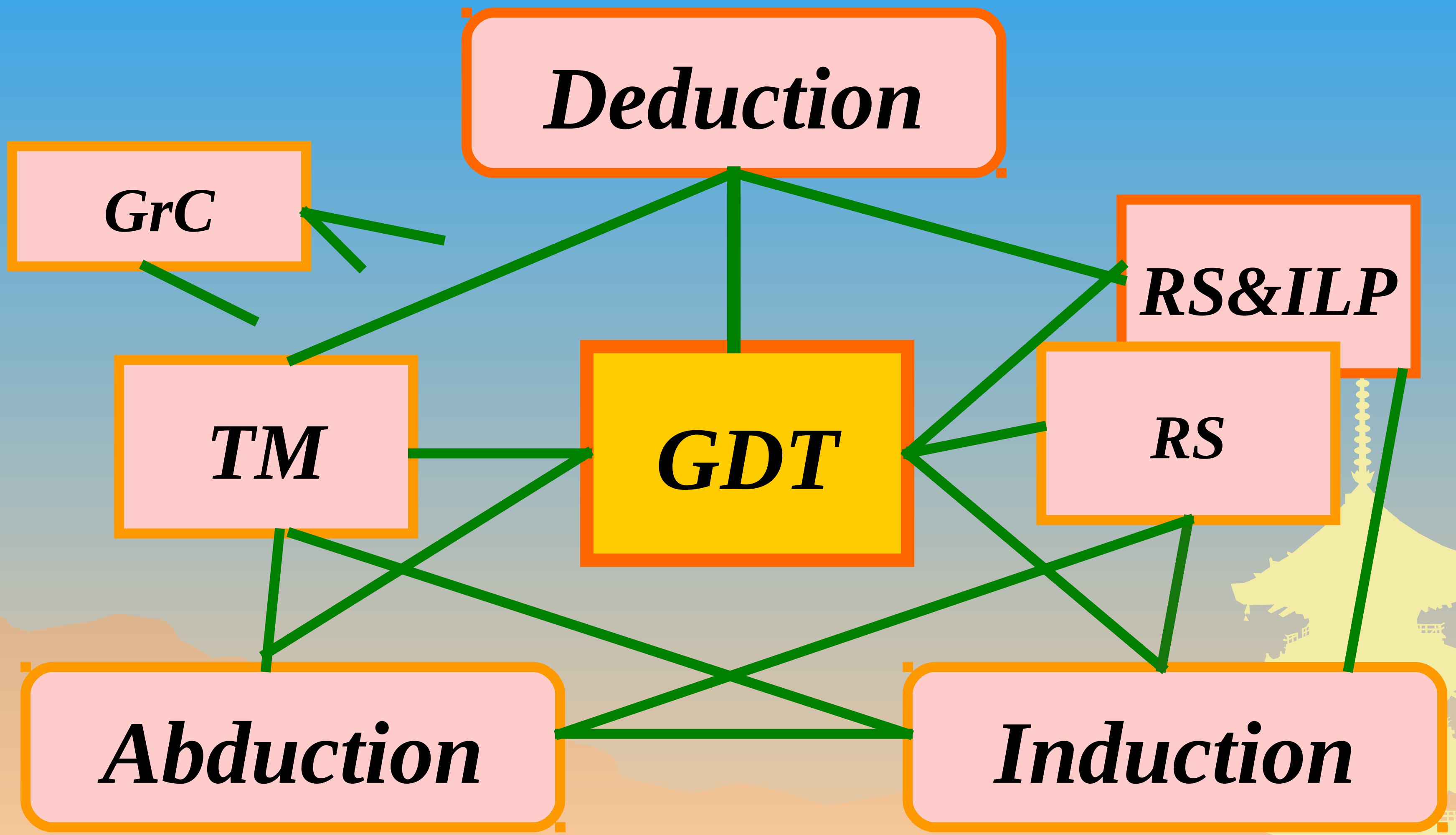
Soft Techniques for KDD



Soft Techniques for KDD (2)



A Hybrid Model



GDT : Generalization Distribution Table

RS : Rough Sets

TM: Transition Matrix

ILP : Inductive Logic Programming

GrC : Granular Computing

A Rough Set Based KDD Process

- ***Discretization based on RS and Boolean Reasoning (RSBR).***
- Attribute selection based RS with Heuristics (RSH).
- Rule discovery by GDT-RS.

Observations

- A real world data set always contains mixed types of data such as continuous valued, symbolic data, etc.
- When it comes to analyze attributes with real values, they must undergo a process called discretization, which divides the attribute’s value into intervals.
- There is a lack of the unified approach to discretization problems so far, and the choice of method depends heavily on data considered.

Discretization based on RSBR

- In the discretization of a decision table $T = (U, A \cup \{d\})$, where $V_a = [v_a, w_a)$ is an interval of real values, we search for a partition P_a of V_a for any $a \in A$.
- Any partition of V_a is defined by a sequence of the so-called *cuts* $v_1 < v_2 < \ldots < v_k$ from V_a.
- Any family of partitions $\{P_a\}_{a \in A}$ can be identified with a set of cuts.

Discretization Based on RSBR (2)

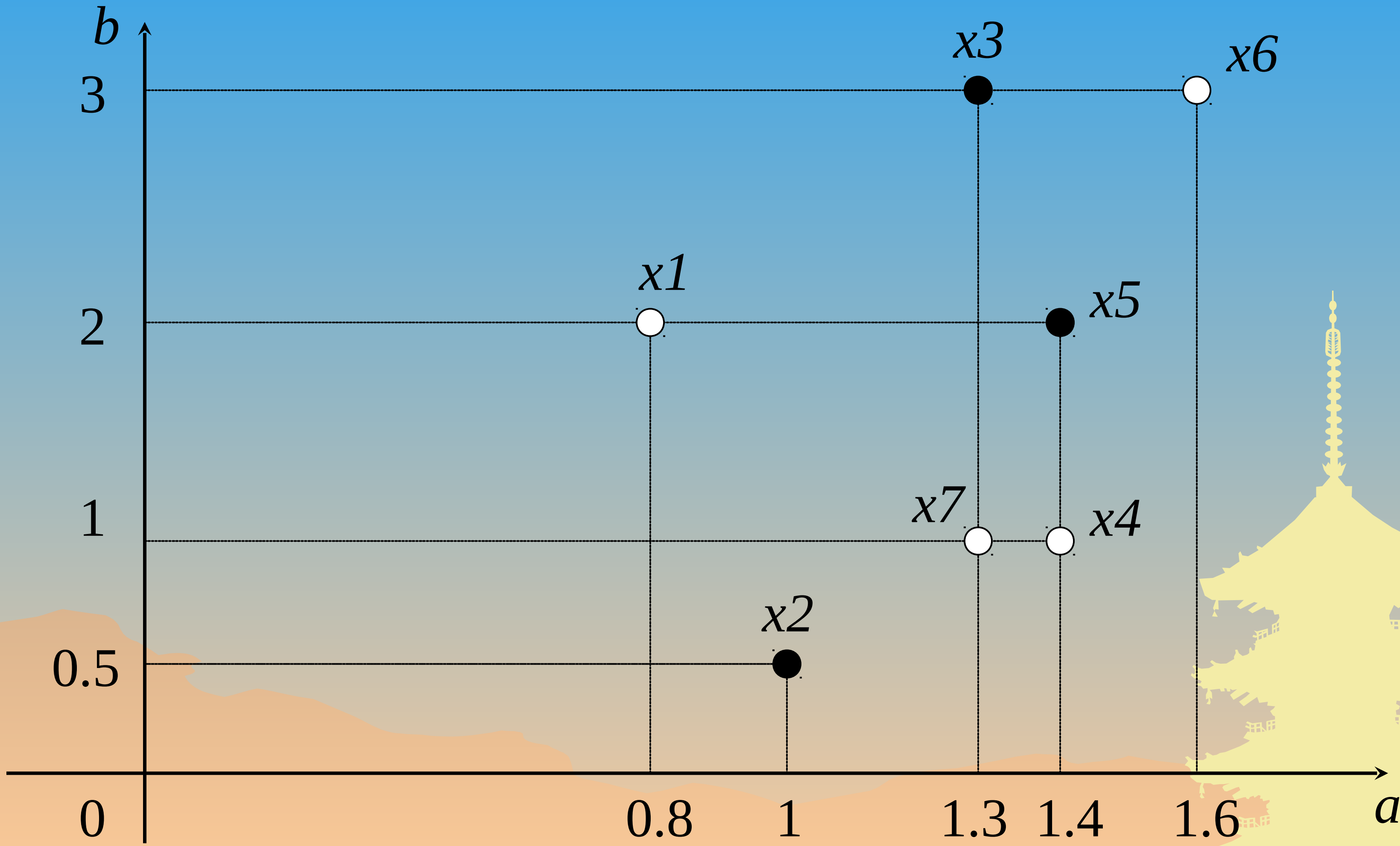
In the discretization process, we search for a set of cuts satisfying some natural conditions.

U	a	b	d
x1	0.8	2	1
x2	1	0.5	0
x3	1.3	3	0
x4	1.4	1	1
x5	1.4	2	0
x6	1.6	3	1
x7	1.3	1	1

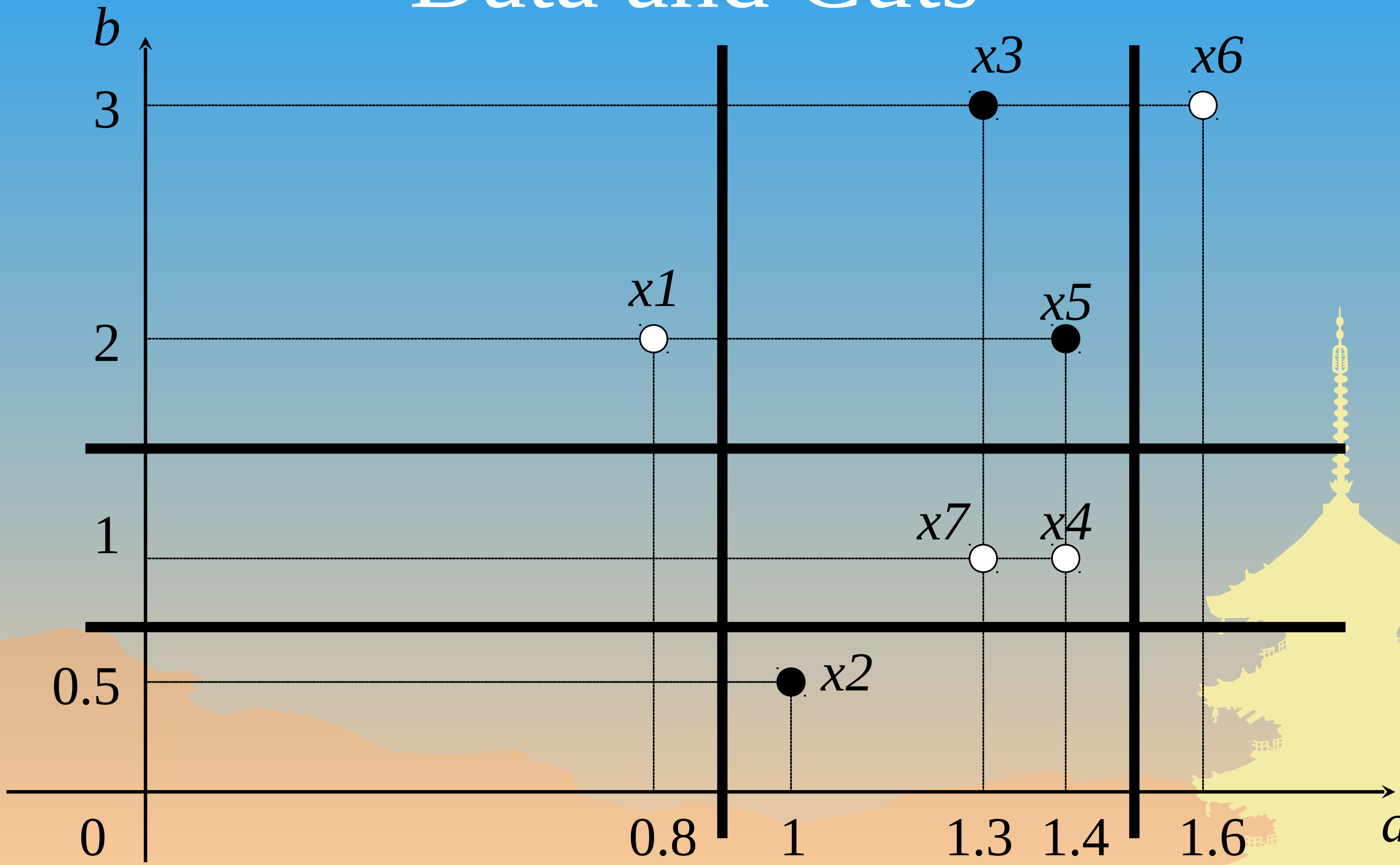
$P = \{(a, 0.9),$
$(a, 1.5),$
$(b, 0.75),$
$(b, 1.5)\}$

U	a^P	b^P	d
x1	0	2	1
x2	1	0	0
x3	1	2	0
x4	1	1	1
x5	1	2	0
x6	2	2	1
x7	1	1	1

A Geometrical Representation of Data



A Geometrical Representation of Data and Cuts



Discretization Based on RSBR (3)

- The sets of possible values of *a* and *b* are defined by

$$V_a = [0, 2); \qquad V_b = [0, 4).$$

- The sets of values of *a* and *b* on objects from *U* are given by

$$a(U) = \{0.8, 1, 1.3, 1.4, 1.6\};$$

$$b(U) = \{0.5, 1, 2, 3\}.$$

Discretization Based on RSBR (4)

- The discretization process returns a partition of the value sets of condition attributes into intervals.

A Discretization Process

- ***Step 1***: define a set of Boolean variables,

$$BV(U) = \{p_1^a, p_2^a, p_3^a, p_4^a, p_1^b, p_2^b, p_3^b\}$$

where

p_1^a corresponds to the interval [0.8, 1) of a

p_2^a corresponds to the interval [1, 1.3) of a

p_3^a corresponds to the interval [1.3, 1.4) of a

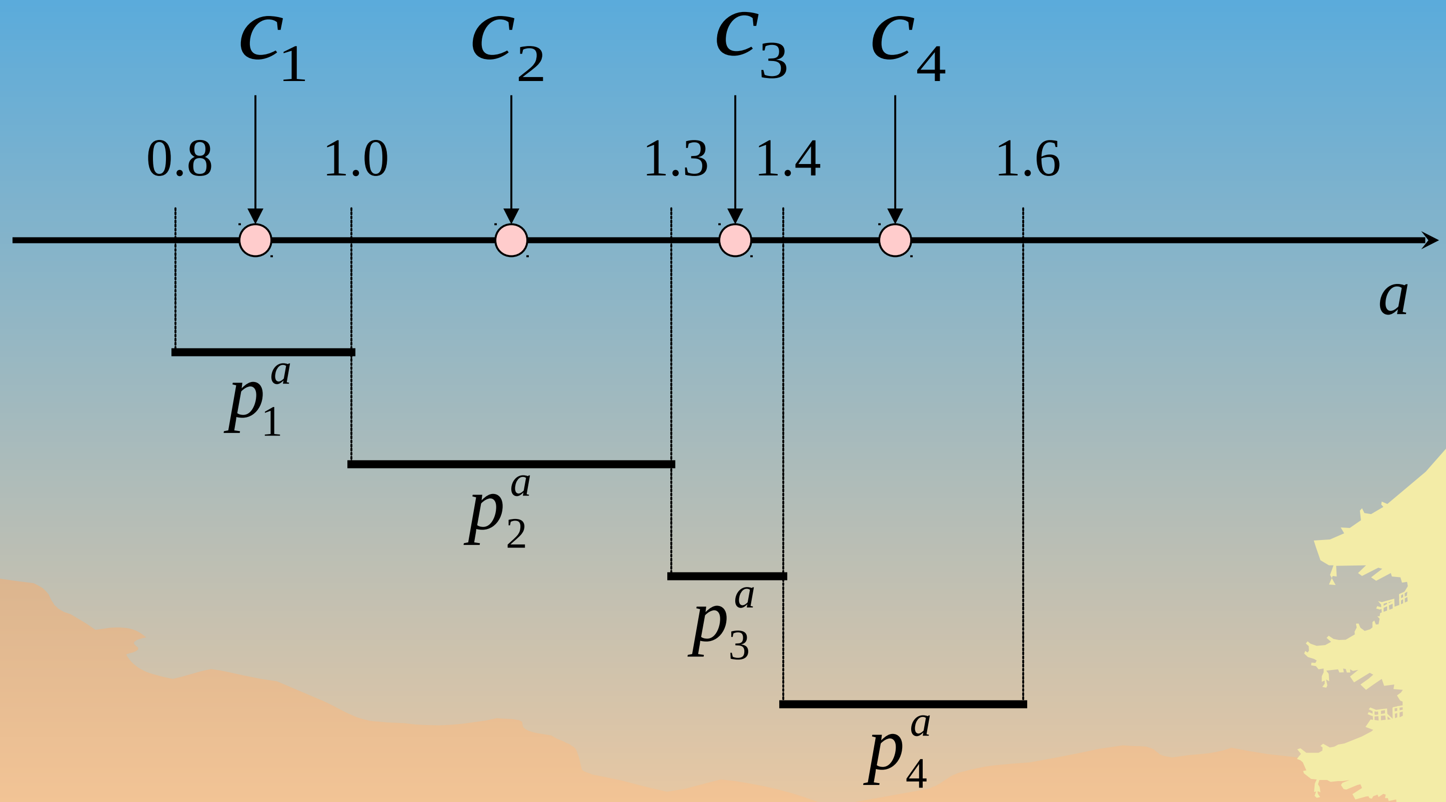
p_4^a corresponds to the interval [1.4, 1.6) of a

p_1^b corresponds to the interval [0.5, 1) of b

p_2^b corresponds to the interval [1, 2) of b

p_3^b corresponds to the interval [2, 3) of b

The Set of Cuts on Attribute a



A Discretization Process (2)

- ***Step 2:*** create a new decision table by using the set of Boolean variables defined in *Step 1*.

  Let $T^P = (U, A \cup \{d\})$ be a decision table, p_k^a be a propositional variable corresponding to the interval $[v_k^a, v_{k+1}^a)$ for any $k \in \{1, \dots, n_a - 1\}$ and $a \in A$.

A Sample T^P Defined in *Step 2*

U^*	p_1^a	p_2^a	p_3^a	p_4^a	p_1^b	p_2^b	p_3^b
(x1,x2)	1	0	0	0	1	1	0
(x1,x3)	1	1	0	0	0	0	1
(x1,x5)	1	1	1	0	0	0	0
(x4,x2)	0	1	1	0	1	0	0
(x4,x3)	0	0	1	0	0	1	1
(x4,x5)	0	0	0	0	0	1	0
(x6,x2)	0	1	1	1	1	1	1
(x6,x3)	0	0	1	1	0	0	0
(x6,x5)	0	0	0	1	0	0	1
(x7,x2)	0	1	0	0	1	0	0
(x7,x3)	0	0	0	0	0	1	0
(x7,x5)	0	0	1	0	0	1	0

The Discernibility Formula

- The discernibility formula

$$\psi(x_1, x_2) = p_1^a \vee p_1^b \vee p_2^b$$

means that in order to discern object *x1* and *x2*, at least one of the following cuts must be set,

a cut between *a*(0.8) and *a*(1)
a cut between *b*(0.5) and *b*(1)
a cut between *b*(1) and *b*(2).

The Discernibility Formulae for All Different Pairs

$$\psi(x_1, x_2) = p_1^a \vee p_1^b \vee p_2^b$$
$$\psi(x_1, x_3) = p_1^a \vee p_2^a \vee p_3^b$$
$$\psi(x_1, x_5) = p_1^a \vee p_2^a \vee p_3^a$$
$$\psi(x_4, x_2) = p_2^a \vee p_3^a \vee p_1^b$$
$$\psi(x_4, x_3) = p_2^a \vee p_2^b \vee p_3^b$$
$$\psi(x_4, x_5) = p_2^b$$

The Discernibility Formulae for All Different Pairs (2)

$$\psi(x_6, x_2) = p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b$$
$$\psi(x_6, x_3) = p_3^a \vee p_4^a$$
$$\psi(x_6, x_5) = p_4^a \vee p_3^b$$
$$\psi(x_7, x_2) = p_2^a \vee p_1^b$$
$$\psi(x_7, x_3) = p_2^b \vee p_3^b$$
$$\psi(x_7, x_5) = p_3^a \vee p_2^b$$

A Discretization Process (3)

- ***Step 3:*** find the minimal subset of *p* that discerns all objects in different decision classes.

  The discernibility boolean propositional formula is defined as follows,

$$\Phi^U = \wedge \{\psi(i.j) : d(x_i) \neq d(x_j)\}.$$

The Discernibility Formula in CNF Form

$$
\begin{aligned}
\Phi^U = {} & (p_1^a \vee p_1^b \vee p_2^b) \wedge (p_1^a \vee p_2^a \vee p_3^b) \\
& \wedge (p_1^a \vee p_2^a \vee p_3^a) \\
& \wedge (p_2^a \vee p_3^a \vee p_1^b) \wedge (p_2^a \vee p_2^b \vee p_3^b) \\
& \wedge (p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b) \\
& \wedge (p_3^a \vee p_4^a) \wedge (p_4^a \vee p_3^b) \wedge (p_2^a \vee p_1^b) \\
& \wedge (p_2^b \vee p_3^b) \wedge (p_3^a \vee p_2^b) \wedge p_2^b.
\end{aligned}
$$

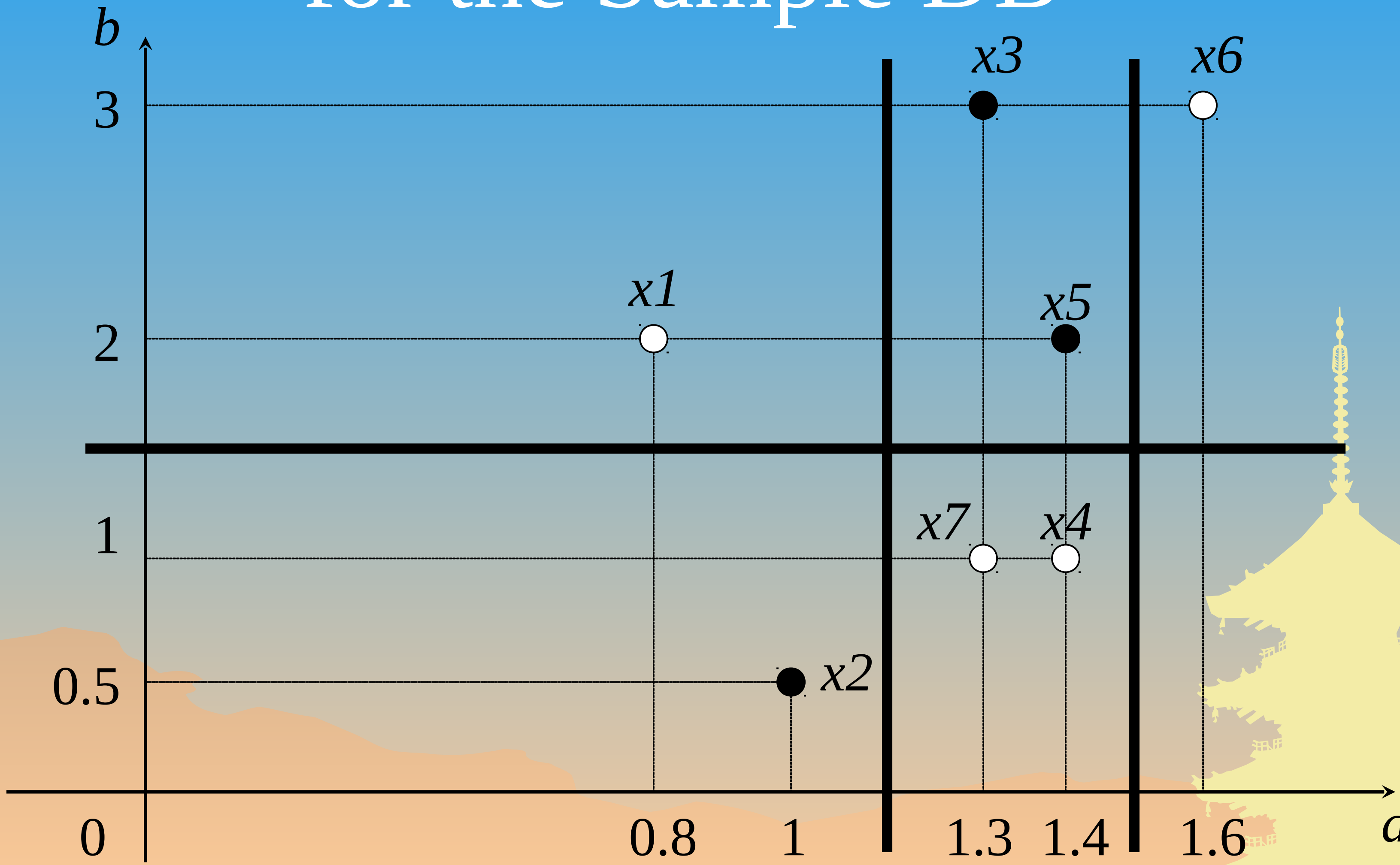
The Discernibility Formula in DNF Form

- We obtain four prime implicants,

$$\Phi^U = (\underline{p_2^a \wedge p_4^a \wedge p_2^b}) \vee (p_2^a \wedge p_3^a \wedge p_2^b \wedge p_3^b)$$
$$\vee (p_3^a \wedge p_1^b \wedge p_2^b \wedge p_3^b) \vee (p_1^a \wedge p_4^a \wedge p_1^b \wedge p_2^b).$$

$\underline{\{p_2^a, p_4^a, p_2^b\}}$ is the optimal result, because it is the minimal subset of P.

The Minimal Set Cuts for the Sample DB



A Result

U	a	b	d
x1	0.8	2	1
x2	1	0.5	0
x3	1.3	3	0
x4	1.4	1	1
x5	1.4	2	0
x6	1.6	3	1
x7	1.3	1	1

P = {(a, 1.2),
(a, 1.5),
(b, 1.5)}

U	a^P	b^P	d
x1	0	1	1
x2	0	0	0
x3	1	1	0
x4	1	0	1
x5	1	1	0
x6	2	1	1
x7	1	0	1

A Rough Set Based KDD Process

- Discretization based on RS and Boolean Reasoning (RSBR).
- ***Attribute selection based RS with Heuristics (RSH).***
- Rule discovery by GDT-RS.

Observations

- A database always contains a lot of attributes that are redundant and not necessary for rule discovery.
- If these redundant attributes are not removed, not only the time complexity of rule discovery increases, but also the quality of the discovered rules may be significantly depleted.

The Goal of Attribute Selection

Finding an optimal subset of attributes in a database according to some criterion, so that a classifier with the highest possible accuracy can be induced by learning algorithm using information about data available only from the subset of attributes.

Attribute Selection

U	*Headache*	*Muscle-pain*	*Temp.*	*Flu*
U1	Yes	Yes	Normal	No
U2	Yes	Yes	High	Yes
U3	Yes	Yes	Very-high	Yes
U4	No	Yes	Normal	No
U5	No	No	High	No
U6	No	Yes	Very-high	Yes

U	*Muscle-pain*	*Temp.*	*Flu*
U1	Yes	Normal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	Yes	Normal	No
U5	No	High	No
U6	Yes	Very-high	Yes

U	*Headache*	*Temp.*	*Flu*
U1	Yes	Normal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	No	Normal	No
U5	No	High	No
U6	No	Very-high	Yes

The Filter Approach

- Preprocessing
- The main strategies of attribute selection:
  - The minimal subset of attributes
  - Selection of the attributes with a higher rank
- Advantage
  - Fast
- Disadvantage
  - Ignoring the performance effects of the induction algorithm

The Wrapper Approach

- Using the induction algorithm as a part of the search evaluation function
- Possible attribute subsets 2^{N-1} (N-number of attributes)
- The main search methods:
  - Exhaustive/Complete search
  - Heuristic search
  - Non-deterministic search
- Advantage
  - Taking into account the performance of the induction algorithm
- Disadvantage
  - The time complexity is high

Basic Ideas:
Attribute Selection using RSH

- Take the attributes in *CORE* as the initial subset.
- Select one attribute each time using the rule evaluation criterion in our rule discovery system, GDT-RS.
- Stop when the subset of selected attributes is a *reduct.*

Why Heuristics ?

- The number of possible reducts can be 2^{N-1} where N is the number of attributes.
  Selecting the optimal reduct from all of possible reducts is time-complex and heuristics must be used.

The Rule Selection Criteria in GDT-RS

- Selecting the rules that cover as many instances as possible.
- Selecting the rules that contain as little attributes as possible, if they cover the same number of instances.
- Selecting the rules with larger strengths, if they have same number of condition attributes and cover the same number of instances.

Attribute Evaluation Criteria

- Selecting the attributes that cause the number of consistent instances to increase faster
  - To obtain the subset of attributes as small as possible
- Selecting an attribute that has smaller number of different values
  - To guarantee that the number of instances covered by rules is as large as possible.

Main Features of RSH

- It can select a better subset of attributes quickly and effectively from a large DB.
- The selected attributes do not damage the performance of induction so much.

An Example of Attribute Selection

U	a	b	c	d	e
u_1	1	0	2	1	1
u_2	1	0	2	0	1
u_3	1	2	0	0	2
u_4	1	2	2	1	0
u_5	2	1	0	0	2
u_6	2	1	1	0	2
u_7	2	1	2	1	1

Condition Attributes:

a: $V_a = \{1, 2\}$

b: $V_b = \{0, 1, 2\}$

c: $V_c = \{0, 1, 2\}$

d: $V_d = \{0, 1\}$

Decision Attribute:

e: $V_e = \{0, 1, 2\}$

Searching for *CORE*

Removing attribute *a*

U	*b*	*c*	*d*	e
u_1	0	2	1	1
u_2	0	2	0	1
u_3	2	0	0	2
u_4	2	2	1	0
u_5	1	0	0	2
u_6	1	1	0	2
u_7	1	2	1	1

Removing attribute *a* does not cause inconsistency.

Hence, *a* is not used as *CORE*.

Searching for *CORE* (2)

Removing attribute *b*

U	a	c	d	e
u_1	1	2	1	1
u_2	1	2	0	1
u_3	1	0	0	2
u_4	1	2	1	0
u_5	2	0	0	2
u_6	2	1	0	2
u_7	2	2	1	1

Removing attribute *b* cause inconsistency.

$$u_1 : a_1 c_2 d_1 \rightarrow e_1$$
$$u_4 : a_1 c_2 d_1 \rightarrow e_0$$

Hence, *b* is used as CORE.

Searching for *CORE* (3)

Removing attribute *c*

U	*a*	*b*	*d*	*e*
u_1	1	0	1	1
u_2	1	0	0	1
u_3	1	2	0	2
u_4	1	2	1	0
u_5	2	1	0	2
u_6	2	1	0	2
u_7	2	1	1	1

Removing attribute *c* does not cause inconsistency.

Hence, *c* is not used as *CORE*.

Searching for *CORE* (4)

Removing attribute *d*

U	a	b	c	e
u_1	1	0	2	1
u_2	1	0	2	1
u_3	1	2	0	2
u_4	1	2	2	0
u_5	2	1	0	2
u_6	2	1	1	2
u_7	2	1	2	1

Removing attribute *d* does not cause inconsistency.

Hence, *d* is not used as *CORE*.

Searching for *CORE* (5)

Attribute b is the unique indispensable attribute.

$$CORE(C)=\{b\}$$

Initial subset $R = \{b\}$

$R=\{b\}$

T

U	a	*b*	*c*	*d*	*e*
u_1	1	0	2	1	1
u_2	1	0	2	0	1
u_3	1	2	0	0	2
u_4	1	2	2	1	0
u_5	2	1	0	0	2
u_6	2	1	1	0	2
u_7	2	1	2	1	1

T'

U'	*b*	*e*
u_1	0	1
u_2	0	1
u_3	2	2
u_4	2	0
u_5	1	2
u_6	1	2
u_7	1	1

$\because b_0 \rightarrow e_1$

The instances containing *b0* will not be considered.

Attribute Evaluation Criteria

- Selecting the attributes that cause the number of consistent instances to increase faster
  - To obtain the subset of attributes as small as possible
- Selecting the attribute that has smaller number of different values
  - To guarantee that the number of instances covered by a rule is as large as possible.

Selecting Attribute from $\{a,c,d\}$

1. Selecting $\{a\}$

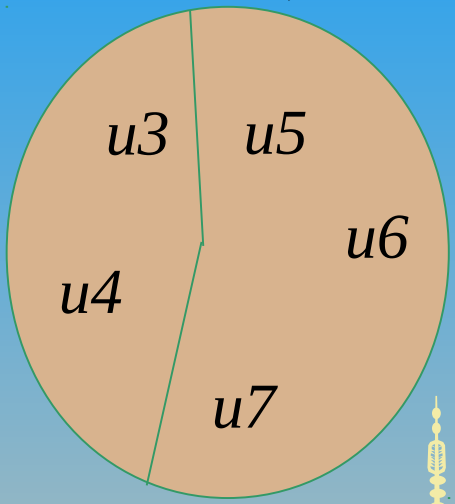
$R = \{a,b\}$

U'	a	b	e
u_3	1	2	2
u_4	1	2	0
u_5	2	1	2
u_6	2	1	2
u_7	2	1	1

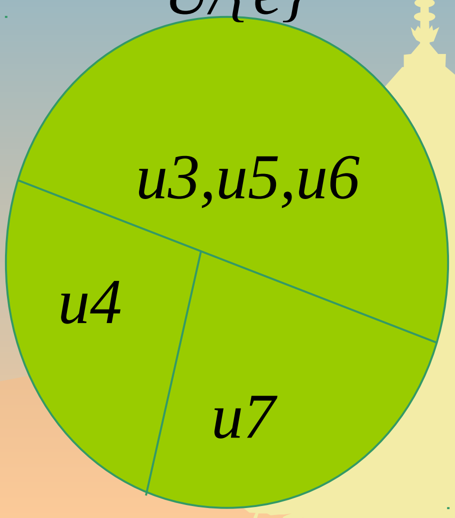
$$a1b2 \rightarrow e2$$
$$a1b2 \rightarrow e0$$

$$a2b1 \rightarrow e2$$
$$a2b1 \rightarrow e1$$

$U/\{a,b\}$



$U/\{e\}$



$$\bigcup_{X \in U/\{e\}} POS_{\{a,b\}}(X) = \phi$$

Selecting Attribute from {*a*,*c*,*d*} (2)

2. Selecting {*c*}

$R = \{b,c\}$

U'	b	c	e
$u3$	2	0	2
$u4$	2	2	0
$u5$	1	0	2
$u6$	1	1	2
$u7$	1	2	1

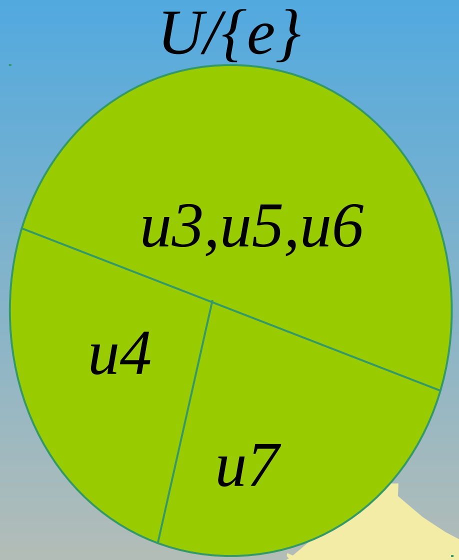
$$b_2c_0 \longrightarrow e_2$$

$$b_2c_2 \longrightarrow e_0$$

$$b_1c_0 \longrightarrow e_2$$

$$b_1c_1 \longrightarrow e_2$$

$$b_1c_2 \longrightarrow e_1$$



$$\bigcup_{X \in U/\{e\}} POS_{\{b,c\}}(X) = \{u3, u4, u5, u6, u7\};$$

Selecting Attribute from $\{a,c,d\}$ (3)

3. Selecting $\{d\}$

$$R = \{b,d\}$$

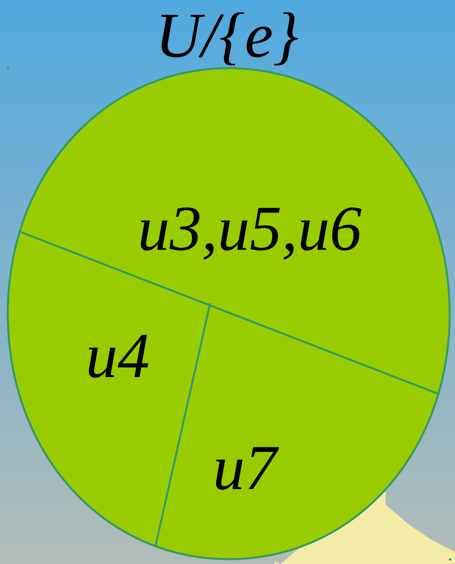
U'	b	d	e
$u3$	2	0	2
$u4$	2	1	0
$u5$	1	0	2
$u6$	1	0	2
$u7$	1	1	1

$$b_2 d_0 \to e_2$$

$$b_2 d_1 \to e_0$$

$$\boxed{b_1 d_0 \to e_2}$$

$$b_1 d_1 \to e_1$$

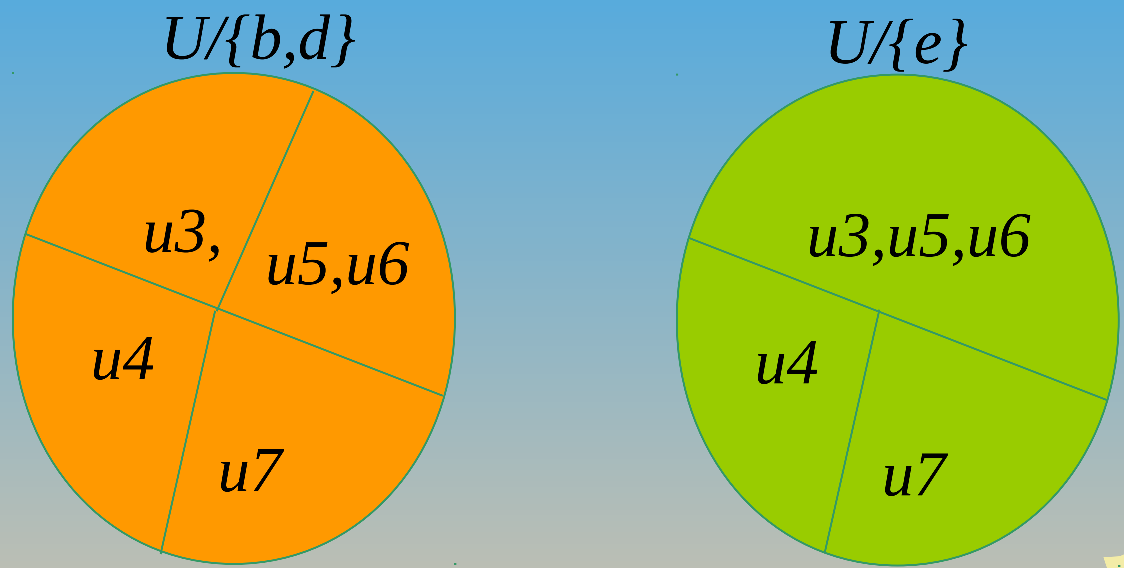


$$\bigcup_{X \in U/\{e\}} POS_{\{b,d\}}(X) = \{u3, u4, u5, u6, u7\};$$

Selecting Attribute from *{a,c,d}* (4)

3. Selecting *{d}*

$R = \{b,d\}$



$$POS_{\{b,d\}}(\{u3,u5,u6\})/\{b,d\} = \{\{u3\},\underline{\{u5,u6\}}\}$$

$$\max\_size(POS_{\{b,d\}}(\{u3,u5,u6\})/\{b,d\}) = 2$$

Result: Subset of attributes ＝ {b, d}

A Heuristic Algorithm for Attribute Selection

- Let *R* be a set of the selected attributes, *P* be the set of unselected condition attributes, *U* be the set of all instances, *X* be the set of contradictory instances, and *EXPECT* be the threshold of accuracy.
- In the initial state, $R = CORE(C)$,

$$P = C - CORE(C),\ \ X = U - POS_R(D)$$

$k = 0.$

A Heuristic Algorithm for Attribute Selection (2)

- ***Step 1.*** If $k >= EXPECT$, finish, otherwise calculate the *dependency degree*, k,

$$k = \frac{|POS_R(D)|}{|U|}.$$

- ***Step 2.*** For each p in P, calculate

$$v_p = |POS_{(R \cup \{p\})}(D)|$$

$$m_p = \max\_size(POS_{(R \cup \{p\})}(D) / (R \cup \{p\} \cup D))$$

where *max_size* denotes the cardinality of the maximal subset.

A Heuristic Algorithm for Attribute Selection (3)

- ***Step 3.*** Choose the best attribute *p* with the largest $v_p * m_p$, and let

$$R = R \cup \{p\}$$

$$P = P - \{p\}.$$

- ***Step 4.*** Remove all consistent instances *u* in $POS_R(D)$ from *X*.
- ***Step 5.*** Go back to *Step 1*.

Experimental Results

Data sets	Attribute Number	Instance Number	Attri. N. In Core	Selected Attri. N.
Monk1	6	124	3	3
Monk3	6	122	4	4
Mushroom	22	8124	0	4
Breast cancer	10	699	1	4
Earthquake	16	155	0	3
Meningitis	30	140	1	4
Bacterial examination	57	20920	2	9
Slope-collapse	23	3436	6	8
Gastric cancer	38	7520	2	19

A Rough Set Based KDD Process

- Discretization based on RS and Boolean Reasoning (RSBR).
- Attribute selection based RS with Heuristics (RSH).
- ***Rule discovery by GDT-RS.***

Main Features of GDT-RS

- Unseen instances are considered in the discovery process, and the uncertainty of a rule, including its ability to predict possible instances, can be explicitly represented in the strength of the rule.
- Biases can be flexibly selected for search control, and background knowledge can be used as a bias to control the creation of a GDT and the discovery process.

A Sample DB

U	**a**	**b**	**c**	**d**
u1	a0	b0	c1	y
u2	a0	b1	c1	y
u3	a0	b0	c1	y
u4	a1	b1	c0	n
u5	a0	b0	c1	n
u6	a0	b2	c1	y
u7	a1	b1	c1	y

Condition attributes： ***a, b, c***

Va = *{a0, a1}* *V**b*** = *{b0, b1, b2}* *V**c*** = *{c0, c1}*

Decision attribute： ***d,*** ***Vd*** = *{y , n}*

A Sample GDT

F(x) / *G(x)*	a0b0c0	a0b0c1	… …	a1b0c0	…...	a1b2c1
***b0c0**	**1/2**		**……**	**1/2**	**……**	
***b0c1**		**1/2**			**……**	
***b1c0**					**……**	
***b1c1**					**……**	
***b2c0**					**……**	
***b2c1**					**……**	**1/2**
a0*c0						
a1b1*						
a1b2*						
	1/6			**1/6**	**……**	
	……				**……**	
a0**	**1/6**	**1/6**			**……**	
a1**				**1/6**	**……**	**1/6**

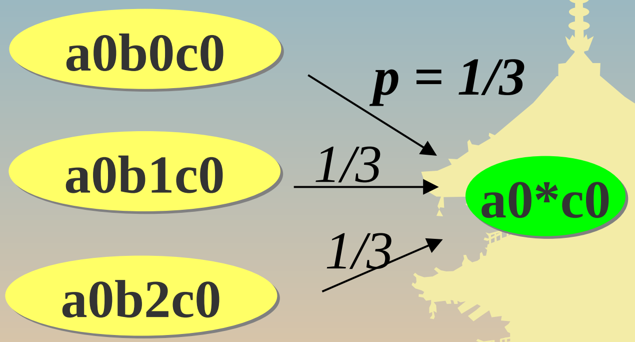
Explanation for GDT

- *F(x):* the possible instances (*PI*)
- *G(x):* the possible generalizations (*PG*)
- $G(x) \rightarrow F(x)$: the probability relationships between *PI* & *PG*.

Probabilistic Relationship Between PIs and PGs

$$p(PI_j \mid PG_i) = \begin{cases} \dfrac{1}{N_{PG_i}} & \text{if } PI_j \in PG_i \\ 0 & \text{otherwise} \end{cases}$$

$$N_{PG_i} = \prod_{k \in \{l \mid PG[l]=*\}} n_k$$



$$N_{\{a0c0\}} = n_b = 3$$

N_{PG_i} is the number of *PI* satisfying the *i*th *PG*.

Unseen Instances

Possible Instances:

U	Headache	Muscle-pain	Temp.	Flu
U1	Yes	Yes	Normal	No
U2	Yes	Yes	High	Yes
U3	Yes	Yes	Very-high	Yes
U4	No	Yes	Normal	No
U5	No	No	High	No
U6	No	Yes	Very-high	Yes

yes , no , normal
yes, no, high
yes, no, very-high
no, yes, high
no, no, normal
no, no, very-high

Closed world → Open world

Rule Representation

$$\boldsymbol{X \longrightarrow Y \textit{ with } S}$$

- *X* denotes the conjunction of the conditions that a concept must satisfy
- *Y* denotes a concept that the rule describes
- *S* is a "measure of strength" of which the rule holds

Rule Strength (1)

$$S(X \to Y) = s(X)(1 - r(X \to Y))$$

- The strength of the generalization X (BK is no used),

$$s(X) = s(PG_k) =$$

$$\sum_l p(PI_l \mid PG_k) = \frac{N_{ins-rel}(PG_k)}{N_{PG_k}}$$

$N_{ins-rel}(PG_k)$ is the number of the observed instances satisfying the ith generalization.

Rule Strength (2)

- The strength of the generalization *X* (BK is used),

$$s(X) = s(PG_k) = \sum_l p_{bk}(PI_l \mid PG_k)$$

$$= \frac{\sum_l BKF(PI_l \mid PG_k)}{N_{PG_k}}$$

Rule Strength (3)

- The rate of noises

$$r(X \rightarrow Y) = \frac{N_{ins-rel}(X) - N_{ins-class}(X,Y)}{N_{ins-rel}(X)}$$

$N_{ins-class}(X,Y)$ is the number of instances belonging to the class *Y* within the instances satisfying the generalization *X*.

Rule Discovery by GDT-RS

U	a	b	c	d
u1	a0	b0	c1	y
u2	a0	b1	c1	y
u3	a0	b0	c1	y
u4	a1	b1	c0	n
u5	a0	b0	c1	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

Condition Attrs.： *a*, *b*, *c*

a: *Va* = {*a0*, *a1*}

b: *Vb* = {*b0*, *b1*, *b2*}

c: *Vc* = {*c0*, *c1*}

Class： *d*:

d: *Vd* = {*y* , *n*}

Regarding the Instances (Noise Rate = 0)

U	a	b	c	d
u1' {u1, u3, u5}	a0	b0	c1	y, y, n
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

U	a	b	c	d
u1'	a0	b0	c1	$\perp$
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$r_{\{y\}}(u1') = 1 - \frac{2}{3} = 0.33$$

$$r_{\{n\}}(u1') = 1 - \frac{1}{3} = 0.67$$

Let $T_{noise} = 0$

$\because$ $r_{\{y\}}(u1') > T_{noise}$ と

$r_{\{n\}}(u1') > T_{noise}$

$\therefore$ $d(u1') = \perp$

Generating Discernibility Vector for *u2*

U	a	b	c	d
u1'	a0	b0	c1	⊥
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$m_{2,1'} = \{b\}$$

$$m_{2,2} = \lambda$$

$$m_{2,4} = \{a, c\}$$

$$m_{2,6} = \{b\}$$

$$m_{2,7} = \lambda$$

	u1'	*u2*	*u4*	*u6*	*u7*
u2	*b*	λ	*a,c*	*b*	λ

Obtaining Reducts for *u2*

U	a	b	c	d
u1’	a0	b0	c1	⊥
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

	u1’	*u2*	*u4*	*u6*	*u7*
u2	*b*	λ	*a,c*	*b*	λ

$$f_T(u2) = (b) \wedge T \wedge (a \vee c) \wedge (b) \wedge T$$

$$= (b) \wedge (a \vee c)$$

$$= \underline{(a \wedge b)} \vee \underline{(b \wedge c)}$$

Generating Rules from *u2*

U	a	b	c	d
u1’	a0	b0	c1	⊥
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$f_T(u2) = \underbrace{(a \wedge b)}_{\{a0,b1\}} \vee \underbrace{(b \wedge c)}_{\{b1,c1\}}$$

$\{a0b1\} \leftarrow a0b1c0$

$\{a0b1\} \xleftarrow{y} a0b1c1(u2)$

s({a0b1}) = 0.5

$r(\{a0b1\} \rightarrow y) = 0$

$\{b1c1\} \xleftarrow{y} a0b1c1(u2)$

$\{b1c1\} \xleftarrow{y} a1b1c1(u7)$

s({b1c1}) = 1

$r(\{b1c1\} \rightarrow y) = 0$

Generating Rules from *u2* (2)

U	a	b	c	d
u1’	a0	b0	c1	⊥
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$\{a0b1\} \rightarrow y \quad \text{with} \quad S = (1 \times \frac{1}{2}) \times (1 - 0) = 0.5$$

$$\{b1c1\} \rightarrow y \quad \text{with} \quad S = (2 \times \frac{1}{2}) \times (1 - 0) = 1$$

Generating Discernibility Vector for *u4*

U	a	b	c	d
u1’	a0	b0	c1	⊥
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$m_{4,1'} = \{a, b, c\}$$

$$m_{4,2} = \{a, c\}$$

$$m_{4,4} = \lambda$$

$$m_{4,6} = \lambda$$

$$m_{4,7} = \{c\}$$

	u1’	*u2*	*u4*	*u6*	*u7*
u4	*a,b,c*	*a,c*	λ	λ	*c*

Obtaining Reducts for *u4*

U	a	b	c	d
u1'	a0	b0	c1	$\perp$
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

	u1'	*u2*	*u4*	*u6*	*u7*
u4	*a,b,c*	*a,c*	λ	λ	*c*

$$f_T(u4) = (a \vee b \vee c) \wedge (a \vee c) \wedge \mathrm{T} \wedge \mathrm{T} \wedge (c)$$

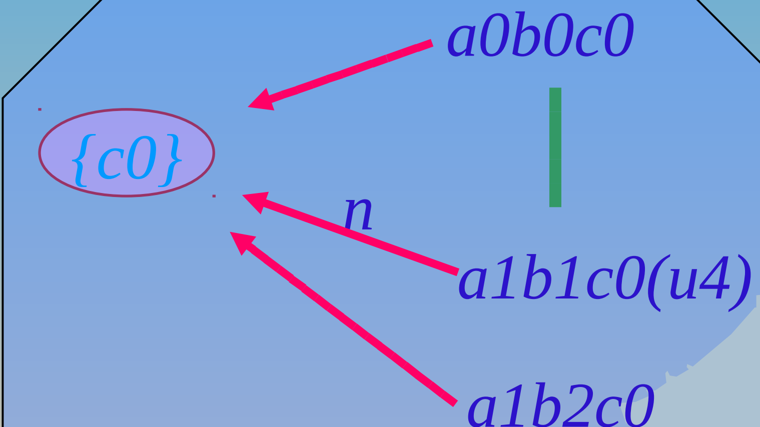
$$= \underline{(c)}$$

Generating Rules from *u4*

U	a	b	c	d
u1’	a0	b0	c1	$\perp$
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$f_T(u4) = (\underline{c})$$

$$\{c0\}$$



$$s(c0) = \frac{1}{6}$$

$$r(\{c0\} \rightarrow n) = 0$$

Generating Rules from *u4* (2)

U	a	b	c	d
u1’	a0	b0	c1	⊥
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$\{c0\} \rightarrow n \quad \text{with} \quad S = (1 \times \frac{1}{6}) \times (1 - 0) = 0.167$$

Generating Rules from All Instances

U	a	b	c	d
u1’	a0	b0	c1	⊥
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

u2: {a0b1} $\longrightarrow$ *y, S = 0.5*

{b1c1} $\longrightarrow$ *y, S =1*

u4: {c0} $\longrightarrow$ *n, S = 0.167*

u6: {b2} $\longrightarrow$ *n, S=0.25*

u7: {a1c1} $\longrightarrow$ *y, S=0.5*

{b1c1} $\longrightarrow$ *y, S=1*

The Rule Selection Criteria in GDT-RS

- Selecting the rules that cover as many instances as possible.
- Selecting the rules that contain as little attributes as possible, if they cover the same number of instances.
- Selecting the rules with larger strengths, if they have same number of condition attributes and cover the same number of instances.

Generalization Belonging to Class *y*

	u2	*u7*
	***a0b1c1* (√)**	***a1b1c1* (√)**
****b1c1***	1/2	1/2
a1*c1		1/3
a0b1*	1/2	

$\{b1c1\} \longrightarrow y$ with $S = 1$ *u2*, *u7*

~~$\{a1c1\} \longrightarrow y$ with $S = 1/2$ *u7*~~

~~$\{a0b1\} \longrightarrow y$ with $S = 1/2$ *u2*~~

Generalization Belonging to Class *n*

	u4 a0b2c1 (n)	***u6*** a1b1c0 (n)
****c0**		1/6
b2	1/4	

c0 $\longrightarrow$ ***n*** with ***S = 1/6*** ***u4***

b2 $\longrightarrow$ ***n*** with ***S = 1/4*** ***u6***

Results from the Sample DB (Noise Rate = 0)

- Certain Rules: Instances Covered

$\{c0\} \longrightarrow n$ with $S = 1/6$	***u4***
$\{b2\} \longrightarrow n$ with $S = 1/4$	***u6***
$\{b1c1\} \longrightarrow y$ with $S = 1$	***u2 , u7***

Results from the Sample DB (2) (Noise Rate > 0)

- Possible Rules:

b0 ⟶ *y with S = (1/4)(1/2)*

<u>a0 & b0 ⟶ y with S = (1/2)(2/3)</u>

a0 & c1 ⟶ *y with S = (1/3)(2/3)*

<u>b0 & c1 ⟶ y with S = (1/2)(2/3)</u>

Instances Covered: u1, u3, u5

Regarding Instances (Noise Rate > 0)

U	a	b	c	d
u1' {u1, u3, u5}	a0	b0	c1	y, y, n
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

U	a	b	c	d
u1'	a0	b0	c1	y
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$r_{\{y\}}(u1') = 1 - \frac{2}{3} = 0.33$$

$$r_{\{n\}}(u1') = 1 - \frac{1}{3} = 0.67$$

Let $T_{noise} = 0.5$

$\because \quad r_{\{y\}}(u1') < T_{noise}$

$\therefore \quad d(u1') = y$

Rules Obtained from All Instacnes

U	a	b	c	d
u1’	a0	b0	c1	y
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

*u1’:{b0}→y, S=1/4*2/3=0.167*

u2: {a0b1} → y, S=0.5
{b1c1} → y, S=1

u4: {c0} → n, S=0.167

u6: {b2} → n, S=0.25

u7: {a1c1} → y, S=0.5
{b1c1} → y, S=1

Example of Using BK

	a0b0c0	*a0b0c1*	*a0b1c0*	*a0b1c1*	*a0b2c0*	*a0b2c1*	*…*	*a1b2c1*
*a0b0**	1/2	1/2						
*a0b1**			1/2	1/2				
*a0*c1*		1/3		1/3		1/3		
*a0***	1/6	1/6	1/6	1/6	1/6	1/6		

BK : a0 => c1, 100%

	a0b0c0	*a0b0c1*	*a0b1c0*	*a0b1c1*	*a0b2c0*	*a0b2c1*	*…*	*a1b2c1*
*a0b0**	0	1						
*a0b1**			0	1				
*a0*c1*		1/3		1/3		1/3		
*a0***	0	1/3	0	1/3	0	1/3		

Changing Strength of Generalization by BK

U	a	b	c	d
u1’	a0	b0	c1	⊥
u2	a0	b1	c1	y
u4	a1	b1	c0	n
u6	a0	b2	c1	n
u7	a1	b1	c1	y

$$f_T(u2) = \underbrace{(a \wedge b)}_{\{a0,b1\}} \vee \underbrace{(b \wedge c)}_{\{b1,c1\}}$$

$\{a0b1\} \xleftarrow{1/2}$ a0b1c0

$\{a0b1\} \xleftarrow{1/2}$ a0b1c1(u2)

s({a0b1}) = 0.5

$r(\{a0b1\} \rightarrow y) = 0$

a0 => c1, 100%

$\{a0b1\} \xleftarrow{0\%}$ a0b1c0

$\{a0b1\} \xleftarrow{100\%}$ a0b1c1(u2)

*s({a0b1}) = **1***

$r(\{a0b1\} \rightarrow y) = 0$

Algorithm 1
Optimal Set of Rules

- ***Step 1.*** Consider the instances with the same condition attribute values as one instance, called a *compound instance*.
- ***Step 2.*** Calculate the rate of noises r for each compound instance.
- ***Step 3.*** Select one instance u from U and create a discernibility vector for u.
- ***Step 4.*** Calculate all reducts for the instance u by using the discernibility function.

Algorithm 1
Optimal Set of Rules (2)

- ***Step 5.*** Acquire the rules from the reducts for the instance *u*, and revise the strength of generalization of each rule.
- ***Step 6.*** Select better rules from the rules (for *u*) acquired in *Step* 5, by using the heuristics for rule selection.
- ***Step 7.*** $U = U - \{u\}$. If ‘$\phi \neq \cap$ then go back to *Step* 3. Otherwise go to *Step* 8.

Algorithm 1
Optimal Set of Rules (3)

- ***Step 8.*** Finish if the number of rules selected in *Step 6* for each instance is 1. Otherwise find a minimal set of rules, which contains all of the instances in the decision table.

The Issue of Algorithm 1

It is not suitable for the database with a large number of attributes.

Methods to Solve the Issue:

- Finding a reduct (subset) of condition attributes in a pre-processing.
- Finding a sub-optimal solution using some efficient heuristics.

Algorithm 2
Sub-Optimal Solution

- ***Step1:*** Set $R = \{\}$, $COVERED = \{\}$, and $SS = \{all\ instances\ IDs\}$.
  For each class D_c, divide the decision table T into two parts: current class T_+ and other classes T_-.
- ***Step2:*** From the attribute values v_{ij} of the instances I_k (where v_{ij} means the *j*th value of attribute *i*, $I_k \in T_+$, $I_k \in SS$),

Algorithm 2
Sub-Optimal Solution (2)

choose a value *v* with the maximal number of occurrence within the instances contained in *T*+ ，and the minimal number of occurrence within the instances contained in *T*-.

- ***Step3:*** Insert *v* into *R*.
- ***Step4:*** Delete the instance ID from *SS* if the instance does not contain *v*.

Algorithm 2
Sub-Optimal Solution (3)

- ***Step5:*** Go back to ***Step2*** until the noise rate is less than the threshold value.
- ***Step6:*** Find out a minimal sub-set *R'* of *R* according to their strengths. Insert $(R' \rightarrow D_c)$ into *RS*. Set *R* = *{}*, copy the instance IDs in *SS* to *COVERED* **,** and set *SS* = {all instance IDs}- *COVERED*.

Algorithm 2
Sub-Optimal Solution (4)

- ***Step8:*** Go back to ***Step2*** until all instances of *T*+ are in *COVERED*.
- ***Step9:*** Go back to ***Step1*** until all classes are handled.

Time Complexity of Alg.1&2

- Time Complexity of Algorithm 1:

$$O(mn^3 + mn^2 N(G_T))$$

- Time Complexity of Algorithm 2:

$$O(m^2 n^2)$$

Let n be the number of instances in a DB,

m the number of attributes,

$N(G_T)$ the number of generalizations and is less than $O(2^{m-1})$.

Experiments

- DBs that have been tested:

  meningitis, bacterial examination, cancer, mushroom, slope-in-collapse, earth-quack, contents-sell, ……

- Experimental methods:
  - Comparing GDT-RS with C4.5
  - Using background knowledge or not
  - Selecting different allowed noise rates as the threshold values
  - Auto-discretization or BK-based discretization.

Experiment 1 (meningitis data)

- C4.5:

$$CellPoly(>220) \rightarrow Bacteria$$

$$CTFind(abnormal) \wedge CellMono(\leq 12) \rightarrow Bacteria$$

$$CellPoly(\leq 220) \wedge CellMono(>12) \rightarrow Virus$$

$$CTFind(normal) \wedge CellPoly(\leq 220) \rightarrow Virus$$

(from a meningitis DB with 140 records, and 38 attributes)

Experiment 1 (meningitis data) (2)

- GDT-RS (auto-discretization):

$$cellPoly(\geq 221) \rightarrow bacteria$$

$$ctFind(abnormal) \wedge cellMono(< 15) \rightarrow bacteria$$

$$sex(m) \wedge CRP(\geq 4) \rightarrow bacteria$$

$$onset(acute) \wedge CCourse(negative) \wedge riskGrounp(p)$$
$$\rightarrow bacteria$$

$$seizure(0) \wedge onset(acute) \wedge riskGrounp(p) \rightarrow bacteria$$

$$culture(strepto) \rightarrow bacteria$$

Experiment 1 (meningitis data) (3)

- GDT-RS (auto-discretization):

$$CRP(<4) \wedge cellPoly(<221) \wedge cellMono(\geq 15) \rightarrow virus$$

$$CRP(<4) \wedge ctFind(normal) \wedge cellPoly(<221) \rightarrow virus$$

$$cellPoly(<221) \wedge cellMono(\geq 15) \wedge risk(n) \rightarrow virus$$

Using Background Knowledge (meningitis data)

- **Never occurring together:**

  EEGwave(normal) $\Longleftrightarrow$ *EEGfocus(+)*

  CSFcell(low) $\Longleftrightarrow$ *Cell_Poly(high)*

  CSFcell(low) $\Longleftrightarrow$ *Cell_Mono(high)*

- **Occurring with lower possibility:**

  WBC(low) $\Longrightarrow$ *CRP(high)*

  WBC(low) $\Longrightarrow$ *ESR(high)*

  WBC(low) $\Longrightarrow$ *CSFcell(high)*

Using Background Knowledge (meningitis data) (2)

- **Occurring with higher possibility:**

WBC(high) $\Longrightarrow$ *CRP(high)*

WBC(high) $\Longrightarrow$ *ESR(high)*

WBC(high) $\Longrightarrow$ *CSF_CELL(high)*

EEGfocus(+) $\Longrightarrow$ *FOCAL(+)*

EEGwave(+) $\Longrightarrow$ *EEGfocus(+)*

CRP(high) $\Longrightarrow$ *CSF_GLU(low)*

CRP(high) $\Longrightarrow$ *CSF_PRO(low)*

Explanation of BK

- If the brain wave (*EEGwave*) is normal, the focus of brain wave (*EEGfocus*) is never abnormal.
- If the number of white blood cells (*WBC)* is high, the inflammation protein (*CRP*) is also high.

Using Background Knowledge (meningitis data) (3)

- *rule1* is generated by BK

rule1:

$$ONSET(acute) \wedge ESR(\leq 5) \wedge CSFcell(>10)$$
$$\wedge\ CULTURE(-) \rightarrow VIRUS(E);$$
$$S = 30(384/E)4$$

Using Background Knowledge (meningitis data) (4)

- ***rule2*** is replaced by ***rule2'***

rule2:

$$DIAG(VIRUS(E)) \wedge LOC[4,7] \rightarrow EEGabnormal;$$

$$S = 30/E.$$

$$\Uparrow$$

rule2':

$$EEGfocus(+) \wedge LOC[4,7] \rightarrow EEGabnormal;$$

$$S = (10/E)4.$$

Experiment 2
(bacterial examination data)

- Number of instances: 20,000
- Number of condition attributes: 60
- Goals:
  - analyzing the relationship between the *bacterium-detected* attribute and other attributes
  - analyzing what attribute-values are related to the sensitivity of antibiotics when the value of bacterium-detected is (+).

Attribute Selection (bacterial examination data)

- Class-1： bacterium-detected（+、−）
  condition attributes： 11
- Class-2： antibiotic-sensibility
  (resistant (R), sensibility(S))
  condition attributes： 21

Some Results (bacterial examination data)

- Some of rules discovered by GDT-RS are the same as C4.5, e.g.,

  $\beta\text{-}\,lactamese(3+) \rightarrow$ *bacterium-detected(+)*

  $urine\text{-}\,quantity(<10^3) \rightarrow$ *bacterium-detected(-)*

- Some of rules can only be discovered by GDT-RS, e.g.,

  $disease1(pneumonia) \rightarrow$ *bacterium-detected(-).*

Experiment 3
(gastric cancer data)

- Instances number ：7520
- Condition Attributes: 38
- Classes ：
  - cause of death (specially, the direct death)
  - post-operative complication
- Goals ：
  - analyzing the relationship between the direct death and other attributes
  - analyzing the relationship between the post-operative complication and other attributes.

Result of Attribute Selection
(gastric cancer data)

- Class： the direct death

  sex, location_lon1, location_lon2, location_cir1, location_cir2, serosal_inva, peritoneal_meta, lymphnode_diss, reconstruction, pre_oper_comp1, post_oper_comp1, histological, structural_atyp, growth_pattern, depth, lymphatic_inva, vascular_inva, ln_metastasis, chemotherapypos

  (19 attributes are selected)

Result of Attribute Selection (2)

（gastric cancer data ）

- Class： post-operative complication

  multi-lesions, sex, location_lon1, location_cir1, location_cir2, lymphnode_diss, maximal_diam, reconstruction, pre_oper_comp1, histological, stromal_type, cellular_atyp, structural_atyp, growth_pattern, depth, lymphatic_inva, chemotherapypos

  (17 attributes are selected)

Experiment 4
(slope-collapse data)

- Instances number ：3436
  - (430 places were collapsed, and 3006 were not)
- Condition attributes: 32
- Continuous attributes in condition attributes: 6
  - extension of collapsed steep slope, gradient, altitude, thickness of surface of soil, No. of active fault, distance between slope and active fault.
- Goal： find out what is the reason that causes the slope to be collapsed.

Result of Attribute Selection
(slope-collapse data)

- 9 attributes are selected from 32 condition attributes:

  altitude, slope azimuthal, slope shape, direction of high rank topography, shape of transverse section, position of transition line, thickness of surface of soil, kind of plant, distance between slope and active fault.

(3 continuous attributes in red color)

The Discovered Rules (slope-collapse data)

- s_azimuthal(2) ∧ s_shape(5) ∧ direction_high(8) ∧ plant_kind(3) S = (4860/E)
- altitude[21,25) ∧ s_azimuthal(3) ∧ soil_thick(>=45) S = (486/E)
- s_azimuthal(4) ∧ direction_high(4) ∧ t_shape(1) ∧ tl_position(2) ∧ s_f_distance(>=9) S = (6750/E)
- altitude[16,17) ∧ s_azimuthal(3) ∧ soil_thick(>=45) ∧ s_f_distance(>=9) S = (1458/E)
- altitude[20,21) ∧ t_shape(3) ∧ tl_position(2) ∧ plant_kind(6) ∧ s_f_distance(>=9) S = (12150/E)
- altitude[11,12) ∧ s_azimuthal(2) ∧ tl_position(1) S = (1215/E)
- altitude[12,13) ∧ direction_high(9) ∧ tl_position(4) ∧ s_f_distance[8,9) S = (4050/E)
- altitude[12,13) ∧ s_azimuthal(5) ∧ t_shape(5) ∧ s_f_distance[8,9) S = (3645/E)
- ……

Other Methods for Attribute Selection

(download from http://www.iscs/nus.edu.sg/liuh/)

- **LVW**: A stochastic wrapper feature selection algorithm
- **LVI**: An incremental multivariate feature selection algorithm
- **WSBG**/C4.5: Wrapper of sequential backward generation
- **WSFG**/C4.5: Wrapper of sequential forward generation

Results of *LVW*

- Rule induction system: C4.5
- Executing times: 10
- Class: direct death
- Number of selected attributes for each time：

  20, 19, 21, 26, 22, 31, 21, 19, 31, 28
- Result-2 (19 attributes are selected)：

  multilesions, *sex*, location_lon3, location_cir4, liver_meta, *lymphnode_diss,* proximal_surg, resection_meth, combined_rese2, *reconstruction, pre_oper_comp1,* post_oper_com2, post_oper_com3, spec_histologi, cellular_atyp, *depth,* eval_of_treat, *ln_metastasis,* othertherapypre

Result of *LVW* (2)

- Result-2 (19 attributes are selected)：

  age, typeofcancer, location_cir3, location_cir4, liver_meta, *lymphnode_diss,* maximal_diam, distal_surg, combined_rese1, combined_rese2, pre_oper_comp2, *post_oper_com1, histological*, spec_histologi, *structural_atyp, depth, lymphatic_inva, vascular_inva, ln_metastasis*

  (only the attributes in red color are selected by our method)

Result of *WSFG*

- Rule induction system:

  C4.5

- Results

  the best relevant attribute first

Result of *WSFG* (2)

(class: direct death）

eval_of_treat, liver_meta, *peritoneal_meta,* typeofcancer, *chemotherapypos,* combined_rese1, *ln_metastasis*, *location_lon2, depth, pre_oper_comp1*, *histological, growth_pattern,vascular_inva*, *location_cir1*,location_lon3, cellular_atyp, maximal_diam, pre_oper_comp2, *location_lon1,* location_cir3, *sex,* post_oper_com3, age, *serosal_inva*, spec_histologi, proximal_surg, location_lon4, chemotherapypre, *lymphatic_inva, lymphnode_diss, structural_atyp*, distal_surg,resection_meth, combined_rese3, chemotherapyin, location_cir4, *post_oper_comp1,* stromal_type, combined_rese2, othertherapypre, othertherapyin, othertherapypos, *reconstruction,* multilesions, *location_cir2,* pre_oper_comp3

(the best relevant attribute first)

Result of *WSBG*

- Rule induction system:

  C4.5

- Result

  the least relevant attribute first

Result of *WSBG* (2)
(class: direct death)

peritoneal_meta, liver_meta, eval_of_treat, *lymphnode_diss, reconstruction,* chemotherapypos, *structural_atyp,* typeofcancer, *pre_oper_comp1*, maximal_diam, *location_lon2*, combined_rese3, othertherapypos, post_oper_com3, stromal_type, cellular_atyp, resection_meth, location_cir3, multilesions, location_cir4, proximal_surg, *location_cir1, sex, lymphatic_inva*, location_lon4, *location_lon1, location_cir2*, distal_surg, post_oper_com2, location_lon3, *vascular_inva*, combined_rese2, age, pre_oper_comp2, *ln_metastasis, serosal_inva, depth, growth_pattern*, combined_rese1, chemotherapyin, spec_histologi, *post_oper_com1, chemotherapypre*, pre_oper_comp3, *histological,* othertherapypre

Result of LVI
(gastric cancer data)

Number of allowed inconsistent instances	Executing times	Number of inconsistent instances	Number of selected attributes
80	1	79	19
	2	68	16
	3	49	20
	4	61	18
	5	66	20
20	1	7	49
	2	19	26
	3	19	28
	4	20	23
	5	18	26

Some Rules

Related to Direct Death

- peritoneal_meta(2) ∧ pre_oper_comp1(.) ∧ post_oper_com1(L) ∧ chemotherapypos(.) S= 3*(7200/E)
- location_lon1(M) ∧ post_oper_com1(L) ∧ ln_metastasis(3) ∧ chemotherapypos(.) S= 3*(2880/E)
- sex(F) ∧ location_cir2(.) ∧ post_oper_com1(L) ∧ growth_pattern(2) ∧ chemotherapypos(.) S= 3*(7200/E)
- location_cir1(L) ∧ location_cir2(.) ∧ post_oper_com1(L) ∧ ln_metastasis(2) ∧ chemotherapypos(.) S= 3*(25920/E)
- pre_oper_comp1(.) ∧ post_oper_com1(L) ∧ histological(MUC) ∧ growth_pattern(3) ∧ chemotherapypos(.) S= 3*(64800/E)
- sex(M) ∧ location_lon1(M) ∧ reconstruction(B2) ∧ pre_oper_comp1(.) ∧ structural_atyp(3) ∧ lymphatic_inva(3) ∧ vascular_inva(0) ∧ ln_metastasis(2) S=3*(345600/E)
- sex(F) ∧ location_lon2(M) ∧ location_cir2(.) ∧ pre_oper_comp1(A) ∧ depth(S2) ∧ chemotherapypos(.) S= 3*(46080/E)

GDT-RS vs. Discriminant Analysis

- if -then rules
- multi-class, high-dimension, large-scale data can be processed
- BK can be used easily
- the stability and uncertainty of a rule can be expressed explicitly
- continuous data must be discretized.

- algebraic expressions
- difficult to deal with the data with multi-class.
- difficult to use BK
- the stability and uncertainty of a rule cannot be explained clearly
- symbolic data must be quantized.

GDT-RS vs. ID3 (C4.5)

- BK can be used easily
- the stability and uncertainty of a rule can be expressed explicitly
- unseen instances are considered
- the minimal set of rules containing all instances can be discovered

- difficult to use BK
- the stability and uncertainty of a rule cannot be explained clearly
- unseen instances are not considered
- not consider whether the discovered rules are the minimal set covered all instances

Rough Sets in ILP and GrC
-- An Advanced Topic --

- Background and goal
- The normal problem setting for ILP
- Issues, observations, and solutions
- Rough problem settings
- Future work on RS (GrC) in ILP

ILP: Inductive Logic Programming
GrC: Granule Computing

Advantages of ILP
(Compared with Attribute-Value Learning)

- It can learn knowledge which is more expressive because it is in predicate logic
- It can utilize background knowledge more naturally and effectively because in ILP the examples, the background knowledge, as well as the learned knowledge are all expressed within the same logic framework.

Weak Points of ILP
(Compared with Attribute-Value Learning)

- It is more difficult to handle numbers (especially continuous values) prevailing in real-world databases.
- The theory, techniques are much less mature for ILP to deal with imperfect data (uncertainty, incompleteness, vagueness, impreciseness, etc. in examples, background knowledge as well as the learned rules).

Goal

- Applying Granular Computing (GrC) and a special form of GrC: Rough Sets to ILP to deal with some kinds of imperfect data which occur in large real-world applications.

Normal Problem Setting for ILP

- Given:
  - The target predicate p
  - The positive examples E^+ and the negative examples E^- (two sets of ground atoms of p)
  - Background knowledge B (a finite set of definite clauses)

Normal Problem Setting for ILP (2)

- To find:
  - Hypothesis *H* (the defining clauses of *p*) which is correct with respect to E^+ and E^-, i.e.

1. $H \cup B$ is complete with respect to E^+

(i.e. $\forall_{e \in E^+} H \cup B \models e$)

We also say that $H \cup B$ covers all positive examples.

2. $H \cup B$ is consistent with respect to E^-

(i.e. $\forall_{e \in E^-} H \cup B \not\models e$)

We also say that $H \cup B$ rejects any negative examples.

Normal Problem Setting for ILP (3)

- Prior conditions:

  1'. B is not complete with respect to E^+

  (Otherwise there will be no learning task at all)

  2'. $B \cup E^+$ is consistent with respect to E^-

  (Otherwise there will be no solution)

Everything is assumed correct and perfect.

Issues

- In large, real-world empirical learning, uncertainty, incompleteness, vagueness, impreciseness, etc. are frequently observed in training examples, in background knowledge, as well as in the induced hypothesis.
- Too strong bias may miss some useful solutions or have no solution at all.

Imperfect Data in ILP

- Imperfect output
  - Even the input (Examples and BK) are "perfect", there are usually several Hs that can be induced.
  - If the input is imperfect, we have imperfect hypotheses.
- Noisy data
  - Erroneous argument values in examples.
  - Erroneous classification of examples as belonging to E^+ or E^-.

Imperfect Data in ILP (2)

- Too sparse data
  - The training examples are too sparse to induce reliable *H*.
- Missing data
  - Missing values: some arguments of some examples have unknown values.
  - ***Missing predicates***: *BK* lacks essential predicates (or essential clauses of some predicates) so that no non-trivial *H* can be induced.

Imperfect Data in ILP (3)

- ***Indiscernible data***
  - Some examples belong to both E^+ and E^-.

This presentation will focus on

(1) Missing predicates

(2) Indiscernible data

Observations

- *H* should be "*correct with respect to* E^+ *and* E^- " needs to be relaxed, otherwise there will be no (meaningful) solutions to the ILP problem.
- While it is impossible to differentiate distinct objects, we may consider granules: sets of objects drawn together by similarity, indistinguishability, or functionality.

Observations (2)

- Even when precise solutions in terms of individual objects can be obtained, we may still prefect to granules in order to have an efficient and practical solution.
- When we use granules instead of individual objects, we are actually relaxing the strict requirements in the standard normal problem setting for ILP, so that rough but useful hypotheses can be induced from imperfect data.

Solution

- Granular Computing (GrC) can pay an important role in dealing with imperfect data and/or too strong bias in ILP.
- GrC is a superset of various theories (such as rough sets, fuzzy sets, interval computation) used to handle incompleteness, uncertainty, vagueness, etc. in information systems

  (Zadeh, 1997).

Why GrC?
A Practical Point of View

- With incomplete, uncertain, or vague information, it may be difficult to differentiate some elements and one is forced to consider granules.
- It may be sufficient to use granules in order to have an efficient and practical solution.
- The acquisition of precise information is too costly, and coarse-grained information reduces cost.

Solution (2)

- Granular Computing (GrC) may be regarded as a label of theories, methodologies, techniques, and tools that make use of granules, i.e., groups, classes, or clusters of a universe, in the process of problem solving.
- We use a special form of GrC: ***rough sets*** to provide a “rough” solution.

Rough Sets

- **Approximation space** $A = (U, R)$

  U is a set (called the universe)

  R is an *equivalence relation* on U (called an indiscernibility relation).

- In fact, U is partitioned by R into *equivalence classes*, elements within an equivalence class are indistinguishable in A.

Rough Sets (2)

- ***Lower* and *upper approximations***. For an equivalence relation *R*, the *lower* and *upper approximations* of $X \subseteq U$ are defined by

$$\underline{Apr_A}(X) = \bigcup_{[x]_R \subseteq X} [x]_R = \{x \in U \mid [x]_R \in X\}$$

$$\overline{Apr_A}(X) = \bigcup_{[x]_R \cap X \neq \phi} [x]_R = \{x \in U \mid [x]_R \cap X \neq \phi\}$$

where $[x]_R$ denotes the equivalence class containing *x*.

Rough Sets (3)

- **Boundary.** $Bnd_A(X) = \overline{Apr_A}(X) - \underline{Apr_A}(X)$ is called the *boundary* of X in A.
- **Rough membership.**
  - elements x surely belongs to X in A if $x \in \underline{Apr_A}(X)$
  - elements x possibly belongs to X in A if $x \in \overline{Apr_A}(X)$
  - elements x surely does not belong to X in A if $x \notin \overline{Apr_A}(X)$.

An Illustrating Example

Given:

The target predicate:

customer(Name, Age, Sex, Income)

The positive examples	**The negative examples**
customer(a, 30, female, 1).	*customer(c, 50, female, 2).*
customer(b, 53, female, 100).	*customer(g, 20, male, 2).*
customer(d, 50, female, 2).	
customer(e, 32, male, 10).	
customer(f, 55, male, 10).	

Background knowledge *B* defining *married_to(H, W)*
by *married_to(e, a).* *married_to(f, d).*

An Illustrating Example (2)

To find:

Hypothesis *H* (*customer/4*) which is correct with respect to E^+ and E^-.

The normal problem setting is perfectly suitable for this problem, and an ILP system can induce the following hypothesis *H* defining *customer/4*:

customer(N, A, S, I) :- I >= 10.

customer(N, A, S, I) :- married_to(N', N),
customer(N', A', S', I').

Rough Problem Setting for Insufficient BK

- **Problem**: If *married_to/2* is missing in *BK*, no hypothesis will be induced.
- **Solution**: Rough Problem Setting 1.
- Given:
  - The target predicate p
    (the set of all ground atoms of p is U).
  - An equivalence relation R on U
    (we have the approximation space $A = (U, R)$).
  - $E^+ \subseteq U$ and $E^- \subseteq U$ satisfying the prior condition:
    $B \cup E^+$ is consistent with respect to E^-.
  - BK, B (may lack essential predicates/clauses).

Rough Problem Setting for Insufficient BK (2)

Considering the following rough sets:

- $E^{++} = \overline{Apr_A}(E^{+})$ containing all positive examples, and those negative examples $E^{-+} = \{e' \in E^{-} \mid \exists_{e \in E^{+}} e \operatorname{R} e'\}$
- $E^{--} = E^{-} - E^{-+}$ containing the "pure" (remaining) negative examples.
- $E_{++} = \underline{Apr_A}(E^{+})$ containing "pure" positive examples. That is, $E_{++} = E^{+} - E^{+-}$ where $E^{+-} = \{e \in E^{+} \mid \exists_{e' \in E^{-}} e \operatorname{R} e'\}$

Rough Problem Setting for Insufficient BK (3)

- $E_{--} = E^{-} + E^{+-}$ containing all negative examples and “non-pure” positive examples.

To find:

- Hypothesis H^{+} (the defining clauses of p) which is correct with respect to E^{++} and E^{--} i.e.

1. $H^{+} \cup B$ covers all examples of E^{++}
2. $H^{+} \cup B$ rejects any examples of E^{--}

Rough Problem Setting for Insufficient BK (4)

- Hypothesis H^- (the defining clauses of p) which is correct with respect to E_{++} and E_{--} i.e.

1. $H^- \cup B$ covers all examples of E_{++}
2. $H^- \cup B$ rejects any examples of E_{--}

Example Revisited

Married_to/2 is missing in *B*. Let *R* be defined as "*customer(N, A, S, I) R customer(N', A, S, I)*", with the Rough Problem Setting 1, we may induce H^+ as:

customer(N, A, S, I) :- I >= 10.
customer(N, A, S, I) :- S = female.

which covers all positive examples and the negative example "*customer(c, 50, female, 2)*", rejecting other negative examples.

Example Revisited (2)

We may also induce H^- as:

customer(N, A, S, I) :- I >= 10.
customer(N, A, S, I) :- S = female, A < 50.

which covers all positive examples except
"*customer(d, 50, female, 2)*",
rejecting all negative examples.

Example Revisited (3)

- These hypotheses are rough (because the problem itself is rough), but still useful.
- On the other hand, if we insist in the normal problem setting for ILP, these hypothese are not considered as “solutions”.

Rough Problem Setting for Indiscernible Examples

- **Problem**: Consider *customer(Age, Sex, Income)*, we have *customer(50, female, 2)* belonging to E^+ as well as to E^-.
- **Solution**: Rough Problem Setting 2.
- Given:
  - The target predicate *p* (the set of all ground atoms of *p* is *U*).
  - $E^+ \subseteq U$ and $E^- \subseteq U$ where $E^+ \cap E^- \neq \phi$
  - Background knowledge *B*.

Rough Problem Setting for Indiscernible Examples (2)

- **Rough sets to consider and the hypotheses to find:**

  Taking the identity relation *I* as a special equivalence relation *R*, the remaining description of Rough Problem Setting 2 is the same as in Rough Problem Setting 1.

Rough Sets (GrC) for Other Imperfect Data in ILP

- Noisy data
- Too sparse data
- Missing values
- Discretization of continuous values

Future Work on RS (GrC) in ILP

- Trying to find more concrete formalisms and methods to deal with noisy data, too sparse data, missing values, etc. in ILP within the framework of RS (GrC).
- Giving quantitative measures associated with hypotheses induced of ILP, either in its normal problem setting or in the new rough setting.
- Developing ILP algorithms and systems based on rough problem settings.

Summary

- Rough sets offers mathematical tools and constitutes a sound basis for KDD.
- We introduced the basic concepts of (classical) rough set theory.
- We described a rough set based KDD process.
- We discussed the problem of imperfect data handling in ILP using some ideas, concepts and methods of GrC (or a particular form of GrC: Rough Sets).

Advanced Topics (to deal with real world problems)

- Recent extensions of rough set theory (rough mereology: approximate synthesis of objects) have developed new methods for decomposition of large datasets, data mining in distributed and multi-agent systems, and fusion of information granules to induce complex information granule approximation.

Advanced Topics (2)
(to deal with real world problems)

- Combining rough set theory with logic (including non-classical logic), ANN, GA, probabilistic and statistical reasoning, fuzzy set theory to construct a hybrid approach.

References and Further Readings

- Z. Pawlak, "Rough Sets", *International Journal of Computer and Information Sciences*, Vol.11, 341-356 (1982).
- Z. Pawlak, *Rough Sets - Theoretical Aspect of Reasoning about Data*, Kluwer Academic Publishers (1991).
- L. Polkowski and A. Skowron (eds.) *Rough Sets in Knowledge Discovery*, Vol.1 and Vol.2., Studies in Fuzziness and Soft Computing series, Physica-Verlag (1998).
- L. Polkowski and A. Skowron (eds.) *Rough Sets and Current Trends in Computing*, LNAI 1424. Springer (1998).
- T.Y. Lin and N. Cercone (eds.), *Rough Sets and Data Mining*, Kluwer Academic Publishers (1997).
- K. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Methods for Knowledge Discovery*, Kluwer Academic Publishers (1998).

References and Further Readings

- R. Slowinski, *Intelligent Decision Support*, Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers (1992).
- S.K. Pal and S. Skowron (eds.) *Rough Fuzzy Hybridization:* A New Trend in Decision-Making, Springer (1999).
- E. Orlowska (ed.) Incomplete Information: Rough Set Analysis, Physica-Verlag (1997).
- S. Tsumolto, et al. (eds.) *Proceedings of the 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery*, The University of Tokyo (1996).
- J. Komorowski and S. Tsumoto (eds.) *Rough Set Data Analysis in Bio-medicine and Public Health*, Physica-Verlag (to appear).

References and Further Readings

- W. Ziarko, "Discovery through Rough Set Theory", Knowledge Discovery: viewing wisdom from all perspectives, *Communications of the ACM*, Vol.42, No. 11 (1999).
- W. Ziarko (ed.) *Rough Sets, Fuzzy Sets, and Knowledge Discovery*, Springer (1993).
- J. Grzymala-Busse, Z. Pawlak, R. Slowinski, and W. Ziarko, "Rough Sets", *Communications of the ACM*, Vol.38, No. 11 (1999).
- Y.Y. Yao, "A Comparative Study of Fuzzy Sets and Rough Sets", Vol.109, 21-47, *Information Sciences* (1998).
- Y.Y. Yao, "Granular Computing: Basic Issues and Possible Solutions", *Proceedings of JCIS 2000*, Invited Session on Granular Computing and Data Mining, Vol.1, 186-189 (2000).

References and Further Readings

- N. Zhong, A. Skowron, and S. Ohsuga (eds.), *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, LNAI 1711, Springer (1999).
- A. Skowron and C. Rauszer, "The Discernibility Matrices and Functions in Information Systems", in R. Slowinski (ed) *Intelligent Decision Support*, Handbook of Applications and Advances of the Rough Sets Theory, 331-362, Kluwer (1992).
- A. Skowron and L. Polkowski, "Rough Mereological Foundations for Design, Analysis, Synthesis, and Control in Distributive Systems", *Information Sciences*, Vol.104, No.1-2, 129-156, North-Holland (1998).
- C. Liu and N. Zhong, "Rough Problem Settings for Inductive Logic Programming", in N. Zhong, A. Skowron, and S. Ohsuga (eds.), *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, LNAI 1711, 168-177, Springer (1999).

References and Further Readings

- J.Z. Dong, N. Zhong, and S. Ohsuga, “Rule Discovery by Probabilistic Rough Induction”, *Journal of Japanese Society for Artificial Intelligence*, Vol.15, No.2, 276-286 (2000).
- N. Zhong, J.Z. Dong, and S. Ohsuga, “GDT-RS: A Probabilistic Rough Induction System”, *Bulletin of International Rough Set Society*, Vol.3, No.4, 133-146 (1999).
- N. Zhong, J.Z. Dong, and S. Ohsuga,“Using Rough Sets with Heuristics for Feature Selection”, *Journal of Intelligent Information Systems* (to appear).
- N. Zhong, J.Z. Dong, and S. Ohsuga, “Soft Techniques for Rule Discovery in Data”, *NEUROCOMPUTING, An International Journal*, Special Issue on Rough-Neuro Computing (to appear).

References and Further Readings

- H.S. Nguyen and S.H. Nguyen, "Discretization Methods in Data Mining", in L. Polkowski and A. Skowron (eds.) *Rough Sets in Knowledge Discovery*, Vol.1, 451-482, Physica-Verlag (1998).
- T.Y. Lin, (ed.) *Journal of Intelligent Automation and Soft Computing*, Vol.2, No. 2, Special Issue on Rough Sets (1996).
- T.Y. Lin (ed.) *International Journal of Approximate Reasoning*, Vol.15, No. 4, Special Issue on Rough Sets (1996).
- Z. Ziarko (ed.) *Computational Intelligence, An International Journal*, Vol.11, No. 2, Special Issue on Rough Sets (1995).
- Z. Ziarko (ed.) *Fundamenta Informaticae, An International Journal*, Vol.27, No. 2-3, Special Issue on Rough Sets (1996).

References and Further Readings

- A. Skowron et al. (eds.) *NEUROCOMPUTING, An International Journal*, Special Issue on Rough-Neuro Computing (to appear).
- A. Skowron, N. Zhong, and N. Cercone (eds.) *Computational Intelligence, An International Journal*, Special Issue on Rough Sets, Data Mining, and Granular Computing (to appear).
- J. Grzymala-Busse, R. Swiniarski, N. Zhong, and Z. Ziarko (eds.) *International Journal of Applied Mathematics and Computer Science*, Special Issue on Rough Sets and Its Applications (to appear).

Related Conference and Web Pages

- RSCTC'2000 will be held in October 16-19, Banff, Canada

  http://www.cs.uregina.ca/~yyao/RSCTC200/
- International Rough Set Society

  http://www.cs.uregina.ca/~yyao/irss/bulletin.html
- BISC/SIG-GrC

  http://www.cs.uregina.ca/~yyao/GrC/

Thank You!