Introduction to Rough sets and Data mining

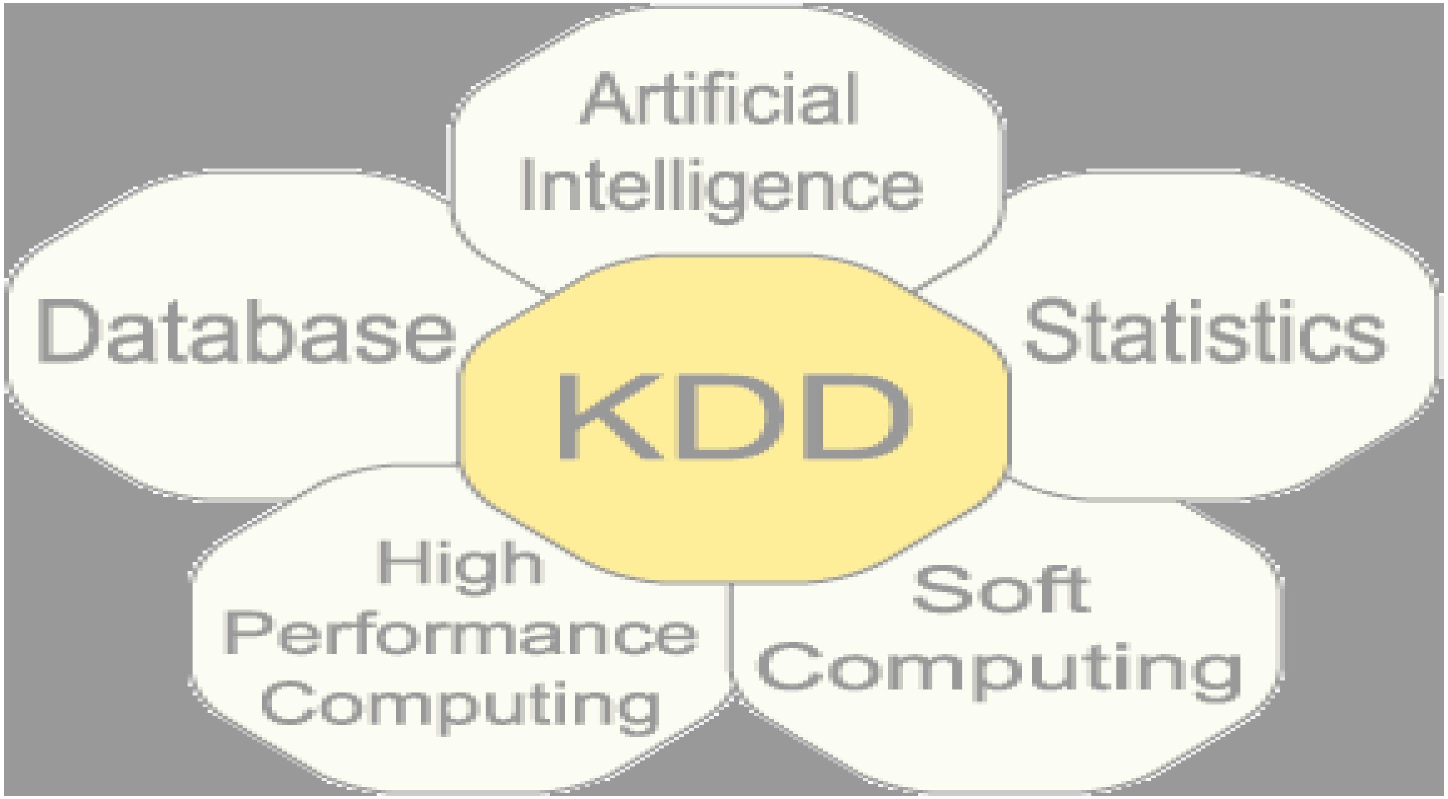
Nguyen Hung Son

http://www.mimuw.edu.pl/~son/

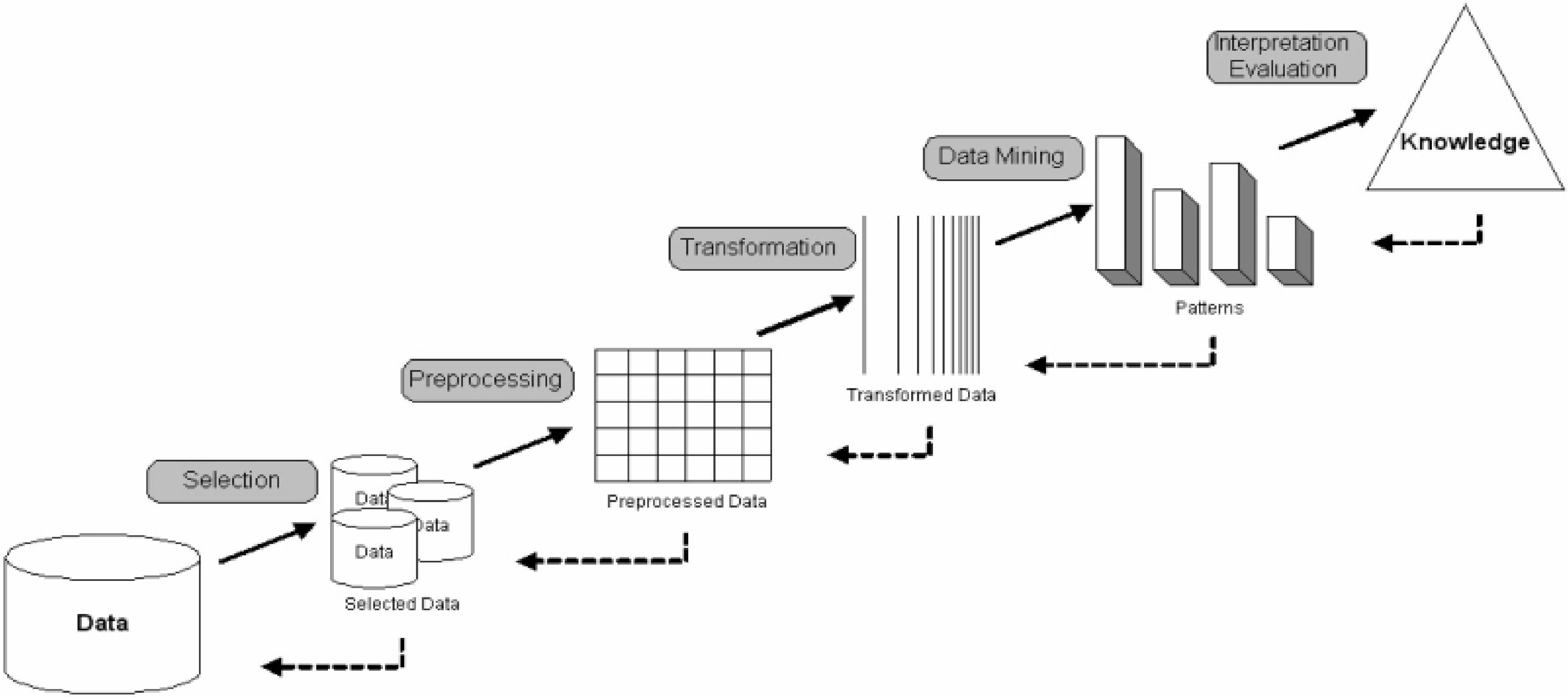
Outline

1. Knowledge discovery and data mining
    1. KDD processes
    2. Data mining techniques
    3. Data mining issues
2. Rough set theory
    1. Basic notions
    2. Applications of rough sets theory
    3. Rough set methodology to data mining

KDD


Artificial Intelligence
Database
KDD
Statistics
High Performance Computing
Soft Computing

Data Mining: a KDD process



Data mining is not …

- Generating multidimensional cubes of a relational table
- Searching for a phone number in a phone book
- Searching for keywords on Google
- Generating a histogram of salaries for different age groups
- Issuing SQL query to a database, and reading the reply

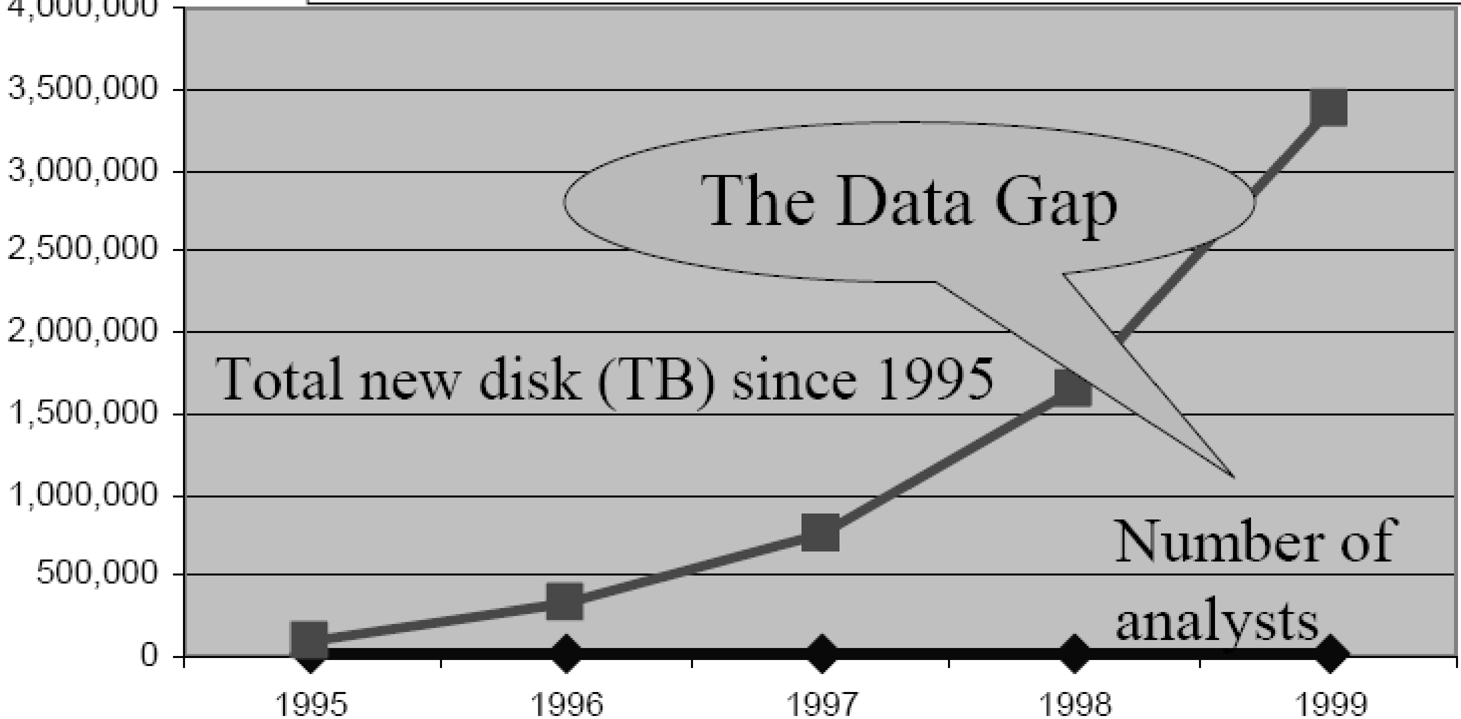
Data mining is …

- Finding groups of people with similar hobbies
- Are chances of getting cancer higher if you live near a power line?

Why is Data Mining prevalent?

- Lots of data is collected and stored in data warehouses
  - Business: Wal-Mart logs nearly 20 million transactions per day
  - Astronomy: Telescope collecting large amounts of data (SDSS)
  - Space: NASA is collecting peta bytes of data from satellites
  - Physics: High energy physics experiments are expected to generate 100 to 1000 tera bytes in the next decade
- Quality and richness of data collected is improving
  - Ex. Retailers, E-commerce, Science
- The gap between data and analysts is increasing
  - Hidden information is not always evident
  - High cost of human labor
  - Much of data is never analyzed at all

Ref: R. Grossman, C. Kamath, V. Kumar, *Data Mining for Scientific and Engineering Applications*



Steps of a KDD Process

1. Learning the application domain:
   - relevant prior knowledge and goals of application
2. Creating a target data set: data selection
3. Data cleaning and preprocessing: (may take 60% of effort!)
4. Data reduction and transformation:
   - Find useful features, dimensionality/variable reduction, invariant representation.
5. Choosing functions of data mining
   - summarization, classification, regression, association, clustering.
6. Choosing the mining algorithm(s)
7. Data mining: search for patterns of interest
8. Pattern evaluation and knowledge presentation
   - visualization, transformation, removing redundant patterns, etc.
9. Use of discovered knowledge

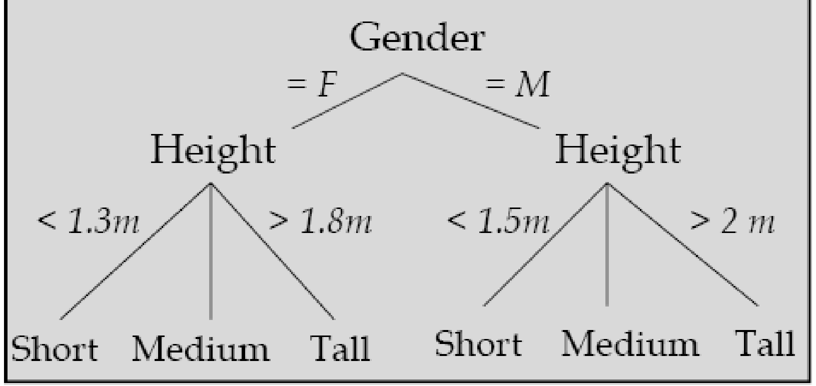
Data mining tasks

- Classification (predictive)
- Clustering (descriptive)
- Association Rule Discovery (descriptive)
- Sequential Pattern Discovery (descriptive)
- Regression (predictive)
- Deviation Detection (predictive)

Classification

- Modeling a class attribute, using other attributes
- Applications
  - Targeted marketing
  - Customer attrition

Decision Tree



Source: Data Mining – Introductory and Advanced topics by Margaret Dunham

Name	Gender	Height	Output
Kristina	F	1.6 m	Medium
Jim	M	2 m	Medium
Maggie	F	1.9 m	Tall
Martha	F	1.88 m	Tall
Stephanie	F	1.7 m	Medium
Bob	M	1.85 m	Medium
Kathy	F	1.6 m	Medium
Dave	M	1.7 m	Medium
Worth	M	2.2 m	Tall
Steven	M	2.1 m	Tall
Debbie	F	1.8 m	Medium
Todd	M	1.95 m	Medium
Kim	F	1.9 m	Tall
Amy	F	1.8 m	Medium
Lynette	F	1.75 m	Medium

Applications of Classification

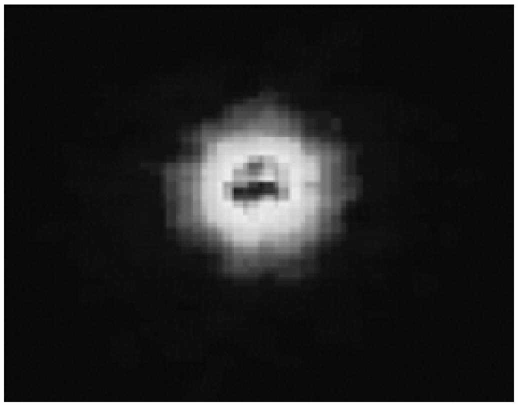
- **Marketing**
  - *Goal*: Reduce cost of mailing by targeting a set of consumers likely to buy a new cell phone product
  - *Approach*:
    - Use the data collected for a similar product introduced in the recent past.
    - Use the profiles of customers along with their {buy, didn't buy} decision. The profile of the information may consist of demographic, lifestyle and company interaction.
- **Fraud Detection**
  - *Goal:* Predict fraudulent cases in credit card transactions
  - *Approach*:
    - Use credit card transactions and the information on its account holders as attributes (important information: when and where the card was used)
    - Label past transactions as {fraud, fair} transactions to form the class attribute
    - Learn a model for the class of transactions and use this model to detect fraud by observing credit card transactions on an account

Application: Sky survey cataloging

- *Goal*: To predict class {star, galaxy} of sky objects, especially visually faint ones, based on the telescopic survey images (from Palomar Observatory)
  - 3000 images with 23,040 x 23,040 pixels per image
- *Approach*:
  - Segment the image
  - Measure image attributes (40 of them) per object
  - Model the class based on these features
- *Success story*: Could find 16 new high red-shift quasars (some of the farthest objects that are difficult to find) !!!

Classifying galaxies

Early



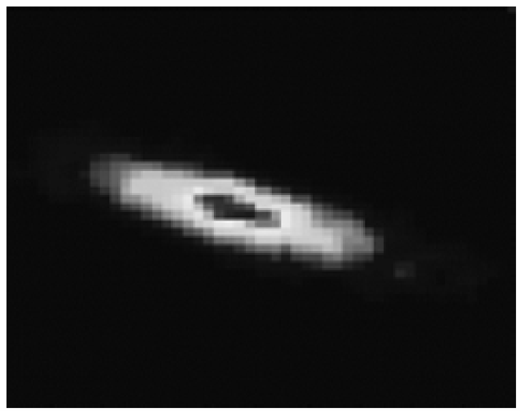
Class:

- Stages of Formation

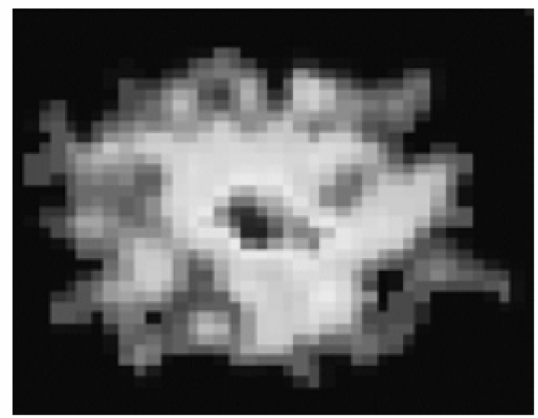
Attributes:

- Image features,
- Characteristics of light waves received, etc.

Intermediate



Late



Data Size:

- 72 million stars, 20 million galaxies
- Object Catalog: 9 GB
- Image Database: 150 GB

Source: Minnesota Automated Plate Scanner Catalog, http://aps.umn.edu

Regresion

- Linear regression
  - Data is modeled using a straight line of a form

    $Y = a + bX$
- Non-linear regression
  - Data is modeled using a nonlinear function

    $Y = a + b \cdot f(X)$

Association rules

Source: Data Mining – Introductory and Advanced topics by Margaret Dunham

Transaction	Items	Transaction	Items
T1	Blouse	T11	T-Shirt
T2	Shoes, Skirt, T-Shirt	T12	Blouse, Jeans, Shoes, Skirt, T-Shirt
T3	Jeans, T-Shirt	T13	Jeans, Shoes, Shorts, T-Shirt
T4	Jeans, Shoes, T-Shirt	T14	Shoes, Skirt, T-Shirt
T5	Jeans, Shorts	T15	Jeans, T-Shirt
T6	Shoes, T-Shirt	T16	Skirt, T-Shirt
T7	Jeans, Skirt	T17	Blouse, Jeans, Skirt
T8	Jeans, Shoes, Shorts, T-Shirt	T18	Jeans, Shoes, Shorts, T-Shirt
T9	Jeans	T19	Jeans
T10	Jeans, Shoes, T-Shirt	T20	Jeans, Shoes, Shorts, T-Shirt

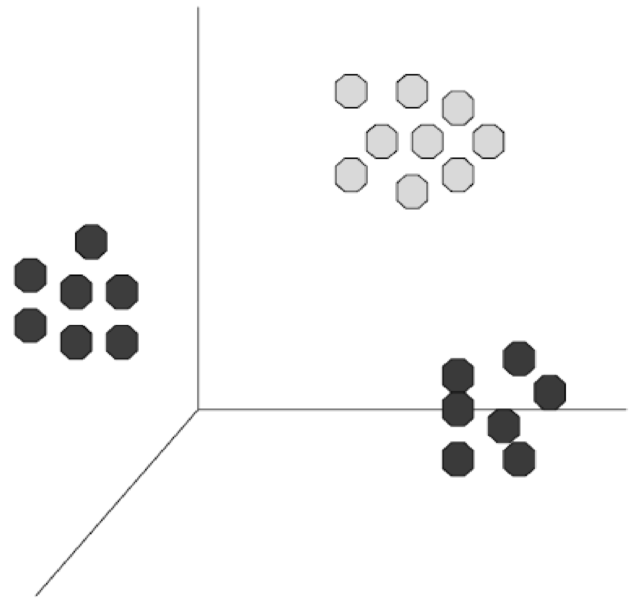
{Jeans, T-Shirt, Shoes} → { Shorts}
Support: 20% Confidence: 100%

Application of association analysis

- Marketing and sales promotion
  - *Item as a consequent*: can be used to determine what products will boost its sales
  - *Item as an antecedent*: can be used to see which products will be impacted if the store stops selling an item (e.g. cheap soda is a "loss leader" for many grocery stores.)
  - $Item_1$ => $Item_2$: can be used to see what products should be stocked along with $Item_1$ to promote the sale of $Item_2$
- Super market shelf management
  - *Example*
    - If a customer buys Jelly, then he is very likely to buy Peanut Butter.
    - So don't be surprised if you find Peanut Butter next to Jelly on an aisle in the super market.
- Inventory Management

Clustering

- Determine object groupings such that objects within the same cluster are similar to each other, while objects in different groups are not
- Problem with similarity measures:
  - Euclidean distance if attributes are continuous
  - Other problem-specific measures
- Example: Euclidean distance based clustering in 3D space
  - Intra cluster distances are minimized
  - Inter cluster distances are maximized



Application of Clustering

- Market Segmentation:
  - To subdivide a market into distinct subset of customers where each subset can be targeted with a distinct marketing mix
- Document Clustering
  - To find groups of documents that are similar to each other based on important terms appearing in them
- Stock market:
  - Observe stock movements everyday
  - Clustering points: Stock – {UP / DOWN}
  - Similarity measure: Two points are more similar if the events described by them frequently happen together on the same day
- Deviation/Anomaly Detection: detect significant deviations from normal behavior
  - Ex. detection of fraudulent credit card transactions
  - Detection of intrusion of a computer network

Sequential Pattern Discovery:

- Given is a set of objects, with each object associated with its own timeline of events, find rules that predict strong sequential dependencies among different events
- Applications:
  - Telecommunication alarm logs
    - (Inverter_Problem Excessive_Line_Current) (Rectifier_Alarm) → (Fire_Alarm)
  - Point of sale transaction sequences
    - (Intro_to_Visual_C) (C++ Primer)→(Perl_For_Dummies, Tcl_Tk)
    - (Shoes) (Racket, Racket ball)→(Sports_Jacket)

Summary on KDD and data mining

- Knowledge discovery in databases is the process of identifying valid, novel, potentially useful, and ultimately understandable patterns/models in data.

- Data mining is a step in the knowledge discovery process consisting of particular data mining algorithms that, under some acceptable computational efficiency limitations, finds patterns or models in data.

Rough sets: Introduction

- **Rough set theory** was developed by Zdzislaw Pawlak in the early 1980's.
- Pioneering 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).

Rough sets: Introduction

- 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.

Rough sets: Introduction

- Recent extensions of rough set theory:
  - **Rough mereology**
  - **Ontology-based rough sets**

  have developed new methods for
  - decomposition of large data sets,
  - data mining in distributed and multi-agent systems, and
  - granular computing.

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
x1	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

$$a : U \to V_a$$

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

Decision Systems/Tables

	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

- 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.

Upper Approximation:

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

Lower Approximation:

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

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 $\{u1\}, \{u2\}, \{u3\}, \{u4\}, \{u5, u7\}, \{u6, u8\}$.

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

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

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

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

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

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

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

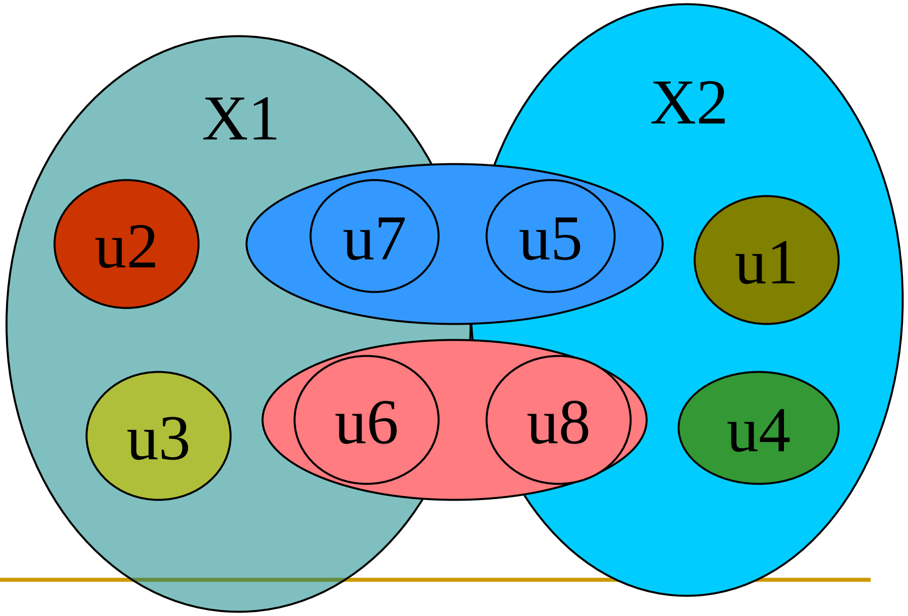
$\overline{R}X2 = \{u1, u4, u5, u8, \mathbf{u7, u6}\}$

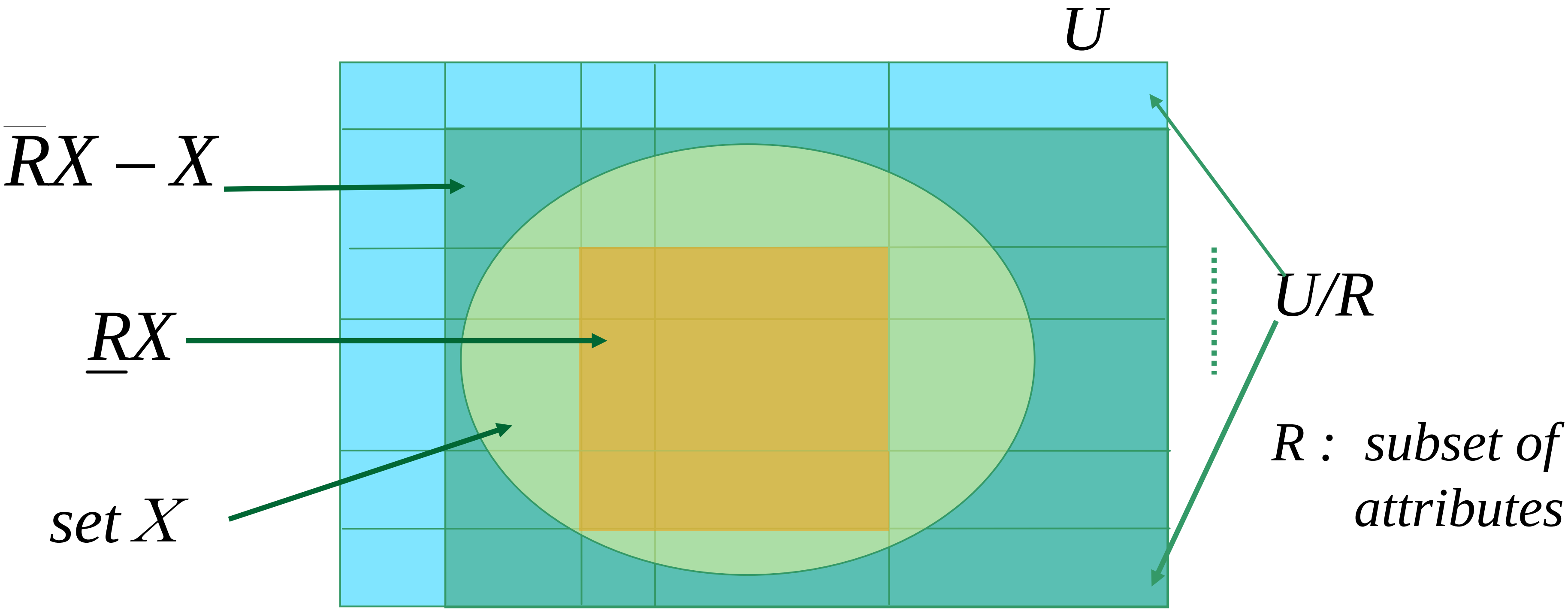
R = {Headache, Temp.}
U/R = { {u1}, {u2}, {u3}, {u4}, {u5, u7}, {u6, u8}}

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,* ***u8, u5****}*

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




U
$\overline{R}X - X$
U/R
$\underline{R}X$
R : subset of attributes
set X

Properties of Approximations

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

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

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

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

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

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

Accuracy of Approximation

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

where $|X|$ denotes the cardinality of $X \neq \varnothing$

Obviously $\alpha_B(X) < 1$,

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

If $0 \le \alpha_B \le 1$. X is *rough* with respect to B.

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.

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

- 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 $\pi = 1$.

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} ⋂ {MusclePain, Temp} = {Temp}**