The Use of Rough Set Methods in Knowledge Discovery in Databases

Marcin Szczuka
The University of Warsaw
Poland

Why KDD?

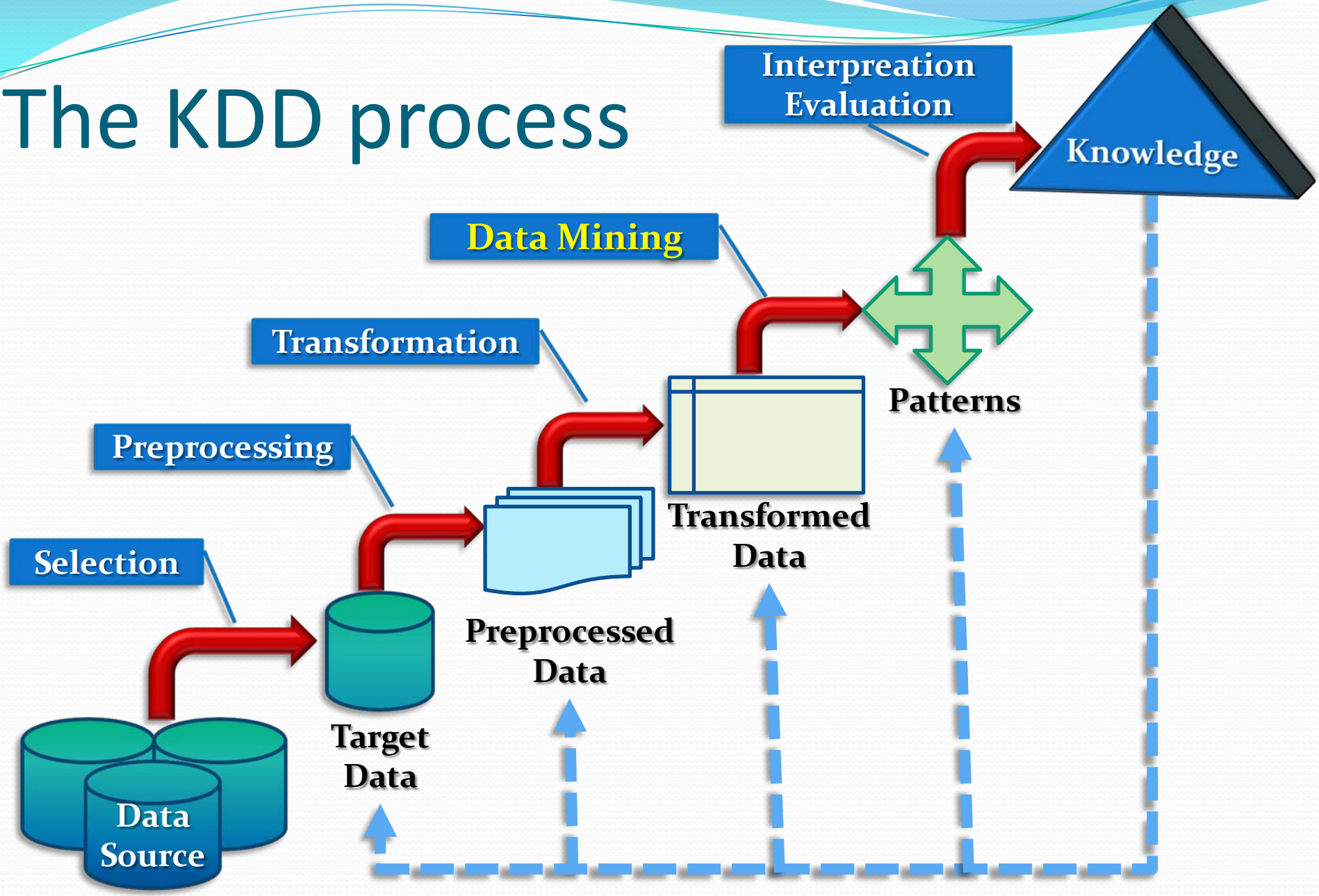
*„We are drowning in the sea of data, but what we really want is **knowledge**"*

PROBLEM: How to retrieve useful information (knowledge) from massive data sources?

SOLUTION: Data Warehouses + Data Mining

- Acquisition of data in real time
- Discovery of interesting knowledge (patterns, dependences, rules, regularities, ...) from large data sets.

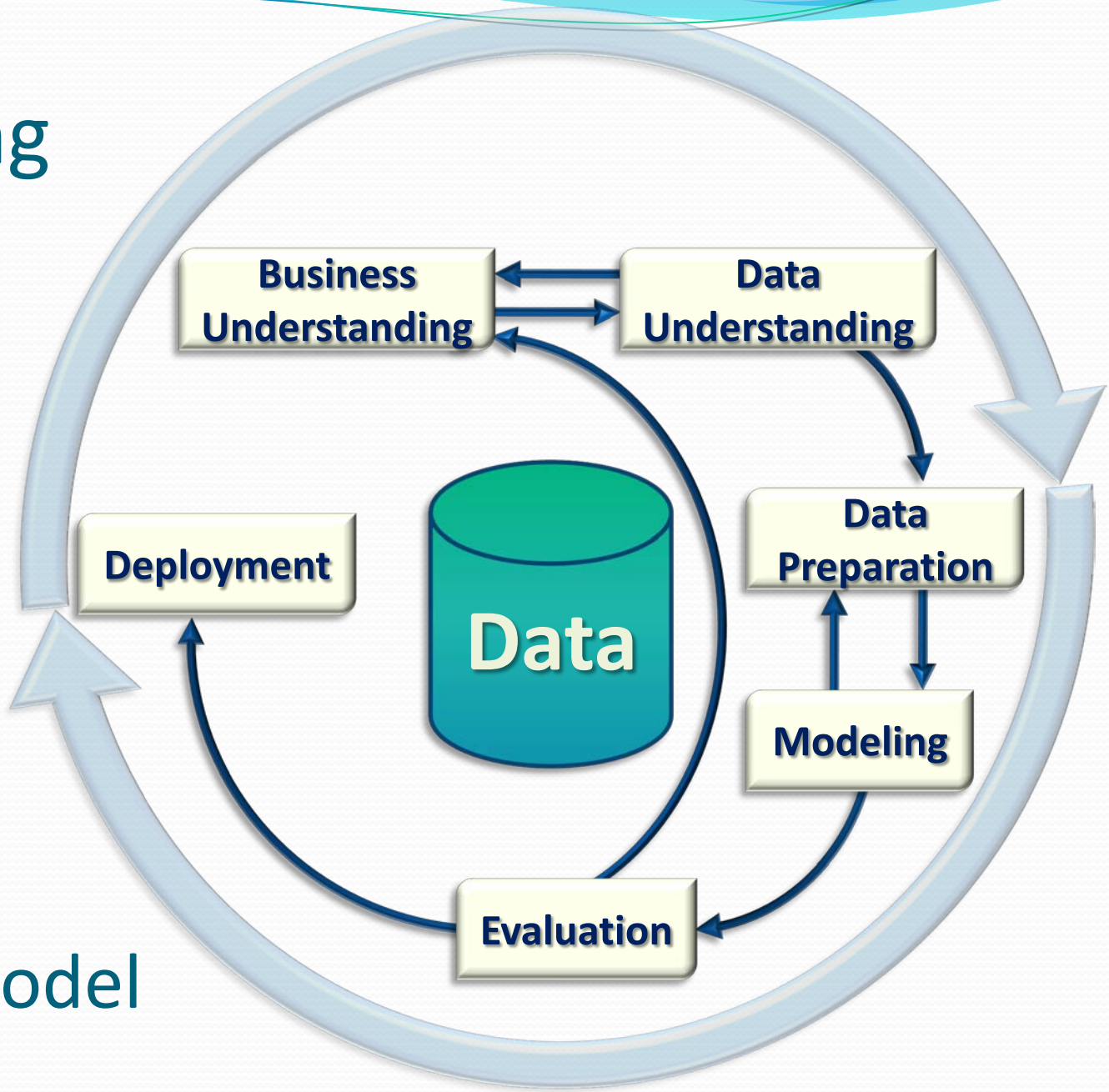
The KDD process



Data mining

- **Data mining** = the iterative and interactive process of discovering **non-trivial, implicit, previously unknown and potentially useful**(interesting) information or patterns from data in large databases.
- **Wikipedia: Data mining** (the analysis step of the Knowledge Discovery in Databases process, or KDD), a relatively young and interdisciplinary field of computer science, is the process of extracting patterns from large data sets by combining methods from statistics and artificial intelligence with database management.

Data Mining



CRISP-DM Model

Challenges and Competitions

- Since 1997 ACM KDD Cup attracted numerous researchers in the area.
- In 2009 Netflix Prize ($1 million) has opened the gate for really serious business-oriented competitions.
- Several platforms for on-line KDD/DM competitions are now existing: **kaggle.com**, **tunedit.org**, …
- Crowdsourcing of highly advanced DM tasks.
- There are several opportunities to win thousands of dollars (or Euros), and that does not include …

Heritage Health Prize

The goal of the prize is to develop a predictive algorithm that can identify patients who will be admitted to the hospital within the next year, using historical claims data.

- **$3 million (US) Grand Prize**.
- Six Milestone Prizes totalling $230,000, which are awarded in varying amounts at three designated intervals during the Competition.
- **Ends in April 2013**

http://www.heritagehealthprize.com/

Rough Set Methodology

What Rough Sets are all about and how to make them work for you?

Rough Set Methodology

- Rough set methodology is based on indiscernibility between objects.
- Rough set methods utilize the comparison between elements, e.g.., discernibility, indiscernibility, similarity, …
- In KDD, rough set methodology can be applied to:
  - implement efficient methods for mining interesting templates from data: data reduction, minimal decision rules, decomposition, hierarchical learning …
  - cooperate and improve existing methods in data mining like decision trees, association rules, clustering, kNN, neural networks, Bayesian networks…
  - design practical KDD projects for complex application domains like: search engines, medical data analysis, bioinformatics, …

A simple example

Id.	Age	LEM	Walk
u_1	16-30	50+	Yes
u_2	16-30	0	No
u_3	31-45	1-25	No
u_4	31-45	1-25	Yes
u_5	46-60	26-49	No
u_6	16-30	26-49	Yes
u_7	46-60	26-49	No

Information system:

Rows = objects

Columns = attributes (features)

If decision (attribute) is present the information system becomes **decision table**.

We usually denote it by:

$S=(U,A)$, where

U – the set (universe) of objects

A – the set of attributes

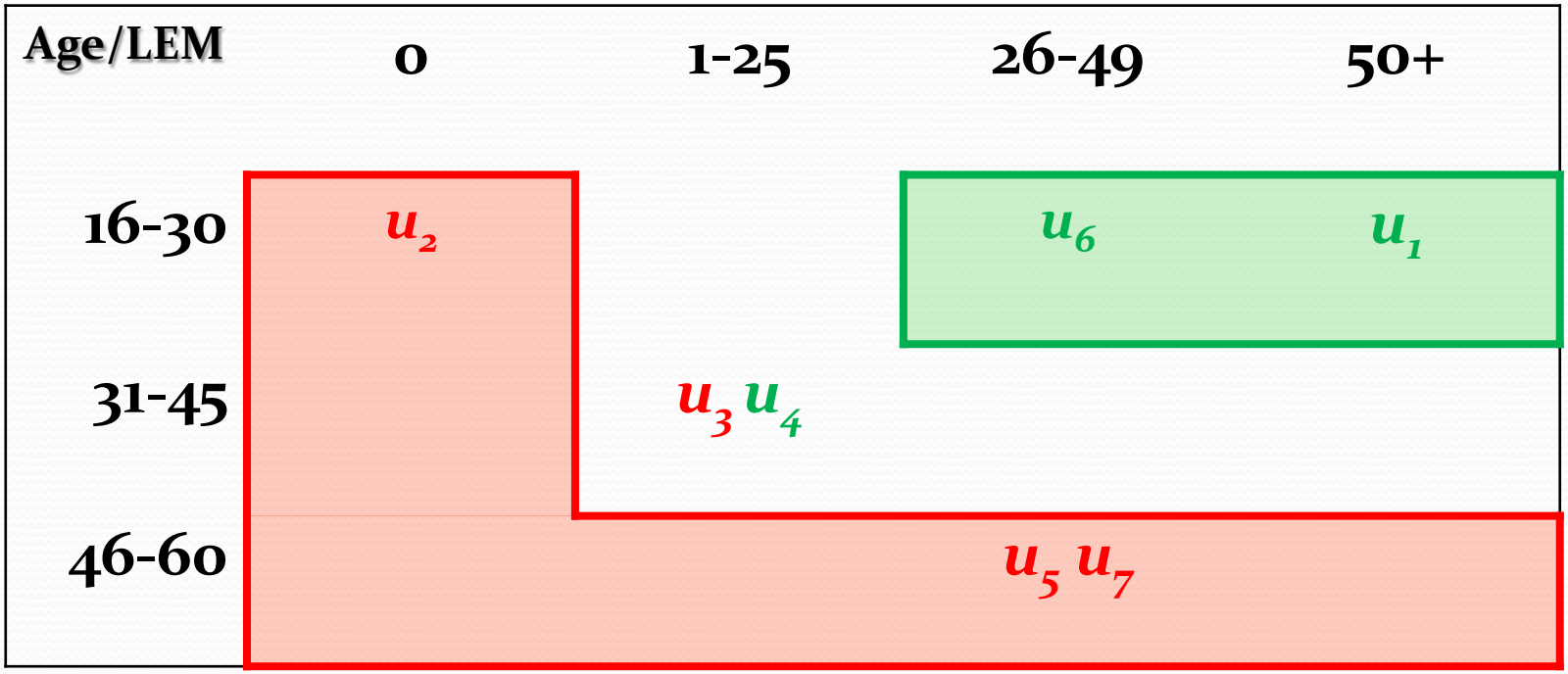
(In)Discernibility

In our example the information is not precise (complete) enough. We cannot discern precisely walking from non-walking patients. So, the concepts of walking and non-walking patient are partly ***indiscernible***.

Age/LEM	0	1-25	26-49	50+
16-30	u_2		u_6	u_1
31-45		u_3 u_4		
46-60			u_5 u_7	

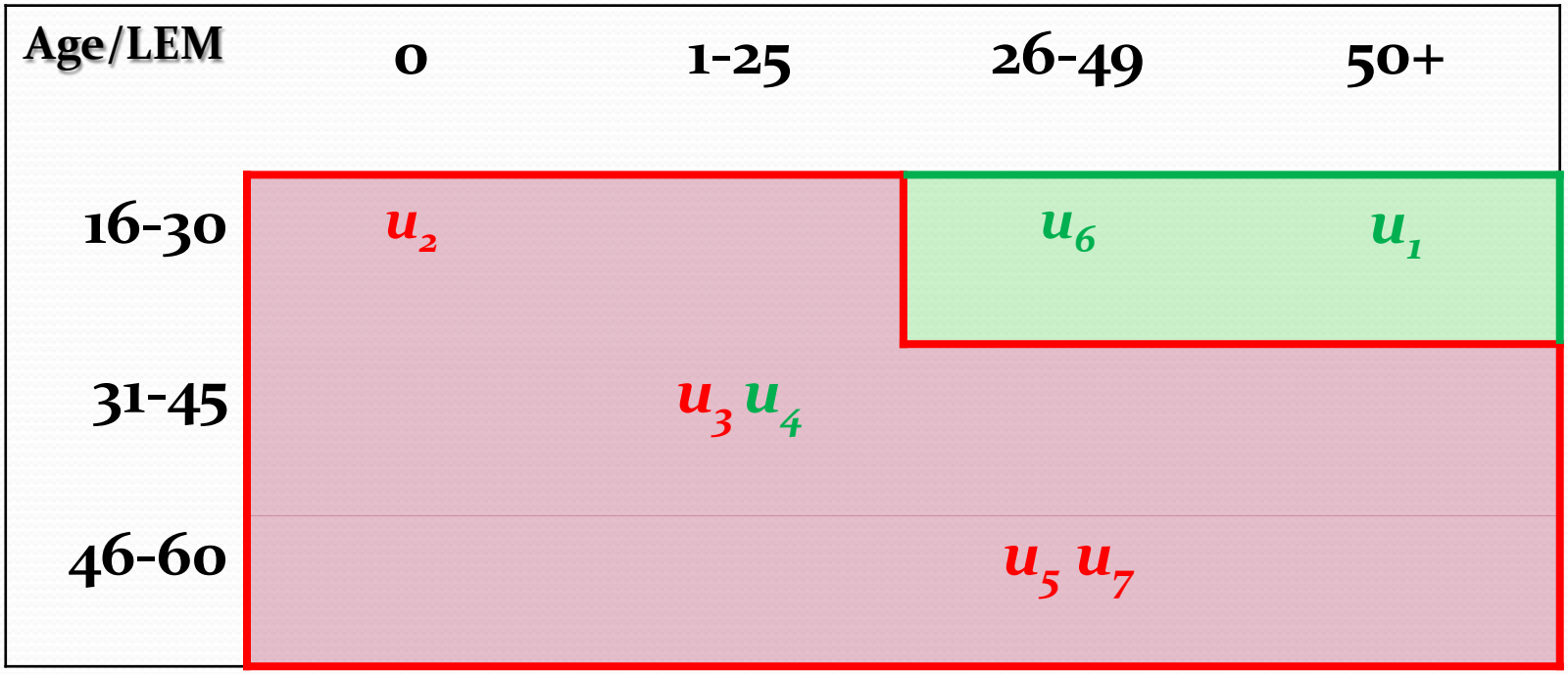
Approximations

Lower approximation corresponds to certainty about the object belonging to a concept (set). They **definitely** belong to the set.



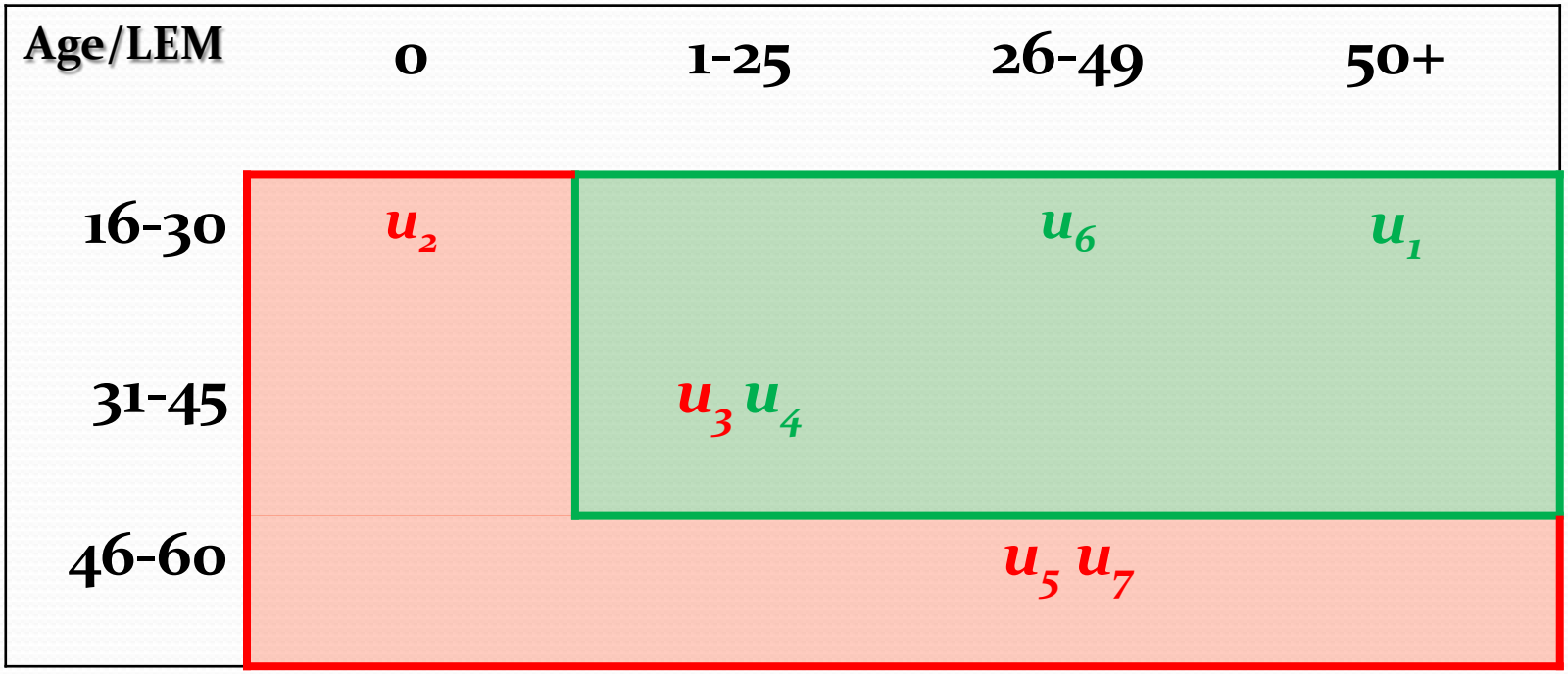
Approximations

Upper approximation corresponds to possibility of the object belonging to a concept (set). It is possible (likely, feasible) that they belong to the set, i.e., they **roughly** are in the set.



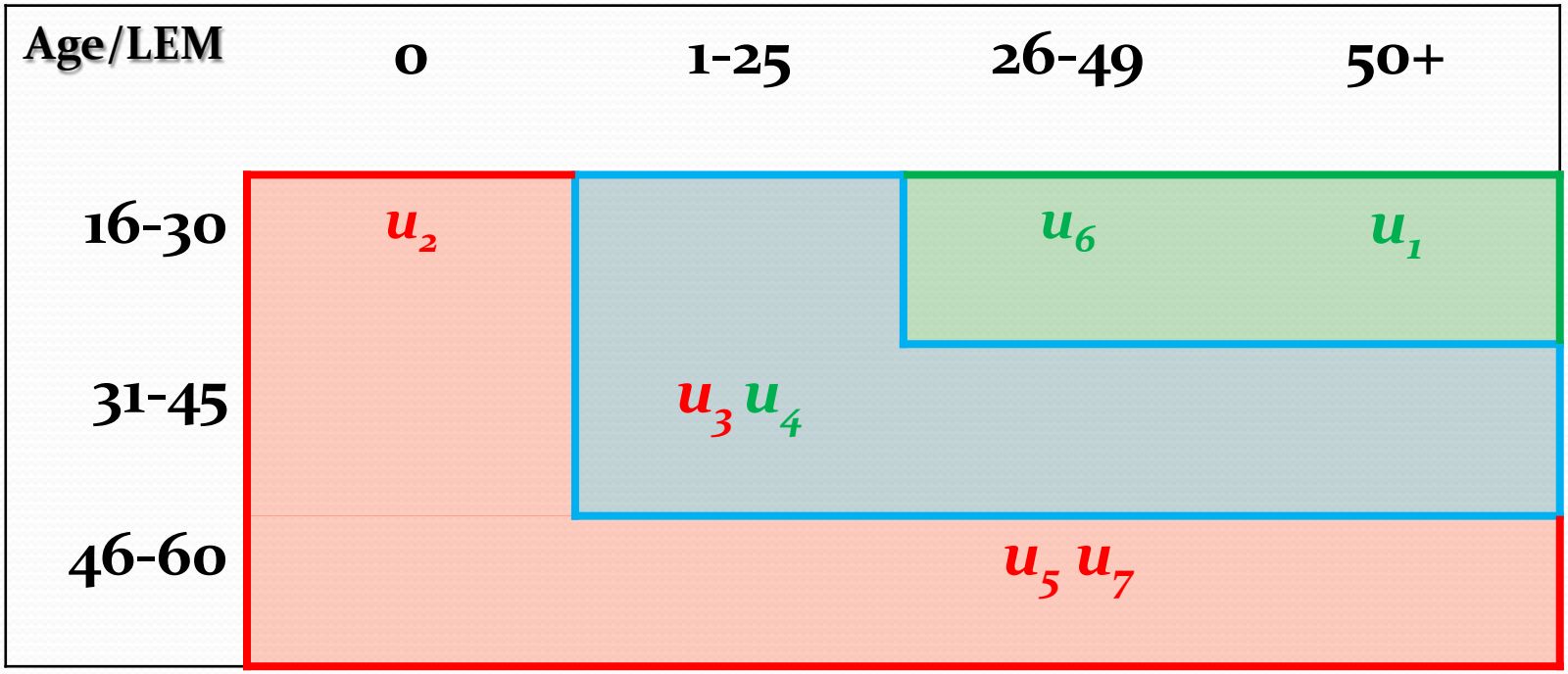
Approximations

Upper approximation corresponds to possibility of the object belonging to a concept (set). It is possible (likely, feasible) that they belong to the set, i.e., they **rougly** are in the set.

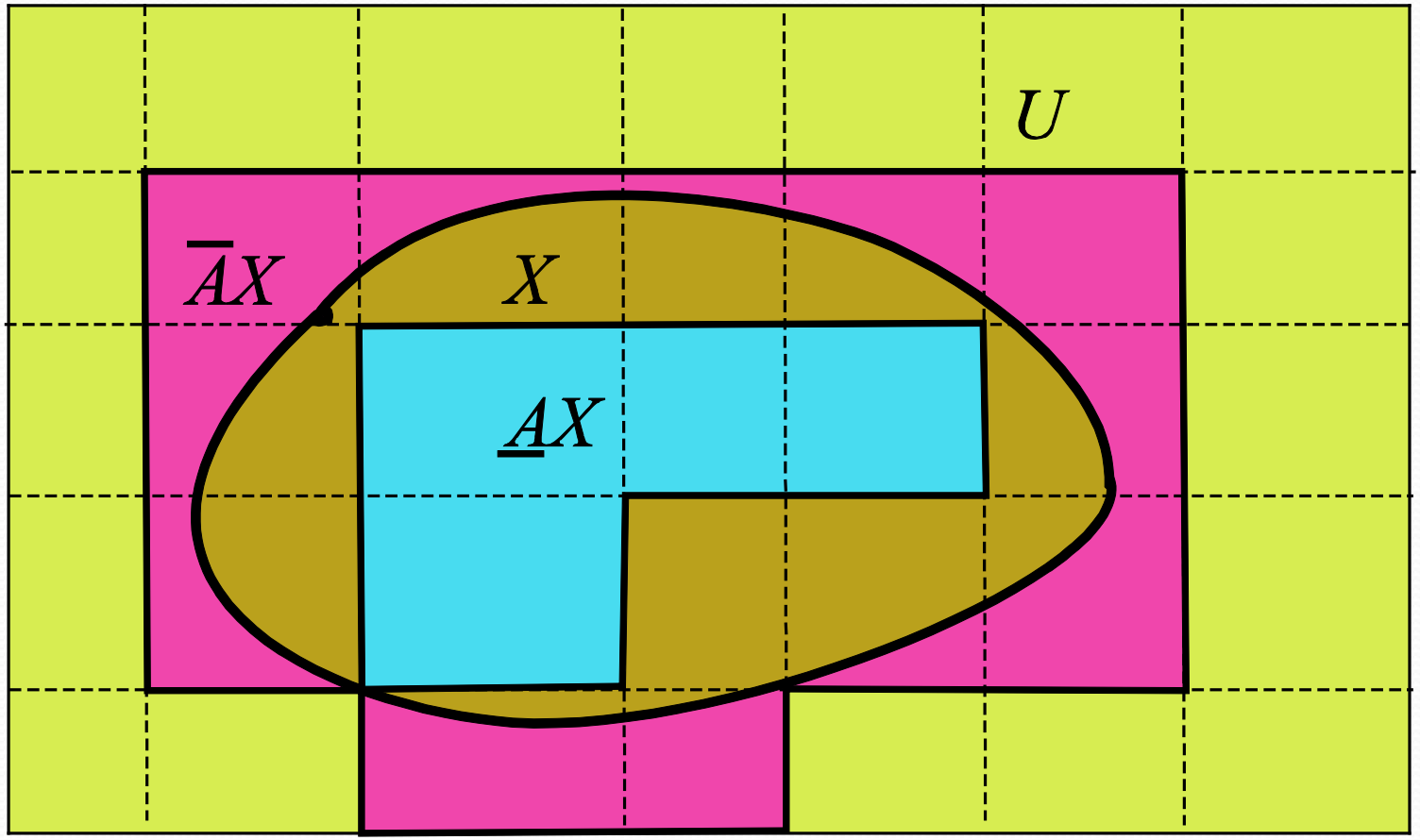


Boundary

The boundary region represents the uncertain portion of our data set. We do not have enough information to definitely establish the status of objects in this area.



The donut



Rough Set Techniques in DM

Over the years work of several “rough setters” resulted in creation of algorithmic methods for analysis of various kinds of data.

These methods utilize fundamental rough set notions such as discernibility, approximation, information function and reduct.

These methods are dealing with:

- Reduction of data size and complexity.
- Discovery of frequent patterns and decision rule discovery.
- Continuous attributes’ discretization.
- Data decomposition.
- Others - including new feature construction, instance-based learning, …

Reduction

- *Do we need all attributes?*
- *Do we need to store the entire data?*
- *Is it possible to avoid a costly test?*

Reducts are subsets of attributes that preserve the same amount of information. They are, however, (NP-)hard to find.

- Efficient and robust heuristics exist for reduct construction task.
- Searching for reducts may be done efficiently with the use of , e.g.., evolutionary computation.
- Overfitting can be avoided by considering several reducts, pruning rules and lessening discernibility constraints.

Data reduction with RS

What is a reduct?

Reducts are minimal subsets of attributes which contain a necessary portion of information from the set of all attributes.

Given an information system $S = (U,A)$ and a monotone evaluation function:

$$\mu_S : \wp(A) \to \Re_+$$

The set $B \subset A$ is called ***μ-reduct*** of A, iff:

1. $\mu(B) = \mu(A)$,
2. for any proper subset $B_o \subset B$ we have $\mu(B_o) < \mu(B)$.

The set $B \subset A$ is called ***approximate reduct***, iff:

1. $\mu(B) \leq \mu(A) - \varepsilon$,
2. for any proper subset ...

Some types of reducts

- **Information reduct:**

  $\mu_I(B)$ = number of pairs of objects discerned by B

- **Decision-oriented reduct:**

  $\mu_D(B)$ = number of pairs of ***conflicting objects*** discerned by B

- **Object-oriented reduct:**

  $\mu_x(B)$ = number of objects discerned from x by B

- **Frequent reducts;**
- **α-reducts;**
- **…**

Pattern and rule discovery

By examining the structure of indiscernibility classes and reducts one can summarize information carried by objects using patterns of the form:

$$(a_{i1} = v_1) \wedge \ldots \wedge (a_{ik} = v_k)$$

These patterns may be further converted into associations.

In classification problems, we can produce decision rules of the form:

$$(a_{i1} = v_1) \wedge \ldots \wedge (a_{ik} = v_k) \Rightarrow d = v_d$$

Patterns and rules can be filtered, pruned, generalized, and composed. That permits management of discovered knowledge.

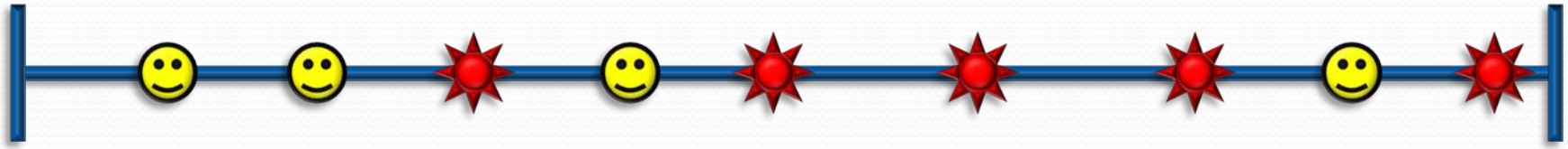
Discretization

Attributes that have many different values, e.g.., real-valued, may pose a technical problem for some algorithmic DM methods. **Discretization** (quantization) of attribute values can be done using rough set framework.

We consider all pairs of objects. Then we consider all possible cuts on attributes in discourse. We choose the cut that induces best split w.r.t. the number of objects from different decision classes that are discerned by this split.

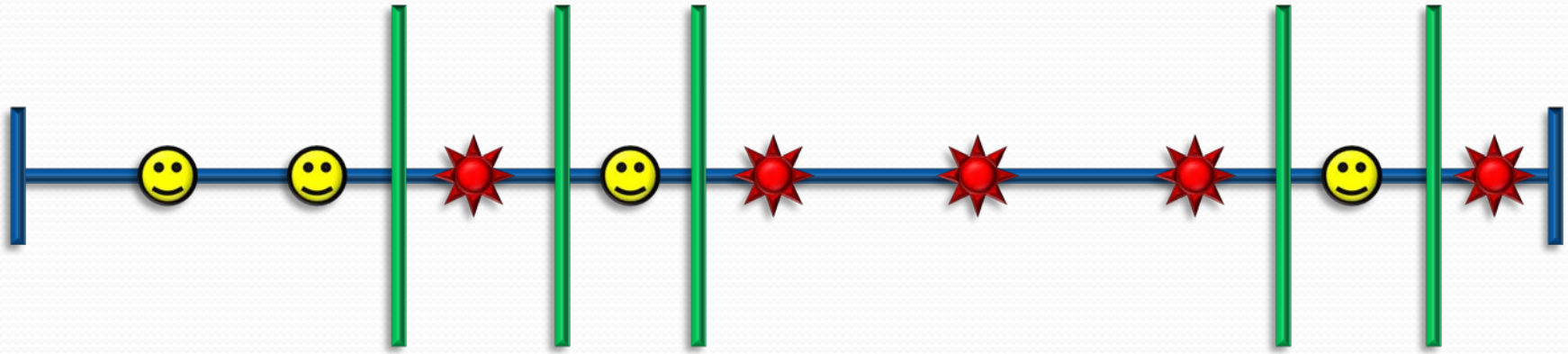
This is called **Maximal Discernibility (MD)** heuristic. Various modifications, especially concerning the choice of best cut, exist.

Discretization - example



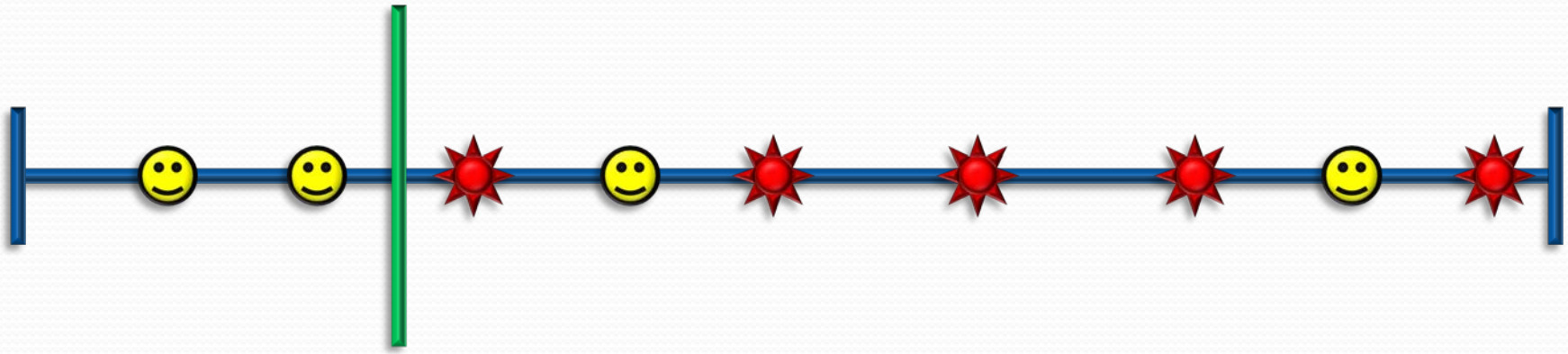
Number of conflicts: **20**

Discretization - example



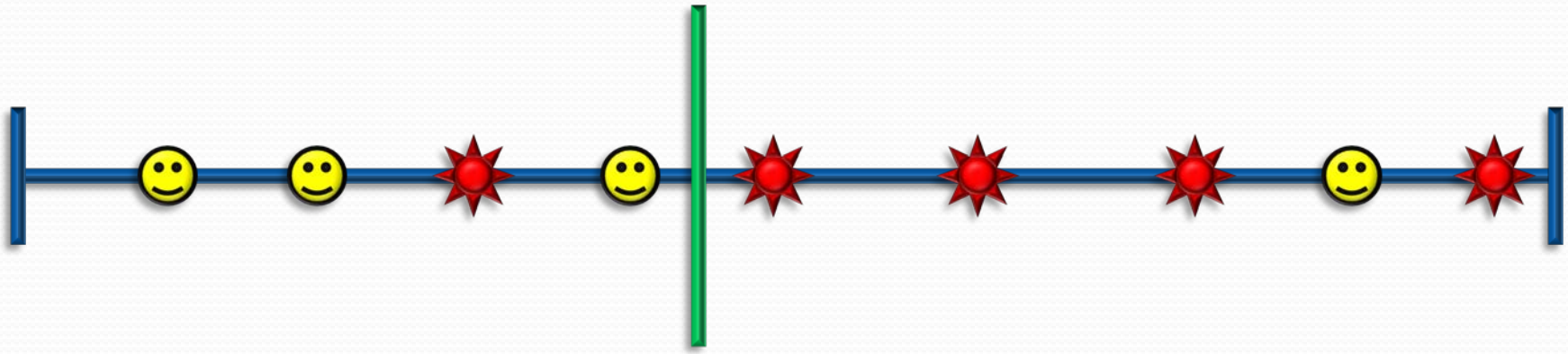
Number of conflicts: **20**

Discretization - example



Number of conflicts: **10**

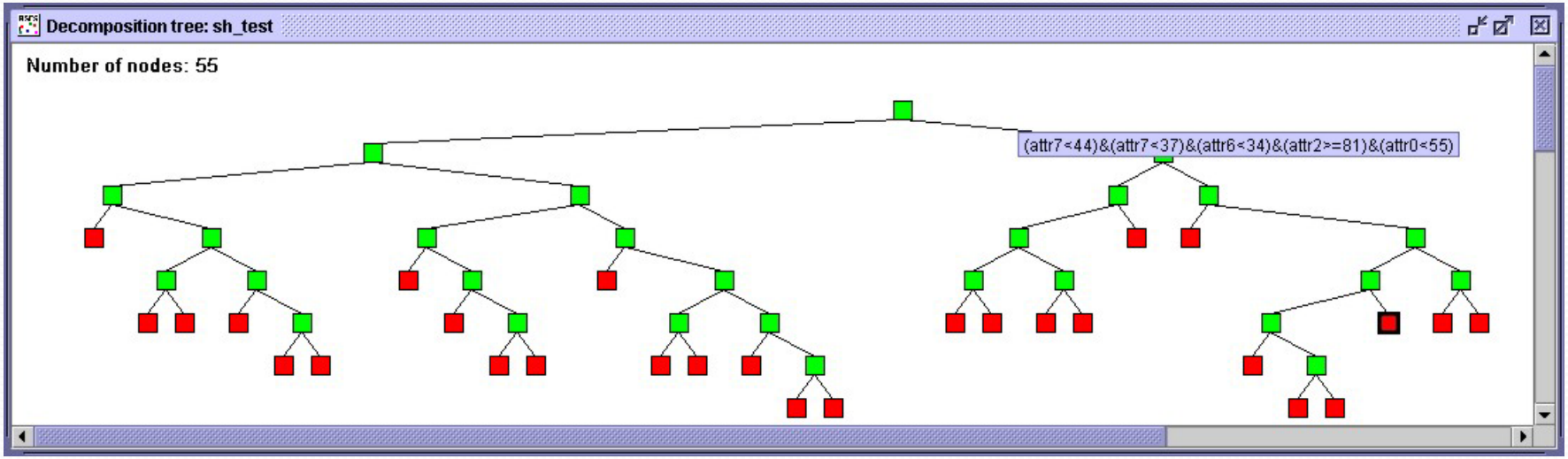
Discretization - example



Number of conflicts: **7**

Data decomposition with RS

Large data sets may not be possible to process as-is. The ability to decompose data set into smaller chunks is a requirement. These fragments, after decomposition represented as leafs in decomposition tree, are supposed to be more uniform and easier to cope with decision-wise.



Rough Sets vs. Others

How RS methods fit the bigger picture and how they compare with some popular DM techniques?

Rough Sets in Decision Tree Construction

Main problem: Search for minimal decision tree compatible with a given decision table.
This problem is NP-hard

Heuristics:

- Decision tree are constructed from a given set of candidate partitions;
- Best-first searching strategy, a quality measure must be defined, e.g..,
  - Entropy gain;
  - Gini's index;
- Rough set based measure:
  discernibility measure = number of resolved conflicts
- In our research: decision trees constructed by discernibility measure have many interesting properties.

Association Rule Generation

Association rule generation methods consist of two steps:

1. Generate as many large templates as possible:
   $T = D_1 \wedge D_2 \wedge \ldots \wedge D_k$
   i.e, $support(T) > s$ and $support(T \wedge D) < s$ for any descriptor D)
2. For any template T, search for a partition $T = P \wedge Q$ such that:
   - $support(P) < support(T) / c$
   - P is the smallest template satisfying the previous condition

Surprisingly: the second step can be solved by rough set methods (using α-reducts).

Rough Sets vs. k-NN

- How to define the measure function or neighbourhood?
- Similarity relation can be learned from data using rough set methods.
- A simple idea:
  the more reducts an attribute appears in the more important this attribute is.
- Filtering methods in rule based classifiers can simulate both decision tree and k-NN classifiers.

Multivariate Analysis

- Multiple Regression: analyzes the relationship between several attributes and the decision;
- Principal Components Analysis and Factor Analysis – A linear dimensionality reduction technique, which:
  - identifies orthogonal directions of maximum variance in the original data,
  - projects the data into a lower-dimensionality space formed of a sub-set of the highest-variance components.
- Discriminant Analysis: searching for the best set of attributes that discriminates objects from two or more decision classes.
- Cluster Analysis: grouping similar objects.

Multivariate Analysis & RS

- PCA and clustering methods can be applied as pre-processing step to rough set methods. e.g., to extract new features from the original set of attributes.
- Experimental results: they can improve the quality of rough set classifiers.

Rough Sets in Network Construction

A Bayesian network is an acyclic directed graph that models probabilistic dependencies between the domain variables:

Q: How to construct Bayesian networks from data?
In many cases, the problem is NP-hard.

A: Searching for structure + probability distribution;

RS: Structure can be reconstructed by calculating frequent reducts!

- Rough sets vs. neural networks:
- Rough sets vs. Petri net
- Rough sets vs. Belief (cause-and-effect) networks.

Software tools

How to do your own experiments with Rough Set methods?

Tools overview

- The past:
  - Rosetta,
  - RSES,
  - ROSE, 4eMka
- The present:
  - Rseslib et consortes
  - jRS, jMAF, jAMM
  - Debellor
  - TunedIT
- Related tools and perspectives for the future (one of possible scenarios)

Rosetta

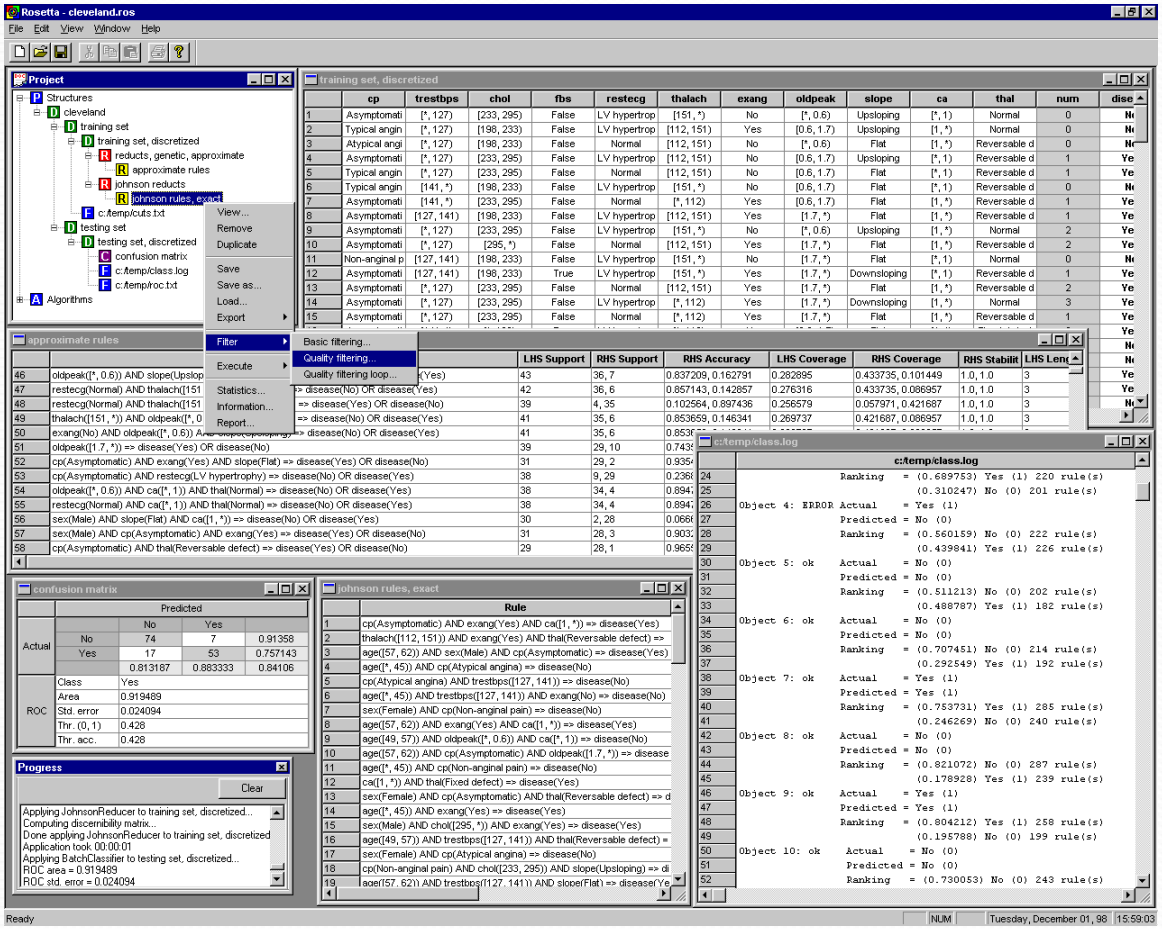
http://www.lcb.uu.se/tools/rosetta/

Supported OS:

Microsoft Windows

Partly Supported OS:

Linux



RSES

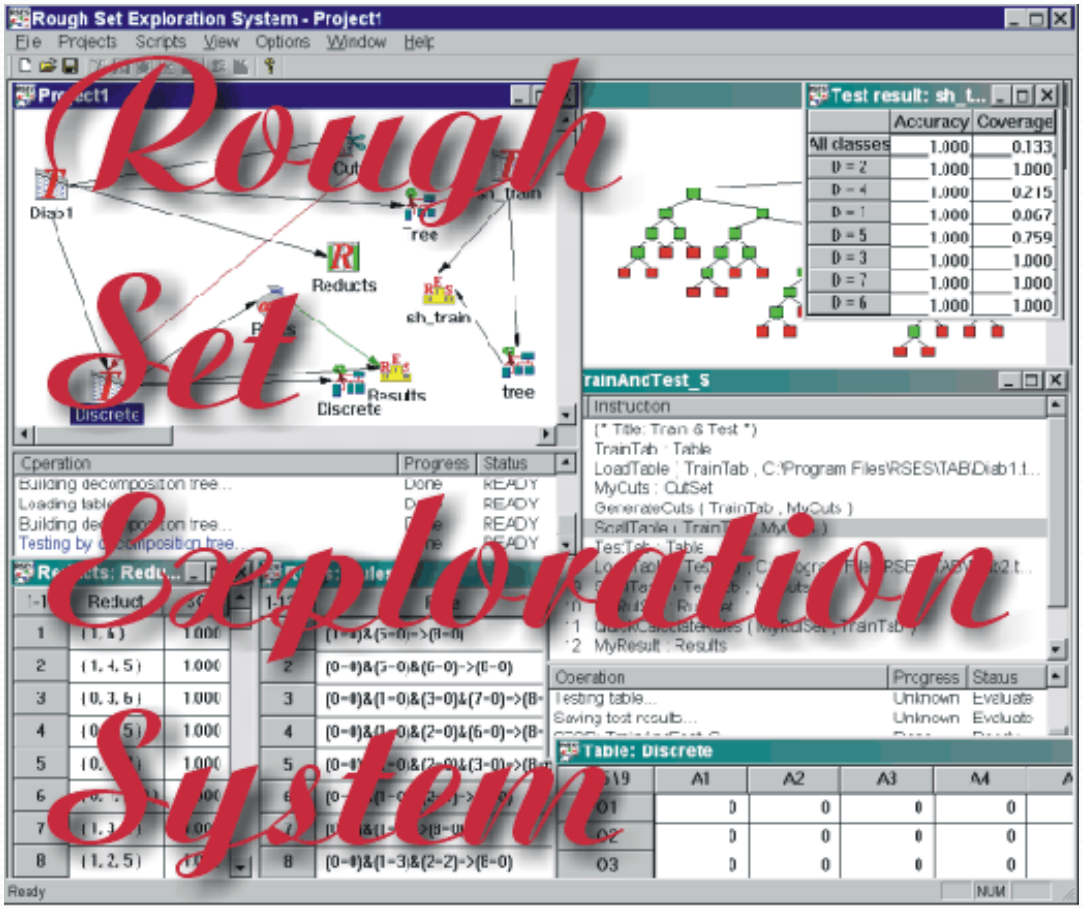
http://logic.mimuw.edu.pl/~rses/

Java™ POWERED

Supported OS:

Microsoft Windows

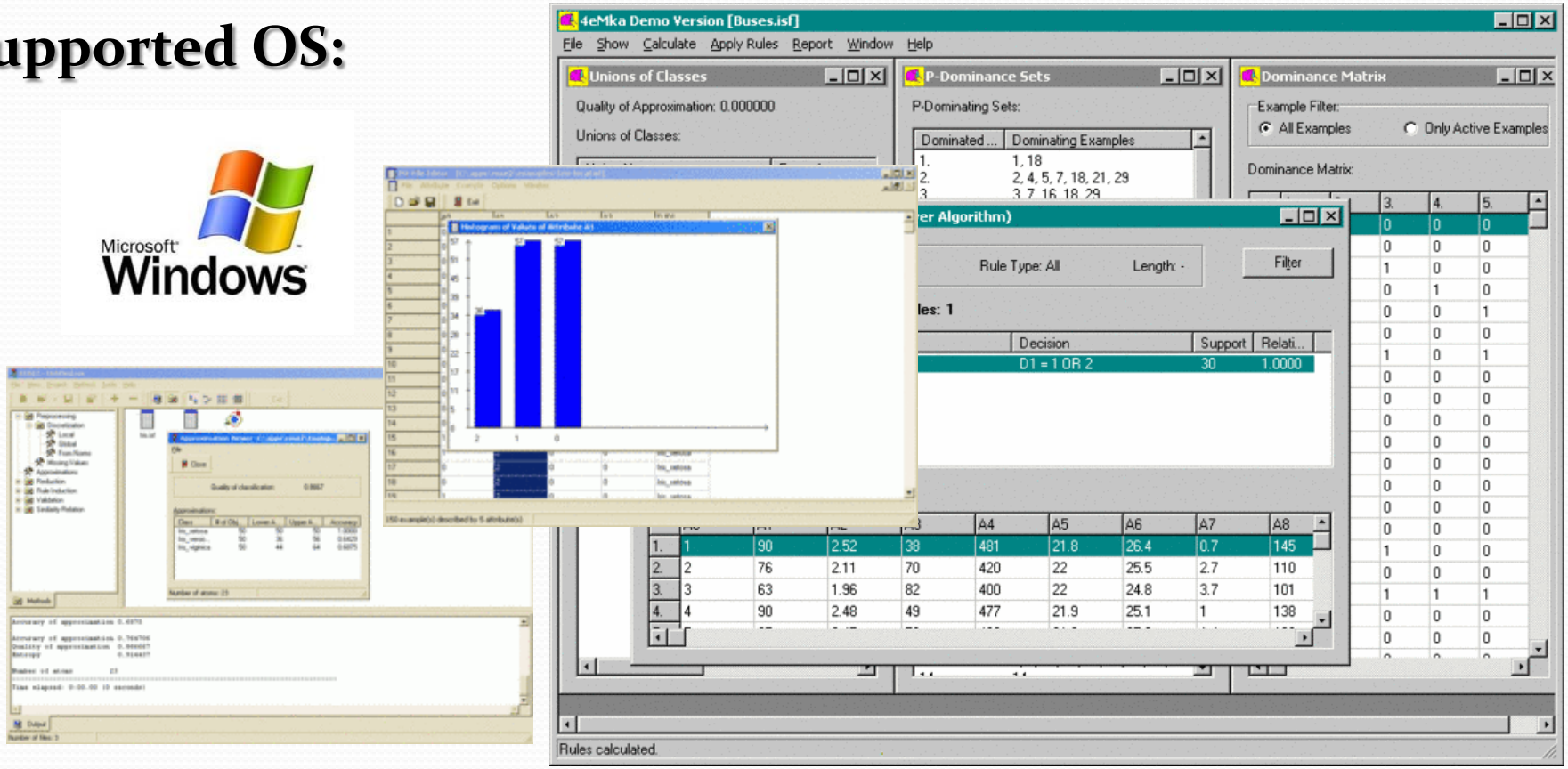
Linux



ROSE 2 & 4eMka2

http://idss.cs.put.poznan.pl/site/software.html

Supported OS:



Rseslib

http://rsproject.mimuw.edu.pl

- Open Source (GNU GPL) library of RS and related classifications algorithms and data structures, written in Java.
- Accepts CSV+, ARFF (Weka), and RSES formats.
- Several small GUI-based tools to make usage easier : Visual Rseslib, Qmak, Trickster.

jRS, jMAF, jAMM

http://idss.cs.put.poznan.pl/site/software.html

- jRS is a Java library implementing methods of analysis provided by the Dominance-based Rough Set Approach and Variable Consistency Dominance-based Rough Set Approach.
- jMAF is a Rough Set Data Analysis Framework written in Java language. It is based on jRS library.
- JAMM is a GUI decision support tool designed to guide the user through analysis and solving of multiple-criteria classification problems using jRS.

Debellor

http://www.debellor.org

- Debellor is an open source framework for scalable data mining and machine learning.
- The unique feature of Debellor is *data streaming*, which enables efficient processing of massive data and is essential for scalability of algorithms.
- You may implement your own algorithms in Debellor's architecture and combine them with pre-existing ones (over 100), in particular:
  - Rseslib algorithms;
  - Most of WEKA algorithms.

TunedIT platform

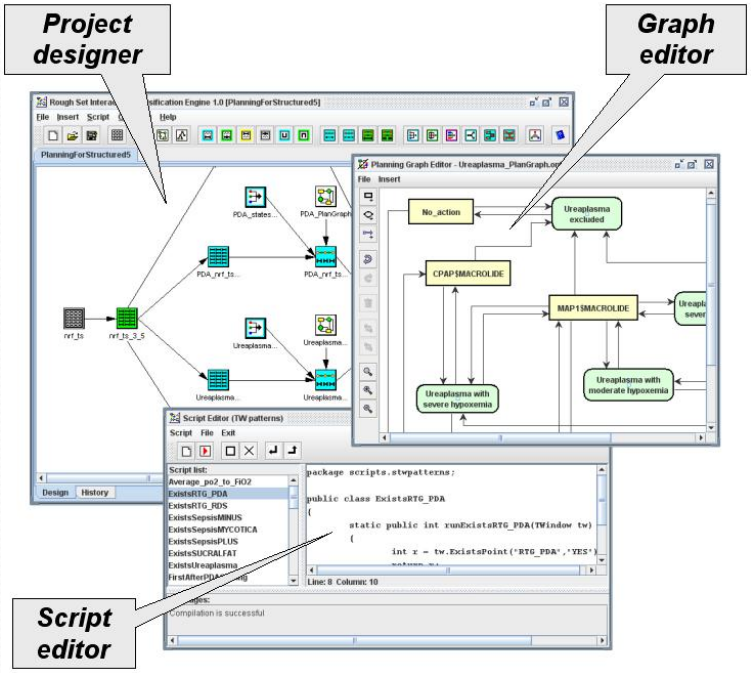
http://www.tunedit.org/

- TunedIT Research delivers web-based tools to help data mining scientists conduct repeatable experiments and easily evaluate data-driven algorithms.
- TunedIT Challenges platform for hosting data competitions - for educational, scientific and business purposes.
- Repository of data sets, methods and algorithms, including:
  - UCI, StatLog, KDD Cup, PROMISE, and other data sets;
  - Algorithms libraries: Debellor ,Weka, and Rseslib;
  - TunedTester framework and various examples.
- Knowledge base gathering various experimental results.

RoughICE

http://www.mimuw.edu.pl/~bazan/roughice/

- The Rough Set Interactive Classification Engine
- Software platform supporting the approximation of spatio-temporal complex concepts in the given concept ontology.
- Support for the dialogue with the user (domain expert).
- Design classifiers and knowledge flow for complex concepts.



Infobright

http://www.infobright.org/

- Column-oriented database (warehouse) engine.
- Suitable for large data thanks to compression methods.
- Supporting analytic processing of large amounts of data.
- Build around rough set concepts of approximation and supporting approximate SQL query answering.
- Based on MySQL architecture.
- Both Open Source (ICE) and commercial (IEE) versions available.

Thank you!

Questions and comments are welcome.

Bibliography and links

Where to look for more information?

- W. Frawley, G. Piatetsky-Shapiro, C. Matheus: *Knowledge Discovery in Databases: An Overview.* AI Magazine , Fall 1992, pgs 213-228.
- D. Hand, H. Mannila, P. Smyth: *Principles of Data Mining.* MIT Press, Cambridge, MA, 2001
- Z. Pawlak: *Rough Sets: theoretical aspects of reasoning about data*. Kluwer AP, Dordrecht, 1991
- L. Polkowski, A. Skowron (eds.): *Rough Sets in Knowledge Discovery*, part 1 and 2 Physica-Verlag, Heidelberg, 1998
- S.K. Pal, L. Polkowski, A. Skowron (eds.): *Rough-Neural Computing*. Springer-Verlag, Heidelberg, 2004

- A. Wojna: *Analogy-based Reasoning in Classifier Construction*. Ph. D. Thesis, Department of Mathematics, Informatics and Mechanics, The University of Warsaw, Warsaw, 2004
- J. Bazan, Nguyen H. S., Nguyen S. H., P. Synak, J. Wróblewski: *Rough set algorithms in classification problem*. In: L. Polkowski, S. Tsumoto, T.Y. Lin (eds.), *Rough Set Methods and Applications*, Physica-Verlag, Heidelberg, 2000, pp. 49–88
- Nguyen H.S., Nguyen S.H.: *Rough Sets and Association Rule Generation*. Fundamenta Informaticae, Vol. 40/4, 1999, pp. 310-318

Links

- International Rough Set Society
  http://www.roughsets.org
- Rough Set Database System (RSDS)
  http://rsds.univ.rzeszow.pl/
- KDNuggets
  http://www.kdnuggets.com/

Software

- Rosetta - http://www.lcb.uu.se/tools/rosetta/
- RSES - http://logic.mimuw.edu.pl/~rses/
- ROSE, 4eMka, jRS, JAMM, etc. - http://idss.cs.put.poznan.pl/site/software.html
- Rseslib - http://rsproject.mimuw.edu.pl
- Debellor - http://www.debellor.org
- TunedIT-http://www.tunedit.org/
- Infobright (ICE) – http://www.infobright.org