Rough sets in Data Science

Part 1: Basic rough set methods for data analysis

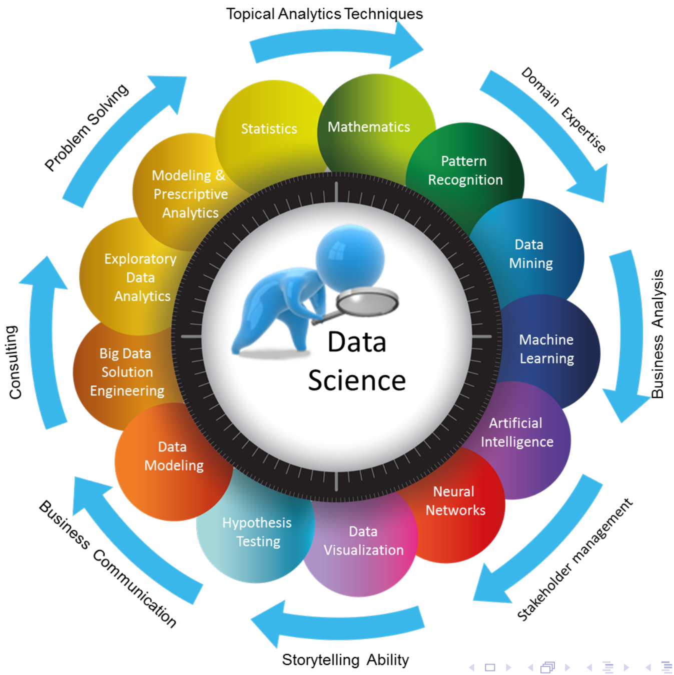
Nguyen Hung Son

University of Warsaw, Poland

2 grudnia 2016

Outline

Outline



Topical Analytics Techniques
Domain Expertise
Business Analysis
Stakeholder management
Storytelling Ability
Business Communication
Consulting
Problem Solving
Statistics
Mathematics
Pattern Recognition
Data Mining
Machine Learning
Artificial Intelligence
Neural Networks
Data Visualization
Hypothesis Testing
Data Modeling
Big Data Solution Engineering
Exploratory Data Analytics
Modeling & Prescriptive Analytics
Data Science

Data Science Process

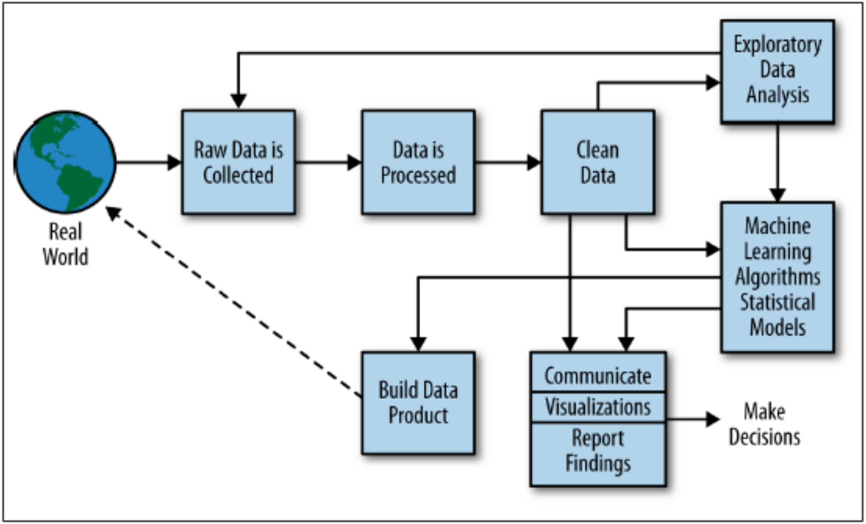
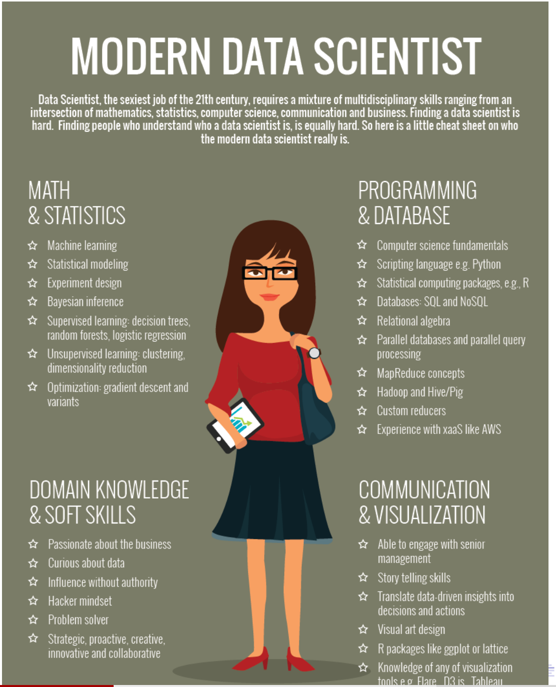


Figure 2-2. The data science process


MODERN DATA SCIENTIST
Data Scientist, the sexiest job of the 21th century, requires a mixture of multidisciplinary skills ranging from an intersection of mathematics, statistics, computer science, communication and business. Finding a data scientist is hard. Finding people who understand who a data scientist is, is equally hard. So here is a little cheat sheet on who the modern data scientist really is.
MATH & STATISTICS
Machine learning
Statistical modeling
Experiment design
Bayesian inference
Supervised learning: decision trees, random forests, logistic regression
Unsupervised learning: clustering, dimensionality reduction
Optimization: gradient descent and variants
PROGRAMMING & DATABASE
Computer science fundamentals
Scripting language e.g. Python
Statistical computing packages, e.g., R
Databases: SQL and NoSQL
Relational algebra
Parallel databases and parallel query processing
MapReduce concepts
Hadoop and Hive/Pig
Custom reducers
Experience with xaaS like AWS
DOMAIN KNOWLEDGE & SOFT SKILLS
Passionate about the business
Curious about data
Influence without authority
Hacker mindset
Problem solver
Strategic, proactive, creative, innovative and collaborative
COMMUNICATION & VISUALIZATION
Able to engage with senior management
Story telling skills
Translate data-driven insights into decisions and actions
Visual art design
R packages like ggplot or lattice
Knowledge of any of visualization tools e.g. Flare, D3.js, Tableau

The Need for Approximate Reasoning

Many tasks in data mining can be formulated as an approximate reasoning problem.

Assume that there are

- Two agents A_1 and A_2;

The Need for Approximate Reasoning

Many tasks in data mining can be formulated as an approximate reasoning problem.

Assume that there are

- Two agents A_1 and A_2;
- They are talking about objects from a common universe $\mathcal{U}$;

The Need for Approximate Reasoning

Many tasks in data mining can be formulated as an approximate reasoning problem.

Assume that there are

- Two agents A_1 and A_2;
- They are talking about objects from a common universe $\mathcal{U}$;
- They use different languages $\mathcal{L}_1$ and $\mathcal{L}_2$;

The Need for Approximate Reasoning

Many tasks in data mining can be formulated as an approximate reasoning problem.

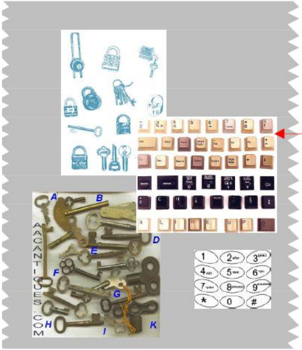
Assume that there are

- Two agents A_1 and A_2;
- They are talking about objects from a common universe $\mathcal{U}$;
- They use different languages $\mathcal{L}_1$ and $\mathcal{L}_2$;
- *Every formula ψ in $\mathcal{L}_1$ (and $\mathcal{L}_2$) describes a set C_ψ of objects from $\mathcal{U}$.*

Each agent, who wants to understand the other, should perform

- an approximation of concepts used by the other;
- an approximation of reasoning scheme, e.g., derivation laws;

Concept approximation problem



An universe of keys

TEACHER

$\mathcal{L}_1 = \{\text{keyboard, ...}\}$

LEARNER

$\mathcal{L}_2 = \{\text{black, brown, white, metal, plastic, ...}\}$

Each agent, who wants to understand the other, should perform

- an approximation of concepts used by the other;
- an approximation of reasoning scheme, e.g., derivation laws;

Classification Problem

Given

- A concept $C \subset \mathcal{U}$ used by teacher;
- A sample $U = U^+ \cup U^-$, where
  - $U^+ \subset C$: positive examples;
  - $U^- \subset \mathcal{U} \setminus C$: negative examples;
- Language $\mathcal{L}_2$ used by learner;

Goal

build an approximation of C in terms of $\mathcal{L}_2$

- with simple description;
- with high quality of approximation;
- using efficient algorithm.

Decision table
$\mathbb{S} = (U, A \cup \{dec\})$
describes training data set.

	a_1	a_2	...	dec
u_1	1	0	...	0
u_2	1	1	...	1
...	...	...	...	...
u_n	0	1	...	0

Clustering Problem

- **Original definition:** Division of data into groups of similar objects.

- **In terms of approximate reasoning:** Looking for approximation of a similarity relation (i.e., a concept of being similar):
  - Universe: the set of pairs of objects;
  - Teacher: a partial knowledge about similarity + optimization criteria;
  - Learner: describes the similarity relation using available features;

Association Discovery

- **Basket data analysis:** looking for approximation of customer behavior in terms of association rules;
  - Universe: the set of transactions;
  - Teacher: hidden behaviors of individual customers;
  - Learner: uses association rules to describe some common trends;

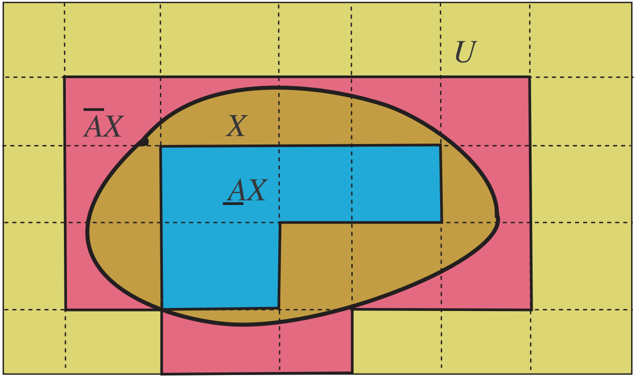
Association Discovery

- **Basket data analysis:** looking for approximation of customer behavior in terms of association rules;
  - Universe: the set of transactions;
  - Teacher: hidden behaviors of individual customers;
  - Learner: uses association rules to describe some common trends;

- **Time series data analysis:**
  - Universe: Sub-sequences obtained by windowing with all possible frame sizes.
  - Teacher: the actual phenomenon behind the collection of timed measurements, e.g., stock market, earth movements.
  - Learner: trends, variations, frequent episodes, extrapolation.

Rough set approach to Concept approximations

- Lower approximation – we are sure that these objects are in the set.
- Upper approximation - it is possible (likely, feasible) that these objects belong to our set (concept). They *roughly* belong to the set.



Generalized definition

Rough approximation of the concept C (induced by a sample X):

any pair $\mathbb{P} = (\mathbf{L}, \mathbf{U})$ satisfying the following conditions:

1. $\mathbf{L} \subseteq \mathbf{U} \subseteq \mathcal{U}$;
2. $\mathbf{L}, \mathbf{U}$ are subsets of $\mathcal{U}$ expressible in the language $\mathcal{L}_2$;
3. $\mathbf{L} \cap X \quad \subseteq \quad C \cap X \quad \subseteq \quad \mathbf{U} \cap X$;
4. $^{(*)}$ the set $\mathbf{L}$ is maximal (and $\mathbf{U}$ is minimal) in the family of sets definable in $\mathcal{L}$ satisfying (3).

Generalized definition

Rough approximation of the concept C (induced by a sample X):

any pair $\mathbb{P} = (\mathbf{L}, \mathbf{U})$ satisfying the following conditions:

1. $\mathbf{L} \subseteq \mathbf{U} \subseteq \mathcal{U}$;
2. $\mathbf{L}, \mathbf{U}$ are subsets of $\mathcal{U}$ expressible in the language $\mathcal{L}_2$;
3. $\mathbf{L} \cap X \quad \subseteq \quad C \cap X \quad \subseteq \quad \mathbf{U} \cap X$;
4. $^{(*)}$ the set $\mathbf{L}$ is maximal (and $\mathbf{U}$ is minimal) in the family of sets definable in $\mathcal{L}$ satisfying (3).

Rough membership function of concept C:

any function $f : \mathcal{U} \to [0, 1]$ such that the pair $(\mathbf{L}_f, \mathbf{U}_f)$, where

- $\mathbf{L}_f = \{x \in \mathcal{U} : f(x) = 1\}$ and
- $\mathbf{U}_f = \{x \in \mathcal{U} : f(x) > 0\}$.

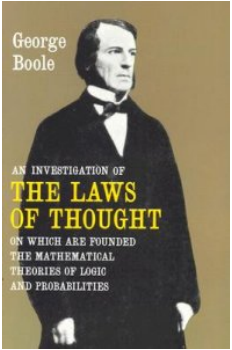
is a rough approximation of C (induced from sample U)

Example of Rough Set models

- **Standard rough sets defined by attributes:**
  - lower and upper approximation of X by attributes from B are defined by indiscernible classes.
- **Tolerance based rough sets:**
  - Using *tolerance* relation (also similarity relation) instead of indiscernibility relation.
- **Variable Precision Rough Sets (VPRS)**
  - allowing some admissible level $0 \leq \beta \leq 1$ of classification inaccuracy.
- **Generalized approximation space**

Outline

Boolean algebra in Computer Science



George Boole (1815-1864)

- George Boole was truly one of the founders of computer science;
- Boolean algebra was an attempt to use algebraic techniques to deal with expressions in the propositional calculus.
- Boolean algebras find many applications in electronic and computer design.
- They were first applied to switching by Claude Shannon in the 20th century.
- Boolean Algebra is also a convenient notation for representing Boolean functions.

Word Problem:

Madison has a pocket full of nickels and dimes.

- She has 4 more dimes than nickels.
- The total value of the dimes and nickels is $1.15.

How many dimes and nickels does she have?

Algebraic approach to problem solving

Word Problem:

Madison has a pocket full of nickels and dimes.

- She has 4 more dimes than nickels.
- The total value of the dimes and nickels is $1.15.

How many dimes and nickels does she have?

- **Problem modeling:**

$$N = \text{number of nickels}$$
$$D = \text{number of dimes}$$
$$D = N + 4$$
$$10D + 5N = 115$$

Algebraic approach to problem solving

Word Problem:

Madison has a pocket full of nickels and dimes.

- She has 4 more dimes than nickels.
- The total value of the dimes and nickels is $1.15.

How many dimes and nickels does she have?

- **Problem modeling:**

$$N = \text{number of nickels}$$
$$D = \text{number of dimes}$$
$$D = N + 4$$
$$10D + 5N = 115$$

- **Solving algebraic problem:**

$$\ldots \Rightarrow D = 9; N = 5$$

Algebraic approach to problem solving

Word Problem:

Madison has a pocket full of nickels and dimes.

- She has 4 more dimes than nickels.
- The total value of the dimes and nickels is \$1.15.

How many dimes and nickels does she have?

- **Problem modeling:**

$$N = \text{number of nickels}$$
$$D = \text{number of dimes}$$
$$D = N + 4$$
$$10D + 5N = 115$$

- **Solving algebraic problem:**

$$\ldots \Rightarrow D = 9; N = 5$$

- **Hura:** 9 dimes and 5 nickels!

Boolean Algebra:

a tuple

$$\mathcal{B} = (B, +, \cdot, 0, 1)$$

satisfying following axioms:

- **Commutative laws:**
  $(a + b) = (b + a)$ and
  $(a \cdot b) = (b \cdot a)$
- **Distributive laws:**
  $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
  $a + (b \cdot c) = (a + b) \cdot (a + c)$
- **Identity elements:**
  $a + 0 = a$ and $a \cdot 1 = a$
- **Complementary:**
  $a + \overline{a} = 1$ and $a \cdot \overline{a} = 0$

$$\mathcal{B}_2 = (\{0, 1\}, +, \cdot, 0, 1)$$

is the smallest, but the most important, model of general Boolean Algebra.

x	y	$x + y$	$x \cdot y$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

x	$\neg x$
0	1
1	0

Applications:

- circuit design;
- propositional calculus;

Boolean Algebra:

a tuple

$$\mathcal{B} = (B, +, \cdot, 0, 1)$$

satisfying following axioms:

- **Commutative laws:**
  $(a+b) = (b+a)$ and
  $(a \cdot b) = (b \cdot a)$
- **Distributive laws:**
  $a \cdot (b+c) = (a \cdot b) + (a \cdot c)$
  $a + (b \cdot c) = (a+b) \cdot (a+c)$
- **Identity elements:**
  $a + 0 = a$ and $a \cdot 1 = a$
- **Complementary:**
  $a + \overline{a} = 1$ and $a \cdot \overline{a} = 0$

Binary Boolean algebra

$$\mathcal{B}_2 = (\{0,1\}, +, \cdot, 0, 1)$$

is the smallest, but the most important, model of general Boolean Algebra.

x	y	$x+y$	$x \cdot y$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

x	$\neg x$
0	1
1	0

Applications:

- circuit design;
- propositional calculus;

Boolean function

- Any function $\boxed{f : \{0,1\}^n \to \{0,1\}}$ is called *a Boolean function*;

x	y	z	f
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Boolean function

- Any function $\boxed{f : \{0,1\}^n \to \{0,1\}}$ is called *a Boolean function*;
- *An implicant* of function f is a term $\boxed{t = x_1 ... x_m \overline{y_1} ... \overline{y_k}}$ such that

$$\forall_{x_1,...,x_n} t(x_1, ..., x_n) = 1 \Rightarrow f(x_1, ..., x_n) = 1$$

x	y	z	f
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Boolean function

- Any function $\boxed{f : \{0,1\}^n \to \{0,1\}}$ is called *a Boolean function*;
- *An implicant* of function f is a term $\boxed{t = x_1 ... x_m \overline{y_1} ... \overline{y_k}}$ such that

$$\forall_{x_1,...,x_n} t(x_1,...,x_n) = 1 \Rightarrow f(x_1,...,x_n) = 1$$

- *Prime implicant:* an implicant that ceases to be so if any of its literal is removed.

x	y	z	f
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Boolean function

- Any function $\boxed{f : \{0,1\}^n \to \{0,1\}}$ is called *a Boolean function*;
- *An implicant* of function f is a term $\boxed{t = x_1 ... x_m \overline{y_1} ... \overline{y_k}}$ such that

$$\forall_{x_1,...,x_n} t(x_1,...,x_n) = 1 \Rightarrow f(x_1,...,x_n) = 1$$

- *Prime implicant:* an implicant that ceases to be so if any of its literal is removed.

A Boolean function can be represented by many Boolean formulas;

- $\phi_1 = xy\overline{z} + x\overline{y}z + \overline{x}yz + xyz$

x	y	z	f
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Boolean function

- Any function $\boxed{f : \{0,1\}^n \to \{0,1\}}$ is called *a Boolean function*;
- *An implicant* of function f is a term $\boxed{t = x_1 ... x_m \overline{y_1} ... \overline{y_k}}$ such that

$$\forall_{x_1,...,x_n} t(x_1,...,x_n) = 1 \Rightarrow f(x_1,...,x_n) = 1$$

- *Prime implicant:* an implicant that ceases to be so if any of its literal is removed.

A Boolean function can be represented by many Boolean formulas;

- $\phi_1 = xy\overline{z} + x\overline{y}z + \overline{x}yz + xyz$
- $\phi_2 = (x+y+z)(\overline{x}+y+z)(x+\overline{y}+z)(x+y+\overline{z})$

x	y	z	f
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Boolean function

- Any function $\boxed{f : \{0,1\}^n \to \{0,1\}}$ is called *a Boolean function*;
- *An implicant* of function f is a term $\boxed{t = x_1 ... x_m \overline{y_1} ... \overline{y_k}}$ such that

$$\forall_{x_1,...,x_n} t(x_1,...,x_n) = 1 \Rightarrow f(x_1,...,x_n) = 1$$

- *Prime implicant:* an implicant that ceases to be so if any of its literal is removed.

A Boolean function can be represented by many Boolean formulas;

- $\phi_1 = xy\overline{z} + x\overline{y}z + \overline{x}yz + xyz$
- $\phi_2 = (x+y+z)(\overline{x}+y+z)(x+\overline{y}+z)(x+y+\overline{z})$
- $\phi_3 = xy + xz + yz$

x	y	z	f
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Boolean function

- Any function $\boxed{f : \{0,1\}^n \to \{0,1\}}$ is called *a Boolean function*;
- *An implicant* of function f is a term $\boxed{t = x_1 ... x_m \overline{y_1} ... \overline{y_k}}$ such that

$$\forall_{x_1,...,x_n} t(x_1,...,x_n) = 1 \Rightarrow f(x_1,...,x_n) = 1$$

- *Prime implicant:* an implicant that ceases to be so if any of its literal is removed.

A Boolean function can be represented by many Boolean formulas;

- $\phi_1 = xy\overline{z} + x\overline{y}z + \overline{x}yz + xyz$
- $\phi_2 = (x+y+z)(\overline{x}+y+z)(x+\overline{y}+z)(x+y+\overline{z})$
- $\phi_3 = xy + xz + yz$
- $xy\overline{z}$ is an implicant

x	y	z	f
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Boolean function

- Any function $\boxed{f : \{0,1\}^n \rightarrow \{0,1\}}$ is called *a Boolean function*;
- *An implicant* of function f is a term $\boxed{t = x_1 ... x_m \overline{y_1} ... \overline{y_k}}$ such that

$$\forall_{x_1,...,x_n} t(x_1,...,x_n) = 1 \Rightarrow f(x_1,...,x_n) = 1$$

- *Prime implicant:* an implicant that ceases to be so if any of its literal is removed.

A Boolean function can be represented by many Boolean formulas;

- $\phi_1 = xy\overline{z} + x\overline{y}z + \overline{x}yz + xyz$
- $\phi_2 = (x+y+z)(\overline{x}+y+z)(x+\overline{y}+z)(x+y+\overline{z})$
- $\phi_3 = xy + xz + yz$
- $xy\overline{z}$ is an implicant
- xy is a prime implicant

x	y	z	f
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Boolean Reasoning Approach

Theorem (Blake Canonical Form)

A Boolean function can be represented as a disjunction of all of its prime implicants: $f = t_1 + t_2 + ... + t_k$

Boolean Reasoning Approach

Theorem (Blake Canonical Form)

A Boolean function can be represented as a disjunction of all of its prime implicants: $f = t_1 + t_2 + ... + t_k$

Boolean Reasoning Schema

1. **Modeling:** Represent the problem by a collection of Boolean equations
2. **Reduction:** Condense the equations into a single Boolean equation

$$f = 0 \quad \text{or} \quad f = 1$$

3. **Development:** Construct the Blake Canonical form, i.e., generate the prime implicants of f
4. **Reasoning:** Apply a sequence of reasoning to solve the problem

Boolean Reasoning – Example

Problem:

A, B, C, D are considering going to a party. Social constrains:

- If A goes than B won't go and C will;
- If B and D go, then either A or C (but not both) will go
- If C goes and B does not, then D will go but A will not.

Boolean Reasoning – Example

Problem:

A, B, C, D are considering going to a party. Social constrains:

- If A goes than B won't go and C will;
- If B and D go, then either A or C (but not both) will go
- If C goes and B does not, then D will go but A will not.

Problem modeling:

$$
\begin{aligned}
A \to \overline{B} \wedge C \rightsquigarrow A(B + \overline{C}) &= 0 \\
\ldots \rightsquigarrow BD(AC + \overline{AC}) &= 0 \\
\ldots \rightsquigarrow \overline{B}C(A + \overline{D}) &= 0
\end{aligned}
$$

Boolean Reasoning – Example

Problem:

A, B, C, D are considering going to a party. Social constrains:

- If A goes than B won't go and C will;
- If B and D go, then either A or C (but not both) will go
- If C goes and B does not, then D will go but A will not.

Problem modeling:

$$
\begin{aligned}
A \to \overline{B} \wedge C \leftrightsquigarrow A(B+\overline{C}) &= 0 \\
\ldots \leftrightsquigarrow BD(AC+\overline{AC}) &= 0 \\
\ldots \leftrightsquigarrow \overline{B}C(A+\overline{D}) &= 0
\end{aligned}
$$

- **After reduction:** $f = A(B+\overline{C}) + BD(AC+\overline{AC}) + \overline{B}C(A+\overline{D}) = 0$

Boolean Reasoning – Example

Problem:

A, B, C, D are considering going to a party. Social constrains:

- If A goes than B won't go and C will;
- If B and D go, then either A or C (but not both) will go
- If C goes and B does not, then D will go but A will not.

Problem modeling:

$$
\begin{aligned}
A \rightarrow \overline{B} \wedge C \rightsquigarrow A(B+\overline{C}) &= 0 \\
\ldots \rightsquigarrow BD(AC+\overline{AC}) &= 0 \\
\ldots \rightsquigarrow \overline{B}C(A+\overline{D}) &= 0
\end{aligned}
$$

- **After reduction:** $f = A(B+\overline{C}) + BD(AC + \overline{AC}) + \overline{B}C(A+\overline{D}) = 0$
- **Blake Canonical form:** $f = B\overline{C}D + \overline{B}C\overline{D} + A = 0$

Boolean Reasoning – Example

Problem:

A, B, C, D are considering going to a party. Social constrains:

- If A goes than B won't go and C will;
- If B and D go, then either A or C (but not both) will go
- If C goes and B does not, then D will go but A will not.

Problem modeling:

$$
\begin{aligned}
A \rightarrow \overline{B} \wedge C \leadsto A(B+\overline{C}) &= 0 \\
\ldots \leadsto BD(AC+\overline{A}\overline{C}) &= 0 \\
\ldots \leadsto \overline{B}C(A+\overline{D}) &= 0
\end{aligned}
$$

- **After reduction:** $f = A(B+\overline{C}) + BD(AC + \overline{A}\overline{C}) + \overline{B}C(A+\overline{D}) = 0$
- **Blake Canonical form:** $f = B\overline{C}D + \overline{B}C\overline{D} + A = 0$
- **Facts:**

$$
\begin{aligned}
BD &\longrightarrow C \\
C &\longrightarrow B + D \\
A &\longrightarrow 0
\end{aligned}
$$

Boolean Reasoning – Example

Problem:

A, B, C, D are considering going to a party. Social constrains:

- If A goes than B won't go and C will;
- If B and D go, then either A or C (but not both) will go
- If C goes and B does not, then D will go but A will not.

Problem modeling:

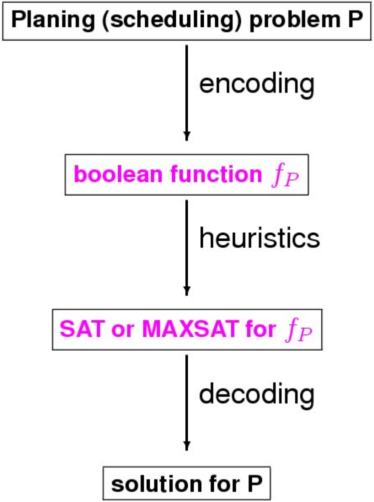
$$\begin{aligned} A \to \overline{B} \wedge C \rightsquigarrow A(B+\overline{C}) &= 0 \\ \ldots \rightsquigarrow BD(AC+\overline{AC}) &= 0 \\ \ldots \rightsquigarrow \overline{B}C(A+\overline{D}) &= 0 \end{aligned}$$

- **After reduction:** $f = A(B+\overline{C}) + BD(AC + \overline{AC}) + \overline{B}C(A+\overline{D}) = 0$
- **Blake Canonical form:** $f = B\overline{C}D + \overline{B}C\overline{D} + A = 0$
- **Facts:**

$$\begin{aligned} BD &\longrightarrow C \\ C &\longrightarrow B + D \\ A &\longrightarrow 0 \end{aligned}$$

- **Reasoning:** (theorem proving) e.g., show that
  "C cannot go alone."

Boolean reasoning for decision problems

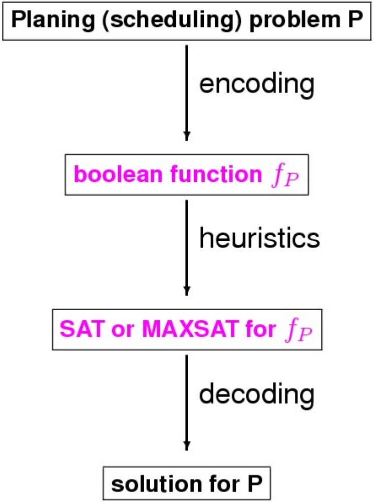


- SAT: whether an equation

$$f(x_1, ..., x_n) = 1$$

has a solution?

Boolean reasoning for decision problems

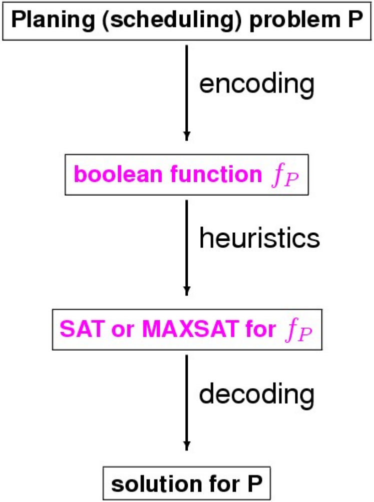


- SAT: whether an equation

$$f(x_1, ..., x_n) = 1$$

  has a solution?
- SAT is the first problem which has been proved to be NP-complete (the Cook's theorem).

Boolean reasoning for decision problems

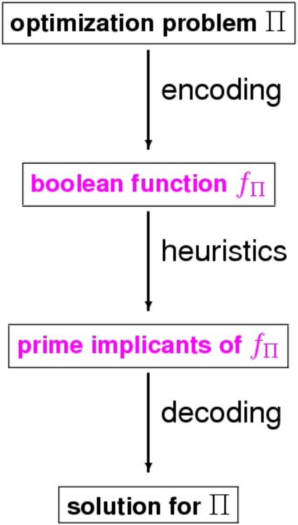


- SAT: whether an equation

  $$f(x_1, ..., x_n) = 1$$

  has a solution?
- SAT is the first problem which has been proved to be NP-complete (the Cook's theorem).
- E.g., scheduling problem may be solved by SAT-solver.

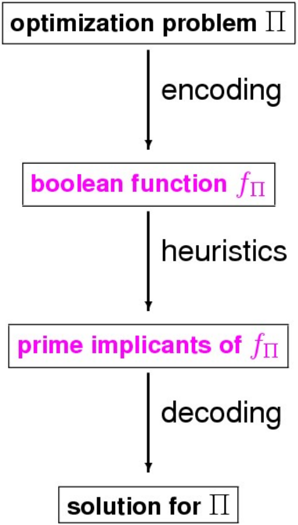
Boolean reasoning for optimization problems



- A function $\phi : \{0,1\}^n \to \{0,1\}$ is "*monotone*" if

$$\forall_{\mathbf{x},\mathbf{y}\in\{0,1\}^n} (\mathbf{x} \leqslant \mathbf{y}) \Rightarrow (\phi(\mathbf{x}) \leqslant \phi(\mathbf{y}))$$

Boolean reasoning for optimization problems

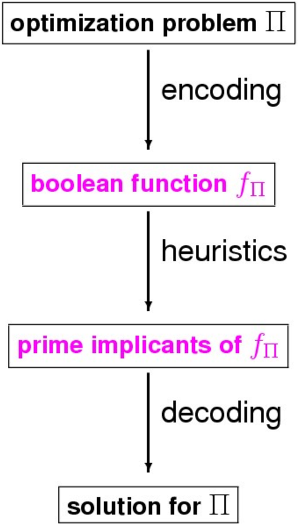


- A function $\phi : \{0,1\}^n \rightarrow \{0,1\}$ is "*monotone*" if

$$\forall_{\mathbf{x},\mathbf{y}\in\{0,1\}^n} (\mathbf{x} \leqslant \mathbf{y}) \Rightarrow (\phi(\mathbf{x}) \leqslant \phi(\mathbf{y}))$$

- Monotone functions can be represented by a boolean expression without negations.

Boolean reasoning for optimization problems



- A function $\phi : \{0,1\}^n \to \{0,1\}$ is "*monotone*" if

$$\forall_{\mathbf{x},\mathbf{y}\in\{0,1\}^n} (\mathbf{x} \leqslant \mathbf{y}) \Rightarrow (\phi(\mathbf{x}) \leqslant \phi(\mathbf{y}))$$

- Monotone functions can be represented by a boolean expression without negations.
- **Minimal Prime Implicant Problem:**

input: Monotone Boolean function f of n variables.

output: A prime implicant of f with the minimal length.

is NP-hard.

Heuristics for minimal prime implicants

Example

$f = (x_1 + x_2 + x_3)(x_2 + x_4)(x_1 + x_3 + x_5)(x_1 + x_5)(x_4 + x_6)$

The prime implicant can be treated as a set covering problem.

1. **Greedy algorithm:** In each step, select the variable that most frequently occurs within clauses
2. **Linear programming:** Convert the given function into a system of linear inequations and applying the Integer Linear Programming (ILP) approach to this system.
3. **Evolutionary algorithms:**
   The search space consists of all subsets of variables
   the cost function for a subset X of variables is defined by (1) the number of clauses that are uncovered by X, and (2) the size of X,

Boolean Reasoning Approach to Rough sets

- Reduct calculation;
- Decision rule generation;
- Real value attribute discretization;
- Symbolic value grouping;
- Hyperplanes and new direction creation;

Outline

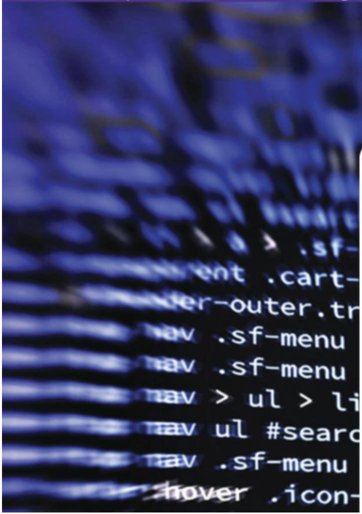
Reduction

- **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 evolutionary computation.
- Overfitting can be avoided by considering several reducts, pruning rules and lessening discernibility constraints.

Special Theme: Machine Learning



Introduction to the Special Theme

Modern Machine Learning: More with Less, Cheaper and Better

by Sander Bohte and Hung Son Nguyen

While the discipline of machine learning is often conflated with the general field of AI, machine learning specifically is concerned with the question of how to program computers to automatically recognise complex patterns and make intelligent decisions based on data. This includes such diverse approaches as probability theory, logic, combinatorial optimisation, search, statistics, reinforcement learning and control theory. In this day and age with an abundance of sensors and computers, applications are ubiquitous, ranging from vision to language processing, forecasting, pattern recognition, games, data mining, expert systems and robotics.

What is a reduct?

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

- Given an information system $\mathbb{S} = (U, A)$ and a monotone evaluation function

$$\mu_{\mathbb{S}} : \mathcal{P}(A) \longrightarrow \Re^{+}$$

- The set $B \subset A$ is called *μ-reduct*, if
  - $\mu(B) = \mu(A)$,
  - for any proper subset $B' \subset B$ we have $\mu(B') < \mu(B)$;
- The set $B \subset A$ is called *approximated reduct*, if
  - $\mu(B) \geq \mu(A) - \varepsilon$,
  - for any proper subset ...

Example

- Consider the *playing tennis* decision table
- Let us try to predict the decision for last two objects
- RS methodology:
  - Reduct calculation
  - Rule calculation
  - Matching
  - Voting

$\mathbb{A}$	a_1	a_2	a_3	a_4	dec
ID	outlook	temp.	hum.	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	?
14	rainy	mild	high	TRUE	?

Example: Decision reduct

$\mathbb{A}$	a_1	a_2	a_3	a_4	dec
ID	outlook	temp.	hum.	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	?
14	rainy	mild	high	TRUE	?

Methodology

1. Discernibility matrix;
2. Discernibility Boolean function
3. Prime implicants $\Longrightarrow$ reducts

Example: Decision reduct

$\mathbb{A}$	a_1	a_2	a_3	a_4	dec
ID	outlook	temp.	hum.	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	?
14	rainy	mild	high	TRUE	?

Methodology

1. Discernibility matrix;
2. Discernibility Boolean function
3. Prime implicants $\Longrightarrow$ reducts

Example: Decision reduct

$\mathbb{A}$	a_1	a_2	a_3	a_4	dec
ID	outlook	temp.	hum.	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	?
14	rainy	mild	high	TRUE	?

Methodology

1. Discernibility matrix;
2. Discernibility Boolean function
3. Prime implicants $\Longrightarrow$ reducts

Discernibility matrix;

$\mathbb{M}$	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

Example: Decision reduct

M	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

$$
\begin{aligned}
f = &(\alpha_1)(\alpha_1 + \alpha_4)(\alpha_1 + \alpha_2)(\alpha_1 \vee \alpha_2 + \alpha_3 + \alpha_4) \\
&(\alpha_1 + \alpha_2 + \alpha_4)(\alpha_2 + \alpha_3 + \alpha_4)(\alpha_1 + \alpha_2 + \alpha_3) \\
&(\alpha_4)(\alpha_2 + \alpha_3)(\alpha_2 + \alpha_4)(\alpha_1 + \alpha_3)(\alpha_3 + \alpha_4)
\end{aligned}
$$

Example: Decision reduct

M	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

$$
\begin{aligned}
f =&(\alpha_1)(\alpha_1+\alpha_4)(\alpha_1+\alpha_2)(\alpha_1 \vee \alpha_2+\alpha_3+\alpha_4)\\
&(\alpha_1+\alpha_2+\alpha_4)(\alpha_2+\alpha_3+\alpha_4)(\alpha_1+\alpha_2+\alpha_3)\\
&(\alpha_4)(\alpha_2+\alpha_3)(\alpha_2+\alpha_4)(\alpha_1+\alpha_3)(\alpha_3+\alpha_4)
\end{aligned}
$$

- simplifying the function by *absorbtion law* (i.e. $p \wedge (p+q) \equiv p$):

$$f \quad = \quad (\alpha_1)(\alpha_4)(\alpha_2+\alpha_3)$$

Example: Decision reduct

M	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

$$
\begin{aligned}
f = &(\alpha_1)(\alpha_1 + \alpha_4)(\alpha_1 + \alpha_2)(\alpha_1 \vee \alpha_2 + \alpha_3 + \alpha_4) \\
&(\alpha_1 + \alpha_2 + \alpha_4)(\alpha_2 + \alpha_3 + \alpha_4)(\alpha_1 + \alpha_2 + \alpha_3) \\
&(\alpha_4)(\alpha_2 + \alpha_3)(\alpha_2 + \alpha_4)(\alpha_1 + \alpha_3)(\alpha_3 + \alpha_4)
\end{aligned}
$$

- simplifying the function by *absorbtion law* (i.e. $p \wedge (p + q) \equiv p$):

$$f = (\alpha_1)(\alpha_4)(\alpha_2 + \alpha_3)$$

- Transformation from CNF to DNF: $f = \alpha_1\alpha_4\alpha_2 + \alpha_1\alpha_4\alpha_3$

Example: Decision reduct

M	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

$$
\begin{aligned}
f = &(\alpha_1)(\alpha_1+\alpha_4)(\alpha_1+\alpha_2)(\alpha_1 \vee \alpha_2+\alpha_3+\alpha_4)\\
&(\alpha_1+\alpha_2+\alpha_4)(\alpha_2+\alpha_3+\alpha_4)(\alpha_1+\alpha_2+\alpha_3)\\
&(\alpha_4)(\alpha_2+\alpha_3)(\alpha_2+\alpha_4)(\alpha_1+\alpha_3)(\alpha_3+\alpha_4)
\end{aligned}
$$

- simplifying the function by *absorbtion law* (i.e. $p \wedge (p+q) \equiv p$):

$$f = (\alpha_1)(\alpha_4)(\alpha_2+\alpha_3)$$

- Transformation from CNF to DNF: $f = \alpha_1\alpha_4\alpha_2 + \alpha_1\alpha_4\alpha_3$
- Each component corresponds to a reduct: $R_1 = \{a_1, a_2, a_4\}$ and $R_2 = \{a_1, a_3, a_4\}$

Outline

Boolean reasoning approach

- Reducts
- Decision rules
- Discretization
- Feature selection and Feature extraction

Outline

Example: Decision Rule Extraction

$\mathbb{M}$	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

$$f_{u_3} = (\alpha_1)(\alpha_1 \vee \alpha_4)(\alpha_1 \vee \alpha_2 \vee \alpha_3 \vee \alpha_4)(\alpha_1 \vee \alpha_2) = \alpha_1$$

Decision rule:

$$(a_1 = \text{overcast}) \implies dec = \text{no}$$

Example: Decision Rule Extraction

$\mathbb{M}$	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

$$\begin{aligned} f_{u_8} &= (\alpha_1+\alpha_2)(\alpha_1)(\alpha_1+\alpha_2+\alpha_3)(\alpha_1+\alpha_2+\alpha_3+\alpha_4)(\alpha_2+\alpha_3) \\ &\quad (\alpha_1+\alpha_3)(\alpha_3+\alpha_4)(\alpha_1+\alpha_4) \\ &= \alpha_1(\alpha_2+\alpha_3)(\alpha_3 \vee \alpha_4) = \alpha_1\alpha_3 + \alpha_1\alpha_2\alpha_4 \end{aligned}$$

Decision rules:

- $(a_1 = \text{sunny}) \wedge (a_3 = \text{high}) \implies dec = \text{no}$
- $(a_1 = \text{sunny}) \wedge (a_2 = \text{mild}) \wedge (a_4 = FALSE) \implies dec = \text{no}$

Example: all conssistent decision rules

Rid	Condition ⇒	Decision	supp.
1	outlook(overcast)⇒	yes	4
2	humidity(normal) AND windy(FALSE)⇒	yes	4
3	outlook(sunny) AND humidity(high)⇒	no	3
4	outlook(rainy) AND windy(FALSE)⇒	yes	3
5	outlook(sunny) AND temp.(hot)⇒	no	2
6	outlook(rainy) AND windy(TRUE)⇒	no	2
7	outlook(sunny) AND humidity(normal)⇒	yes	2
8	temp.(cool) AND windy(FALSE)⇒	yes	2
9	temp.(mild) AND humidity(normal)⇒	yes	2
10	temp.(hot) AND windy(TRUE)⇒	no	1
11	outlook(sunny) AND temp.(mild) AND windy(FALSE)⇒	no	1
12	outlook(sunny) AND temp.(cool)⇒	yes	1
13	outlook(sunny) AND temp.(mild) AND windy(TRUE)⇒	yes	1
14	temp.(hot) AND humidity(normal)⇒	yes	1

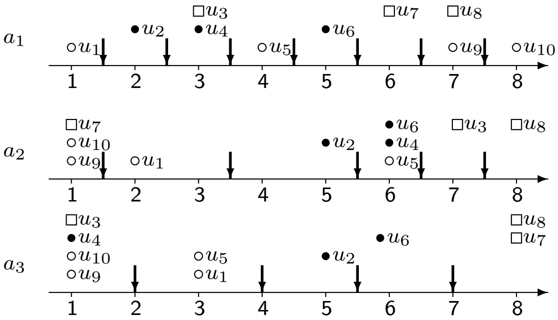
Outline

Discretization problem

Given a decision table $\mathbb{S} = (U, A \cup \{d\})$ where

$$U = \{x_1, \ldots, x_n\};\ A = \{a_1, ..., a_k : U \to \Re\} \text{ and } d : U \to \{1, ..., r(d)\}$$

A	a_1	a_2	a_3	d
u_1	1.0	2.0	3.0	0
u_2	2.0	5.0	5.0	1
u_3	3.0	7.0	1.0	2
u_4	3.0	6.0	1.0	1
u_5	4.0	6.0	3.0	0
u_6	5.0	6.0	5.0	1
u_7	6.0	1.0	8.0	2
u_8	7.0	8.0	8.0	2
u_9	7.0	1.0	1.0	0
u_{10}	8.0	1.0	1.0	0



Discretization problem

- A cut (a, c) on an attribute $a \in A$ discerns a pair of objects $x, y \in U$ if
$$(a(x) - c)(a(y) - c) < 0.$$
- A set of cuts $\mathbf{C}$ is consistent with $\mathbb{S}$ (or $\mathbb{S}$–consistent, for short) if and only if for any pair of objects $x, y \in U$ such that $dec(x) = dec(y)$, the following condition holds:

  IF x, y are discernible by $\mathbb{S}$ **THEN** x, y are discernible by $\mathbf{C}$.
- The consistent set of cuts $\mathbf{C}$ is called *irreducible* iff $\mathbf{Q}$ is not consistent for any proper subset $\mathbf{Q} \subset \mathbf{C}$.
- The consistent set of cuts $\mathbf{C}$ is called it optimal iff $card(C) \leq card(Q)$ for any consistent set of cuts $\mathbf{Q}$.

Discretization problem

OptiDisc: optimal discretization problem
input: A decision table $\mathbb{S}$.
output: $\mathbb{S}$-optimal set of cuts.

The corresponding decision problem can be formulated as:

DiscSize: k-cuts discretization problem
input: A decision table $\mathbb{S}$ and an integer k.
question: Decide whether there exists a $\mathbb{S}$-irreducible set of cuts $\mathbf{P}$ such that $card(\mathbf{P}) < k$.

Theorem

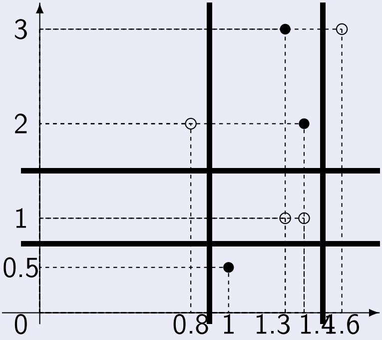
Computational complexity of discretization problems

- *The problem* DiscSize *is NP–complete.*
- *The problem* OptiDisc *is NP–hard.*

Boolean reasoning method for discretization

Example of a consistent set of cuts

$\mathbb{S}$	a	b	d
u_1	0.8	2	1
u_2	1	0.5	0
u_3	1.3	3	0
u_4	1.4	1	1
u_5	1.4	2	0
u_6	1.6	3	1
u_7	1.3	1	1



$$\mathbf{C} = \{(a;0.9), (a;1.5), (b;0.75), (b;1.5)\}$$

The discernibility formulas $\psi_{i,j}$ for different pairs (u_i, u_j) of objects:

$$\begin{array}{ll}
\psi_{2,1} = p_1^a + p_1^b + p_2^b; & \psi_{2,4} = p_2^a + p_3^a + p_1^b; \\
\psi_{2,6} = p_2^a + p_3^a + p_4^a + p_1^b + p_2^b + p_3^b; & \psi_{2,7} = p_2^a + p_1^b; \\
\psi_{3,1} = p_1^a + p_2^a + p_3^b; & \psi_{3,4} = p_2^a + p_2^b + p_3^b; \\
\psi_{3,6} = p_3^a + p_4^a; & \psi_{3,7} = p_2^b + p_3^b; \\
\psi_{5,1} = p_1^a + p_2^a + p_3^a; & \psi_{5,4} = p_2^b; \\
\psi_{5,6} = p_4^a + p_3^b; & \psi_{5,7} = p_3^a + p_2^b.
\end{array}$$

The discernibility formula $\Phi_{\mathbb{S}}$ in CNF form is given by

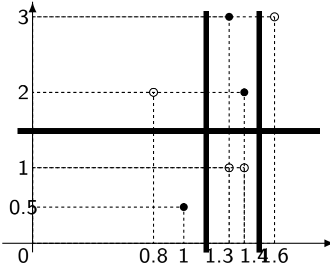
$$\begin{aligned}
\Phi_{\mathbb{S}} = \quad & \left(p_1^a + p_1^b + p_2^b\right)\left(p_1^a + p_2^a + p_3^b\right)\left(p_1^a + p_2^a + p_3^a\right)\left(p_2^a + p_3^a + p_1^b\right) p_2^b \\
& \left(p_2^a + p_2^b + p_3^b\right)\left(p_2^a + p_3^a + p_4^a + p_1^b + p_2^b + p_3^b\right)\left(p_3^a + p_4^a\right)\left(p_4^a + p_3^b\right) \\
& \left(p_2^a + p_1^b\right)\left(p_2^b + p_3^b\right)\left(p_3^a + p_2^b\right).
\end{aligned}$$

Transforming the formula $\Phi_{\mathbb{S}}$ into its DNF form we obtain four prime implicants:

$$\Phi_{\mathbb{S}} = \quad p_2^a p_4^a p_2^b + p_2^a p_3^a p_2^b p_3^b + p_3^a p_1^b p_2^b p_3^b + p_1^a p_4^a p_1^b p_2^b.$$

Discretization by reduct calculation

$\mathbb{S}^*$	p_1^a	p_2^a	p_3^a	p_4^a	p_1^b	p_2^b	p_3^b	d^*
(u_1, u_2)	1	0	0	0	1	1	0	1
(u_1, u_3)	1	1	0	0	0	0	1	1
(u_1, u_5)	1	1	1	0	0	0	0	1
(u_4, u_2)	0	1	1	0	1	0	0	1
(u_4, u_3)	0	0	1	0	0	1	1	1
(u_4, u_5)	0	0	0	0	0	1	0	1
(u_6, u_2)	0	1	1	1	1	1	1	1
(u_6, u_3)	0	0	1	1	0	0	0	1
(u_6, u_5)	0	0	0	1	0	0	1	1
(u_7, u_2)	0	1	0	0	1	0	0	1
(u_7, u_3)	0	0	0	0	0	1	1	1
(u_7, u_5)	0	0	1	0	0	1	0	1
new	0	0	0	0	0	0	0	0



Outline

Information systems and Decision tables

	Diploma	Experience	French	Reference	Decision
x_1	MBA	Medium	Yes	Excellent	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_4	MSc	High	Yes	Neutral	Accept
x_5	MSc	Medium	Yes	Neutral	Reject
x_6	MSc	High	Yes	Excellent	Accept
x_7	MBA	High	No	Good	Accept
x_8	MCE	Low	No	Excellent	Reject

$$\mathbb{D} = (U, A \cup \{d\})$$

- For any $B \subset A$:

$$x\ IND(B)\ y \Longleftrightarrow inf_B(x) = inf_B(y)$$

  $IND(B$ is a equivalent relation.
- $[u]_B = \{v : u\ IND(B)\ v\}$ – the equivalent class of $IND(B)$.
- $B \subseteq A$ defines a partition of U:

$$U|_B = \{[u]_B : u \in U\}$$

- For any subsets $P, Q \subseteq A$:

$$U|_P = U|_Q \Longleftrightarrow \forall_{u \in U} [u]_P = [u]_Q \quad (1)$$

$$U|_P \preceq U|_Q \Longleftrightarrow \forall_{u \in U} [u]_P \subseteq [u]_Q \quad (2)$$

- Properties:

$$P \subseteq Q \Longrightarrow U|_P \preceq U|_Q \quad (3)$$

$$\forall_{u \in U} \quad [u]_{P \cup Q} = [u]_P \cap [u]_Q \quad (4)$$

What are reducts?

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

- Given an information system $\mathbb{S} = (U, A)$ and a monotone evaluation function

$$\mu_{\mathbb{S}} : \mathcal{P}(A) \longrightarrow \Re^+$$

- The set $B \subset A$ is called *μ-reduct*, if
  - $\mu(B) = \mu(A)$,
  - for any proper subset $B' \subset B$ we have $\mu(B') < \mu(B)$;
- The set $B \subset A$ is called *approximated reduct*, if
  - $\mu(B) \geq \mu(A) - \varepsilon$,
  - for any proper subset ...

Definition (CORE and RED)

$$\mu\text{-RED} = \text{set off all } \mu\text{-reducts}; \quad \mu\text{-CORE} = \bigcap_{B \in \mu\text{-RED}} B$$

Positive Region Based Reducts

- For any $B \subseteq A$ and $X \subseteq U$:

$$\underline{B}(X) = \{u : [u]_B \subseteq X\}; \qquad \overline{B}(X) = \{u : [u]_B \cap X \neq \emptyset\}$$

- Let $\mathbb{S} = (U, A \cup \{dec\})$ be a decision table, let $B \subseteq A$, and let $U|_{dec} = \{X_1, ..., X_k\}$:

$$POS_B(dec) = \bigcup_{i=1}^{k} \underline{B}(X_i)$$

- If $R \subseteq A$ satisfies
  1. $POS_R(dec) = POS_A(dec)$
  2. For any $a \in R : POS_{R-\{a\}}(dec) \neq POS_A(dec)$

  then R is called the *reduct of A based on positive region*.
- $PRED(A) =$ set of reducts based on positive region;
- This is the μ-reduct, where $\mu(B) = |POS_B(dec)|$

Reducts

- Indiscernibility relation

$$(x,y) \in IND(B) \iff \forall_{a \in A} a(x) = a(y)$$
$$(x,y) \in IND_{dec}(B) \iff dec(x) = dec(y) \vee \forall_{a \in A} a(x) = a(y)$$

- A *decision-relative reduct* is a minimal set of attributes $R \subseteq A$ such that $IND_{dec}(R) = IND_{dec}(A)$.
- The set of all reducts is denoted by:

$$\mathcal{RED}(\mathbb{D}) = \{R \subseteq A : R \text{ is a reduct of } \mathbb{D}\}$$

Outline

The importance of attributes

$$\mathcal{RED}(\mathbb{D}) = \{R \subseteq A : R \text{ is a reduct of } \mathbb{D}\}$$

- Core attributes:

$$CORE(\mathbb{D}) = \bigcap_{R \in \mathcal{RED}(\mathbb{D})} R$$

- An attribute $a \in A$ is called **reduct attribute** if it occurs in at least one of reducts

$$REAT(\mathbb{D}) = \bigcup_{R \in \mathcal{RED}(\mathbb{D})} R$$

- The attribute is called *redundant attribute* if it is not a reductive attribute.
- An attribute b is redundant $\Leftrightarrow b \in A - REAT$

The problem setting

It is obvious that for any reduct $R \in \mathcal{RED}(\mathbb{D})$:

$$CORE(\mathbb{D}) \subseteq R \subseteq REAT(\mathbb{D})$$

The problem

For a given a decision table $\mathbb{S} = (U, A \cup \{dec\})$ calculate

$$CORE(\mathbb{D}) = \bigcap_{R \in \mathcal{RED}(\mathbb{D})} R \quad \text{and} \quad REAT(\mathbb{D}) = \bigcup_{R \in \mathcal{RED}(\mathbb{D})} R$$

Example

	a_1	a_2	a_3	a_4	Decision
x_1	MBA	Medium	Yes	Excellent	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_4	MSc	High	Yes	Neutral	Accept
x_5	MSc	Medium	Yes	Neutral	Reject
x_6	MSc	High	Yes	Excellent	Accept
x_7	MBA	High	No	Good	Accept
x_8	MCE	Low	No	Excellent	Reject

In this example:

- the set of all reducts $\mathcal{RED}(\mathbb{D}) = \{\{a_1, a_2\}, \{a_2, a_4\}\}$
- Thus

$$CORE(\mathbb{D}) = \{a_2\} \quad REAT(\mathbb{D}) = \{a_1, a_2, a_4\}$$

- the redundant attribute: a_3

Outline

Discernibility matrix

	a_1	a_2	a_3	a_4	Decision
x_1	MBA	Medium	Yes	Excellent	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_4	MSc	High	Yes	Neutral	Accept
x_5	MSc	Medium	Yes	Neutral	Reject
x_6	MSc	High	Yes	Excellent	Accept
x_7	MBA	High	No	Good	Accept
x_8	MCE	Low	No	Excellent	Reject

	x_1	x_4	x_6	x_7
x_2	a_2, a_4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4
x_3	a_1, a_2, a_4	a_1, a_2, a_4	a_1, a_2, a_4	a_1, a_2, a_3
x_5	a_1, a_4	a_2	a_2, a_4	a_1, a_2, a_3, a_4
x_8	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1, a_2, a_4

Boolean approach to reduct problem

- Boolean discernibility function:

$$\begin{aligned}\Delta_{\mathbb{D}}(a_1, ..., a_4) = {} & (a_2 + a_4)(a_1 + a_2)(a_1 + a_2 + a_4)(a_2 + a_3 + a_4) \\ & (a_1 + a_2 + a_4)(a_1 + a_2 + a_4)(a_1 + a_2 + a_4)(a_1 + a_2 + a_3) \\ & (a_1 + a_4)(a_2)(a_2 + a_4)(a_1 + a_2 + a_3 + a_4)(a_1 + a_2 + a_3) \\ & (a_1 + a_2 + a_3 + a_4)(a_1 + a_2 + a_3)(a_1 + a_2 + a_4)\end{aligned}$$

- In general: $R = \{a_{i_1}, ...a_{i_j}\}$ is a reduct in $\mathbb{D} \Leftrightarrow$ the monomial

$$m_R = a_{i_1} \cdot ... \cdot a_{i_j}$$

is a prime implicant of $\Delta_{\mathbb{D}}(a_1, ..., a_k)$

Theorem

For any attribute $a \in A$, a is a core attribute if and only if a occurs in discernibility matrix as a singleton. As a consequence, the problem of searching for core attributes can be solved in polynomial time

Simplifying the discernibility function

- Absorption law:

$$x + (x \cdot y) = x \qquad x \cdot (x + y) = x$$

- In our example: irreducible CNF of the discernibility function is as follows:

$$\Delta_{\mathbb{D}}(a_1, ..., a_4) = a_2 \cdot (a_1 + a_4)$$

- Complexity of searching for irreducible CNF: $O(n^4 k)$ steps.

Calculation of reductive attribute

Theorem

For any decision table $\mathbb{D} = (U, A \cup \{d\})$. *If*

$$\Delta_{\mathbb{D}}(a_1, ..., a_k) = \left(\sum_{a \in C_1} a\right) \cdot \left(\sum_{a \in C_2} a\right) \ldots \left(\sum_{a \in C_m} a\right)$$

is the irreducible CNF of discernibility function $\Delta_{\mathbb{D}}(a_1, ..., a_k)$, *then*

$$REAT(\mathbb{D}) = \bigcup_{i=1}^{m} C_i \tag{5}$$

Therefore the problem of calculation of all reductive attributes can be solved in $O(n^4 k)$ *steps.*

Outline

Boolean Reasoning Approach to Rough sets

Complexity of encoding functions

Given a decision table with n objects and m attributes

Problem	**Nr of variables**	**Nr of clauses**
minimal reduct	$O(m)$	$O(n^2)$
decision rules	$O(n)$ functions	
	$O(m)$	$O(n)$
discretization	$O(mn)$	$O(n^2)$
grouping	$O(\sum_{a\in A} 2^{\|V_a\|})$	$O(n^2)$
hyperplanes	$O(n^m)$	$O(n^2)$

Greedy algorithm:

time complexity of searching for the best variable:

$$O(\#variables \times \#clauses)$$

Data Mining

The iterative and interactive process of discovering *non-trivial, implicit, previously unknown* and *potentially useful (interesting) information or patterns* from large databases.

W. Frawley and G. Piatetsky-Shapiro and C. Matheus,(1992)

The science of extracting *useful information* from large data sets or databases.

D. Hand, H. Mannila, P. Smyth (2001)

Rough set algorithms based on BR reasoning:

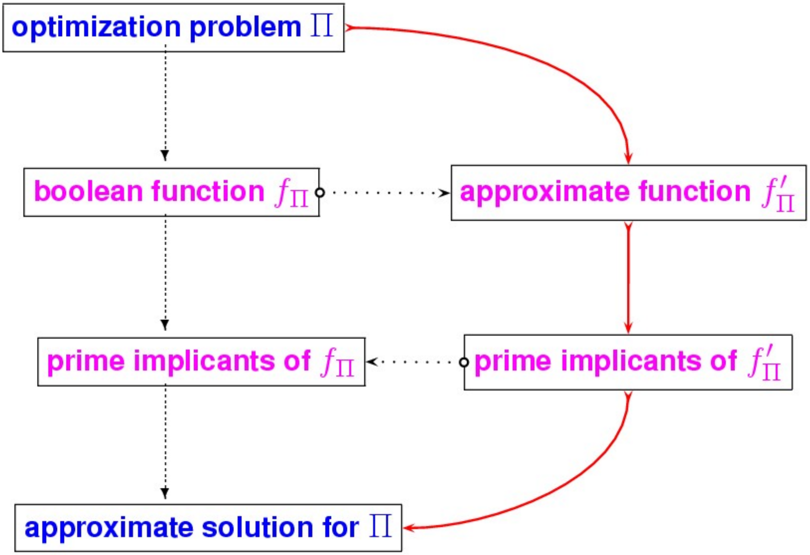
Advantages:

- accuracy: high;
- interpretability: high;
- adjustability: high;
- etc.

Disadvantages:

- Complexity: high;
- Scalability: low;
- Usability of domain knowledge: weak;

Approximate Boolean Reasoning



Example: Decision reduct

$\mathbb{A}$	a_1	a_2	a_3	a_4	dec
ID	outlook	temp.	hum.	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	?
14	rainy	mild	high	TRUE	?

Methodology

1. Discernibility matrix;
2. Discernibility Boolean function
3. Prime implicants $\Longrightarrow$ reducts

Discernibility matrix;

$\mathbb{M}$	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

The set R is a reduct if (1) it has nonempty intersection with each cell of the discernibility matrix and (2) it is minimal.

MD heuristics

- First we have to calculate the number of occurrences of each attributes in the discernibility matrix:

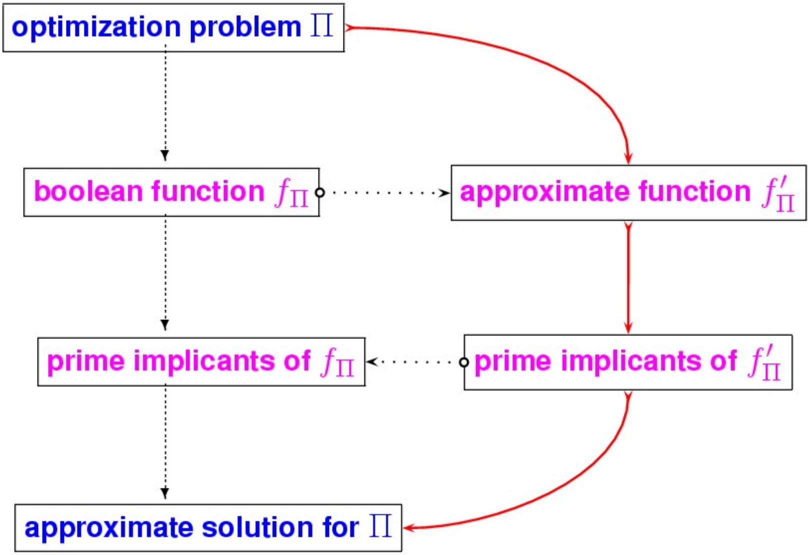
$$eval(a_1) = disc_{dec}(a_1) = 23 \qquad eval(a_2) = disc_{dec}(a_2) = 23$$
$$eval(a_3) = disc_{dec}(a_3) = 18 \qquad eval(a_4) = disc_{dec}(a_4) = 16$$

  Thus a_1 and a_2 are the two most preferred attributes.
- Assume that we select a_1. Now we remove those cells that contain a_1. Only 9 cells remain, and the number of occurrences are:

$$eval(a_2) = disc_{dec}(a_1, a_2) - disc_{dec}(a_1) = 7$$
$$eval(a_3) = disc_{dec}(a_1, a_3) - disc_{dec}(a_1) = 7$$
$$eval(a_4) = disc_{dec}(a_1, a_4) - disc_{dec}(a_1) = 6$$

- If this time we select a_2, then the are only 2 remaining cells, and, both are containing a_4;
- Therefore, the greedy algorithm returns the set $\{a_1, a_2, a_4\}$ as a reduct of sufficiently small size.

Approximate Boolean Reasoning



MD heuristics for reducts without discernibility matrix?

$\mathbb{A}$	a_1	a_2	a_3	a_4	dec
ID	outlook	temp.	hum.	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes

13	overcast	hot	normal	FALSE	?
14	rainy	mild	high	TRUE	?

1. Number of occurences of attibutes in $\mathbb{M}$;
2. Number of occurences of a set of attibutes in $\mathbb{M}$;

- Contingence table for a_1:

a_1	$dec = no$	$dec = yes$	$total$
$sunny$	3	2	5
$overcast$	0	3	3
$rainy$	1	3	4
$total$	4	8	12

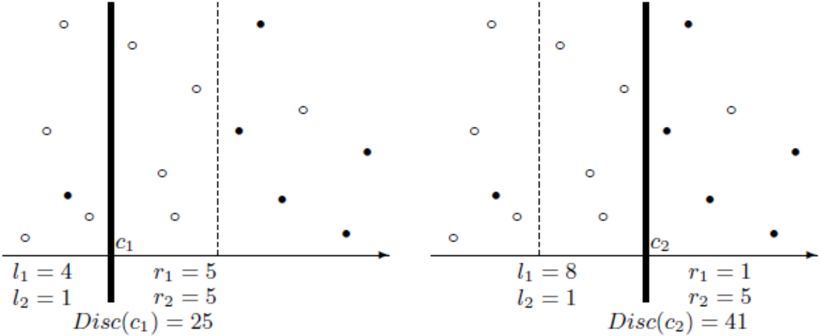
$$\boxed{disc_{dec}(a_1) = 4 \cdot 8 - 3 \cdot 2 - 0 \cdot 3 - 1 \cdot 3 = 23}$$

- Contingence table for $\{a_1, a_2\}$:

(a_1, a_2)	no	yes	$total$
$sunny, hot$	2	0	2
$sunny, mild$	1	1	2
$sunny, cool$	0	1	1
$overcast$	0	3	3
$rainy, mild$	0	2	2
$rainy, cool$	1	1	2
$total$	4	8	12

$$\boxed{disc_{dec}(a_1, a_2) = 4 \cdot 8 - 2 \cdot 0 - \ldots = 30}$$

Discernibility measure for discretization



- number of conflicts in a set of objects X: $conflict(X) = \sum_{i<j} N_i N_j$
- the discernibility of a cut (a, c):

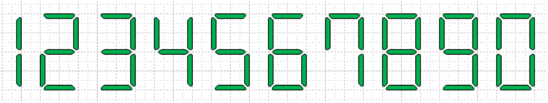
$$W(c) = conflict(U) - conflict(U_L) - conflict(U_R)$$

where $\{U_L, U_R\}$ is a partition of U defined by c.

Outline

Exercise 1: Digital Clock Font

Each digit in Digital Clock is made of a certain number of dashes, as shown in the image below. Each dash is displayed by a LED (light-emitting diode)



Propose a decision table to store the information about those digits and use the rough set methods to solve the following problems:

1. Assume that we want to switch off some LEDs to save the energy, but we still want to recognise the parity of the shown digit based on the remaining dashes. What is the minimal set of dashes you want to display?
2. The same question for the case we want to recognise all digits.

Exercise 2: Core attribute

Propose an algorithm of searching for all core attributes that does not use the discernibility matrix and has time complexity of $O(k \cdot n \log n)$.

Exercise 3: Decision table with maximal number of reducts

We know that the number of reducts for any decision table $\mathbb{S}$ with m attributes can not exceed the upper bound

$$N(m) = \binom{m}{\lfloor m/2 \rfloor}.$$

For any integer m construct a decision table with m attributes such that the number of reducts for this table equals to $N(m)$.