Approximate Boolean Reasoning Approach to Rough Sets and Data Mining

Hung Son Nguyen

Institute of Mathematics, Warsaw University
son@mimuw.edu.pl

RSFDGrC, September 3, 2005

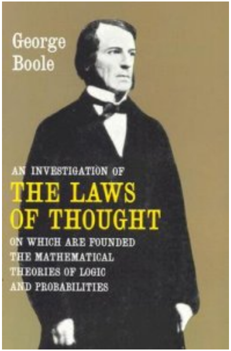
Outline

1. Boolean Reasoning Methodology
   - Introduction
   - Boolean Reasoning Approach to AI
2. Rough Set Approach to Data Mining
   - Concept Approximation Problem
   - Rough approximation of concepts
3. Approximate Boolean Reasoning
   - Motivation
   - ABR and Reducts vs. Association Rules

Outline

1. Boolean Reasoning Methodology
   - Introduction
   - Boolean Reasoning Approach to AI
2. Rough Set Approach to Data Mining
   - Concept Approximation Problem
   - Rough approximation of concepts
3. Approximate Boolean Reasoning
   - Motivation
   - ABR and Reducts vs. Association Rules

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.

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?

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

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:**
  $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$, and
  $a + (b \cdot c) = (a + b) \cdot (a + c)$
- **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;

Associative law: $(x+y)+z = x+(y+z)$ and $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

Idempotence: $x + x = x \quad and \quad x \cdot x = x(dual)$

Op. with 0 and 1: $x + 1 = 1 \quad and \quad x \cdot 0 = 0(dual)$

Absorption laws: $(y \cdot x) + x = x \quad and \quad (y + x) \cdot x = x(dual)$

Involution laws: $\overline{(\overline{x})} = x$

DeMorgan's laws:

$$\neg(x+y) = \neg x \cdot \neg y \quad and \quad \neg(x \cdot y) = \neg x + \neg y(dual)$$

Consensus laws:

$$(x+y) \cdot (\overline{x} + z) \cdot (y + z) = (x+y) \cdot (\overline{x} + z) \text{ and}$$
$$(x \cdot y) + (\overline{x} \cdot z) + (y \cdot z) = (x \cdot b) + (\overline{x} \cdot z)$$

Duality principle: Any algebraic equality derived from the axioms of Boolean algebra remains true when the operators $+$ and $\cdot$ are interchanged and the identity elements 0 and 1 are interchanged

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 $\boxed{f : \{0,1\}^n \rightarrow \{0,1\}}$ 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.

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 $\boxed{f : \{0,1\}^n \to \{0,1\}}$ 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
- 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

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:

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

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:

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

- **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:

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

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

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

- **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:**

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

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:

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

- **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:**

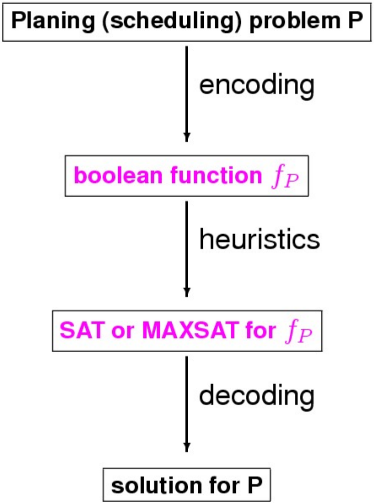
$$\begin{aligned} BD &\longrightarrow C \\ C &\longrightarrow B \vee D \\ A &\longrightarrow 0 \end{aligned}$$

- **Reasoning:** (theorem proving) e.g., show that
  "nobody can go alone."

Outline

1. Boolean Reasoning Methodology
   - Introduction
   - Boolean Reasoning Approach to AI
2. Rough Set Approach to Data Mining
   - Concept Approximation Problem
   - Rough approximation of concepts
3. Approximate Boolean Reasoning
   - Motivation
   - ABR and Reducts vs. Association Rules

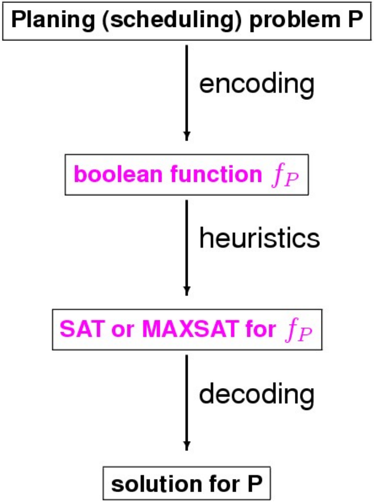
Boolean reasoning for decision problems



- SAT: whether an equation

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

has a solution?



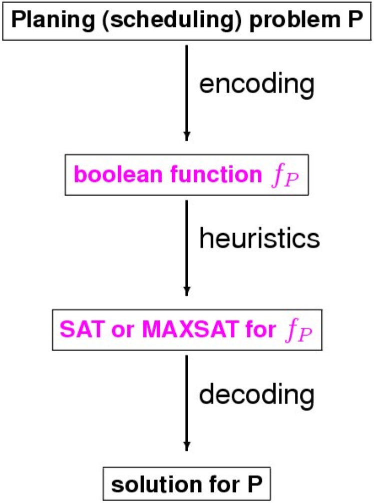
- SAT: whether an equation

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

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

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

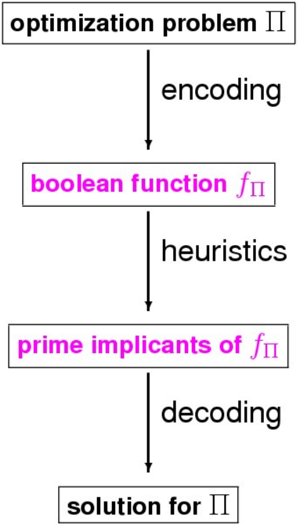
has a solution?

- 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.

procedure DPLL(ϕ, t)

```
//SAT:
  if φ/t is empty then
    return SATISFIABLE;
  end if
//Conflict:
  if φ/t contains an empty clause then
    return UNSATISFIABLE;
  end if
//Unit Clause:
  if φ/t contains a unit clause {p} then
    return DPLL(φ, tp);
  end if
//Pure Literal:
  if φ/t has a pure literal p then
    return DPLL( φ, tp);
  end if
//Branch:
  Let p be a literal from a minimum size clause of φ/t
  if DPLL( φ, tp ) then
    return SATISFIABLE;
  else
    return DPLL( φ, tp̄ );
  end if
```

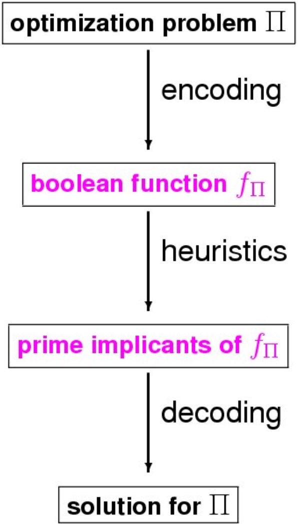
Boolean reasoning for optimization problems



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

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

Boolean reasoning for optimization problems

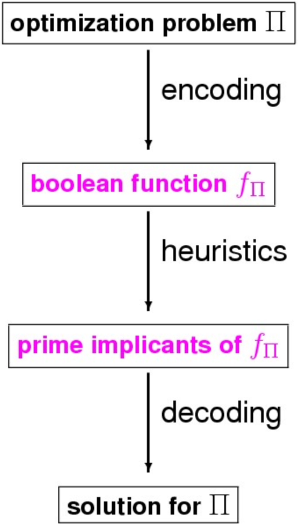


- A function $\phi : \{0,1\}^n \rightarrow \{0,1\}$ is "*monotone*" if

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

- 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

1. Boolean Reasoning Methodology
   - Introduction
   - Boolean Reasoning Approach to AI
2. Rough Set Approach to Data Mining
   - Concept Approximation Problem
   - Rough approximation of concepts
3. Approximate Boolean Reasoning
   - Motivation
   - ABR and Reducts vs. Association Rules

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

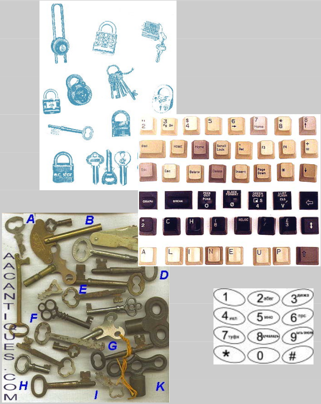
Many tasks in data mining can be formulated as an APPROXIMATE REASONING PROBLEM.

Assume that there are

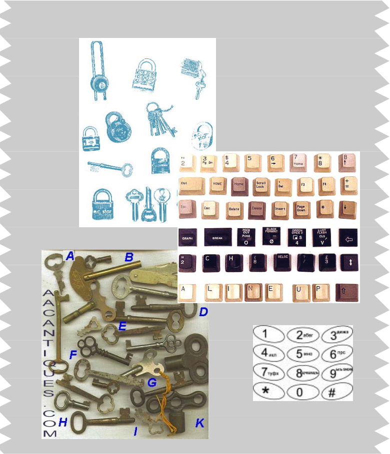
- Two agents A_1 and A_2;
- They are talking about objects from a common universe $\mathcal{U}$;
- They use different languages $\mathcal{L}_1$ and $\mathcal{L}_2$;
- *Every formula ψ in $\mathcal{L}_1$ (and $\mathcal{L}_2$) describes a set C_ψ of objects from $\mathcal{U}$.*

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;



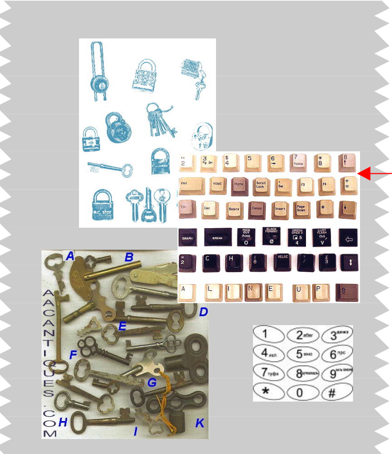
An universe of keys



An universe of keys

Teacher

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



An universe of keys

Teacher

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

Learner

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

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

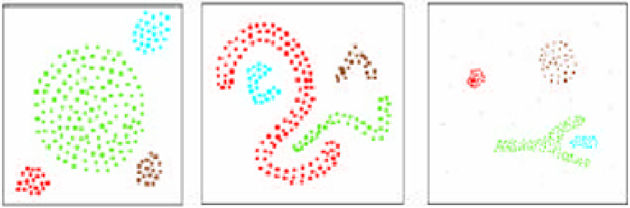
build an approximation of C in terms of $\mathcal{L}_2$

- 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

- **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;

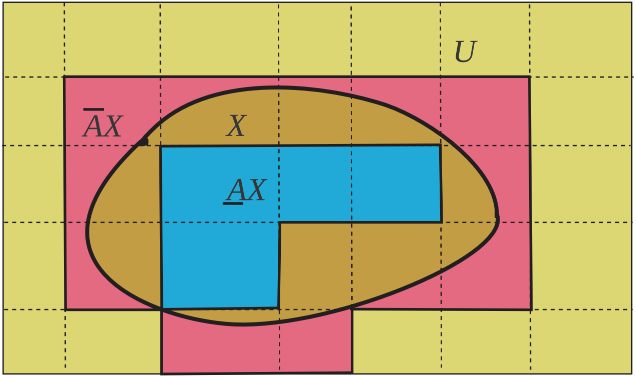
- **Time series data analysis:**
  - Universe: Sub-sequences obtained by windowing with all possible frame sizes.
  - Teacher: the actual phenomenon behind the collection of timed measurements, e.g., stock market, earth movements.
  - Learner: trends, variations, frequent episodes, extrapolation.

Outline

1. Boolean Reasoning Methodology
   - Introduction
   - Boolean Reasoning Approach to AI
2. Rough Set Approach to Data Mining
   - Concept Approximation Problem
   - Rough approximation of concepts
3. Approximate Boolean Reasoning
   - Motivation
   - ABR and Reducts vs. Association Rules

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**

Variable Precision Rough Sets (VPRS)

- Using *tolerance* relation (also similarity relation) instead of indiscernibility relation.
- If we allow weaker indiscernibility (tolerance) the indiscernibility classes may overlap.
- The family of sets which are definable using tolerance classes is richer than in case of equivalence classes.
- We may also extend the lower approximation of a set, allowing some admissible level $0 \leq \beta \leq 1$ of classification inaccuracy.

$$\underline{A}_{\beta}X = \bigcup\{[x]_A | \frac{|[x]_A \cap X|}{|[x]_A|} \geq \beta\}$$

Generalized approximation space

is a quadruple $\mathcal{A} = (\mathcal{U}, I, \nu, P)$, where

1. $\mathcal{U}$ is a non-empty set of objects (*an universe*),
2. $I : \mathcal{U} \to \mathcal{P}(\mathcal{U})$ is an *uncertainty function* satisfying conditions:
   - $x \in I(x)$ for $x \in \mathcal{U}$
   - $y \in I(x) \iff x \in I(y)$ for any $x, y \in \mathcal{U}$.

   Thus, the relation $xRy \iff y \in I(x)$ is a tolerance relation (reflexive and symmetric) and $I(x)$ is a tolerance class of x,
3. $\nu : \mathcal{P}(\mathcal{U}) \times \mathcal{P}(\mathcal{U}) \to [0, 1]$ is a *vague inclusion function*, which is a kind of membership function defined over $\mathcal{P}(\mathcal{U}) \times \mathcal{P}(\mathcal{U})$ to measure degree of inclusion between two sets. Vague inclusion must be *monotone* with respect to the second argument, i.e., if $Y \subseteq Z$ then $\nu(X, Y) \leq \nu(X, Z)$ for $X, Y, Z \subseteq \mathcal{U}$.
4. $P : I(\mathcal{U}) \to \{0, 1\}$ is a *structurality function*

Generalized Approximation Space

- Together with uncertainty function I, vague inclusion function ν defines the *rough membership function* for $x \in \mathcal{U}, X \subseteq \mathcal{U}$:

$$\mu_{I,\nu}(x, X) = \nu(I(x), X)$$

- The vague inclusion function ν is approximately constructed from the finite set of examples $U \in \mathcal{U}$.
- Lower and upper approximations in $\mathcal{A}$ of $X \subseteq \mathcal{U}$ are then defined as

$$\mathbf{L}_{\mathcal{A}}(X) = \{x \in \mathcal{U} : P(I(x)) = 1 \ \wedge \nu(I(x), X) = 1\}$$
$$\mathbf{U}_{\mathcal{A}}(X) = \{x \in \mathcal{U} : P(I(x)) = 1 \ \wedge \nu(I(x), X) > 0\}$$

- The structurality function allows us to enforce additional global conditions on sets $I(x)$ considered in approximations. Only sets $X \in I(\mathcal{U})$ for which $P(X) = 1$ (referred as *P-structural elements* in U) are considered.
- For example, function $P_\alpha(X) = 1 \iff |X \cup U|/|U| > \alpha$ will discard all relatively small subsets of U (given by α).

Classifier

Result of a concept approximation method.

It is also called the *classification algorithm* featured by

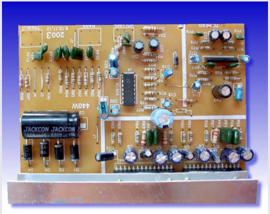
- **Input:** information vector of an object;
- **Output:** whether an object belong to the concept;
- **Parameters:** are necessary for tuning the quality of classifier;

Rough classifier

Outside look: 4 possible answers

- YES (lower approximation)
- POSSIBLY YES (boundary region)
- NO
- DON'T KNOW

Inside:



- Feature selection/reduction;
- Feature extraction (discretization, value grouping, hyperplanes ...);
- Decision rule extraction;
- Data decomposition;
- Reasoning scheme approximation;

Outline

1 Boolean Reasoning Methodology
- Introduction
- Boolean Reasoning Approach to AI

2 Rough Set Approach to Data Mining
- Concept Approximation Problem
- Rough approximation of concepts

3 Approximate Boolean Reasoning
- Motivation
- ABR and Reducts vs. Association Rules

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

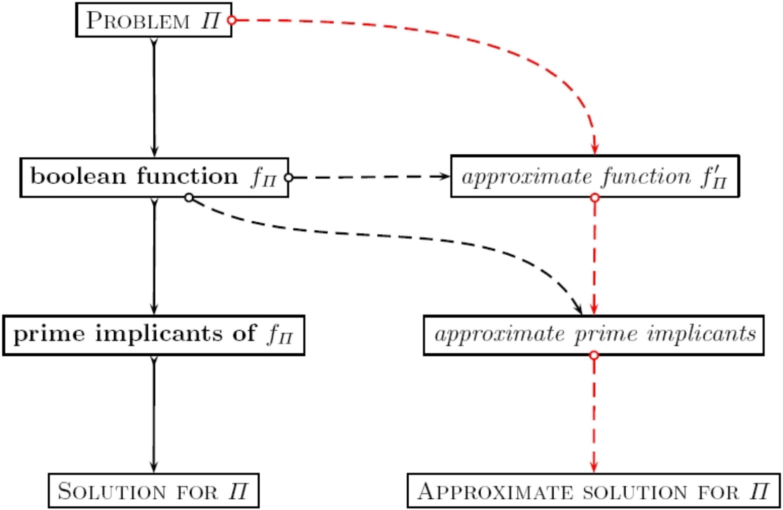


Figure: The Boolean reasoning scheme for optimization problems

Outline

1 Boolean Reasoning Methodology
- Introduction
- Boolean Reasoning Approach to AI

2 Rough Set Approach to Data Mining
- Concept Approximation Problem
- Rough approximation of concepts

3 Approximate Boolean Reasoning
- Motivation
- ABR and Reducts vs. Association Rules

Data reduction in Rough sets

What is 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
  - $\mu(B) \geq \mu(A)(1 - \varepsilon)$,
  - for any proper subset ...

Some types of reducts

- Information reduct:

$$\mu_1(B) = \text{number of pairs of objects discerned by } B$$

- Decision oriented reduct:

$$\mu_2(B) = \text{number of pairs of conflict objects discerned by } B$$

- Object oriented reduct:

$$\mu_x(B) = \text{number of objects discerned with } x \text{ by } B$$

- Frequent reducts;
- α-reducts: $(1-\alpha)$ approximation reduct with respect to the discernibility measure;
- ...

Example

$\mathbb{A}$	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	?

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

Reducts

After reducing of all repeated clauses we have:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) = & (x_1)(x_1 + x_4)(x_1 + x_2)(x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + x_4) \\ & (x_2 + x_3 + x_4)(x_1 + x_2 + x_3)(x_4)(x_2 + x_3)(x_2 + x_4) \\ & (x_1 + x_3)(x_3 + x_4)(x_1 + x_2 + x_4) \end{aligned}$$

- remove those clauses that are absorbed by some other clauses (using absorbtion rule: $p(p+q) \equiv p$):

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

- Translate f from CNF to DNF

$$f = x_1 x_4 x_2 + x_1 x_4 x_3$$

- Every monomial corresponds to a reduct. Thus we have 2 reducts: $R_1 = \{a_1, a_2, a_4\}$ and $R_2 = \{a_1, a_3, a_4\}$

counting table

By *contingency table* of a set of attributes B we denote the two-dimensional array $Count(B) = [n_{v,k}]_{v \in INF(B), k \in V_{dec}}$, where

$$n_{v,k} = card(\{x \in U : inf_B(x) = v \text{ and } dec(x) = k\})$$

Discernibility measure:

$$disc_{dec}(B) = \frac{1}{2} \sum_{v \neq v', k \neq k'} n_{v,k} \cdot n_{v',k'} \qquad (1)$$

$$disc_{dec}(B) = conflict(U) - \sum_{[x] \in U/IND(B)} conflict([x]_{IND(B)}) \tag{2}$$

Thus, the discernibility measure can be determined in $O(S)$ time:

$$disc_{dec}(B) = \frac{1}{2}\left(n^2 - \sum_{k=1}^{d} n_k^2\right) - \frac{1}{2}\sum_{v \in INF(B)}\left[\left(\sum_{k=1}^{d} n_{v,k}\right)^2 - \sum_{k=1}^{d} n_{v,k}^2\right] \tag{3}$$

where $n_k = |CLASS_k| = \sum_v n_{v,k}$ is the size of k^{th} decision class.

ABR approach to reducts

- 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 are taking under consideration only those cells of the discernibility matrix which are not containing a_1. There are 9 such cells only, and the number of occurrences are as following:

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

- 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.