Boolean Methods in Operations Research and Related Areas

Yves Crama

QuantOM Center for Quantitative Methods and Operations Management
HEC Management School, University of Liège, Belgium

IFORS Distinguished Lecture
INFORMS Annual Meeting, Charlotte, NC, November 2011

A few facts about Belgium

- Chocolate
- Beer
- Tintin
- Smurfs
- Justine Henin and Kim Clijsters
- Magritte
- Adolphe Sax (saxophone)
- $>$ 500 days without government (world record)
- Boolean functions

Outline

1. Boolean functions
2. Two fundamental models
3. Two applications
4. Conclusions

Boolean functions

Consider a function:

$$(x_1, x_2, \ldots, x_n) \to F(x_1, x_2, \ldots, x_n).$$

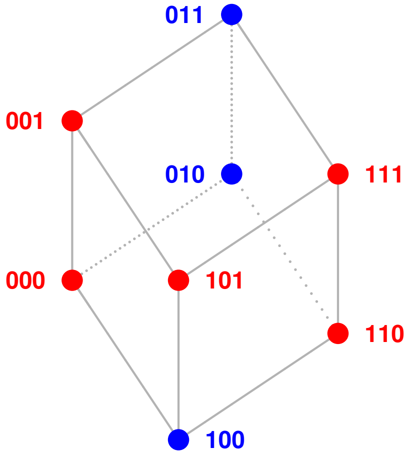
To be interesting,

- each variable should take at least two distinct values, and
- the function should take at least two distinct values.

So: the most elementary interesting functions are those for which each variable takes exactly two values, and for which the function itself can take exactly two values.

Named Boolean functions, after George Boole (1815-1864).

Example



red = 0

blue = 1

Boolean functions

- Boole was interested in modeling human reasoning (the *Laws of Thought* – 1854).
- Each variable and output can be interpreted as "True" or "False".
- In spite of their simplicity, Boolean functions possess a rich theory and have found an amazing array of applications over the last 150 years.

Applications of Boolean functions

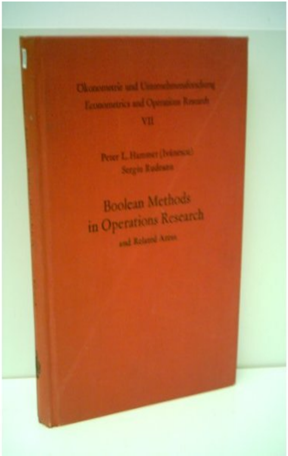
- Electrical and electronic engineering: signal goes through a network (Yes or No) depending on the state of intermediate gates (Open or Closed).
- Computer science: computation output is 0 or 1 depending on the initial input (in binary format: string of 0's and 1's).
- Game theory and social choice: a resolution is adopted (Yes or No) by a governing body depending on the votes (Yes or No) of individual members.
- Artificial intelligence: an action is taken (Yes or No) depending on the presence or absence of certain features (e.g, medical diagnosis: prescribe additional tests or not).
- Reliability: complex system operates (Yes or No) depending on the state of its elements (operating or failed).

Links with operations research

A “classic” on Boolean methods in operations research:

BOOLEAN METHODS
IN OPERATIONS RESEARCH
and Related Areas

Peter L. HAMMER and Sergiu RUDEANU
Springer-Verlag, New York, 1968

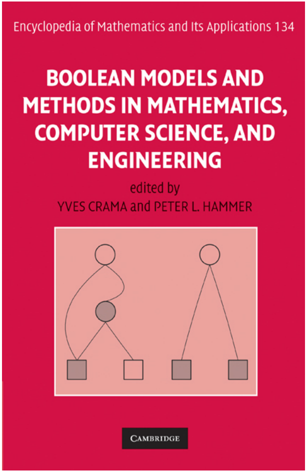


Two recent books

A collection of surveys:

BOOLEAN MODELS AND METHODS in Mathematics, Computer Science and Engineering

Yves CRAMA and Peter L. HAMMER, editors.
Cambridge University Press, 2010
780 pages



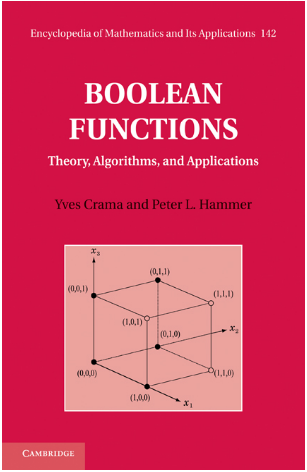
Two recent books

A research monograph:

BOOLEAN FUNCTIONS
Theory, Algorithms, and Applications

Yves CRAMA and Peter L. HAMMER
Cambridge University Press, 2011
710 pages

with contributions by C. Benzaken, E. Boros, N. Brauner, M.C. Golumbic, V. Gurvich, L. Hellerstein, T. Ibaraki, A. Kogan, K. Makino, B. Simeone



BOOLEAN FUNCTIONS
Theory, Algorithms, and Applications

Extensive coverage, with emphasis on normal form representations, fundamental results, combinatorial and structural properties, algorithms and complexity, variety of applications.

Foundations:

1. Fundamental concepts and applications
2. Boolean equations
3. Prime implicants and minimal DNFs
4. Duality theory

BOOLEAN FUNCTIONS
Theory, Algorithms, and Applications

State of the field

Very dynamic 150-year old field!!

- Two books (1,500 pages) are not enough (but there are other books...)
- Many concepts have been repeatedly rediscovered in different fields.
- Much recent progress on certain fundamental problems.
- Many remaining open problems!

Boolean methods are alive and well...

Yves Crama

QuantOM Center for Quantitative Methods and Operations Management
HEC Management School, University of Liège, Belgium

IFORS Distinguished Lecture
INFORMS Annual Meeting, Charlotte, NC, November 2011

Definitions

Definitions

- Boolean function: mapping $\varphi : \{0,1\}^n \to \{0,1\}$.
- Variables: $x_1, x_2, \ldots, x_n$
- Operations: disjunction $\vee$ (logical OR), conjunction $\wedge$ or product (logical AND), complementation $\overline{x} = 1 - x$ (logical NOT)
- Literal: variable x or its complement $\overline{x}$
- Elementary disjunction (clause): $x_1 \vee \overline{x_2} \vee \ldots \vee x_{n-1} \vee \overline{x_n}$
- Elementary conjunction (term): $\overline{x_1}\, x_2 \ldots x_{n-1}\, \overline{x_n}$

Disjunctive normal form

How do we represent $\varphi(X)$?

Disjunctive normal form

A Boolean function φ is in DNF if it is expressed as a disjunction of elementary conjunctions (terms).

Example: $\varphi = x_1\, x_2 \;\vee\; \overline{x_1}\, x_2\, \overline{x_3} \;\vee\; \overline{x_2}\, x_3$.

Note: every Boolean function has a DNF (and a CNF) representation.

Definitions

Boolean equation

A Boolean equation is an equation of the form $\varphi(X) = 0$.

Its complexity depends on how $\varphi(X)$ is represented.

Disjunctive equations

DNF equation

A DNF equation is an equation of the form $\varphi(X) = 0$ where φ is a DNF.

Example: $\varphi = x_1\, x_2 \vee \overline{x_1}\, x_2\, \overline{x_3} \vee \overline{x_2}\, x_3 = 0.$
Solution: $x_1 = 0$, $x_2 = 1$, $x_3 = 1$.

A solution “hits” every term with a 0.

Note: DNF equations are equivalent to satisfiability problems, and hence are NP-complete.

Boolean equations: State of the art

State of the art

1. Enormous progress in the solution of Boolean equations over the last 15 years: mix of classical methods (resolution, Davis-Putnam) and of modern heuristic approaches (stochastic search).
2. Practical applications in classical areas (e.g., circuit design), but also: successful reformulation of hard combinatorial problems (e.g., scheduling, timetabling).
3. Deep theoretical questions remain. For instance: regarding the existence of a density threshold which sharply separates consistent from inconsistent equations (*threshold conjecture*).

Threshold conjecture.

Threshold conjecture.

For each $k \geq 2$, there exists a constant c_k^* such that random DNF equations with n variables, cn terms, and k variables per term, are

- consistent with probability approaching 1 as n goes to infinity when $c < c_k^*$, and
- inconsistent with probability approaching 1 as n goes to infinity when $c > c_k^*$.

Positive functions

Positive functions

Function φ is *positive* (monotone) if

$$X \leq Y \Rightarrow \varphi(X) \leq \varphi(Y).$$

Positive Boolean functions can be represented by DNFs without complemented variables.

Example: $\varphi = x_1 x_2 \vee x_1 x_3 x_4 \vee x_2 x_3.$

Minimal true points and maximal false points

Example: $\varphi = x_1x_2 \vee x_1x_3x_4 \vee x_2x_3$.

φ takes value 1 if $x_1 = x_2 = 1$, or $x_1 = x_3 = x_4 = 1$, or $x_2 = x_3 = 1$ (or more variables take value 1).

(1,1,0,0), (1,0,1,1) and (0,1,1,0) are the minimal true points (MTP) of φ.

φ takes value 0 if $x_1 = x_2 = 0$, or $x_1 = x_3 = 0$, or $x_2 = x_3 = 0$, or $x_2 = x_4 = 0$ (or more variables take value 0).

(0,0,1,1), (0,1,0,1), (1,0,0,1) and (1,0,1,0) are the maximal false points (MFP) of φ.

Dualization

Observation: A positive function φ is completely defined by the list of its MTPs or by the list of its MFPs.

A fundamental algorithmic problem:

Dualization

- Input: the list of minimal true points of a positive function.
- Output: the list of maximal false points of the function.

Problem investigated in Boolean theory, game theory, integer programming, electrical engineering, artificial intelligence, reliability, combinatorics, etc.

Dualization

Dualization amounts to generating

- all minimal solutions of a set covering problem:

$$\sum_{j \in E_i} x_j \geq 1 \ (i = 1, ..., m);$$

$$x_j \in \{0, 1\} \ (j = 1, ..., n);$$

- all minimal transversals of a hypergraph (V, E), $E = (E_1, \ldots, E_m)$, $E_i \subseteq V$ (in particular: all maximal stable sets of a graph);
- all maximal losing coalitions of a simple game (defined by the list of its minimal winning coalitions).

Complexity

Note: the output is uniquely defined, but its size can be exponentially large in the size of the input.

(Lawler, Lenstra, Rinnooy Kan 1980; Johnson, Papadimitriou, Yannakakis 1988; Bioch, Ibaraki 1995; etc.)

Can positive Boolean functions be dualized in *total polynomial time*, that is, in time polynomial in the combined size of the input and of the output?

Equivalent problem

Dualization is “polynomially equivalent” to the problem:

Test Dual

- Input: the MTPs of a Boolean function φ, and a list L of points.
- Question: is L the list of MFPs of φ ?

- Dualization can be solved in total polynomial time if and only if Test Dual can be solved in polynomial time.
- Test Dual does not require exponential outputs.

Quasi-polynomial algorithm

Fredman and Khachiyan have shown

Fredman and Khachiyan (1996)

Dualization can be solved in time $O(m^{c\,logm})$, where m is the combined size of the input and of the output of the problem.

- Numerous extensions and generalizations by Boros, Elbassioni, Gurvich, Khachiyan, Makino, etc.
- But the central questions remain open:

Open problems

Complexity of Dualization

Can dualization be solved in total polynomial time?

Complexity of Test Dual

Can Test Dual be solved in polynomial time? Is it NP-hard (unlikely)?

Analysis of shareholders' power in corporate networks

Based on:

- Y. Crama and L. Leruth, Control and voting power in corporate networks: Concepts and computational aspects, *European Journal of Operational Research* 178 (2007) 879–893.
- Y. Crama and L. Leruth, Power indices and the measurement of control in corporate structures, to appear in *International Game Theory Review*.

Problem statement

Given a large network of shareholding relations among firms, who owns control in this network?

Objects of study:

- networks of entities (firms, banks, individual owners, pension funds,...) linked by shareholding relationships;
- their structure;
- notion and measurement of control in such networks.

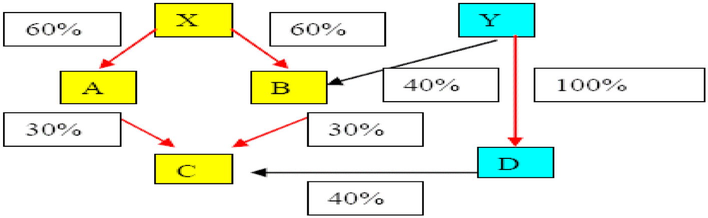
Shareholding networks

Graph model

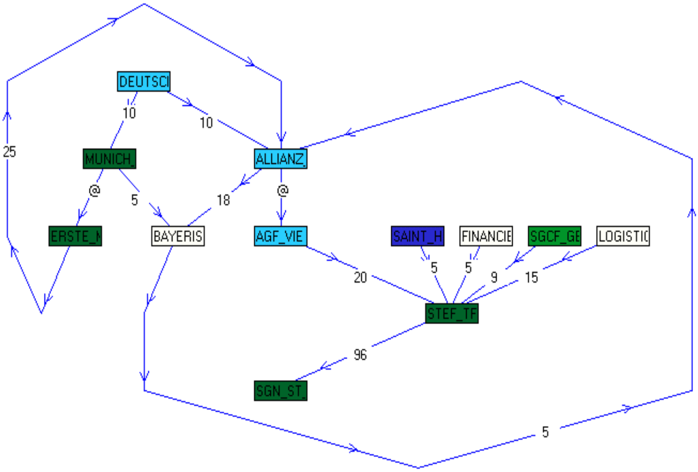
A shareholding network is a weighted graph:

- nodes = firms
- arc (i, j) if firm i is a shareholder of firm j
- $w(i, j)$ = fraction of shares of firm j held by firm i

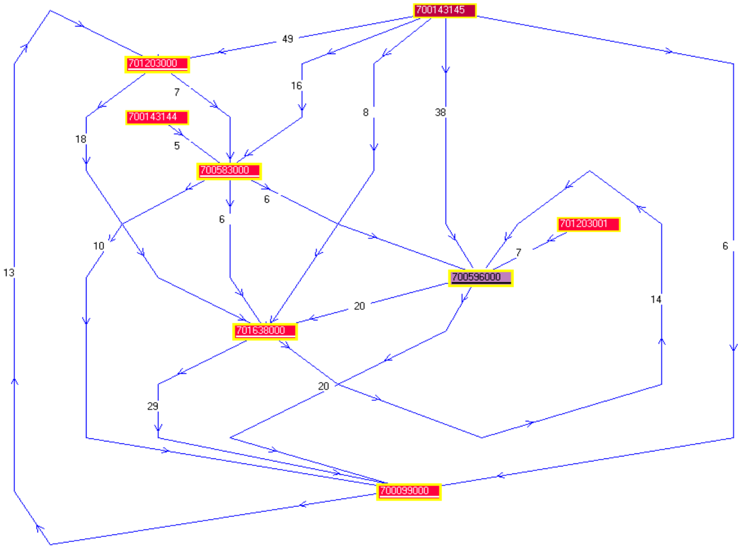
Small shareholders' network (from Eurostat 2010 – Statistical office of the European Union)



Real shareholders' network 1



Real shareholders' network 2



Measurement of control

Issues:

- Who controls who in a network? To what extent?
- In a pyramidal structure, identify “groups” of firms “controlled” by a same shareholder?
- In a pyramidal structure, who are the “ultimate” shareholders of a given firm?

Measurement of control

A game-theoretic model:

- Look at the shareholders of firm j as playing a weighted majority game whenever a decision is to be made by firm j.
- Identify the level of control of player i over j with the Banzhaf power index of i in this game.

Link with Boolean models?

Simple games

Simple game

A simple game is a positive Boolean function v on n variables.

Interpretation

- n players.
- v describes the voting rule which is adopted by the players when a decision is to be made.
- $v(0, 0, 1, 1)$ is the outcome of the voting process when players 3 and 4 say “Yes”.

Weighted majority games

Example:

- Player i carries a voting weight w_i
- Voting threshold t
- $v(X) = 1 \iff \sum_{i=1}^{n} w_i x_i > t$

In Boolean jargon, a weighted majority game is called a threshold function.

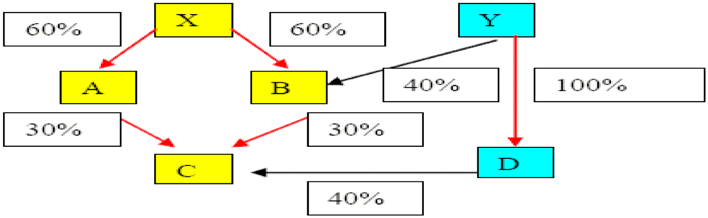
Banzhaf index

Banzhaf index

The Banzhaf index $Z(k)$ of player k in a game is the probability that, for a random voting pattern (uniformly distributed), the outcome of the game changes from 0 to 1 when player k changes her vote from 0 to 1.

- Provides a measure of the influence or power of player k.
- Related to, but different from the Shapley-Shubik index.
- Usually different from the voting weights in a weighted majority game.

Small shareholders' network (from Eurostat 2010 – Statistical office of the European Union)



Chow parameters

Link with Boolean functions:

- While attempting to characterize threshold functions, Chow (1961) has introduced $(n+1)$ parameters associated with a Boolean function $f(x_1, x_2, \ldots, x_n)$:

$$(\omega_1, \omega_2, \ldots, \omega_n, \omega)$$

where

- ω is the number of "ones" of f
- ω_k is the number of "ones" of f where $x_k = 1$.

Chow parameters

- ω_k is the number of "ones" of f where $x_k = 1$

  $\Longrightarrow \omega_k/2^{n-1}$ is the probability that $f = 1$ when $x_k = 1$.

- Can be shown that Banzhaf indices are simple transformations of the Chow parameters:

$$Z(k) = (2\omega_k - \omega)/2^{n-1}.$$

- Similar indices in reliability theory.

Shareholders' voting game

Back to shareholders' networks:

- Look at the shareholders of firm j as playing a weighted majority game (with quota 50%) whenever a decision is to be made by firm j
- Level of control of firm i over firm j = Banzhaf index $Z(i, j)$ in this game.

Single layer of shareholders

- Power indices have been proposed for the measurement of corporate control by many researchers (Shapley and Shubik, Cubbin and Leech, Gambarelli, Zwiebel,...)
- In most applications: single layers of shareholders (weighted majority games)

But real networks are more complex...

- Up to several thousand firms
- Incomplete shareholding data (small holders are unidentified)
- Multilayered (pyramidal) structures
- Cycles
- Ultimate relevant shareholders are not univoquely defined

Analysis of shareholders' power in corporate networks

Combining

- stochastic simulation to mimic the behavior of voters,
- graph-theoretic algorithms to accelerate computations,

allows us to compute power indices in large-scale networks.
See Crama and Leruth (2007).

Computational complexity

Computational complexity of the procedures?
Numerous results on Banzhaf/Chow indices of weighted majority games/threshold functions.

- NP-hard.
- "Efficient" (pseudo-polynomial) dynamic programming algorithms.
- Approaches based on generating functions, Monte-Carlo simulation, multilinear extensions, ...

Computational complexity

Recent papers on computational complexity of Chow parameters:

- Complexity of enumeration algorithms for threshold functions and other functions (Aziz et al. 2009, Klinz and Woeginger 2005).
- Complexity and accuracy of sampling-based algorithms for general functions (Bachrach et al. 2010).
- “Learning” threshold functions from their Chow parameters (O’Donnell and Servedio 2008).

More work is needed, e.g., to evaluate Chow parameters (and related indices) of compositions of threshold functions (complexity, approximation, . . .)

Classification: mining meaningful relations

Problem statement

- Given: collection of observed examples (attributes,class),
- classify/explain/predict new examples based on their attributes.

- Grant / do not grant financial credit based on observed features (level of income, past credit record, education level, etc.).
- Classify benign vs. malign cancers based on collections of tests and past observations.
- Predict polymer properties based on physical and chemical properties.

Classification example

Example: Classification into $\mathbb{T}$ or into $\mathbb{F}$.

$\mathbb{T}$	1	0	0	1	1	0	0	0	0
	1	1	1	1	0	0	1	0	0
	0	1	1	0	1	1	1	0	0
	1	0	1	0	0	1	1	1	1
$\mathbb{F}$	0	0	1	1	1	1	0	0	1
	0	1	0	1	1	1	1	1	1
	0	0	0	0	1	1	0	1	0
	1	1	0	0	0	0	0	1	0
??	0	1	1	0	1	0	0	1	0

Logical Analysis of Data

For binary data: Boolean framework proposed in

- Y. Crama, P.L. Hammer and T. Ibaraki, Cause-effect relationships and partially defined Boolean functions, *Annals of Operations Research* 16, 1988, 299-325.

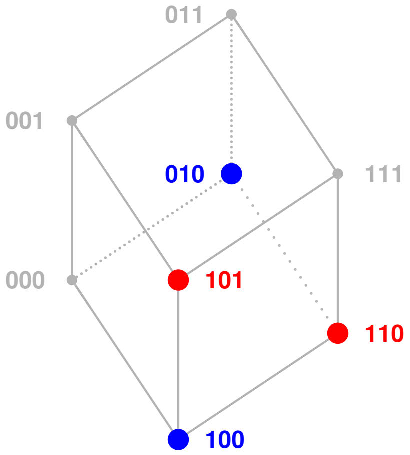
Further developed by G. Alexe, S. Alexe, M. Anthony, E. Boros, A. Kogan, K. Makino, T. Bonates, N. Brauner, and many others.

Partially defined Boolean functions

Basic idea:

- data: partially defined Boolean function

Example



$\mathbf{F} = \{(\mathbf{110}), (\mathbf{101})\}$

$\mathbf{T} = \{(\mathbf{010}), (\mathbf{100})\}$

Partially defined Boolean functions

Basic idea:

- data: partially defined Boolean function
- the relation to be learned is a Boolean function extending the observations
- goal: “guess” the most appropriate functional extension.

Note: related questions have been investigated in electrical engineering (circuit design with don’t cares).

Logical Analysis of Data

Based on the representation of extensions by DNFs and on selection of

- subsets of relevant variables (support sets)
- relevant terms (patterns)
- relevant disjunctions of terms (theories)

Support sets

Find a small (smallest, if possible) subset of the attributes which distinguishes the sets $\mathbb{T}$ and $\mathbb{F}$.

$\mathbb{T}$	1	0	0	1	1	0	0	0	0
	1	1	1	1	0	0	1	0	0
	0	1	1	0	1	1	1	0	0
	1	0	1	0	0	1	1	1	1
$\mathbb{F}$	0	0	1	1	1	1	0	0	1
	0	1	0	1	1	1	1	1	1
	0	0	0	0	1	1	0	1	0
	1	1	0	0	0	0	0	1	0

Support sets

Find a small (smallest, if possible) subset of the attributes which distinguishes the sets $\mathbb{T}$ and $\mathbb{F}$.

$\mathbb{T}$	1	0	0	1	1	0	0	0	0
	1	1	1	1	0	0	1	0	0
	0	1	1	0	1	1	1	0	0
	1	0	1	0	0	1	1	1	1
$\mathbb{F}$	0	0	1	1	1	1	0	0	1
	0	1	0	1	1	1	1	1	1
	0	0	0	0	1	1	0	1	0
	1	1	0	0	0	0	0	1	0

Patterns

- Pattern: combination of attributes which is observed at least once in a “true” example, but never in a “false” example.

Patterns

Pattern: $(11 * 1)$

$\mathbb{T}$	1	0	0	1
	1	1	0	1
	0	1	1	0
	1	0	1	1
$\mathbb{F}$	0	0	1	0
	0	1	0	1
	0	0	0	1
	1	1	0	0

Patterns

Pattern: $(11 * 1) \to$ term $x_1\, x_2\, x_4$

$\mathbb{T}$	1	0	0	1
	1	1	0	1
	0	1	1	0
	1	0	1	1
$\mathbb{F}$	0	0	1	0
	0	1	0	1
	0	0	0	1
	1	1	0	0

Patterns

- Pattern: combination of attributes which is observed at least once in a "true" example, but never in a "false" example.
- Search for patterns (or quasi-patterns) by enumeration, MILP models, etc.

Theories

- Theory: an extension f of the partially defined function which can be represented as a disjunction of patterns.
- Classification: data point X is classified as “true” point if $f(X) = 1$.
- LAD provides systematic approaches for building “good theories”.

Properties

- Classification: data point X is classified as "true" point if $f(X) = 1$, i.e, (at least) one pattern of f is "triggered" by X.
- Justification: $f(X) = 1$ means that we have previously observed another "true" example displaying the same features, and we have never observed a "false" example displaying these features.
- Interpretability: Patterns define understandable classification rules.
- Bitheories: similar justifications hold for "true" and "false" classifications.
- Nearest neighbor, decision trees yield bitheories.

See, e.g., E. Boros et al., Logical Analysis of Data: Classification with justification, *Annals of OR* 188 (2011) 33-61.

In practice...

- Large stream of pure and applied research over the last 20 years, in connection with related developments in data mining and machine learning.
- Many applications, in particular for biomedical datasets.
- Provide excellent, simple and robust classifiers.
- Understandable and justifiable.

A typical application of LAD

G. Alexe et al., Ovarian cancer detection by logical analysis of proteomic data, *Proteomics* 4, 2004, 766-783.

- Observations: 162 ovarian cancer (positive) cases, 91 control (negative) cases
- Features: measurements for 15,154 peptides

Results:

- only 7-9 peptides are needed to describe the classification;
- accuracy of classification is very high: about 98-99% of correct classifications;
- classification rules raise new research questions for biologists (how and why are certain patterns related to the occurrence of cancers?).

Conclusion

Boolean Methods in Operations Research and Related Areas

- Dynamic field of research, many challenging problems at the interface of applications and mathematics.
- Much more in
  Y. Crama and P.L. Hammer, *Boolean Functions: Theory, Algorithms, and Applications*, CUP, 2011.
  Y. Crama and P.L. Hammer, eds., *Boolean Models and Methods in Mathematics, Computer Science and Engineering*, CUP, 2010.

Closing

Thank you!