Pseudo-Boolean Functions and Nonlinear 0-1 Optimization

Yves Crama

HEC Management School
University of Liège

Francqui Lecture, KUL, April 2010

Outline

1. Boolean and Pseudo-Boolean Functions

Outline

Outline

Outline

Objectives

Focus:

Objectives

Focus:

- basic facts about pseudo-Boolean functions

Objectives

Focus:

- basic facts about pseudo-Boolean functions
- some representative applications

Objectives

Focus:

- basic facts about pseudo-Boolean functions
- some representative applications
- including nice proofs!

Objectives

Focus:

- basic facts about pseudo-Boolean functions
- some representative applications
- including nice proofs!

Largely based on

BOOLEAN FUNCTIONS
Theory, Algorithms, and Applications

Yves CRAMA and Peter L. HAMMER
Cambridge University Press
Due to appear: December 2010

Outline

1. Boolean and Pseudo-Boolean Functions
2. MAX CUT
3. MAX SAT
4. Nonlinear 0-1 optimization algorithms

Definitions

Boolean functions

A Boolean function is a mapping $\varphi : \{0,1\}^n \rightarrow \{0,1\}$

Definitions

Boolean functions

A Boolean function is a mapping $\varphi : \{0,1\}^n \rightarrow \{0,1\}$

Pseudo-Boolean functions

A pseudo-Boolean function is a mapping $f : \{0,1\}^n \rightarrow \mathbf{R}$

Examples

x_1	x_2	x_3	φ	f
0	0	0	0	4
0	0	1	1	2
0	1	0	0	-1
0	1	1	1	3
1	0	0	0	-5
1	0	1	0	6
1	1	0	1	3
1	1	1	1	7

Set functions

Set functions:

Boolean and pseudo-Boolean functions on $\{0, 1\}^n$ can also be viewed as *set functions*, that is, functions defined on subsets of $\{1, 2, \ldots, n\}$.

x_1	x_2	x_3	S	φ	f
0	0	0	$\emptyset$	0	4
0	0	1	$\{3\}$	1	2
0	1	0	$\{2\}$	0	-1
0	1	1	$\{2,3\}$	1	3
1	0	0	$\{1\}$	0	-5
1	0	1	$\{1,3\}$	0	6
1	1	0	$\{1,2\}$	1	3
1	1	1	$\{1,2,3\}$	1	7

Definitions

Literals

A Boolean *literal* is a variable x_i or its negation (complementation) $\overline{x}_i = 1 - x_i$.

Definitions

Literals

A Boolean *literal* is a variable x_i or its negation (complementation) $\overline{x}_i = 1 - x_i$.

Terms

A Boolean *term* (conjunction, AND) is a product of literals: $\overline{x}_1\overline{x}_2x_3,\ \overline{x}_1x_2x_3,\ \overline{x}_2x_3$, etc.

Definitions

Literals

A Boolean *literal* is a variable x_i or its negation (complementation) $\overline{x}_i = 1 - x_i$.

Terms

A Boolean *term* (conjunction, AND) is a product of literals: $\overline{x}_1\overline{x}_2x_3,\ \overline{x}_1x_2x_3,\ \overline{x}_2x_3$, etc.

Note that $\overline{x}_1\overline{x}_2x_3 = 1$ exactly when $x_1 = 0$, $x_2 = 0$, $x_3 = 1$.

Definitions

Literals

A Boolean *literal* is a variable x_i or its negation (complementation) $\overline{x}_i = 1 - x_i$.

Terms

A Boolean *term* (conjunction, AND) is a product of literals: $\overline{x}_1\overline{x}_2x_3,\ \overline{x}_1x_2x_3,\ \overline{x}_2x_3$, etc.

Note that $\overline{x}_1\overline{x}_2x_3 = 1$ exactly when $x_1 = 0$, $x_2 = 0$, $x_3 = 1$.

DNFs

A *disjunctive normal form* (DNF) is a disjunction (OR) of terms.

A DNF takes value 1 if at least one of its terms takes value 1.
Example: $\overline{x}_1\overline{x}_2x_3 \vee \overline{x}_1x_2x_3 \vee \overline{x}_2x_3$.

Representation by DNFs

DNFs:

Every Boolean function can be represented – in many ways – by a disjunctive normal form (DNF).

Representation by DNFs

DNFs:

Every Boolean function can be represented – in many ways – by a disjunctive normal form (DNF).

x_1	x_2	x_3	φ	Terms	DNFs
0	0	0	0		
0	0	1	1	$\overline{x}_1\overline{x}_2x_3$	
0	1	0	0		
0	1	1	1	$\overline{x}_1x_2x_3$	$\varphi = \overline{x}_1\overline{x}_2x_3 \vee \overline{x}_1x_2x_3 \vee x_1x_2\overline{x}_3 \vee x_1x_2x_3$
1	0	0	0		$= x_1x_2 \vee x_2x_3 \vee \overline{x}_1x_3$
1	0	1	0		
1	1	0	1	$x_1x_2\overline{x}_3$	
1	1	1	1	$x_1x_2x_3$	

Representation by ANFs

ANFs:

Every pseudo-Boolean function can be represented – in many ways – by an *arithmetic normal form* (ANF), that is, a polynomial in its literals.

Representation by ANFs

ANFs:

Every pseudo-Boolean function can be represented – in many ways – by an *arithmetic normal form* (ANF), that is, a polynomial in its literals.

x_1	x_2	x_3	f	Terms	ANFs
0	0	0	4	$\overline{x}_1\overline{x}_2\overline{x}_3$	
0	0	1	2	$\overline{x}_1\overline{x}_2 x_3$	
0	1	0	-1	$\overline{x}_1 x_2\overline{x}_3$	$f = 4\overline{x}_1\overline{x}_2\overline{x}_3 + 2\overline{x}_1\overline{x}_2 x_3 - \overline{x}_1 x_2\overline{x}_3$
0	1	1	3	$\overline{x}_1 x_2 x_3$	$+ 3\overline{x}_1 x_2 x_3 - 5x_1\overline{x}_2\overline{x}_3 + 6x_1\overline{x}_2 x_3$
1	0	0	-5	$x_1\overline{x}_2\overline{x}_3$	$3x_1 x_2\overline{x}_3 + 7x_1 x_2 x_3$
1	0	1	6	$x_1\overline{x}_2 x_3$	
1	1	0	3	$x_1 x_2\overline{x}_3$	
1	1	1	7	$x_1 x_2 x_3$	

Representation by polynomials

Substituting each $\overline{x}_i$ by $1 - x_i$ yields a polynomial representation.

Representation by polynomials

Substituting each $\overline{x}_i$ by $1 - x_i$ yields a polynomial representation.

x_1	x_2	x_3	f	Terms	ANFs
0	0	0	4	$\overline{x}_1\overline{x}_2\overline{x}_3$	
0	0	1	2	$\overline{x}_1\overline{x}_2 x_3$	
0	1	0	-1	$\overline{x}_1 x_2\overline{x}_3$	$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2$
0	1	1	3	$\overline{x}_1 x_2 x_3$	$+ 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3$
1	0	0	-5	$x_1\overline{x}_2\overline{x}_3$	
1	0	1	6	$x_1\overline{x}_2 x_3$	
1	1	0	3	$x_1 x_2\overline{x}_3$	
1	1	1	7	$x_1 x_2 x_3$	

Representation by polynomials

Polynomials:

Every pseudo-Boolean function can be represented – in a unique way – by a *multilinear polynomial* in its variables.

Representation by polynomials

Polynomials:

Every pseudo-Boolean function can be represented – in a unique way – by a *multilinear polynomial* in its variables.

Extensions:

Note: every polynomial like

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3$$

defines

- a pseudo-Boolean function on $\{0, 1\}^n$;

Representation by polynomials

Polynomials:

Every pseudo-Boolean function can be represented – in a unique way – by a *multilinear polynomial* in its variables.

Extensions:

Note: every polynomial like

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3$$

defines

- a pseudo-Boolean function on $\{0,1\}^n$;
- a function on $[0,1]^n$.

Representation by polynomials

Polynomials:

Every pseudo-Boolean function can be represented – in a unique way – by a *multilinear polynomial* in its variables.

Extensions:

Note: every polynomial like

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3$$

defines

- a pseudo-Boolean function on $\{0, 1\}^n$;
- a function on $[0, 1]^n$.

Example: $f(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}) = \frac{19}{8}$

Applications

Game theory

Applications

Game theory

- *Simple (voting) games* are Boolean functions

Applications

Game theory

- *Simple (voting) games* are Boolean functions
- *Cooperative games in characteristic form* are pseudo-Boolean functions: $f(S)$ = value that can be obtained by coalition S

Applications

Game theory

- *Simple (voting) games* are Boolean functions
- *Cooperative games in characteristic form* are pseudo-Boolean functions: $f(S)$ = value that can be obtained by coalition S
- The polynomial expression of f is called its *multilinear extension* (viewed as function on $[0,1]^n$).

Applications

Game theory

- *Simple (voting) games* are Boolean functions
- *Cooperative games in characteristic form* are pseudo-Boolean functions: $f(S)$ = value that can be obtained by coalition S
- The polynomial expression of f is called its *multilinear extension* (viewed as function on $[0,1]^n$).

Numerous applications in artificial intelligence, operations research, combinatorics, algebra, etc.

Pseudo-Boolean optimization

Complexity

Given a pseudo-Boolean function f in multilinear polynomial form, it is NP-hard to find the maximum of f.

Pseudo-Boolean optimization

Complexity

Given a pseudo-Boolean function f in multilinear polynomial form, it is NP-hard to find the maximum of f.

A useful property:

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ (continuous maximizer) is attained at a 0-1 point (discrete maximizer):

$$\max_{X\in\{0,1\}^n} f(X) = \max_{X\in[0,1]^n} f(X).$$

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0, 1]^n$ is attained at a 0-1 point:

$$\max_{X \in \{0,1\}^n} f(X) = \max_{X \in [0,1]^n} f(X).$$

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ is attained at a 0-1 point:

$$\max_{X \in \{0,1\}^n} f(X) = \max_{X \in [0,1]^n} f(X).$$

Sketch of proof: for

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3;$$

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ is attained at a 0-1 point:

$$\max_{X\in\{0,1\}^n} f(X) = \max_{X\in[0,1]^n} f(X).$$

Sketch of proof: for

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3;$$

- consider any point $(x_1^*, x_2^*, x_3^*) \in [0,1]^3$

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ is attained at a 0-1 point:

$$\max_{X\in\{0,1\}^n} f(X) = \max_{X\in[0,1]^n} f(X).$$

Sketch of proof: for

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3;$$

- consider any point $(x_1^*, x_2^*, x_3^*) \in [0,1]^3$
- fix (x_1^*, x_2^*) in f: the resulting function $g(x_3)$ is linear

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ is attained at a 0-1 point:

$$\max_{X \in \{0,1\}^n} f(X) = \max_{X \in [0,1]^n} f(X).$$

Sketch of proof: for

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3;$$

- consider any point $(x_1^*, x_2^*, x_3^*) \in [0,1]^3$
- fix (x_1^*, x_2^*) in f: the resulting function $g(x_3)$ is linear
- maximize $g(x_3)$ for x_3: this yields a 0-1 maximizer $\hat{x}_3$

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ is attained at a 0-1 point:

$$\max_{X\in\{0,1\}^n} f(X) = \max_{X\in[0,1]^n} f(X).$$

Sketch of proof: for

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3;$$

- consider any point $(x_1^*, x_2^*, x_3^*) \in [0,1]^3$
- fix (x_1^*, x_2^*) in f: the resulting function $g(x_3)$ is linear
- maximize $g(x_3)$ for x_3: this yields a 0-1 maximizer $\hat{x}_3$
- $f(x_1^*, x_2^*, \hat{x}_3) \geq f(x_1^*, x_2^*, x_3^*)$

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ is attained at a 0-1 point:

$$\max_{X\in\{0,1\}^n} f(X) = \max_{X\in[0,1]^n} f(X).$$

Sketch of proof: for

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3;$$

- consider any point $(x_1^*, x_2^*, x_3^*) \in [0,1]^3$
- fix (x_1^*, x_2^*) in f: the resulting function $g(x_3)$ is linear
- maximize $g(x_3)$ for x_3: this yields a 0-1 maximizer $\hat{x}_3$
- $f(x_1^*, x_2^*, \hat{x}_3) \geq f(x_1^*, x_2^*, x_3^*)$
- repeat for x_1, x_2.

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ is attained at a 0-1 point:

$$\max_{X\in\{0,1\}^n} f(X) = \max_{X\in[0,1]^n} f(X).$$

Sketch of proof: for

$$f = 4 - 9x_1 - 5x_2 - 2x_3 + 13x_1x_2 + 13x_1x_3 + 6x_2x_3 - 13x_1x_2x_3;$$

- consider any point $(x_1^*, x_2^*, x_3^*) \in [0,1]^3$
- fix (x_1^*, x_2^*) in f: the resulting function $g(x_3)$ is linear
- maximize $g(x_3)$ for x_3: this yields a 0-1 maximizer $\hat{x}_3$
- $f(x_1^*, x_2^*, \hat{x}_3) \geq f(x_1^*, x_2^*, x_3^*)$
- repeat for x_1, x_2.

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0, 1]^n$ is attained at a 0-1 point:

$$\max_{X \in \{0,1\}^n} f(X) = \max_{X \in [0,1]^n} f(X).$$

Pseudo-Boolean optimization

Rosenberg

The maximum of a multilinear polynomial on $[0,1]^n$ is attained at a 0-1 point:

$$\max_{X\in\{0,1\}^n} f(X) = \max_{X\in[0,1]^n} f(X).$$

Observe: the proof actually shows that, given any point $X^* \in [0,1]^n$, a better point $\hat{X} \in \{0,1\}^n$ can be found in polynomial time.

Outline

Definitions

Cuts

- undirected graph $G = (N, E)$ with $N = \{1, 2, \ldots, n\}$
- capacities $c : E \to \mathbf{R}^+$ on edges
- for $S \subseteq N$, the *cut* $\delta(S)$ is the set of edges having exactly one endpoint in S;
- the capacity of cut $\delta(S)$ is $\sum_{(i,j)\in\delta(S)} c(i,j)$.

Definitions

Cuts

- undirected graph $G = (N, E)$ with $N = \{1, 2, \ldots, n\}$
- capacities $c : E \to \mathbf{R}^+$ on edges
- for $S \subseteq N$, the *cut* $\delta(S)$ is the set of edges having exactly one endpoint in S;
- the capacity of cut $\delta(S)$ is $\sum_{(i,j)\in\delta(S)} c(i,j)$.

MAX CUT problem

Find a cut of maximum capacity in G.

Definitions

Cuts

- undirected graph $G = (N, E)$ with $N = \{1, 2, \ldots, n\}$
- capacities $c : E \to \mathbf{R}^+$ on edges
- for $S \subseteq N$, the *cut* $\delta(S)$ is the set of edges having exactly one endpoint in S;
- the capacity of cut $\delta(S)$ is $\sum_{(i,j)\in\delta(S)} c(i,j)$.

MAX CUT problem

Find a cut of maximum capacity in G.

Note: MAX CUT is NP-hard (as opposed to MIN CUT, which is polynomial).

Pseudo-Boolean formulation

Observe:

- let $x_i = 1$ if vertex i is in S, $x_i = 0$ otherwise;
- edge (i, j) is in the cut $\delta(S)$ if and only if $x_i\overline{x}_j + \overline{x}_i x_j = 1$.

Pseudo-Boolean formulation

Observe:

- let $x_i = 1$ if vertex i is in S, $x_i = 0$ otherwise;
- edge (i, j) is in the cut $\delta(S)$ if and only if $x_i\overline{x}_j + \overline{x}_i x_j = 1$.

Therefore,

MAX CUT problem

MAX CUT is equivalent to the maximization of the quadratic pseudo-Boolean function

$$f(x_1, x_2, \ldots, x_n) = \sum_{1 \le i < j \le n} c(i,j)(x_i\overline{x}_j + \overline{x}_i x_j).$$

MAX CUT and quadratic pseudo-Boolean optimization are closely related problems

Large cuts

Theorem

In every graph, there is a cut with weight at least $\frac{1}{2}\sum_{1\leq i<j\leq n} c(i,j)$ (the sum of all weights).

Pseudo-Boolean proof:

Large cuts

Theorem

In every graph, there is a cut with weight at least $\frac{1}{2}\sum_{1\leq i<j\leq n} c(i,j)$ (the sum of all weights).

Pseudo-Boolean proof:

- let $f(x_1, x_2, \ldots, x_n) = \sum_{1\leq i<j\leq n} c(i,j)(x_i\overline{x}_j + \overline{x}_i x_j)$

Large cuts

Theorem

In every graph, there is a cut with weight at least $\frac{1}{2}\sum_{1\leq i<j\leq n} c(i,j)$ (the sum of all weights).

Pseudo-Boolean proof:

- let $f(x_1, x_2, \ldots, x_n) = \sum_{1\leq i<j\leq n} c(i,j)(x_i\overline{x}_j + \overline{x}_i x_j)$
- $f(\frac{1}{2}, \ldots, \frac{1}{2}) = \sum_{1\leq i<j\leq n} c(i,j)(\frac{1}{4} + \frac{1}{4}) = \frac{1}{2}\sum_{1\leq i<j\leq n} c(i,j)$

Large cuts

Theorem

In every graph, there is a cut with weight at least $\frac{1}{2}\sum_{1\leq i<j\leq n} c(i,j)$ (the sum of all weights).

Pseudo-Boolean proof:

- let $f(x_1, x_2, \ldots, x_n) = \sum_{1\leq i<j\leq n} c(i,j)(x_i\overline{x}_j + \overline{x}_i x_j)$
- $f(\frac{1}{2}, \ldots, \frac{1}{2}) = \sum_{1\leq i<j\leq n} c(i,j)(\frac{1}{4} + \frac{1}{4}) = \frac{1}{2}\sum_{1\leq i<j\leq n} c(i,j)$
- by Rosenberg's theorem, $\max_{X\in\{0,1\}^n} f(X) = \max_{X\in[0,1]^n} f(X) \geq f(\frac{1}{2}, \ldots, \frac{1}{2})$.

Note: the large cut can be found in polynomial time.

Outline

Definitions

DNF SATISFIABILITY

SAT problem:

- Input: a DNF $\varphi(x_1, \ldots, x_n) = \bigvee_{k=1}^{m} T_k$
- Output: "Yes" if there is a point $X^* = (x_1, \ldots, x_n) \in \{0, 1\}^n$ such that $\varphi(X^*) = 0$; "No"' otherwise.

Definitions

DNF SATISFIABILITY

SAT problem:

- Input: a DNF $\varphi(x_1, \ldots, x_n) = \bigvee_{k=1}^{m} T_k$
- Output: "Yes" if there is a point $X^* = (x_1, \ldots, x_n) \in \{0,1\}^n$ such that $\varphi(X^*) = 0$; "No"' otherwise.

Note: $\varphi(X^*) = 0$ iff X^* makes *all terms* T_k of φ equal to 0, or equivalently, iff X^* makes at least one literal equal to 0 in *each term* of φ.

For example, with

$$\varphi(x_1, x_2, x_3) = \overline{x}_1\overline{x}_2 x_3 \vee \overline{x}_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_1 x_2 x_3,$$

we get: $\varphi(1, 0, 1) = 0$.

Cook's theorem

Cook's theorem

SAT is NP-complete even when each term contains at most 3 literals (3SAT).

Definitions

What if there is no X^* such that $\varphi(X^*) = 0$ (i.e., if $\varphi \equiv 1$)?

Definitions

What if there is no X^* such that $\varphi(X^*) = 0$ (i.e., if $\varphi \equiv 1$)?

Maximum Satisfiability

MAX SAT problem:

- Input: a DNF $\varphi(x_1, \ldots, x_n) = \bigvee_{k=1}^{m} T_k$
- Output: a point $X^* = (x_1, \ldots, x_n) \in \{0, 1\}^n$ which maximizes the number of terms T_k such that $T_k(X^*) = 0$.

Definitions

What if there is no X^* such that $\varphi(X^*) = 0$ (i.e., if $\varphi \equiv 1$)?

Maximum Satisfiability

MAX SAT problem:

- Input: a DNF $\varphi(x_1, \ldots, x_n) = \bigvee_{k=1}^{m} T_k$
- Output: a point $X^* = (x_1, \ldots, x_n) \in \{0, 1\}^n$ which maximizes the number of terms T_k such that $T_k(X^*) = 0$.

If SAT has a solution X^*, then X^* is optimal for MAX SAT.

Definitions

What if there is no X^* such that $\varphi(X^*) = 0$ (i.e., if $\varphi \equiv 1$)?

Maximum Satisfiability

MAX SAT problem:

- Input: a DNF $\varphi(x_1, \ldots, x_n) = \bigvee_{k=1}^{m} T_k$
- Output: a point $X^* = (x_1, \ldots, x_n) \in \{0, 1\}^n$ which maximizes the number of terms T_k such that $T_k(X^*) = 0$.

If SAT has a solution X^*, then X^* is optimal for MAX SAT. In fact:

Theorem

MAX SAT is NP-hard even when each term contains at most 2 literals (MAX 2SAT).

Weighted version

Weighted Maximum Satisfiability

Weighted MAX SAT problem:

- Input: a DNF $\varphi(x_1, \ldots, x_n) = \bigvee_{k=1}^{m} T_k$, weights $w_k \in \mathbf{R}^+$ for $k = 1, \ldots, m$.
- Output: a point $X^* = (x_1, \ldots, x_n) \in \{0, 1\}^n$ which maximizes the total weight of the terms canceled by X^*:

$$\text{maximize } \sum_{k=1}^{m} \{ w_k \mid T_k(X^*) = 0 \} \quad \text{subject to } X^* \in \{0, 1\}^n.$$

Weighted version

Weighted Maximum Satisfiability

Weighted MAX SAT problem:

- Input: a DNF $\varphi(x_1, \ldots, x_n) = \bigvee_{k=1}^{m} T_k$, weights $w_k \in \mathbf{R}^+$ for $k = 1, \ldots, m$.
- Output: a point $X^* = (x_1, \ldots, x_n) \in \{0, 1\}^n$ which maximizes the total weight of the terms canceled by X^*:

$$\text{maximize} \sum_{k=1}^{m} \{\, w_k \mid T_k(X^*) = 0 \,\} \quad \text{subject to } X^* \in \{0, 1\}^n.$$

Pseudo-Boolean formulation

MAX CUT problem

MAX SAT is equivalent to the maximization of the pseudo-Boolean function

$$f(x_1, x_2, \dots, x_n) = \sum_{k=1}^{m} w_k \left(1 - T_k \right).$$

Pseudo-Boolean formulation

MAX CUT problem

MAX SAT is equivalent to the maximization of the pseudo-Boolean function

$$f(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{m} w_k \left(1 - T_k \right).$$

Example: with equal weights and

$$\varphi(x_1, x_2, x_3) = \overline{x}_1 \overline{x}_2 x_3 \vee \overline{x}_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_1 x_2 x_3,$$

we get

$$f(x_1, x_2, x_3) = (1 - \overline{x}_1 \overline{x}_2 x_3) + (1 - \overline{x}_1 x_2 x_3) + (1 - x_1 x_2 \overline{x}_3) + (1 - x_1 x_2 x_3),$$

where $\overline{x}_i = (1 - x_i)$.

Pseudo-Boolean formulation

Pseudo-Boolean formulation allows:

- approaches by nonlinear (0-1) optimization
- approximation algorithms

Pseudo-Boolean formulation

Pseudo-Boolean formulation allows:

- approaches by nonlinear (0-1) optimization
- approximation algorithms

Recall

Approximation algorithm

An α*-algorithm* for MAX SAT is a polynomial-time algorithm which, for every instance, produces a solution $\hat{X}$ with value at least α times the optimal value:

$$\sum_{k=1}^{m} \{\, w_k \mid T_k(\hat{X}) = 0 \,\} \geq \alpha \, \mathrm{OPT}.$$

Pseudo-Boolean formulation

Pseudo-Boolean formulation allows:

- approaches by nonlinear (0-1) optimization
- approximation algorithms

Recall

Approximation algorithm

An α-*algorithm* for MAX SAT is a polynomial-time algorithm which, for every instance, produces a solution $\hat{X}$ with value at least α times the optimal value:

$$\sum_{k=1}^{m} \{ w_k \mid T_k(\hat{X}) = 0 \} \geq \alpha \, \mathrm{OPT}.$$

Approximability of MAX SAT

Johnson 1974

There is a $(1 - \frac{1}{2^d})$-approximation algorithm for the restriction of MAX SAT to DNFs in which every term has degree at least d.
In particular, there is a $\frac{1}{2}$-approximation algorithm for MAX SAT.

Approximability of MAX SAT

Johnson 1974

There is a $(1 - \frac{1}{2^d})$-approximation algorithm for the restriction of MAX SAT to DNFs in which every term has degree at least d.
In particular, there is a $\frac{1}{2}$-approximation algorithm for MAX SAT.

Pseudo-Boolean proof.

Approximability of MAX SAT

Johnson 1974

There is a $(1 - \frac{1}{2^d})$-approximation algorithm for the restriction of MAX SAT to DNFs in which every term has degree at least d.
In particular, there is a $\frac{1}{2}$-approximation algorithm for MAX SAT.

Pseudo-Boolean proof.

- let $f(x_1, \dots, x_n) = \sum_{k=1}^{m} w_k \left(1 \; - \; T_k \right)$

Approximability of MAX SAT

Johnson 1974

There is a $(1 - \frac{1}{2^d})$-approximation algorithm for the restriction of MAX SAT to DNFs in which every term has degree at least d.
In particular, there is a $\frac{1}{2}$-approximation algorithm for MAX SAT.

Pseudo-Boolean proof.

- let $f(x_1, \dots, x_n) = \sum_{k=1}^{m} w_k \left(1 - T_k \right)$
- $f(\frac{1}{2}, \dots, \frac{1}{2}) = \sum_{k=1}^{m} w_k \left(1 - (\frac{1}{2})^{|T_k|} \right) \geq (1 - \frac{1}{2^d}) \sum_{k=1}^{m} w_k$

Approximability of MAX SAT

Johnson 1974

There is a $(1 - \frac{1}{2^d})$-approximation algorithm for the restriction of MAX SAT to DNFs in which every term has degree at least d.
In particular, there is a $\frac{1}{2}$-approximation algorithm for MAX SAT.

Pseudo-Boolean proof.

- let $f(x_1, \dots, x_n) = \sum_{k=1}^{m} w_k \left(1 - T_k \right)$
- $f(\frac{1}{2}, \dots, \frac{1}{2}) = \sum_{k=1}^{m} w_k \left(1 - (\frac{1}{2})^{|T_k|} \right) \geq (1 - \frac{1}{2^d}) \sum_{k=1}^{m} w_k$
- as in Rosenberg's theorem, find $X^* \in \{0,1\}^n$ such that $f(X^*) \geq f(\frac{1}{2}, \dots, \frac{1}{2}) \geq (1 - \frac{1}{2^d}) \sum_{k=1}^{m} w_k \geq (1 - \frac{1}{2^d})$ OPT.

Approximability of MAX SAT

Subsequent improvements:

Yannakakis 1994, Goemans and Williamson 1994

There is a $\frac{3}{4}$-approximation algorithm for MAX SAT.

Approximability of MAX SAT

Subsequent improvements:

Yannakakis 1994, Goemans and Williamson 1994

There is a $\frac{3}{4}$-approximation algorithm for MAX SAT.

Sketch of pseudo-Boolean proof.

Approximability of MAX SAT

Subsequent improvements:

Yannakakis 1994, Goemans and Williamson 1994

There is a $\frac{3}{4}$-approximation algorithm for MAX SAT.

Sketch of pseudo-Boolean proof.

- let X^{LP} be the fractional solution of an LP relaxation of MAX SAT, with optimal value Z^{LP}

Approximability of MAX SAT

Subsequent improvements:

Yannakakis 1994, Goemans and Williamson 1994

There is a $\frac{3}{4}$-approximation algorithm for MAX SAT.

Sketch of pseudo-Boolean proof.

- let X^{LP} be the fractional solution of an LP relaxation of MAX SAT, with optimal value Z^{LP}
- show: $f(X^{LP}) \geq \frac{3}{4} Z^{LP} \geq \frac{3}{4}$ OPT,

Approximability of MAX SAT

Subsequent improvements:

Yannakakis 1994, Goemans and Williamson 1994

There is a $\frac{3}{4}$-approximation algorithm for MAX SAT.

Sketch of pseudo-Boolean proof.

- let X^{LP} be the fractional solution of an LP relaxation of MAX SAT, with optimal value Z^{LP}
- show: $f(X^{LP}) \geq \frac{3}{4} Z^{LP} \geq \frac{3}{4}$ OPT,
- as in Rosenberg's theorem, find $X^* \in \{0,1\}^n$ such that $f(X^*) \geq f(X^{LP})$.

Approximability of MAX SAT

Still many subsequent improvements...
Among others:

Approximability of MAX SAT

Still many subsequent improvements...
Among others:

MAX 3SAT

- (Karloff and Zwick 1997) There is an α-approximation algorithm with $\alpha = 0.875$ for MAX 3SAT.

Approximability of MAX SAT

Still many subsequent improvements...
Among others:

MAX 3SAT

- (Karloff and Zwick 1997) There is an α-approximation algorithm with $\alpha = 0.875$ for MAX 3SAT.
- (Håstad 1997) Unless P = NP, there is no α-approximation algorithm with $\alpha > 0.875$ for MAX 3SAT.

Approximability of MAX SAT

Still many subsequent improvements...
Among others:

MAX 3SAT

- (Karloff and Zwick 1997) There is an α-approximation algorithm with $\alpha = 0.875$ for MAX 3SAT.
- (Håstad 1997) Unless P = NP, there is no α-approximation algorithm with $\alpha > 0.875$ for MAX 3SAT.

MAX 2SAT

- (Lewin, Livnat and Zwick 2002) There is an α-approximation algorithm with $\alpha = 0.9401$ for MAX 2SAT.

Approximability of MAX SAT

Still many subsequent improvements...
Among others:

MAX 3SAT

- (Karloff and Zwick 1997) There is an α-approximation algorithm with $\alpha = 0.875$ for MAX 3SAT.
- (Håstad 1997) Unless P = NP, there is no α-approximation algorithm with $\alpha > 0.875$ for MAX 3SAT.

MAX 2SAT

- (Lewin, Livnat and Zwick 2002) There is an α-approximation algorithm with $\alpha = 0.9401$ for MAX 2SAT.
- (Håstad 1997) Unless P = NP, there is no α-approximation algorithm with $\alpha > 0.9545$ for MAX 2SAT.

Outline

Pseudo-Boolean optimization

Problem statement:

PB optimization

Given a pseudo-Boolean function f in multilinear polynomial form, find the maximum of f. (NP-hard)

Many applications:

Pseudo-Boolean optimization

Problem statement:

PB optimization

Given a pseudo-Boolean function f in multilinear polynomial form, find the maximum of f. (NP-hard)

Many applications:

- MAX CUT
- MAX SAT
- computer vision

Computer vision

Basic framework: given a blurred, “noisy" image, restore a “better” version.

Challenge: Restored image should be “similar” to the initial one, “smooth” in “continuous areas”, “crisp” at boundaries.

Formulation

- set $\mathcal{P}$ of *pixels* (points in $\mathbf{R}^2$)
- initial assignment of *colors* (labels) to pixels: $c_0 : \mathcal{P} \to C$
- *energy function*: for every new coloring $c : \mathcal{P} \to C$, $E(c)$ measures the deficiency of c

Formulation

- set $\mathcal{P}$ of *pixels* (points in $\mathbf{R}^2$)
- initial assignment of *colors* (labels) to pixels: $c_0 : \mathcal{P} \to C$
- *energy function*: for every new coloring $c : \mathcal{P} \to C$, $E(c)$ measures the deficiency of c

Typically:

$$E(c) = \sum_{p \in \mathcal{P}} (c_0(p) - c(p))^2 + \sum_{(p,q) \in E} V(c(p), c(q)),$$

where E is a collection of "neighboring pixels".
One may choose for instance

$$V(c(p), c(q)) = 0 \text{ if } c(p) = c(q),\ V(c(p), c(q)) = M \text{ otherwise.}$$

Black-white case

If each pixel can take one of two colors: $c(p) \in \{0, 1\}$ for all p,

Black-white case

If each pixel can take one of two colors: $c(p) \in \{0, 1\}$ for all p,

$$E(c) = \sum_{p \in \mathcal{P}} (c_0(p) - c(p))^2 + \sum_{(p,q) \in E} V(c(p), c(q))$$

is a quadratic pseudo-Boolean function (because each $V(c(p), c(q))$ depends on 2 Boolean variables only).

Black-white case

If each pixel can take one of two colors: $c(p) \in \{0, 1\}$ for all p,

$$E(c) = \sum_{p \in \mathcal{P}} (c_0(p) - c(p))^2 + \sum_{(p,q) \in E} V(c(p), c(q))$$

is a quadratic pseudo-Boolean function (because each $V(c(p), c(q))$ depends on 2 Boolean variables only).

Black-white case

Arises as subproblem for more general versions: given a coloring with C colors,

- find the best possible assignment achievable by *extending* a given color (say, green): that is, each pixel can be either colored green or maintained in its current color.
- find the best possible assignment achievable by *exchanging* two given colors (say, green and blue): that is, each green or blue pixel can be recolored either in green or in blue.

Boykov, Veksler and Zabih (2001) develop efficient heuristics based on such moves.

Quadratic optimization

The quadratic case has attracted most of the attention:

- many examples arise in this form: MAX CUT, MAX 2SAT, computer vision,...
- higher-degree cases can be efficiently reduced to quadratic.

In particular: *roof duality* framework and extensions.

Roof duality: linearization

Given: quadratic pseudo-Boolean maximization problem

$$\max f(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in E} c_{ij}\, x_i x_j.$$

Standard linearization: substitute z_{ij} for each product $x_i x_j$.

Roof duality: linearization

Given: quadratic pseudo-Boolean maximization problem

$$\max f(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in E} c_{ij}\, x_i x_j.$$

Standard linearization: substitute z_{ij} for each product $x_i x_j$.

$$
\begin{aligned}
\text{(SL)} \qquad \max \quad & \sum_{i,j)\in E} c_{ij}\, z_{ij} && (1)\\
\text{subject to} \quad & x_i \geq z_{ij} && (2)\\
& x_j \geq z_{ij} && (3)\\
& x_i + x_j \leq z_{ij} + 1 && (4)\\
& x_i \in \{0,1\} && (5)\\
& z_{ij} \in \{0,1\} && (6)
\end{aligned}
$$

Roof duality: linearization

L_2 bound

The optimal value of the linear relaxation of (SL) provides an upper-bound L_2 on OPT.

Roof duality: complementation

Another approach...
Given: quadratic pseudo-Boolean maximization problem

$$\max f(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in E} c_{ij}\, x_i x_j.$$

Write f in the form (*negaform*)

$$f(x_1, x_2, \ldots, x_n) = a_0 - \sum_{i} a_i\, \tilde{x}_i - \sum_{(i,j)} a_{ij}\, \tilde{x}_i \tilde{x}_j$$

where

- $\tilde{x}_i$ is either x_i or $\overline{x}_i$,
- $a_i \geq 0$, $a_{ij} \geq 0$ holds for all coefficients, except a_0.

e.g., $x_1 x_2 = 1 - (1 - x_2) - (1 - x_1)x_2$

Roof duality: complementation

Write f in the negaform

$$f(x_1, x_2, \ldots, x_n) = a_0 - \sum_i a_i\, \tilde{x}_i - \sum_{(i,j)} a_{ij}\, \tilde{x}_i \tilde{x}_j$$

where

- $\tilde{x}_i$ is either x_i or $\overline{x}_i$,
- $a_i \geq 0$, $a_{ij} \geq 0$ holds for all coefficients, except a_0.

Observations

- This is always possible.
- a_0 is an upper-bound on $\max f$ for every negaform of f.

Properties

Denote by C_2 the best possible upper bound derived from a negaform.

Hammer, Hansen and Simeone 1984

- Standard linearization and negaforms yield the same bound: $L_2 = C_2$

Properties

Denote by C_2 the best possible upper bound derived from a negaform.

Hammer, Hansen and Simeone 1984

- Standard linearization and negaforms yield the same bound: $L_2 = C_2$
- *Weak persistency*: if x_i takes value 1 (0) in the optimal solution of the relaxation of (SL), then it takes value 1 (0) in some maximizer of f.

Properties

Denote by C_2 the best possible upper bound derived from a negaform.

Hammer, Hansen and Simeone 1984

- Standard linearization and negaforms yield the same bound: $L_2 = C_2$
- *Weak persistency*: if x_i takes value 1 (0) in the optimal solution of the relaxation of (SL), then it takes value 1 (0) in some maximizer of f.
- *Strong persistency*: if $a_i\tilde{x}_i, a_i > 0$, is a linear term in the optimal negaform of f, then $\tilde{x}_i$ takes value 0 in all maximizers of f.

Properties

Previous approach has been extended in various ways:

- efficient computation of bounds and of persistent values: Boros and Hammer (2002), Boros, Hammer and Tavares (2005), Rother, Kolmogorov, Lempitsky and Szummer (2007), etc.
- hierarchy of improving bounds: Boros, Crama and Hammer (1990, 1992), Boros and Minoux (2009), etc.
- connections with lift-and-project, Adams-Sherali relaxations: Boros and Minoux (2009), etc.
- higher-degree polynomials (Crama 1993)

Properties

A recent application:

- remarkable success in computer vision (sparse) applications
- based on fast computation of bounds by network flows, persistency properties and further developments.

See Rother, Kolmogorov, Lempitsky and Szummer (2007), Kolmogorov and Rother (2007).

Conclusions

- Pseudo-Boolean functions are ubiquitous in many fields of application.

Conclusions

- Pseudo-Boolean functions are ubiquitous in many fields of application.
- Rich connections with graph theory (cuts, flows, stable sets,...), AI (SAT, belief functions, constraint satisfaction,...), game theory (cooperative games), algebra and matroid theory (submodularity), etc.

Conclusions

- Pseudo-Boolean functions are ubiquitous in many fields of application.
- Rich connections with graph theory (cuts, flows, stable sets,...), AI (SAT, belief functions, constraint satisfaction,...), game theory (cooperative games), algebra and matroid theory (submodularity), etc.
- Much computational and theoretical work to be done.

Conclusions

- Pseudo-Boolean functions are ubiquitous in many fields of application.
- Rich connections with graph theory (cuts, flows, stable sets,...), AI (SAT, belief functions, constraint satisfaction,...), game theory (cooperative games), algebra and matroid theory (submodularity), etc.
- Much computational and theoretical work to be done.

See Boros and Hammer (2002), Crama and Hammer (2010).

Some references

E. Boros, Y. Crama and P.L. Hammer, Upper bounds for quadratic 0–1 maximization, *Operations Research Letters* 9 (1990) 73–79.

E. Boros, Y. Crama and P.L. Hammer, Chvátal cuts and odd cycle inequalities in quadratic 0–1 optimization, *SIAM Journal on Discrete Mathematics* 5 (1992) 163–177.

E. Boros and P.L. Hammer, Pseudo-Boolean optimization, *Discrete Applied Mathematics* 123 (2002) 155–225.

Y. Boykov, O. Veksler and R. Zabih, Fast approximate energy minimization via graph cuts, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001) 1222–1239.

Some references

Y. Crama, Concave extensions for nonlinear 0–1 maximization problems, *Mathematical Programming* 61 (1993) 53–60.

Y. Crama and P.L. Hammer, eds., *Boolean Functions: Theory, Algorithms, and Applications*, Cambridge University Press, New York, to appear.

V. Kolmogorov and C. Rother, Minimizing non-submodular functions with graph cuts - A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007) 1274–1279.

C. Rother, V. Kolmogorov, V. Lempitsky and M. Szummer, Optimizing binary MRFs via extended roof duality, in: *IEEE Conference on Computer Vision and Pattern Recognition* June 2007.