On the Usefulness of Predicates

Per Austrin
austrin@cs.toronto.edu

University of Toronto

Fields Workshop on CSPs, 2011-08-15

(joint work with Johan Håstad)

MAX 4-LIN

The MAX 4-LIN problem:

$$
\begin{array}{lclc}
(x_1 \oplus \overline{x_2} \oplus x_3 \oplus x_4) & \wedge & (\overline{x_1} \oplus x_2 \oplus \overline{x_3} \oplus \overline{x_4}) & \wedge \\
(x_1 \oplus x_2 \oplus \overline{x_3} \oplus \overline{x_5}) & \wedge & (x_1 \oplus \overline{x_2} \oplus x_4 \oplus \overline{x_5}) & \wedge \\
(\overline{x_1} \oplus \overline{x_2} \oplus x_4 \oplus x_5) & \wedge & (x_1 \oplus \overline{x_2} \oplus x_4 \oplus \overline{x_6}) & \wedge \\
(x_1 \oplus \overline{x_2} \oplus x_5 \oplus x_6) & \wedge & (x_1 \oplus x_3 \oplus \overline{x_4} \oplus \overline{x_5}) & \wedge \\
(\overline{x_2} \oplus x_3 \oplus \overline{x_4} \oplus \overline{x_6}) & \wedge & (\overline{x_2} \oplus \overline{x_3} \oplus \overline{x_4} \oplus \overline{x_6}) & \wedge \\
(\overline{x_2} \oplus x_3 \oplus \overline{x_5} \oplus \overline{x_6}) & \wedge & (x_2 \oplus \overline{x_4} \oplus x_5 \oplus x_6) & \wedge \\
(x_3 \oplus x_4 \oplus x_5 \oplus \overline{x_6}) & \wedge & (\overline{x_3} \oplus x_4 \oplus \overline{x_5} \oplus \overline{x_6}) &
\end{array}
$$

The MAX 4-LIN problem:

$$
\begin{array}{llll}
(x_1 \oplus \overline{x_2} \oplus x_3 \oplus x_4) & \wedge & (\overline{x_1} \oplus x_2 \oplus \overline{x_3} \oplus \overline{x_4}) & \wedge \\
(x_1 \oplus x_2 \oplus \overline{x_3} \oplus \overline{x_5}) & \wedge & (x_1 \oplus \overline{x_2} \oplus x_4 \oplus \overline{x_5}) & \wedge \\
(\overline{x_1} \oplus \overline{x_2} \oplus x_4 \oplus x_5) & \wedge & (x_1 \oplus \overline{x_2} \oplus x_4 \oplus \overline{x_6}) & \wedge \\
(x_1 \oplus \overline{x_2} \oplus x_5 \oplus x_6) & \wedge & (x_1 \oplus x_3 \oplus \overline{x_4} \oplus \overline{x_5}) & \wedge \\
(\overline{x_2} \oplus x_3 \oplus \overline{x_4} \oplus \overline{x_6}) & \wedge & (\overline{x_2} \oplus \overline{x_3} \oplus \overline{x_4} \oplus \overline{x_6}) & \wedge \\
(\overline{x_2} \oplus x_3 \oplus \overline{x_5} \oplus \overline{x_6}) & \wedge & (x_2 \oplus \overline{x_4} \oplus x_5 \oplus x_6) & \wedge \\
(x_3 \oplus x_4 \oplus x_5 \oplus \overline{x_6}) & \wedge & (\overline{x_3} \oplus x_4 \oplus \overline{x_5} \oplus \overline{x_6}) &
\end{array}
$$

$$
\begin{array}{lcl}
x_1 & = & \text{FALSE} \\
x_2 & = & \text{FALSE} \\
x_3 & = & \text{FALSE} \\
x_4 & = & \text{FALSE} \\
x_5 & = & \text{TRUE} \\
x_6 & = & \text{TRUE}
\end{array}
$$

Hardness of MAX 4-LIN

By [Hås01], it is NP-hard to

find assignment α satisfying $1/2 + \epsilon$ equations
assuming there is some α satisfying $1 - \epsilon$ equations.

NP-hard to do anything non-trivial!

Example 2: Max GLST

$$GLST(x_1, x_2, x_3, x_4) = \begin{cases} x_2 \oplus x_3 & \text{if } x_1 = \text{False} \\ x_2 \oplus x_4 & \text{if } x_1 = \text{True} \end{cases}$$

The Max GLST problem:

$$\begin{array}{lclc}
GLST(x_1, \overline{x_2}, x_3, x_4) & \wedge & GLST(\overline{x_1}, x_2, \overline{x_3}, \overline{x_4}) & \wedge \\
GLST(x_1, x_2, \overline{x_3}, \overline{x_5}) & \wedge & GLST(x_1, \overline{x_2}, x_4, \overline{x_5}) & \wedge \\
GLST(\overline{x_1}, \overline{x_2}, x_4, x_5) & \wedge & GLST(x_1, \overline{x_2}, x_4, \overline{x_6}) & \wedge \\
GLST(x_1, \overline{x_2}, x_5, x_6) & \wedge & GLST(x_1, x_3, \overline{x_4}, \overline{x_5}) & \wedge \\
GLST(\overline{x_2}, x_3, \overline{x_4}, \overline{x_6}) & \wedge & GLST(\overline{x_2}, \overline{x_3}, \overline{x_4}, \overline{x_6}) & \wedge \\
GLST(\overline{x_2}, x_3, \overline{x_5}, \overline{x_6}) & \wedge & GLST(x_2, \overline{x_4}, x_5, x_6) & \wedge \\
GLST(x_3, x_4, x_5, \overline{x_6}) & \wedge & GLST(\overline{x_3}, x_4, \overline{x_5}, \overline{x_6}) &
\end{array}$$

Example 2: MAX GLST

$$GLST(x_1, x_2, x_3, x_4) = \begin{cases} x_2 \oplus x_3 & \text{if } x_1 = \text{False} \\ x_2 \oplus x_4 & \text{if } x_1 = \text{True} \end{cases}$$

The MAX GLST problem:

$$\begin{array}{llll}
GLST(x_1, \overline{x_2}, x_3, x_4) & \wedge & GLST(\overline{x_1}, x_2, \overline{x_3}, \overline{x_4}) & \wedge \\
GLST(x_1, x_2, \overline{x_3}, \overline{x_5}) & \wedge & GLST(x_1, \overline{x_2}, x_4, \overline{x_5}) & \wedge \\
GLST(\overline{x_1}, \overline{x_2}, x_4, x_5) & \wedge & GLST(x_1, \overline{x_2}, x_4, \overline{x_6}) & \wedge \\
GLST(x_1, \overline{x_2}, x_5, x_6) & \wedge & GLST(x_1, x_3, \overline{x_4}, \overline{x_5}) & \wedge \\
GLST(\overline{x_2}, x_3, \overline{x_4}, \overline{x_6}) & \wedge & GLST(\overline{x_2}, \overline{x_3}, \overline{x_4}, \overline{x_6}) & \wedge \\
GLST(\overline{x_2}, x_3, \overline{x_5}, \overline{x_6}) & \wedge & GLST(x_2, \overline{x_4}, x_5, x_6) & \wedge \\
GLST(x_3, x_4, x_5, \overline{x_6}) & \wedge & GLST(\overline{x_3}, x_4, \overline{x_5}, \overline{x_6}) &
\end{array}$$

$$\begin{array}{lcl}
x_1 & = & \text{FALSE} \\
x_2 & = & \text{TRUE} \\
x_3 & = & \text{TRUE} \\
x_4 & = & \text{TRUE} \\
x_5 & = & \text{TRUE} \\
x_6 & = & \text{TRUE}
\end{array}$$

Hardness of MAX GLST

By [GLST'98], it is NP-hard to

find assignment α satisfying $1/2 + \epsilon$ constraints
assuming there is some α satisfying $1 - \epsilon$ constraints.

NP-hard to do anything non-trivial!

Give up?

Ok so these problems appear as hard as they can be

Give up?

Ok so these problems appear as hard as they can be

Let us relax the conditions of the algorithm

Give up?

Ok so these problems appear as hard as they can be

Let us relax the conditions of the algorithm

1. The algorithm chooses $Q : \{0, 1\}^k \to [0, 1]$

Give up?

Ok so these problems appear as hard as they can be

Let us relax the conditions of the algorithm

1. The algorithm chooses $Q : \{0,1\}^k \to [0,1]$
2. Instead of maximizing

$$\sum P(\ldots)$$

   seek to maximize

$$\sum Q(\ldots)$$

Notation

instance is pair (P, L) for

- k-ary predicate $P : \{0,1\}^k \to \{0,1\}$
- list of k-tuples of literals L

Notation

instance is pair (P, L) for

- k-ary predicate $P : \{0,1\}^k \to \{0,1\}$
- list of k-tuples of literals L

optimum $\mathrm{Opt}(P, L) \in [0,1]$ max fraction satisfied constraints

Notation

instance is pair (P, L) for
- k-ary predicate $P : \{0,1\}^k \to \{0,1\}$
- list of k-tuples of literals L

optimum $\mathrm{Opt}(P, L) \in [0, 1]$ max fraction satisfied constraints

expectation $E_P = \mathbb{E}[P(x)]$ over uniform x

Usefulness

$P : \{0,1\}^k \to \{0,1\}$ is *useful* for $Q : \{0,1\}^k \to [0,1]$ if we can

find assignment α to (Q, L) with value $E_Q + \epsilon$

assuming $\mathrm{Opt}(P, L) \geq 1 - \epsilon$.

Usefulness

$P : \{0,1\}^k \to \{0,1\}$ is *useful* for $Q : \{0,1\}^k \to [0,1]$ if we can

find assignment α to (Q, L) with value $E_Q + \epsilon$

assuming $\mathrm{Opt}(P, L) \geq 1 - \epsilon$.

P is *useless* if it is not useful for any Q.

Motivation

If you can’t win, change the rules.

Motivation

If you can't win, change the rules.

Some analogues

Motivation

If you can’t win, change the rules.

Some analogues

- Balanced separators

Motivation

If you can’t win, change the rules.

Some analogues

- Balanced separators
- Degree bounded spanning trees

Motivation

If you can't win, change the rules.

Some analogues

- Balanced separators
- Degree bounded spanning trees
- Learning concept class FOO by concept class BAR

Motivation

If you can't win, change the rules.

Some analogues

- Balanced separators
- Degree bounded spanning trees
- Learning concept class FOO by concept class BAR

Typically Q would be relaxation of P or at least have something to do with P

- E.g., "weak majority" instead of "strong majority"

But we're generous and allow any Q

Example 1

$$GLST(x_1, x_2, x_3, x_4) = \begin{cases} x_2 \oplus x_3 & \text{if } x_1 = \text{False} \\ x_2 \oplus x_4 & \text{if } x_1 = \text{True} \end{cases}$$

$$\begin{array}{lclcl} \color{green}{GLST(x_1, \overline{x_2}, x_3, x_4)} & \wedge & \color{green}{GLST(\overline{x_1}, x_2, \overline{x_3}, \overline{x_4})} & \wedge \\ \color{green}{GLST(x_1, x_2, \overline{x_3}, \overline{x_5})} & \wedge & \color{green}{GLST(x_1, \overline{x_2}, x_4, \overline{x_5})} & \wedge \\ \color{green}{GLST(\overline{x_1}, \overline{x_2}, x_4, x_5)} & \wedge & \color{green}{GLST(x_1, \overline{x_2}, x_4, \overline{x_6})} & \wedge \\ \color{green}{GLST(x_1, \overline{x_2}, x_5, x_6)} & \wedge & \color{green}{GLST(x_1, x_3, \overline{x_4}, \overline{x_5})} & \wedge \\ \color{green}{GLST(\overline{x_2}, x_3, \overline{x_4}, \overline{x_6})} & \wedge & \color{red}{GLST(\overline{x_2}, \overline{x_3}, \overline{x_4}, \overline{x_6})} & \wedge \\ \color{green}{GLST(\overline{x_2}, x_3, \overline{x_5}, \overline{x_6})} & \wedge & \color{green}{GLST(x_2, \overline{x_4}, x_5, x_6)} & \wedge \\ \color{green}{GLST(x_3, x_4, x_5, \overline{x_6})} & \wedge & \color{green}{GLST(\overline{x_3}, x_4, \overline{x_5}, \overline{x_6})} & \end{array}$$

$$\begin{array}{lcl} x_1 & = & \text{FALSE} \\ x_2 & = & \text{TRUE} \\ x_3 & = & \text{TRUE} \\ x_4 & = & \text{TRUE} \\ x_5 & = & \text{TRUE} \\ x_6 & = & \text{TRUE} \end{array}$$

Example 1

$$GLST(x_1, x_2, x_3, x_4) = \begin{cases} x_2 \oplus x_3 & \text{if } x_1 = \text{False} \\ x_2 \oplus x_4 & \text{if } x_1 = \text{True} \end{cases}$$

$$\begin{array}{llll}
NAE(\quad, \overline{x_2}, x_3, x_4) & \wedge & NAE(\quad, x_2, \overline{x_3}, \overline{x_4}) & \wedge \\
NAE(\quad, x_2, \overline{x_3}, \overline{x_5}) & \wedge & NAE(\quad, \overline{x_2}, x_4, \overline{x_5}) & \wedge \\
NAE(\quad, \overline{x_2}, x_4, x_5) & \wedge & NAE(\quad, \overline{x_2}, x_4, \overline{x_6}) & \wedge \\
NAE(\quad, \overline{x_2}, x_5, x_6) & \wedge & NAE(\quad, x_3, \overline{x_4}, \overline{x_5}) & \wedge \\
NAE(\quad, x_3, \overline{x_4}, \overline{x_6}) & \wedge & NAE(\quad, \overline{x_3}, \overline{x_4}, \overline{x_6}) & \wedge \\
NAE(\quad, x_3, \overline{x_5}, \overline{x_6}) & \wedge & NAE(\quad, \overline{x_4}, x_5, x_6) & \wedge \\
NAE(\quad, x_4, x_5, \overline{x_6}) & \wedge & NAE(\quad, x_4, \overline{x_5}, \overline{x_6}) &
\end{array}$$

$$\begin{array}{lcl}
x_1 & = & \text{FALSE} \\
x_2 & = & \text{TRUE} \\
x_3 & = & \text{TRUE} \\
x_4 & = & \text{TRUE} \\
x_5 & = & \text{TRUE} \\
x_6 & = & \text{TRUE}
\end{array}$$

(NAE = Not All Equal)

Example 1

$$GLST(x_1, x_2, x_3, x_4) = \begin{cases} x_2 \oplus x_3 & \text{if } x_1 = \text{False} \\ x_2 \oplus x_4 & \text{if } x_1 = \text{True} \end{cases}$$

$$\begin{array}{llll}
NAE(\quad, \overline{x_2}, x_3, x_4) & \wedge & NAE(\quad, x_2, \overline{x_3}, \overline{x_4}) & \wedge \\
NAE(\quad, x_2, \overline{x_3}, \overline{x_5}) & \wedge & NAE(\quad, \overline{x_2}, x_4, \overline{x_5}) & \wedge \\
NAE(\quad, \overline{x_2}, x_4, x_5) & \wedge & NAE(\quad, \overline{x_2}, x_4, \overline{x_6}) & \wedge \\
NAE(\quad, \overline{x_2}, x_5, x_6) & \wedge & NAE(\quad, x_3, \overline{x_4}, \overline{x_5}) & \wedge \\
NAE(\quad, x_3, \overline{x_4}, \overline{x_6}) & \wedge & NAE(\quad, \overline{x_3}, \overline{x_4}, \overline{x_6}) & \wedge \\
NAE(\quad, x_3, \overline{x_5}, \overline{x_6}) & \wedge & NAE(\quad, \overline{x_4}, x_5, x_6) & \wedge \\
NAE(\quad, x_4, x_5, \overline{x_6}) & \wedge & NAE(\quad, x_4, \overline{x_5}, \overline{x_6}) &
\end{array}$$

$$\begin{array}{lcl}
x_1 & = & \text{FALSE} \\
x_2 & = & \text{TRUE} \\
x_3 & = & \text{TRUE} \\
x_4 & = & \text{TRUE} \\
x_5 & = & \text{TRUE} \\
x_6 & = & \text{TRUE}
\end{array}$$

(NAE = Not All Equal)

$GLST(x_1, x_2, x_3, x_4)$ is useful for $NAE(\cdot, x_2, x_3, x_4)$.

Example 2

What about MAX 4-LIN?

$$
\begin{array}{lclcl}
(x_1 \oplus \overline{x_2} \oplus x_3 \oplus x_4) & \wedge & (\overline{x_1} \oplus x_2 \oplus \overline{x_3} \oplus \overline{x_4}) & \wedge \\
(x_1 \oplus x_2 \oplus \overline{x_3} \oplus \overline{x_5}) & \wedge & (x_1 \oplus \overline{x_2} \oplus x_4 \oplus \overline{x_5}) & \wedge \\
(\overline{x_1} \oplus \overline{x_2} \oplus x_4 \oplus x_5) & \wedge & (x_1 \oplus \overline{x_2} \oplus x_4 \oplus \overline{x_6}) & \wedge \\
(x_1 \oplus \overline{x_2} \oplus x_5 \oplus x_6) & \wedge & (x_1 \oplus x_3 \oplus \overline{x_4} \oplus \overline{x_5}) & \wedge \\
(\overline{x_2} \oplus x_3 \oplus \overline{x_4} \oplus \overline{x_6}) & \wedge & (\overline{x_2} \oplus \overline{x_3} \oplus \overline{x_4} \oplus \overline{x_6}) & \wedge \\
(\overline{x_2} \oplus x_3 \oplus \overline{x_5} \oplus \overline{x_6}) & \wedge & (x_2 \oplus \overline{x_4} \oplus x_5 \oplus x_6) & \wedge \\
(x_3 \oplus x_4 \oplus x_5 \oplus \overline{x_6}) & \wedge & (\overline{x_3} \oplus x_4 \oplus \overline{x_5} \oplus \overline{x_6}) &
\end{array}
$$

$$
\begin{array}{lcl}
x_1 & = & \text{FALSE} \\
x_2 & = & \text{FALSE} \\
x_3 & = & \text{FALSE} \\
x_4 & = & \text{FALSE} \\
x_5 & = & \text{TRUE} \\
x_6 & = & \text{TRUE}
\end{array}
$$

Example 2

What about MAX 4-LIN?

$$
\begin{array}{llll}
(x_1 \oplus \overline{x_2} \oplus x_3 \oplus x_4) & \wedge & (\overline{x_1} \oplus x_2 \oplus \overline{x_3} \oplus \overline{x_4}) & \wedge \\
(x_1 \oplus x_2 \oplus \overline{x_3} \oplus \overline{x_5}) & \wedge & (x_1 \oplus \overline{x_2} \oplus x_4 \oplus \overline{x_5}) & \wedge \\
(\overline{x_1} \oplus \overline{x_2} \oplus x_4 \oplus x_5) & \wedge & (x_1 \oplus \overline{x_2} \oplus x_4 \oplus \overline{x_6}) & \wedge \\
(x_1 \oplus \overline{x_2} \oplus x_5 \oplus x_6) & \wedge & (x_1 \oplus x_3 \oplus \overline{x_4} \oplus \overline{x_5}) & \wedge \\
(\overline{x_2} \oplus x_3 \oplus \overline{x_4} \oplus \overline{x_6}) & \wedge & (\overline{x_2} \oplus \overline{x_3} \oplus \overline{x_4} \oplus \overline{x_6}) & \wedge \\
(\overline{x_2} \oplus x_3 \oplus \overline{x_5} \oplus \overline{x_6}) & \wedge & (x_2 \oplus \overline{x_4} \oplus x_5 \oplus x_6) & \wedge \\
(x_3 \oplus x_4 \oplus x_5 \oplus \overline{x_6}) & \wedge & (\overline{x_3} \oplus x_4 \oplus \overline{x_5} \oplus \overline{x_6}) &
\end{array}
$$

$$
\begin{array}{lcl}
x_1 & = & \text{FALSE} \\
x_2 & = & \text{FALSE} \\
x_3 & = & \text{FALSE} \\
x_4 & = & \text{FALSE} \\
x_5 & = & \text{TRUE} \\
x_6 & = & \text{TRUE}
\end{array}
$$

MAX 4-LIN *is useless*
– no matter what objective we use we can't do anything useful

Variations

- *adaptive* or *non-adaptive* usefulness:
  does the algorithm choose Q before or after seeing L?

Variations

- *adaptive* or *non-adaptive* usefulness:
  does the algorithm choose Q before or after seeing L?
  – turns out to make very little difference

Variations

- *adaptive* or *non-adaptive* usefulness:
  does the algorithm choose Q before or after seeing L?
  – turns out to make very little difference

- *computational* or *information-theoretic* usefulness.
  – will focus on computational

Variations

- *adaptive* or *non-adaptive* usefulness:
  does the algorithm choose Q before or after seeing L?
  – turns out to make very little difference
- *computational* or *information-theoretic* usefulness.
  – will focus on computational
- more *general classes of CSPs*
  – more on that later

Quick Conclusions

Most approximation resistance results in fact show the stronger property of uselessness

- [Hås'01, ST'99, EH'08, ST'06, AM'09]

Quick Conclusions

Most approximation resistance results in fact show the stronger property of uselessness

- [Hås'01, ST'99, EH'08, ST'06, AM'09]
- but clearly not [GLST'98]

Quick Conclusions

Most approximation resistance results in fact show the stronger property of uselessness

- [Hås'01, ST'99, EH'08, ST'06, AM'09]
- but clearly not [GLST'98]

In particular from [AM'09] follows that:

assuming UGC, *P is useless if it supports a pairwise independent distribution*.

(there exists pairwise independent distribution μ such that $\mathit{Supp}(\mu) \subseteq P^{-1}(1)$)

A dictatorship test

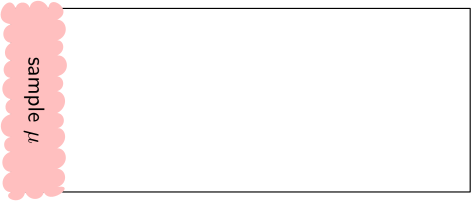
Given $f : \{0, 1\}^n \to \{0, 1\}$

1. Pick random $k \times n$ matrix X over $\{0, 1\}$

A dictatorship test

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

1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently



A dictatorship test

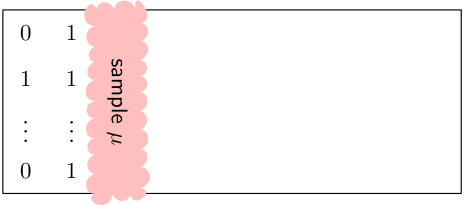
Given $f : \{0,1\}^n \to \{0,1\}$

1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently

A dictatorship test

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

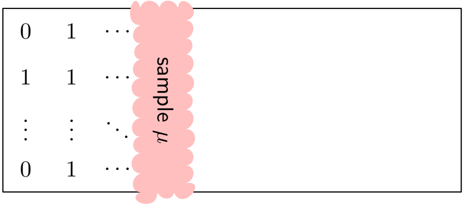
1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently



A dictatorship test

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

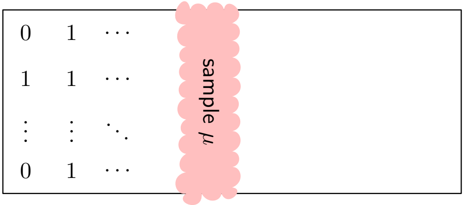
1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently



A dictatorship test

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

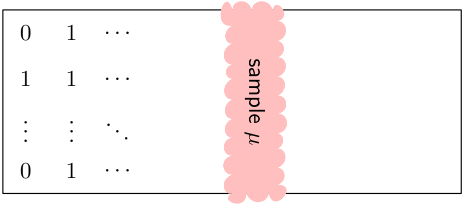
1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently



A dictatorship test

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

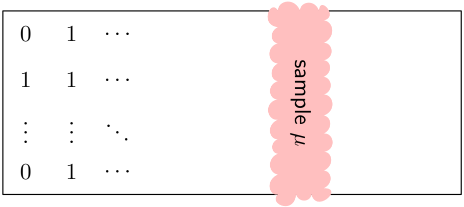
1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently



A dictatorship test

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

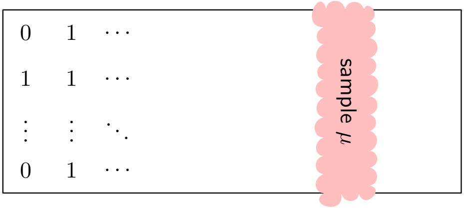
1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently



A dictatorship test

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

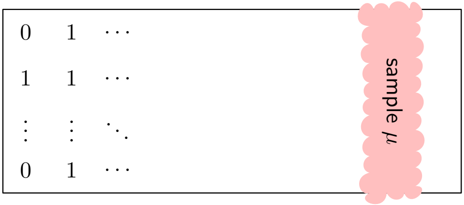
1. Pick random $k \times n$ matrix X over $\{0, 1\}$, each column sampled from μ, independently



A dictatorship test

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

1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently



A dictatorship test

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

1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently

$$\begin{array}{|cccccc}
\hline
0 & 1 & \cdots & & \cdots \\
1 & 1 & \cdots & & \cdots \\
\vdots & \vdots & \ddots & & \ddots \\
0 & 1 & \cdots & & \cdots \\
\hline
\end{array}$$

sample μ

A dictatorship test

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

1. Pick random $k \times n$ matrix X over $\{0, 1\}$, each column sampled from μ, independently

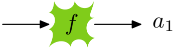
$$\boxed{\begin{array}{ccccc} 0 & 1 & \cdots & \cdots & 0 \\ 1 & 1 & \cdots & \cdots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 1 & \cdots & \cdots & 0 \end{array}}$$

A dictatorship test

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

1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently
2. Let $a = (a_1, \ldots, a_k)$, where $a_i = f(X_i)$

$$\begin{array}{cccccc} 0 & 1 & \cdots & & \cdots & 0 \\ 1 & 1 & \cdots & & \cdots & 1 \\ \vdots & \vdots & \ddots & & \ddots & \vdots \\ 0 & 1 & \cdots & & \cdots & 0 \end{array}$$



A dictatorship test

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

1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently
2. Let $a = (a_1, \ldots, a_k)$, where $a_i = f(X_i)$

$$\begin{array}{|cccccc|}
\hline
0 & 1 & \cdots & & \cdots & 0 \\
1 & 1 & \cdots & & \cdots & 1 \\
\vdots & \vdots & \ddots & & \ddots & \vdots \\
0 & 1 & \cdots & & \cdots & 0 \\
\hline
\end{array}
\qquad
\begin{array}{l}
 a_1 \\
\longrightarrow f \longrightarrow a_2
\end{array}$$

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

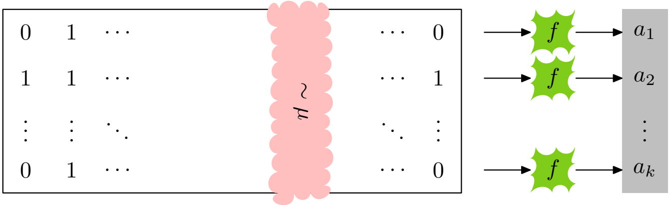
1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently
2. Let $a = (a_1, \ldots, a_k)$, where $a_i = f(X_i)$

$$
\begin{array}{|cccccc|ccc}
\hline
0 & 1 & \cdots & & \cdots & 0 & & & a_1 \\
1 & 1 & \cdots & & \cdots & 1 & & & a_2 \\
\vdots & \vdots & \ddots & & \ddots & \vdots & & & \vdots \\
0 & 1 & \cdots & & \cdots & 0 & \longrightarrow & f \longrightarrow & a_k \\
\hline
\end{array}
$$

A dictatorship test

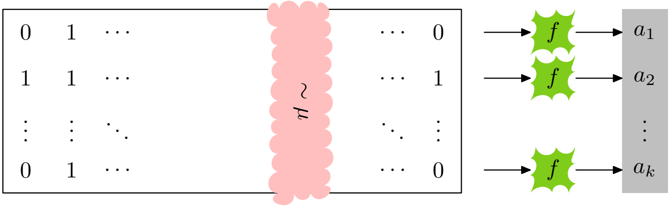
Given $f : \{0,1\}^n \to \{0,1\}$

1. Pick random $k \times n$ matrix X over $\{0,1\}$, each column sampled from μ, independently
2. Let $a = (a_1, \ldots, a_k)$, where $a_i = f(X_i)$
3. Output $(a_1, \ldots, a_k)$



A dictatorship test

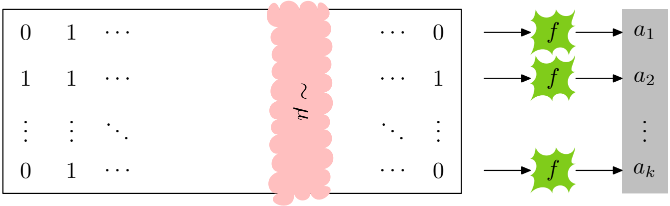
Analysis: study the distribution η of $(a_1, \ldots, a_k)$



A dictatorship test

Analysis: study the distribution η of $(a_1, \ldots, a_k)$

Completeness If f is a dictator, then $\eta = \mu$ so $\mathbb{E}_\eta[P] = 1$

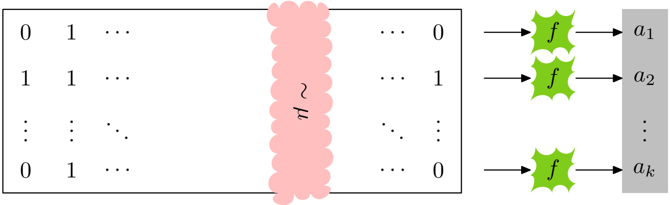


A dictatorship test

Analysis: study the distribution η of $(a_1, \ldots, a_k)$

Completeness If f is a dictator, then $\eta = \mu$ so $\mathbb{E}_\eta[P] = 1$

Soundness If f is far from dictator, then $\eta \approx$ uniform



Intuition behind soundness

- If f is far from dictator, can apply *invariance principle* [MOO05, Mos07]

Intuition behind soundness

- If f is far from dictator, can apply *invariance principle* [MOO05, Mos07]
- Distribution of $(f(X_1), \ldots, f(X_k))$ does not change if μ *replaced by distribution* μ' with *same first and second moments*

Intuition behind soundness

- If f is far from dictator, can apply *invariance principle* [MOO05, Mos07]
- Distribution of $(f(X_1), \ldots, f(X_k))$ does not change if μ *replaced by distribution* μ' with *same first and second moments*
- In particular can use μ' = *uniform*

A Converse

Theorem

If P does not support a pairwise independent distribution there is a Q for which P is useful.

A Converse

Theorem

If P does not support a pairwise independent distribution there is a Q for which P is useful.

Gives complete characterization, assuming UGC.

Proof Sketch

Fact

If P does not support pairwise independence then there is a quadratic function $Q : \{0,1\}^k \to [0,1]$ such that $Q(x) > E_Q$ for all $x \in P^{-1}(1)$.

Proof Sketch

Fact

If P does not support pairwise independence then there is a quadratic function $Q : \{0,1\}^k \to [0,1]$ such that $Q(x) > E_Q$ for all $x \in P^{-1}(1)$.

In particular there is $\epsilon > 0$ such that if $\mathrm{Opt}(P, L) \geq 1 - \epsilon$ then $\mathrm{Opt}(Q, L) \geq E_Q + \epsilon$

Proof Sketch

Fact

If P does not support pairwise independence then there is a quadratic function $Q : \{0,1\}^k \to [0,1]$ such that $Q(x) > E_Q$ for all $x \in P^{-1}(1)$.

In particular there is $\epsilon > 0$ such that if $\mathrm{Opt}(P, L) \geq 1 - \epsilon$ then $\mathrm{Opt}(Q, L) \geq E_Q + \epsilon$

Q is *quadratic* so we can use *standard SDP techniques* to find assignment with value $E_Q + \epsilon'$

Without negations

What if we don't have negations?

E.g., MAX CUT, MAX 4-LIN$^+$

$$
\begin{array}{llll}
(x_1 \oplus x_2 \oplus x_3 \oplus x_4) & \wedge & (x_1 \oplus x_2 \oplus x_3 \oplus x_7) & \wedge \\
(x_1 \oplus x_2 \oplus x_6 \oplus x_7) & \wedge & (x_1 \oplus x_3 \oplus x_4 \oplus x_6) & \wedge \\
(x_1 \oplus x_3 \oplus x_5 \oplus x_7) & \wedge & (x_1 \oplus x_3 \oplus x_6 \oplus x_7) & \wedge \\
(x_1 \oplus x_4 \oplus x_5 \oplus x_6) & \wedge & (x_1 \oplus x_5 \oplus x_6 \oplus x_7) & \wedge \\
(x_2 \oplus x_3 \oplus x_4 \oplus x_5) & \wedge & (x_2 \oplus x_3 \oplus x_4 \oplus x_6) & \wedge \\
(x_2 \oplus x_3 \oplus x_5 \oplus x_7) & \wedge & (x_2 \oplus x_3 \oplus x_6 \oplus x_7) & \wedge \\
(x_2 \oplus x_5 \oplus x_6 \oplus x_7) & \wedge & (x_3 \oplus x_4 \oplus x_6 \oplus x_7) &
\end{array}
$$

Without negations

What if we don't have negations?

E.g., MAX CUT, MAX 4-LIN^{+}

$$
\begin{array}{lll}
(x_1 \oplus x_2 \oplus x_3 \oplus x_4) & \wedge & (x_1 \oplus x_2 \oplus x_3 \oplus x_7) & \wedge \\
(x_1 \oplus x_2 \oplus x_6 \oplus x_7) & \wedge & (x_1 \oplus x_3 \oplus x_4 \oplus x_6) & \wedge \\
(x_1 \oplus x_3 \oplus x_5 \oplus x_7) & \wedge & (x_1 \oplus x_3 \oplus x_6 \oplus x_7) & \wedge \\
(x_1 \oplus x_4 \oplus x_5 \oplus x_6) & \wedge & (x_1 \oplus x_5 \oplus x_6 \oplus x_7) & \wedge \\
(x_2 \oplus x_3 \oplus x_4 \oplus x_5) & \wedge & (x_2 \oplus x_3 \oplus x_4 \oplus x_6) & \wedge \\
(x_2 \oplus x_3 \oplus x_5 \oplus x_7) & \wedge & (x_2 \oplus x_3 \oplus x_6 \oplus x_7) & \wedge \\
(x_2 \oplus x_5 \oplus x_6 \oplus x_7) & \wedge & (x_3 \oplus x_4 \oplus x_6 \oplus x_7) &
\end{array}
$$

$$
\begin{array}{lcl}
x_1 & = & \text{FALSE} \\
x_2 & = & \text{FALSE} \\
x_3 & = & \text{FALSE} \\
x_4 & = & \text{TRUE} \\
x_5 & = & \text{FALSE} \\
x_6 & = & \text{FALSE} \\
x_7 & = & \text{TRUE}
\end{array}
$$

Without negations

What if we don't have negations?

E.g., MAX CUT, MAX 4-LIN$^+$

A *caveat*: need to be careful about how to define what is *trivial*

Without negations

What if we don't have negations?

E.g., MAX CUT, MAX 4-LIN$^+$

A *caveat*: need to be careful about how to define what is *trivial*

Natural definition

$$E_Q^+ = \max_{p \in [0,1]} \mathop{\mathbb{E}}_p[Q(x)]$$

expectation under p-biased distribution

Without negations

What if we don't have negations?

E.g., MAX CUT, MAX 4-LIN$^+$

A *caveat*: need to be careful about how to define what is *trivial*

Natural definition

$$E_Q^+ = \max_{p \in [0,1]} \mathbb{E}_p[Q(x)]$$

expectation under p-biased distribution

Trivial to achieve value E_Q^+

Without negations

Definition

A distribution μ over $\{0,1\}^k$ is *pairwise symmetric* if there is b, ρ such that $\mathbb{E}_\mu[x_i] = b$ for all i and $\mathbb{E}_\mu[x_i x_j] = \rho$ for all $i \neq j$.

Without negations

Definition

A distribution μ over $\{0,1\}^k$ is *pairwise symmetric* if there is b, ρ such that $\mathbb{E}_\mu[x_i] = b$ for all i and $\mathbb{E}_\mu[x_i x_j] = \rho$ for all $i \neq j$.

Positively pairwise symmetric if $\rho \geq b^2$

Without negations

Definition

A distribution μ over $\{0,1\}^k$ is *pairwise symmetric* if there is b, ρ such that $\mathbb{E}_\mu[x_i] = b$ for all i and $\mathbb{E}_\mu[x_i x_j] = \rho$ for all $i \neq j$.

Positively pairwise symmetric if $\rho \geq b^2$

E.g., (possibly biased) pairwise independent distribution

Without negations

Definition

A distribution μ over $\{0,1\}^k$ is *pairwise symmetric* if there is b, ρ such that $\mathbb{E}_\mu[x_i] = b$ for all i and $\mathbb{E}_\mu[x_i x_j] = \rho$ for all $i \neq j$.

Positively pairwise symmetric if $\rho \geq b^2$

Theorem

Assuming the UGC, P without negations is useless if and only if P supports a positively pairwise symmetric distribution.

Without negations

Definition

A distribution μ over $\{0,1\}^k$ is *pairwise symmetric* if there is b, ρ such that $\mathbb{E}_\mu[x_i] = b$ for all i and $\mathbb{E}_\mu[x_i x_j] = \rho$ for all $i \neq j$.

Positively pairwise symmetric if $\rho \geq b^2$

Theorem

Assuming the UGC, P without negations is useless if and only if P supports a positively pairwise symmetric distribution.

Remark: *checkable in time $2^{O(k)}$* by convex optimization.

Algorithm

Claim

If P does not support a positively pairwise symmetric distribution there is a quadratic Q such that $Q(x) > E_Q^+$ for all $x \in P^{-1}(1)$.

Algorithm

Claim

If P does not support a positively pairwise symmetric distribution there is a quadratic Q such that $Q(x) > E_Q^+$ for all $x \in P^{-1}(1)$.

The rest similarly as before.

Algorithm

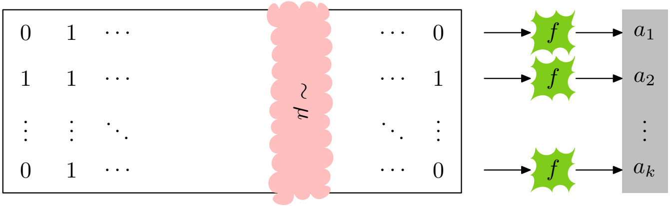
Claim

If P does not support a positively pairwise symmetric distribution there is a quadratic Q such that $Q(x) > E_Q^+$ for all $x \in P^{-1}(1)$.

The rest similarly as before.

Caveat: need to be careful about Q where E_Q^+ attained by $p = 0$ or $p = 1$.

Hardness Without Negations

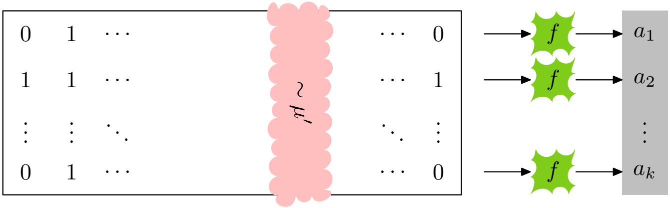


Hardness Without Negations

Analysis: study the distribution η of $(a_1, \ldots, a_k)$

Completeness If f is a dictator, then $\eta = \mu$ so $\mathbb{E}_\eta[P] = 1$

Soundness If f is far from dictator, then can replace μ by μ' with same 1st and 2nd moments



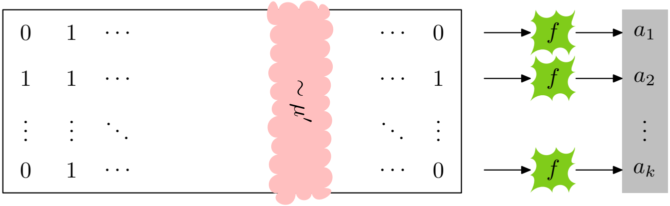
Hardness Without Negations

Analysis: study the distribution η of $(a_1, \ldots, a_k)$

Completeness If f is a dictator, then $\eta = \mu$ so $\mathbb{E}_\eta[P] = 1$

Soundness If f is far from dictator, then can replace μ by μ' with same 1st and 2nd moments

- Can take $\mu' =$ comb. of two product distributions U_a and U_b



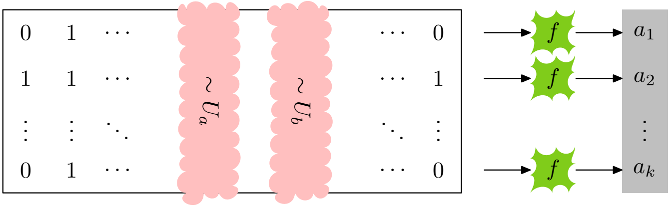
Hardness Without Negations

Analysis: study the distribution η of $(a_1, \ldots, a_k)$

Completeness If f is a dictator, then $\eta = \mu$ so $\mathbb{E}_\eta[P] = 1$

Soundness If f is far from dictator, then can replace μ by μ' with same 1st and 2nd moments

- Can take $\mu' =$ comb. of two product distributions U_a and U_b
- Change each column to either U_a or U_b without losing value.



More general CSPs

What about a *general CSP?*

Not quite clear what the proper definition is.

More general CSPs

What about a *general CSP?*

Not quite clear what the proper definition is.

1. Replace all constraints by same Q

More general CSPs

What about a *general CSP?*

Not quite clear what the proper definition is.

1. Replace all constraints by same Q
2. Replace all constraints of type P_1 by Q_1, all constraints of type P_2 by Q_2, etc

More general CSPs

What about a *general CSP?*

Not quite clear what the proper definition is.

1. Replace all constraints by same Q
2. Replace all constraints of type P_1 by Q_1, all constraints of type P_2 by Q_2, etc
3. Some compromise?

Some Final Comments

New natural relaxation of Max-CSPs

Some Final Comments

New natural relaxation of Max-CSPs

Ultimate hardness of a CSP!
(For real this time)

Some Final Comments

New natural relaxation of Max-CSPs

Ultimate hardness of a CSP!
(For real this time)

Assuming UGC, complete characterization in the single-predicate setting (with or without negations)

Some Final Comments

New natural relaxation of Max-CSPs

Ultimate hardness of a CSP!
(For real this time)

Assuming UGC, complete characterization in the single-predicate setting (with or without negations)

Would be interesting to consider other settings

- Satisfiability
- Robust satisfiability

Thank you!