WOLA

# Testing the Boolean Rank

Michal Parnas

Joint work with: Dana Ron, Adi Shraibman

# The Real rank

The real rank of a matrix $M_{n\times n}$ of size $n \times n$:

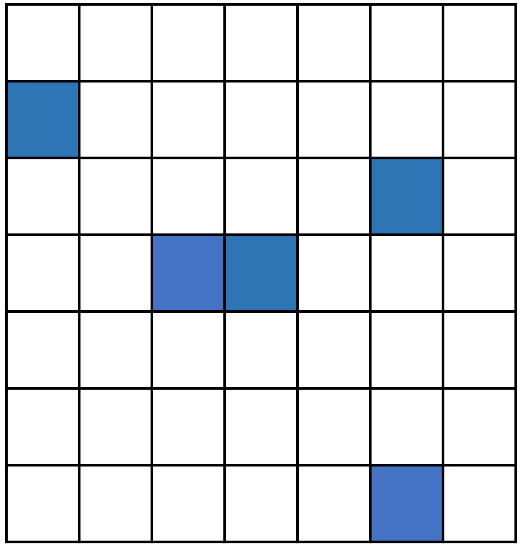
- Maximal # independent rows/columns of M.

- Minimal r such that $M_{n\times n}$ can be decomposed as:

$$\left(M_{n\times n}\right) = \left(X_{n\times r}\right) \cdot \left(Y_{r\times n}\right)$$

- Computing exactly in poly time using Gaussian elimination.

# Testing the Real Rank



**Property Testing Algorithm:**

Does M have rank $\leq$ d or M is **ε-far** from rank $\leq$ d

(at least ε-fraction of the entries should be modified to have rank $\leq$ d).

- Krauthgamer , Sasson 2003: non-adaptive algorithm, query complexity $O(d^2/\varepsilon^2)$.
- Wang, and Woodruff, 2014: adaptive algorithm , query complexity $O(d^2/\varepsilon)$.
- Balcan, Woodruff, Zhang 2018: non-adaptive algorithm, query complexity $\tilde{O}(d^2/\varepsilon)$.

# The Boolean rank

- The Boolean rank of a Boolean matrix $M_{n\times n}$ is the minimal r such that:

$$\left(M_{n\times n}\right)=\left(X_{n\times r}\right)\cdot\left(Y_{r\times n}\right)$$

$X_{n\times r}$ and $Y_{r\times n}$ are Boolean, and operations are Boolean ( $1 + 1 = 1$).

- Computing Boolean rank exactly is NP-hard.
- Testing algorithms for real rank can't be adapted to Boolean rank, since use linearity.

Using theorem of Alon, Fischer, Newman 2007:

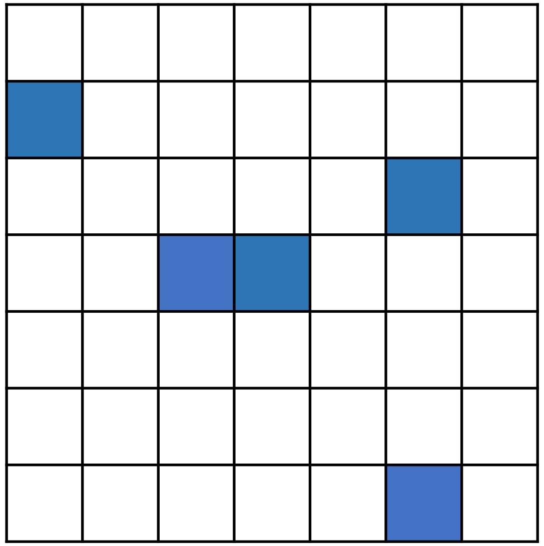
Boolean rank $\leq d$ ⇒ every submatrix of M has $\leq 2^d$ distinct rows/columns.

Boolean rank is testable with $\left(2^d/\varepsilon\right)^{O(2^{4d})}$ queries.

# Our Main Result

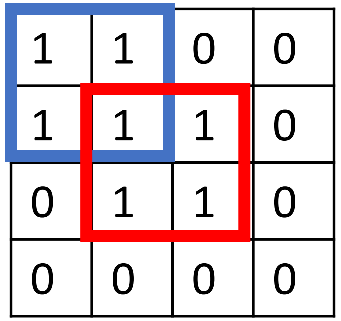
**Theorem:**

There exists a 1-sided error testing algorithm for the Boolean rank with polynomial query complexity of $\tilde{O}\left(d^{4} / \varepsilon^{6}\right)$

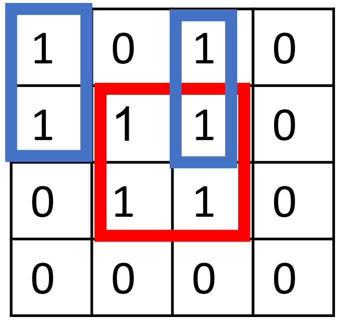


# Alternative Definitions for Boolean rank

- Minimal # monochromatic rectangles to cover all 1's of M.




Boolean rank 2




Boolean rank 2

- Minimal # bipartite bicliques to cover all edges of bipartite graph represented by M.
- Boolean rank related to non-deterministic communication complexity of M.

# Testing the Boolean Rank

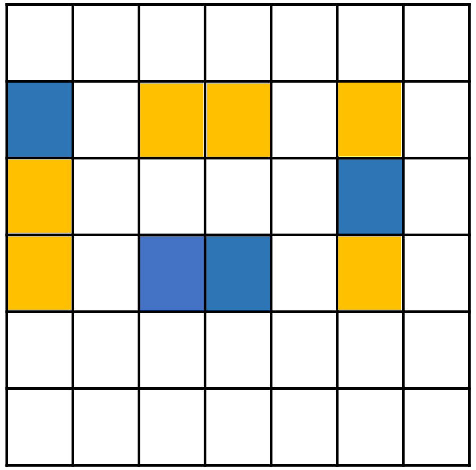
Algorithm (Test M for Boolean rank d, given d and ε):

- Select uniformly, independently, at random $O\left(\frac{d^2}{\varepsilon^3}\log\frac{d}{\varepsilon}\right)$ entries from **M**.
- Let **U** be subset of entries selected, and let **W** be submatrix of **M** induced by **U**.
- Accept if **W** has Boolean rank ≤ d, otherwise reject.

Query complexity: $\tilde{O}\left(d^4/\varepsilon^6\right)$

Running time: exponential in sample size since problem is NP-hard.

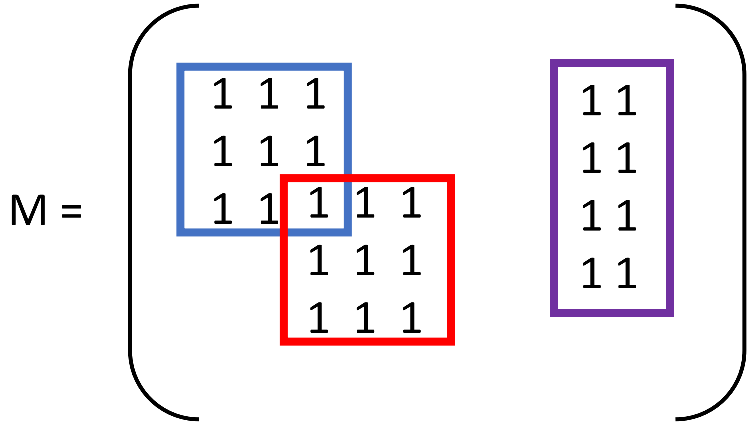
M =



# Proof of Correctness

**Theorem:** The Algorithm is a 1-sided error testing algorithm for the Boolean rank.

- The algorithm always accepts M if it has Boolean rank $\leq d$.

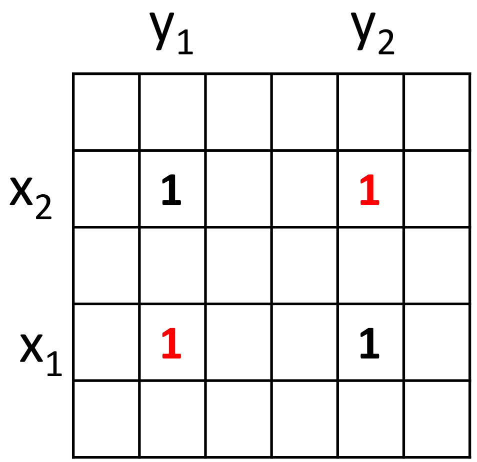



- If M is $\varepsilon$-far from Boolean rank d then algorithm rejects with prob. $\geq 2/3$.

# Basic Concept – Compatible entries

1-entries $(x_1, y_1)$ and $(x_2, y_2)$ are **compatible**

if $M[x_1, y_2] = M[x_2, y_1] = 1$ .




Compatible entries can be in same monochromatic rectangle.

# Skeletons and beneficial entries

Czumaj, Sohler 2005: combinatorial programs.
Parnas, Ron, Rubinfeld 2006: Tolerant testing, skeletons.

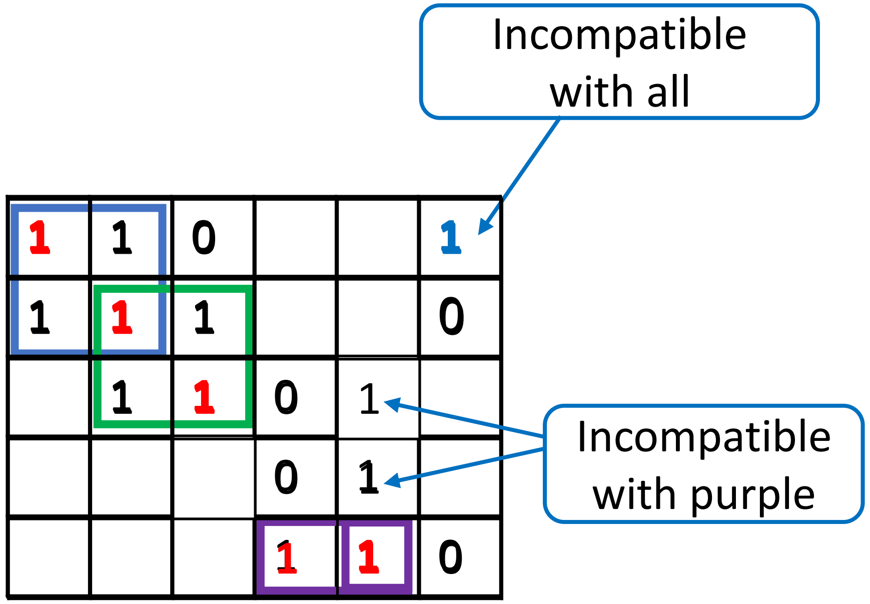
} Separating probabilistic analysis from combinatorial structure

# Skeletons and beneficial entries

Czumaj, Sohler 2005: combinatorial programs.
Parnas, Ron, Rubinfeld 2006: Tolerant testing, skeletons.

Separating probabilistic analysis from combinatorial structure

A **skeleton** for M is a multiset $S = \{S_1,\ldots,S_d\}$ where each $S_i$ contains compatible 1-entries (can be in same monochromatic rectangle).




A 1-entry (x,y) is **beneficial** for skeleton S, if for every $1 \leq i \leq d$:

- (x,y) is incompatible with $S_i$, or
- Adding (x,y) to $S_i$ reduces significantly #entries that can join $S_i$

Skeleton becomes more constrained.

# Proof Sketch for ε-far M

**Main Claim:**

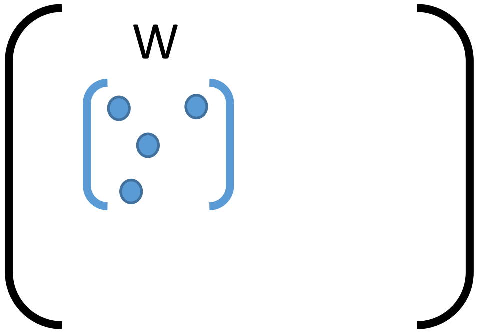
It is possible to define skeletons and beneficial entries such that:

1. M is ε-far from Boolean rank at most d ⟹ every skeleton has $\varepsilon^2 n^2$ beneficial entries.
2. Skeletons are small: Size is $O(d^2/\varepsilon)$.

Using claim ⇓

For a sample of size $O\left(\frac{d^2}{\varepsilon^3}\log\frac{d}{\varepsilon}\right)$ with prob. ≥ 2/3,

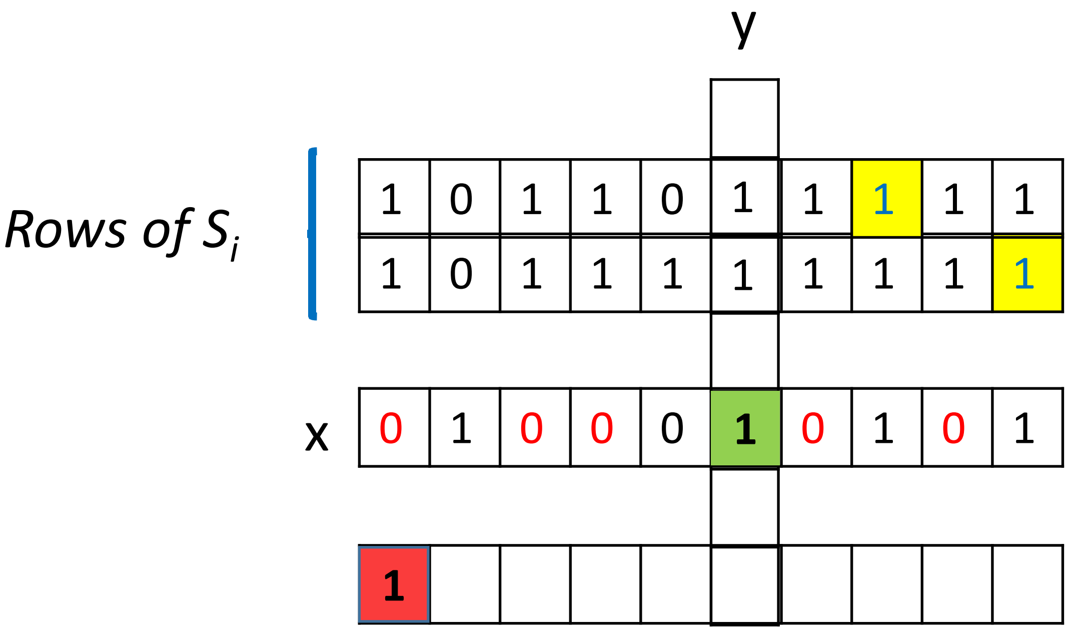
Boolean rank of **W** is > d, and algorithm rejects as required.

M =


# zero heavy row/column

Row x is zero-heavy for $S_i$ if there are $\geq \frac{\varepsilon}{4d} n$ columns with zeros in row x,

that do not have zeros in rows of entries from $S_i$

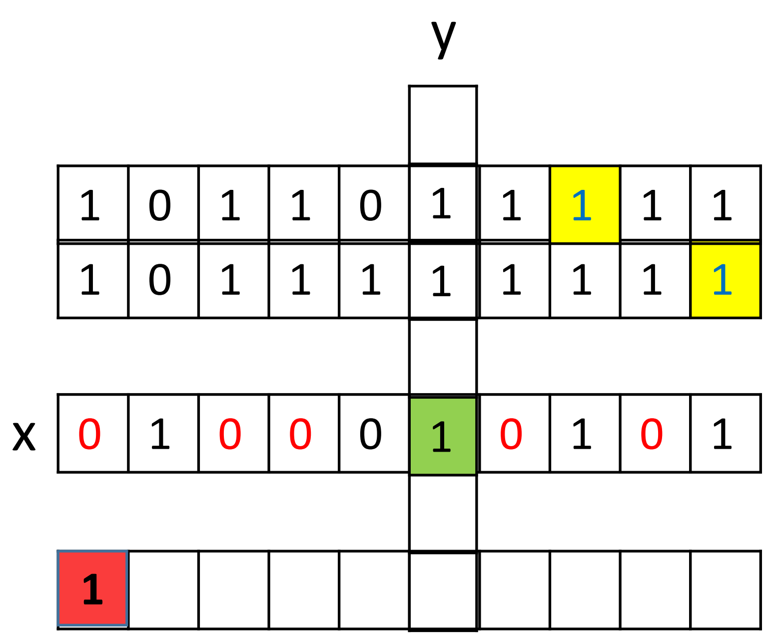



Adding a 1-entry to $S_i$ from a zero-heavy row, reduces significantly #entries that can join $S_i$

# Skeletons and Beneficial entries

A 1-entry (x,y) can be added to $S_i$ if:

- (x,y) is compatible with each entry in $S_i$, and
- row x or column y is zero-heavy for $S_i$




A 1-entry is **beneficial** for skeleton $S = \{S_1,\ldots,S_d\}$, if for every $1 \leq i \leq d$,

the it can be added to $S_i$ or it is incompatible with $S_i$

# Proof of main claim

**Main Claim:**

1. M is $\varepsilon$-far from Boolean rank at most d ⟹ every skeleton has $\varepsilon^2 n^2$ beneficial entries.
2. Skeletons are small: Size is $O(d^2/\varepsilon)$.

⇓

1. Assume there are $< \varepsilon^2 n^2$ beneficial entries ⟹ modify M so that it has Boolean rank $\leq d$.

2. Only entries in zero-heavy rows/columns are added to skeleton
   ⟹ every entry added, disqualifies many other entries.

# Open Problems

- Binary rank:

  Minimal # monochromatic rectangles to partition all 1's of M.

  Minimal # bipartite bicliques to partition all edges of bipartite graph represented by M.

  Related to deterministic communication complexity of M.

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 |

Binary rank 3

> Theorem: Binary rank is testable with $O\left(2^{2^d}/\varepsilon\right)$ queries.

Polynomial query complexity testing algorithm for binary rank?

- Lower bounds on query complexity for Boolean/binary rank.
- Other rank functions: non-negative rank?

WOLA

# THANK YOU!