29 March 2024

Solving SAT.

Plan

* Beating Brute Force for 3SAT
* Announcements
* Orthogonal Vectors solves CNF-SAT

CNF-SAT

Given a CNF $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$

Does there exist $\vec{a} \in \{0,1\}^n$ s.t. $\varphi(\vec{a}) = 1$?

$$C_i = (l_{i_1} \vee l_{i_2} \vee l_{i_3} \vee \ldots \vee l_{i_k})$$

literal

variable x_j

or negation $\neg x_j$

3SAT

CNF-SAT where each clause has at most 3 literals

Brute Force SAT (φ)

For each $a \in \{0,1\}^n$ // 2^n possible assignments

Evaluate $\varphi(a)$ // poly(n, m)

if $\varphi(a) = 1$ $\longrightarrow$ Return ✓

Return ✗

Running Time? $\longrightarrow$ $2^n \cdot \text{poly}(n,m)$

Brute Force SAT (φ)

For each $a \in \{0,1\}^n$ // 2^n possible assignments

  Evaluate $\varphi(a)$ // $poly(n, m)$

  if $\varphi(a) = 1 \longrightarrow$ Return ✓

Return ✗

Running Time? $\longrightarrow$ $2^n \cdot poly(n,m)$

Can we do better?

$\hookrightarrow$ e.g. for 3SAT?

Given φ, $l_i \in \{x_i, \neg x_i\}$

Let $\varphi\big|_{l_i = b}$ be the simplification of φ after setting all occurrences of x_i consistent w/ $l_i = b$.

Given φ, $l_i \in \{x_i, \neg x_i\}$

Let $\varphi|_{l_i=b}$ be the simplification of φ after setting all occurrences of x_i consistent w/ $l_i = b$.

$$\varphi = (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_5) \wedge (x_2 \vee x_4 \vee x_5)$$

$$\varphi|_{\neg x_1 = 1} = (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_5) \wedge (x_2 \vee x_4 \vee x_5)$$

Given φ, $l_i \in \{x_i, \neg x_i\}$

Let $\varphi|_{l_i = b}$ be the simplification of φ after setting all occurrences of x_i consistent w/ $l_i = b$.

$$\varphi = (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_5) \wedge (x_2 \vee x_4 \vee x_5)$$

$$\varphi|_{\neg x_1 = 1} = (\overset{0}{\cancel{x_1}} \vee \neg x_3) \wedge (\overset{1}{\cancel{\neg x_1}} \vee x_5) \wedge (x_2 \vee x_4 \vee x_5)$$

$$\downarrow$$

$$\neg x_3 \wedge (x_2 \vee x_4 \vee x_5)$$

Branch3SAT (φ)

(Monien-Speckmeyer '86)

if φ is a 2-CNF

solve 2SAT(φ) in polynomial time.

Else, find some clause $C = (\ell_1 \vee \ell_2 \vee \ell_3)$

$$\text{Return } \Big(\text{Branch3SAT}(\varphi|_{\ell_1=1}) \vee \text{Branch3SAT}(\varphi|_{\ell_1=0,\ell_2=1}) \vee \text{Branch3SAT}(\varphi|_{\ell_1=0,\ell_2=0,\ell_3=1}) \Big)$$

Correctness

At least 1 of ℓ_1, ℓ_2, ℓ_3 must be set to 1.

Branch3SAT (φ)

Makes 3 Recursive calls

Branch3SAT ($\varphi |_{\ell_1 = 1}$)

Branch3SAT ($\varphi |_{\ell_1 = 0, \ell_2 = 1}$)

Branch3SAT ($\varphi |_{\ell_1 = 0, \ell_2 = 0, \ell_3 = 1}$)

Running Time?

For n-variable 3-CNFs

$$T(n) \leq T(n-1) + T(n-2) + T(n-3) + \text{poly}(n)$$

Branch3SAT (φ)

Makes 3 Recursive calls

Branch3SAT $\left(\varphi \big|_{\ell_1 = 1} \right)$

Branch3SAT $\left(\varphi \big|_{\ell_1 = 0,\, \ell_2 = 1} \right)$

Branch3SAT $\left(\varphi \big|_{\ell_1 = 0,\, \ell_2 = 0,\, \ell_3 = 1} \right)$

Running Time?

For n-variable 3-CNFs

$$T(n) \leq T(n-1) + T(n-2) + T(n-3) + \text{poly}(n)$$

$$\Downarrow$$

$$T(n) \leq 1.833^n$$

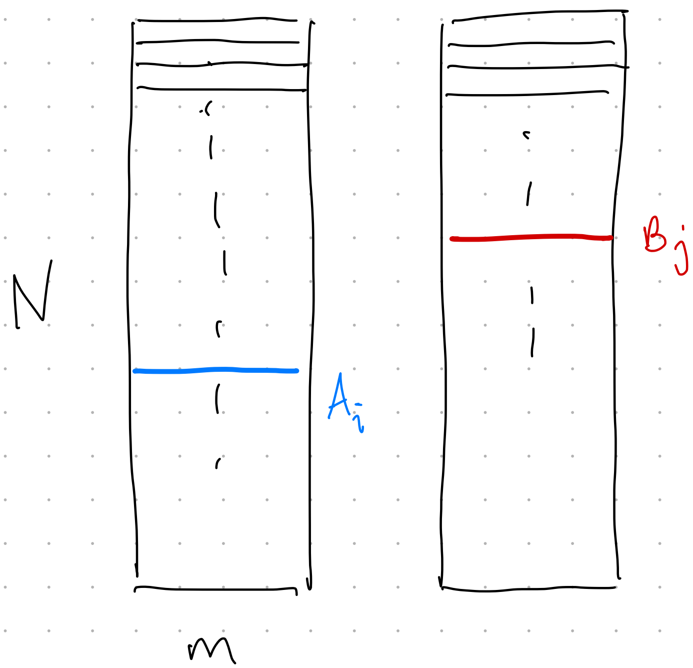
Announcements

* Prelim Review, April 9, 7-9pm Gates G01

* Prelim #2, Aprill 11, 7:30pm
  - ↳ Room assignments announced after break
  - ↳ Covering
    - Divide & Conquer
    - Flow
    - NP-Completeness

* Have a great spring break!

Orthogonal Vectors Problem (OV)

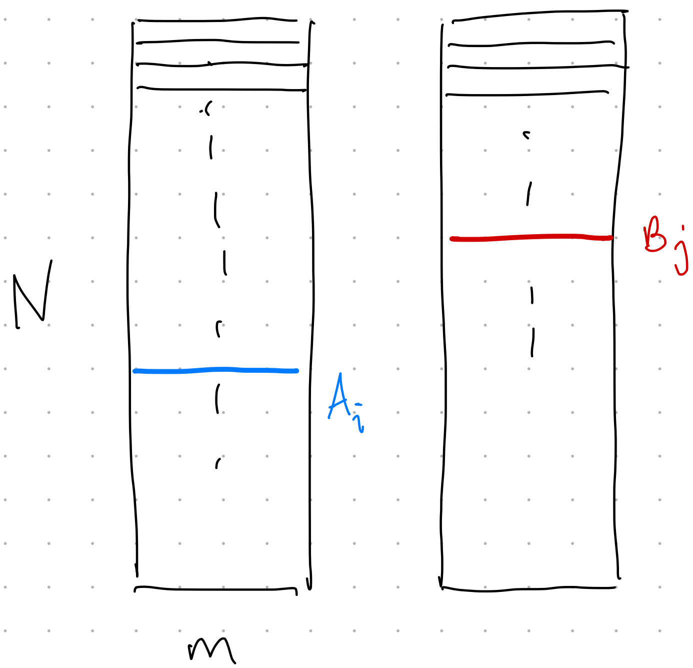


Given: Two lists A, B each of N vectors over $\{0,1\}^m$

Does there exist $1 \le i, j \le N$ s.t. A_i and B_j are orthogonal?

$$A_i \cdot B_j = \sum_{k=1}^{m} A_{ik} \cdot B_{jk} = 0$$

Orthogonal Vectors Problem (OV)



Given. Two lists A, B

each of N vectors over $\{0,1\}^m$

Does there exist

$1 \le i, j \le N$ s.t.

A_i and B_j are **orthogonal**?

$$A_i \cdot B_j = \sum_{k=1}^{m} A_{ik} \cdot B_{jk} = 0$$

Naive OV:

For $i = 1 \ldots N$.

For $j = 1 \ldots N$

Test if $A_i \cdot B_j = 0$

Running Time $N^2 \cdot m$

<u>Theorem</u>. (Due to Ryan Williams, former 482~~0~~ student!)

If there exists an $N^{1.9}$ time algorithm for OV,

then there exists a 1.94^n time algorithm for CNF-SAT.

Theorem. (Due to Ryan Williams, former 4820 student!)

If there exists an $N^{1.9}$ time algorithm for OV, then there exists a 1.94^n time algorithm for CNF-SAT.

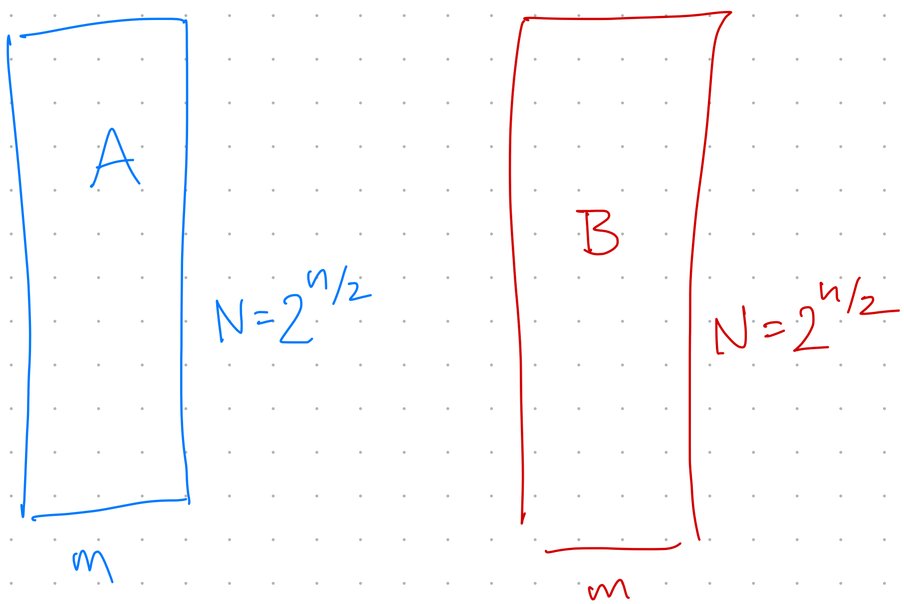
This would be a MAJOR breakthrough in Algorithms & Complexity Theory.

$\sim 2^n$

<u>Idea</u>: $\underbrace{\text{Reduce}}_{\text{Exponential-time reduction}}$ CNF-SAT to OV.

<u>Given</u> $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$

* write down



based on "partial assignments"

Partial Assignments

* Consider splitting the variables in half

$$x_1, x_2, \ldots, x_{n/2} \;\Big|\; x_{n/2+1}, x_{n/2+2}, \ldots, x_n$$

Partial Assignments

$\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$

$(x_1 \vee \neg x_2 \vee \cdots)$

* Consider splitting the variables in half

$x_1, \ldots x_n$

$$x_1, x_2, \ldots, x_{n/2} \;\Big|\; x_{n/2+1}, x_{n/2+2}, \ldots, x_n$$

$\{0,1\}^n$

* For $i \in \{0,1\}^{n/2}$, $j \in \{0,1\}^{n/2}$

$$(x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n) \longleftarrow (i, j)$$

is an assignment to $\vec{x}$

and i, j are <u>partial</u> assignments.

CNF-SAT via OV.

For each $i \in \{0,1\}^{n/2}$

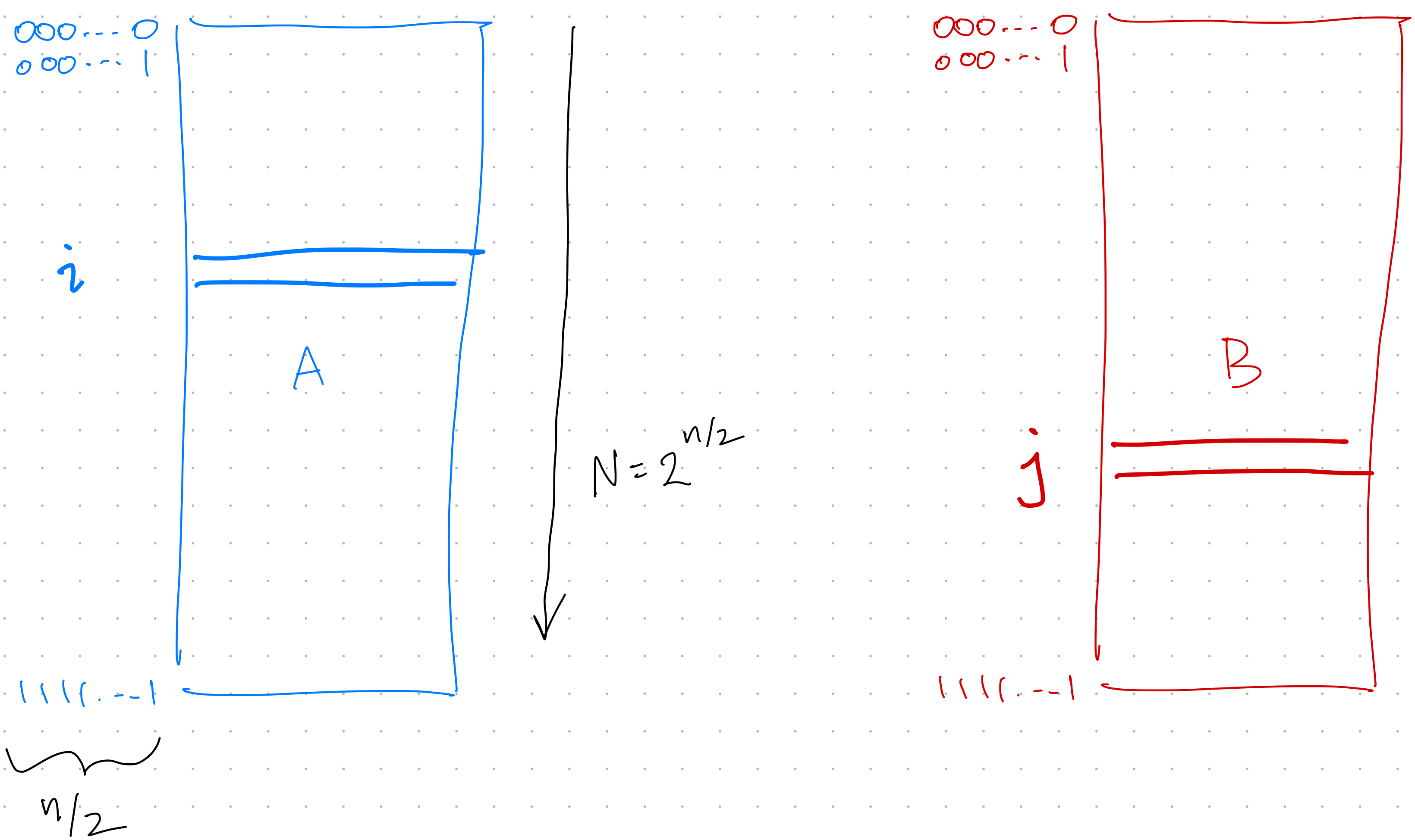
$$A_i \leftarrow \text{Partial Assignment Gadget}\left(x_1, \ldots, x_{n/2}, i\right)$$

For each $j \in \{0,1\}^{n/2}$

$$B_j \leftarrow \text{Partial Assignment Gadget}\left(x_{n/2+1}, \ldots, x_n, j\right)$$

Return $OV(A, B)$

Vectors indexed by partial assignments



Each index $i \in \{0,1\}^{n/2}$ corresponds to an assignment to

$x_1, x_2, \ldots, x_{n/2} \leftarrow i$

Each $j \in \{0,1\}^{n/2}$ corresponds to assignment

$x_{n/2+1}, x_{n/2+2}, \ldots, x_n \leftarrow j$

Vector coordinates determined by satisfying clauses

$$\varphi = C_1 \wedge C_2 \wedge \dots \wedge \overset{C_k}{\dots} \wedge C_m$$

A_i

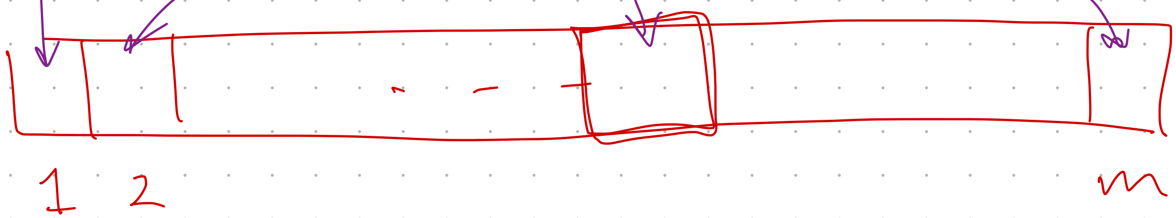
$$A_{ik} = \begin{cases} 0 & \text{if } x_1, x_2, \dots, x_{n/2} \leftarrow i \text{ satisfies clause } C_k \\ 1 & \text{otherwise} \end{cases}$$

$$C_k = (x_2 \vee x_9 \vee \cancel{x_n} \vee \cancel{\neg x_{n-10}} \vee \dots \vee x_7)$$

Vector coordinates determined by satisfying clauses

$$\varphi = C_1 \wedge C_2 \wedge \cdots \overset{C_k}{-} \wedge C_m$$

B_j



$$B_{jk} = \begin{cases} 0 & \text{if } x_{n/2+1}, x_{n/2+2}, \dots, x_n \leftarrow j \\ & \text{satisfies clause } C_k \\ 1 & \text{otherwise} \end{cases}$$

$$C_k = (\cancel{x_2} \vee \cancel{x_9} \vee x_n \vee \neg x_{n-10} \vee \cdots \vee \cancel{x_7})$$

<u>Claim</u>:

$A_{ik} \cdot B_{jk} = 0$ if and only if

the assignment

$$(x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n) \leftarrow (i, j)$$

satisfies the clause C_k

Claim.

$A_{ik} \cdot B_{jk} = 0$ if and only if the assignment

$$(x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n) \longleftarrow (i, j)$$

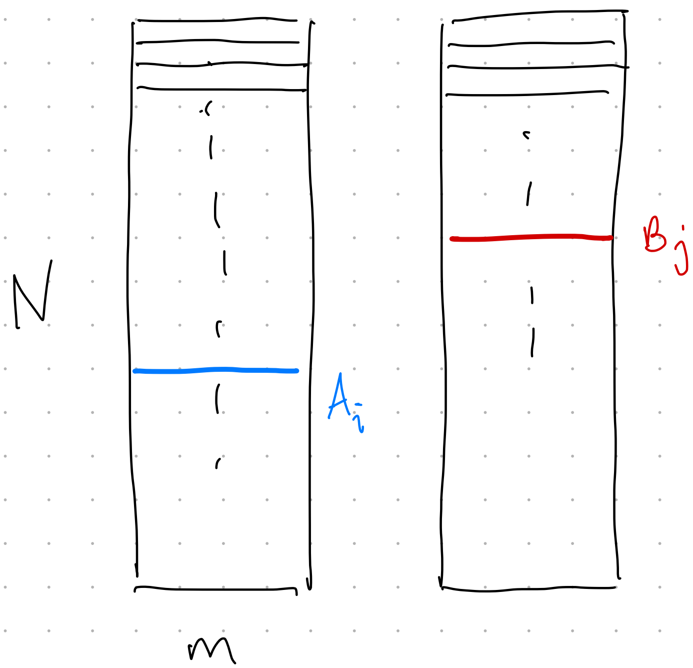
satisfies the clause C_k

Corollary. There exists orthogonal A_i and B_j if and only if φ is satisfiable.

$$i \to \left. 0 \ldots\ldots 2^{n/2}-1 \right\} \{0,1\}^{n/2}$$

Orthogonal Vectors Problem (OV)

Solves CNF-SAT



Reduction

$$2 \times 2^{n/2} \cdot \text{poly}(n, m)$$

$$+ \; T_{OV}(2^{n/2})$$

Suppose $T_{OV}(N) = N^{1.9}$.

$\Rightarrow$ CNF-SAT: $\left(2^{n/2}\right)^{1.9} \leq 1.94^n$

What did we show?

* New algorithmic approach for solving CNF-SAT.
  - ↳ we only need to improve OV.

* Hardness for polynomial-time.
  - ↳ If CNF-SAT requires $\sim 2^n$ time, then OV requires $\sim N^2$ time.

What did we show?

* New algorithmic approach for solving CNF-SAT.

↳ we only need to improve OV.

* Hardness for polynomial-time.

↳ If CNF-SAT requires $\sim 2^n$ time,

then OV requires $\sim N^2$ time.

$\Downarrow$

Theorem. If CNF-SAT requires $\sim 2^n$ time,

(Backurs-Indyk '15) then Edit Distance requires $\tilde{\Omega}(n^2)$ time.