28 March 2025

Solving SAT.

Plan

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

CNF-SAT

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

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

$$C_i = (\underbrace{l_{i_1} \vee l_{i_2} \vee l_{i_3} \vee \dots \vee l_{i_n}}_{\text{up to } n \text{ literals}})$$

k-SAT

CNF-SAT where each clause has at most k 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? $\rightarrow$ $2^n \cdot \text{poly}(n,m)$

Note: Works for all CNFs

Can we do better? e.g. for 3SAT?

3SAT — Satisfiability Problem for 3-CNF formulas

every clause has ≤ 3 literals.

$$\phi = (\ell_{11} \vee \ell_{12} \vee \ell_{13}) \wedge \ldots \wedge (\ell_{m1} \vee \ell_{m2} \vee \ell_{m3})$$

Suppose $\ell_{11} = 1$

Do we care about ℓ_{12} or ℓ_{13}?

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/ $l_i = b$.

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/ $l_i = b$.

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

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

Branch3SAT (φ)

(Monien-Speckmeyer '86)

if φ is a 2-CNF

solve 2SAT(φ) in polynomial time.

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

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

Correctness

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

Branch3SAT (φ)

(Monien - Speckmeyer '86)

if φ is a 2-CNF

solve 2SAT(φ) in polynomial time.

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

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

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

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

Running Time?

* Setting a literal reduces number of variables!

Branch3SAT(φ)

Makes 3 Recursive calls

$$\text{Branch3SAT}\big(\underbrace{\varphi|_{\ell_1=1}}_{n-1}\big)$$

$$\text{Branch3SAT}\big(\underbrace{\varphi|_{\ell_1=0,\,\ell_2=1}}_{n-2}\big)$$

$$\text{Branch3SAT}\big(\underbrace{\varphi|_{\ell_1=0,\,\ell_2=0,\,\ell_3=1}}_{n-3}\big)$$

Running Time $T(n) \equiv$ time on n-variable 3-CNF ϕ

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

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

Guess $T(n) \leq c^n$

$$\leq c^{n-1} + c^{n-2} + c^{n-3}$$

$$= c^{n-3} \cdot \underbrace{(c^2 + c + 1)} \cdot n^k$$

$$c^3 = c^2 + c + 1 \implies c \leq 1.84$$

$$\implies T(n) \leq 1.84^n$$

exponentially faster than Brute Force!

$$T(n) \le T(n-1) + T(n-2) + T(n-3) + poly(n)$$

Guess $T(n) \le c^n$

$$\le c^{n-1} + c^{n-2} + c^{n-3}$$

$$= c^{n-3} \cdot \underbrace{(c^2 + c + 1)} \cdot n^k$$

$$c^3 = c^2 + c + 1 \implies c \le 1.84$$

$$\implies T(n) \le 1.84^n$$

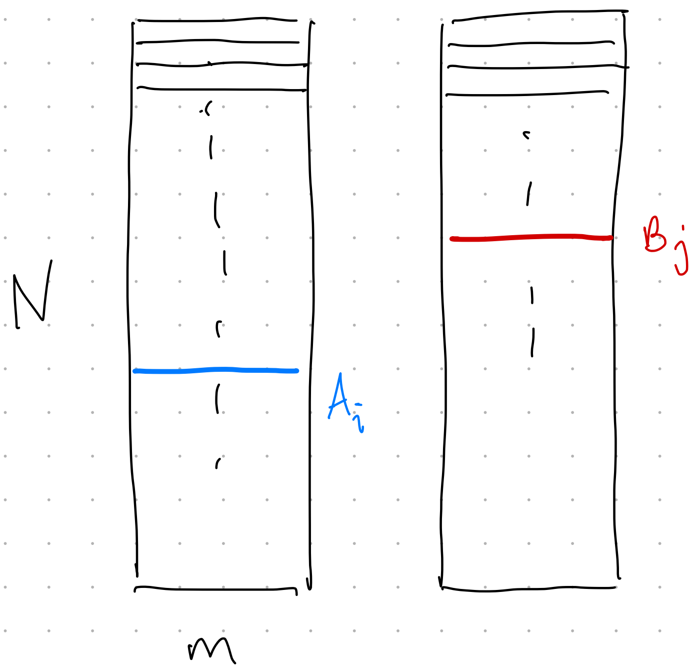
exponentially faster than Brute Force!

Current Best: 1.34^n (Shoening, 1999 Moser-Scheder, 2011)

Announcements

* Have a great Spring break!

Orthogonal Vectors Problem (OV)

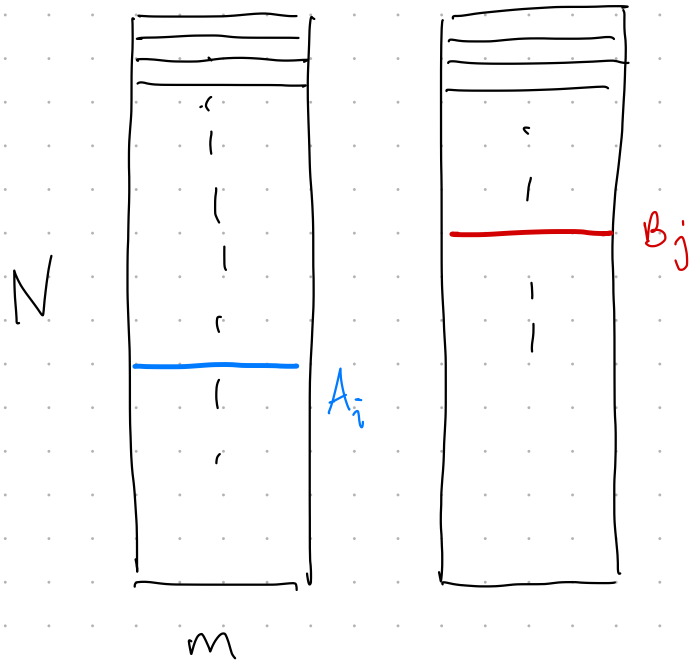


Given: Two lists A, B each of N vectors over $\{0,1\}^m$

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 $N^2 \cdot m$

Theorem. (Due to Ryan Williams, former $482\cancel{0}$ 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.

$\sim 2^n$

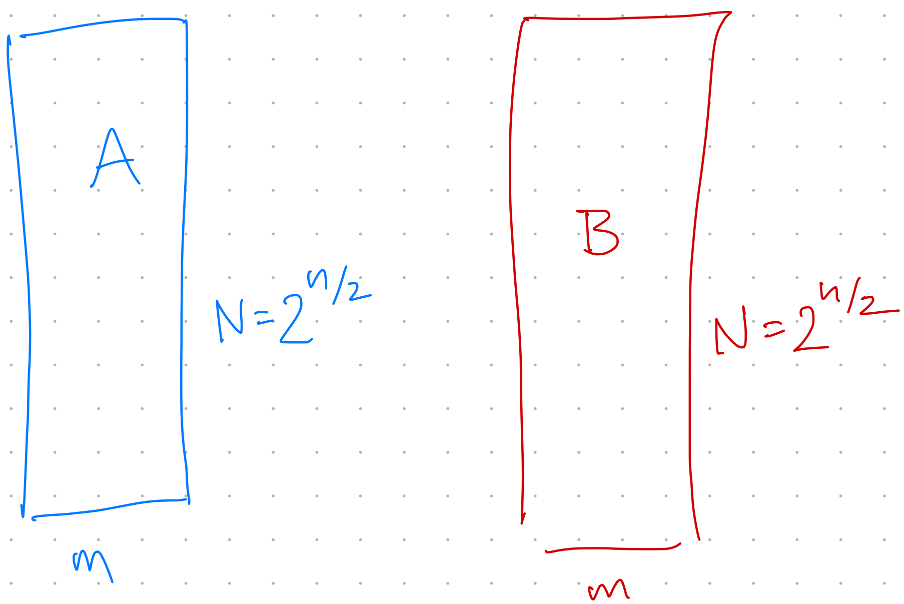
This would be a MAJOR breakthrough in Algorithms & Complexity Theory.

Idea: Reduce CNF-SAT to OV.

Exponential-time reduction

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

* write down



based on "partial assignments"

Partial Assignments

* Consider splitting the variables in half

$$\underbrace{x_1, x_2, \ldots, x_{n/2}}_{\in \{0,1\}^{n/2}} \;\Big|\; \underbrace{x_{n/2+1}, x_{n/2+2}, \ldots, x_n}_{\in \{0,1\}^{n/2}}$$

Each assignment to half the variables

$$\Updownarrow$$

Some "index" in $\{0, 1, \ldots, 2^{n/2}-1\}$

Partial Assignments

* Consider splitting the variables in half

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

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

$$(x_1, x_2, \dots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \dots, x_n) \longleftarrow \Big(\underset{[0,\, 2^{n/2}-1]}{\underset{\uparrow}{i}} \;,\; \underset{[0,\, 2^{n/2}-1]}{\underset{\uparrow}{j}}\Big)$$

is an assignment to $\vec{x}$

and i, j are partial assignments.

CNF-SAT via OV.

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

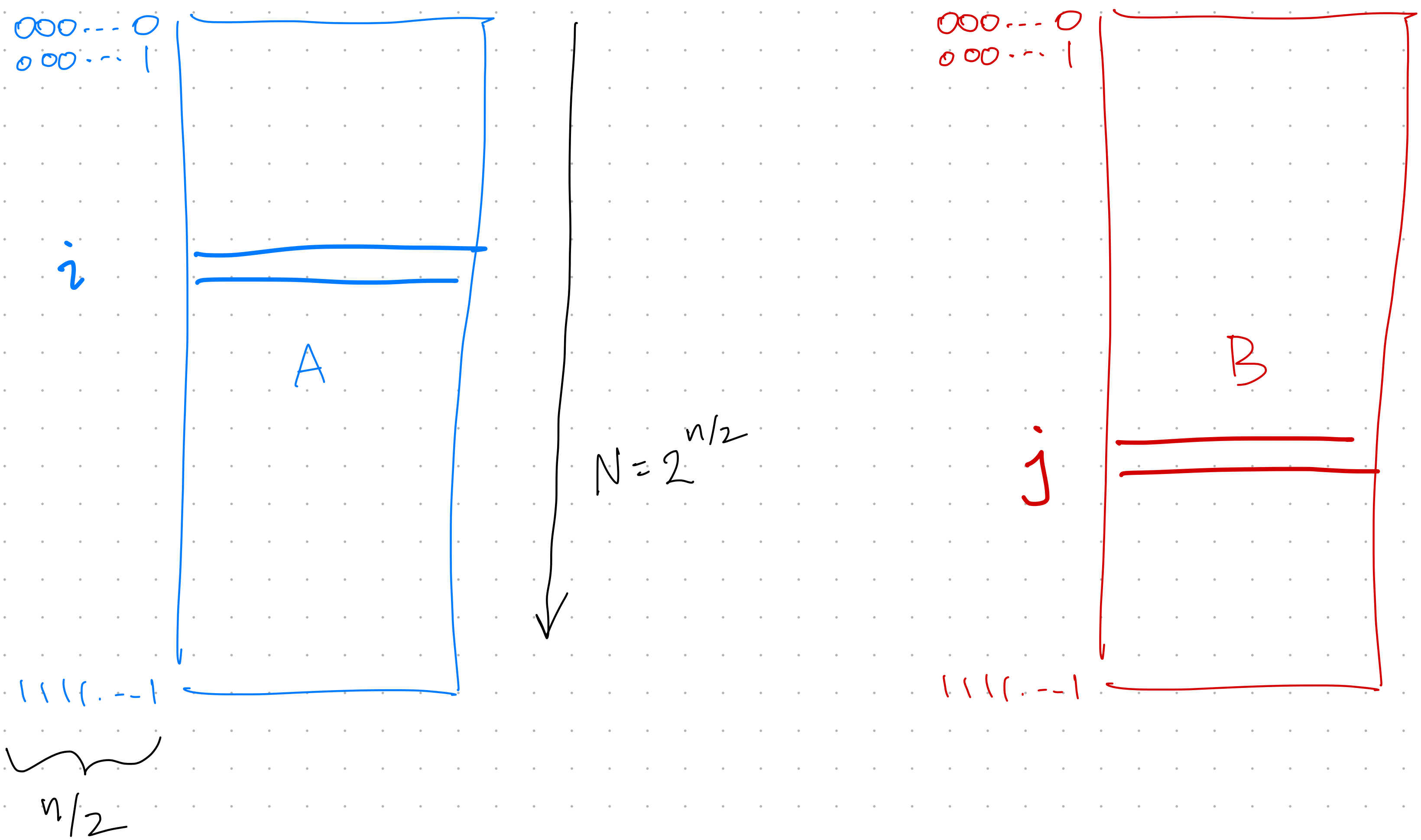
$A_i \leftarrow$ Partial Assignment Gadget $(x_1, \dots, x_{n/2}, i)$

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

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

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

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

A_i

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

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

Vector coordinates determined by satisfying clauses

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

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

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

Claim.

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

the assignment

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

satisfies the clause C_k

Claim.

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

the assignment

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

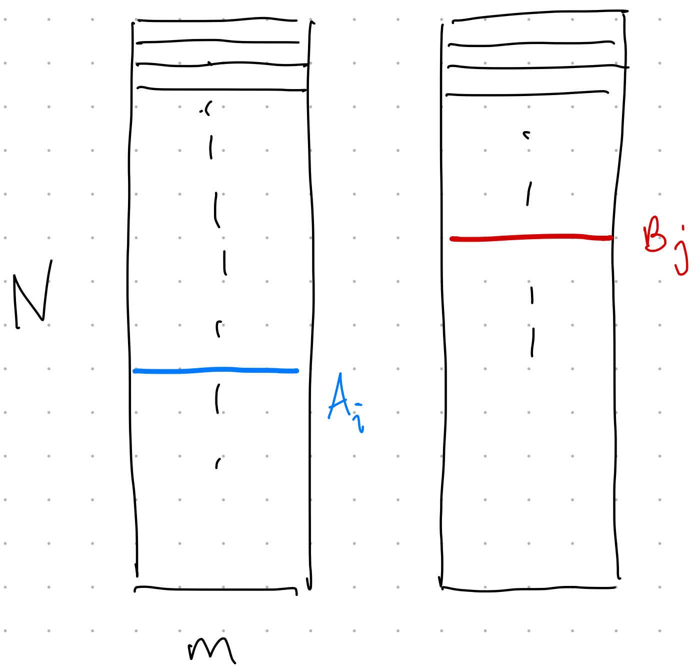
satisfies the clause C_k

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

$$i \to 0, \ldots, 2^{n/2}-1 \;\}\; \{0,1\}^{n/2}$$

Orthogonal Vectors Problem (OV)

Solves CNF-SAT



Reduction

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

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

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

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

What did we show?

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

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