* HW6 due on Monday
* Exam 3 on Nov 20 (Thurs)
* lecture on Nov 18 virtual and optional (not part of Exams)

Today's Plan : Fine grained complexity

Fine Grained Complexity

For problems in P (polytime solvable), we argue that it is unlikely/difficult to design a faster algorithm.

– Usually we start with a different assumption from $P \neq NP$

Orthogonal Vectors Problem :

Given two sets of vectors in $\{0,1\}^d$:

$$V = \{v_1, \dots, v_n\} \qquad W = \{w_1, \dots w_n\}$$

Find $v_i \in V$ and $w_j \in W$ s.t. $v_i \cdot w_j = 0$ (i.e., they don't share a 1)

$$= \sum_{k=1}^{d} v_i(k) \cdot w_j(k)$$

Naive alg takes $O(n^2 d)$ time.

It seems difficult to improve.

SETH $\Longrightarrow$ OV

Strong Exponential Time Hypothesis : There is no alg that solves any $\underbrace{k\text{-CNF}}_{\text{clauses with OR of } \leq k \text{ literals}}$ instance in $2^{n(1-\epsilon)} \cdot poly(n)$ time.

Remark : ETH says $2^{o(n)}$ time not possible.

Thm: If we can solve OV in $n^{2-\delta} \cdot d$ time, where $\delta > 0$ is a constant then we can solve k-CNF in $2^{n(1-\epsilon)} \cdot poly(n)$ time for constant $\epsilon > 0$.

– In other words, SETH $\Rightarrow$ fast OV is impossible.

Pf: We will reduce k-CNF to OV.

- Partition n variables into $\{1, \dots n/2\}$ and $\{\frac{n}{2}+1, \dots n\}$

Let $d = \#$ clauses

Define $2^{n/2}$ vectors V for 1st $n/2$ vars, one for each assignment

$C_1 = x_1 \vee \overline{x_2} \vee x_3$

$C_2 = x_2 \vee x_3 \vee x_4$

$C_3 = \overline{x_1} \vee \overline{x_3} \vee \overline{x_4}$

$\{x_1, x_2\} \quad \{x_3, x_4\}$

Let $d = \#$ clauses

Define $2^{n/2}$ vectors V for 1st $n/2$ vars, one for each assignment

Similarly, $2^{n/2}$ vectors W for 2nd $n/2$ vars, "

ℓ-th coordinate 0 if clause ℓ satisfied and 1 otherwise

$\{x_1, x_2\}$ $\{x_3, x_4\}$

x_1	x_2	x_3	x_4
T	T	T	T
T	F	T	F
F	T	F	T
F	F	F	F

$\begin{bmatrix}0\\0\\1\end{bmatrix}$ $\begin{bmatrix}0\\0\\1\end{bmatrix}$ $\begin{bmatrix}0\\0\\0\end{bmatrix}$

Obs: If $\exists\, v_i$ and w_j that are orthogonal, then all clauses can be satisfied.

— Since each coordinate is 0 or 1, orthogonal means for all ℓ, ℓ-th coordinate 0 in at least one of v_i or w_j.

Runtime: Fast OV implies we can solve this in time

$$\left(2^{n/2}\right)^{2-\delta} d = 2^{n-\delta/2} \cdot \#\text{clauses}.$$

■

Takeaways: Improving runtime of 'simple' problems could be as hard as refuting major conjectures.

Nearest-Neighbor Problem

Given n vectors $V = \{v_1, v_2, \ldots, v_n\}$ in $\{0,1\}^d$

Objective: Preprocess them in $n^{2-\epsilon} d$ time such that now for any query vector $w \in \{0,1\}^d$, we want to find $\min_{i \in [n]} \|v_i - w\|_2$ in $n^{1-\epsilon} \cdot d$ time.

E.g. in 1-d we can sort in $O(n \log n)$ time & binary search answers in $O(\log n)$ time.

Thm: We can reduce OV to Nearest Nbr

$\Rightarrow$ if OV is hard to solve in $O(n^{2-\delta} d)$ time then Nearest Nbr cannot be done for constant ϵ.

Pf: Given OV instance with vector sets V and W in d-dimensions. For each $v \in V$ and $w \in W$, we will define $\tilde{v}$ and $\tilde{w} \in \{0,1\}^{3d}$

Goal: Nearest nbr of $\tilde{w}$ in $\tilde{V}$ is an orthogonal vector of w in V.

Intuition: $\|w - v\|^2 = \|w\|^2 + \|v\|^2 - 2\, w \cdot v$

If every vector has same length then nearest nbr $\arg\min_v \|w-v\| = \arg\max w.v$ ← maximizes dot-product

Define: $\tilde{v} = \begin{bmatrix} v \\ 1-v \\ 0 \end{bmatrix}$ $\tilde{w} = \begin{bmatrix} -w \\ 0 \\ 1-w \end{bmatrix}$

Note: $\|\tilde{v}\| = \|\tilde{w}\| = \sqrt{d}$ and $\tilde{v}.\tilde{w} = -v.w \leq 0$

Hence, if $w \in W$ has an orthogonal vector in V, then nearest nbr of w in V will be the orthogonal vector

$\Longrightarrow$ we can solve OV using nearest nbr.

$$\text{Runtime} = \underbrace{O(n^{2-\delta} \cdot d)}_{\text{Preprocessing}} + \underbrace{n}_{=|W|} \cdot \underbrace{O(n^{1-\delta} \cdot d)}_{\text{time per } w \in W}$$

$$= O(n^{2-\delta} \cdot d)$$

Lecture Notes	Lecture Date
Lec22_Reductions.pdf	Nov 11, 2025
Lec21_OGD_Experts_Minimax.pdf	Nov 6, 2025
Lec20_OnlineGradDescent.pdf	Nov 4, 2025
Lec19_SmoothedGradDescent.pdf	Oct 30, 2025
Lec18_GradientDescent.pdf	Oct 28, 2025
Lec17_ConvexOptimization.pdf	Oct 23, 2025
Lec16_SetCoverRandRounding.pdf	Oct 16, 2025
Lec15_QuickSortMaxCut.pdf	Oct 14, 2025
Lec14_RandAlgMedianFinding.pdf	Oct 9, 2025
Lec13_FacilityLocation.pdf	Oct 2, 2025
Lec12_ApproxAlgStart.pdf	Sep 30, 2025
Lec11_Proof_LP_Duality.pdf	Sep 25, 2025
Lec10_LP_Duality.pdf	Sep 23, 2025
Lec9_LPsStart.pdf	Sep 16, 2025
Lec8_MaxFlow.pdf	Sep 11, 2025
Lec7 MST_Matchings.pdf	Sep 9, 2025
Lec6_GraphAlgStart.pdf	Sep 4, 2025
Lec5_MDP.pdf	Sep 2, 2025
Lec4 DP_Knapsack.pdf	Aug 28, 2025
Lec3 MedianAndDP.pdf	Aug 26, 2025
Lec2 DivideAndConquer.pdf	Aug 21, 2025
Lec1 IntroAndBigO.pdf	Aug 19, 2025