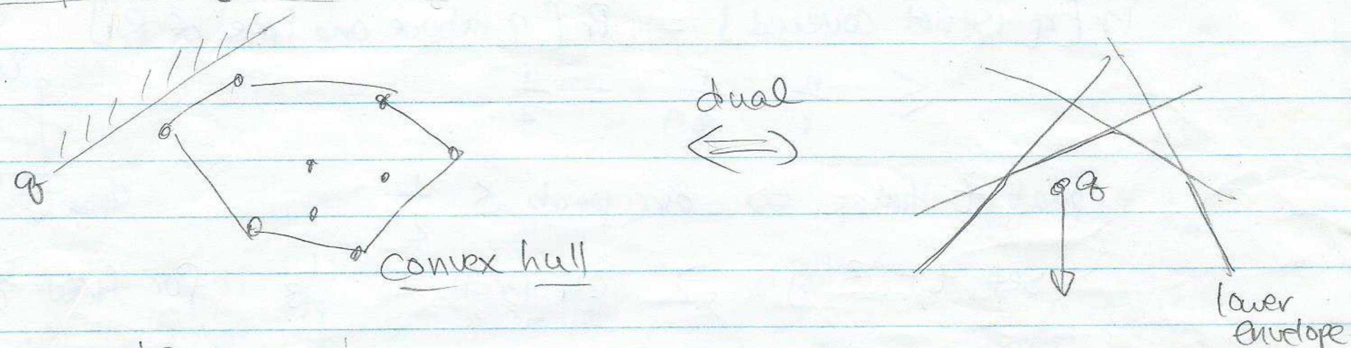
Halfspace Range Reporting in 2D & 3D



e.g. emptiness:

2D: $O(n)$ space, $O(\log n)$ time by binary search

3D: $O(n)$ space, $O(\log n)$ time by 2D point location

lower envelope $\Rightarrow$ convex polyhedron

reporting? (appl - 2D circular range reporting reduces to 3D halfspace reporting)

Shallow Cutting Lemma (Matoušek '92)

Given n lines L in $\mathbb{R}^2$ (or n planes in $\mathbb{R}^3$)

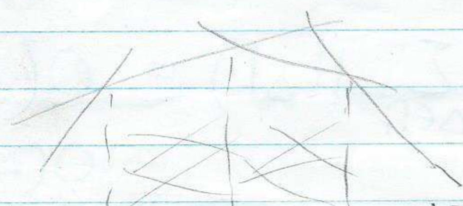
can form $O(r)$ cells each intersecting $O(\frac{n}{r})$ lines s.t. all (n/r)-shallow pts are covered } called $(\frac{1}{r})$-shallow cutting

(if we ignore log factors) pts with $\leq n/r$ lines below

Pf Idea: 1. take random sample $R \subseteq L$ w. prob. $\approx \frac{r}{2n}$

2. return "vertical decomposition" of the lower envelope of R

size $O(r)$



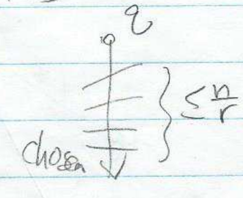
LE(R)

By prev. analysis, each cell intersects $\lesssim \frac{n}{r}$ lines w. good prob.

Fix an $(\frac{n}{r})$-shallow pt q

Pr[q is not covered] = Pr[q above one line of R]

$$\le \frac{n}{r} \cdot \frac{r}{2n} = \frac{1}{2}$$



repeat t times $\Rightarrow$ err prob $\le \frac{1}{2^t}$

set $t = 3\log n$ $\Rightarrow$ err prob $\le \frac{1}{n^3}$ for fixed q

$\Rightarrow$ over err prob $\le \frac{1}{n}$

(work harder to avoid extra log....) $\square$

[Rmk: construction time: $O(n\log n)$ rand. (Ramos '99)
$O(n\log n)$ det. (C.-Tsakalidis '15)]

<u>Cutting Tree Method</u> (in 2D & 3D)

recurse $\Rightarrow$ $O(n^{1+\varepsilon})$ space

$O(\log n + k)$ time

$S(n) = O(r)\,S(\frac{n}{r}) + O(r)$

$Q(n) = \begin{cases} Q(\frac{n}{r}) + O(r) & \text{if shallow} \\ O(k) & \text{if not shallow} \end{cases}$

r large const

switch to brute force

<u>Better Method</u> (C. '00)

idea - hierarchy of cuttings

for $r = 1, 2, 4, 8, \ldots$

compute $(1/r)$-shallow cutting Γ_r

for each cell Δ,

store L_Δ = list of lines (or planes) intersecting Δ

<u>conflict list</u>

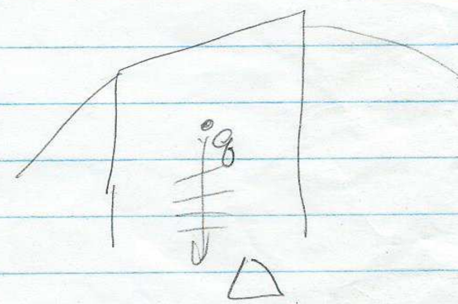
space: $O\left(\sum_{\Delta\in\Gamma_r} |L_\Delta|\right) = O\left(r \cdot \frac{n}{r}\right)$

$= O(n)$ for each r

$\Rightarrow$ total $\boxed{O(n\log n)}$

Query alg'm, given pt q:

1. find r with $k \le n/r < 2k$
2. find $\Delta \in \Gamma_r$ containing q ← 2D point location $O(\log n)$ time
3. search L_Δ by brute force



time $O(|L_\Delta|)$
$= O\left(\frac{n}{r}\right)$
$= O(k)$

$\Rightarrow$ time $O(\log n + k)$

but don't know k in advance

idea: guess!

try $k = k_0, k_1, k_2, \ldots$ with $k_i = 2^i \log n$

total query time $O\left(\sum_{i: k_{i-1} < k} (\cancel{\log n +} k_i) \right)$ (geom series)

$$= \begin{cases} O(k) & \text{if } k > \log n \\ O(\log n) & \text{else} \end{cases}$$

$$= \boxed{O(\log n + k)}$$

by shallow partition thm & combine

Rmks – space improvable to $O(n)$ [Afshani–C. '09]

– also works for 3D orthogonal 3-sided (dominance)

$O(n)$ space, $O(\log\log U + k)$ time

by 2D orthogonal point location

– in higher D, (size of lower envelope)

$O^*(n^{\lfloor d/2 \rfloor})$ space, $O(\log n + k)$ time (shallow cutting)

$O(n)$ space, $O^*(n^{1-1/\lfloor d/2 \rfloor} + k)$ time [shallow partition]

An Improvement:

for $r = r_1, r_2, \ldots$ where $\frac{n}{r_{i+1}} = \left(\frac{n}{r_i}\right)^{2/3}$

...

store L_Δ in a linear-space DS

with $O(|L_\Delta|^{2/3} + k)$ query time

Space: $\boxed{O(n \log\log n)}$

query time: line 1: find r_i s.t. $k \le \frac{n}{r_i}$, $k > \frac{n}{r_{i+1}}$

line 3: $O(|L_\Delta|^{2/3} + k)$

$= O\left(\left(\frac{n}{r_i}\right)^{2/3} + k\right) = O(k)$

$\Rightarrow \boxed{O(\log n + k)}$