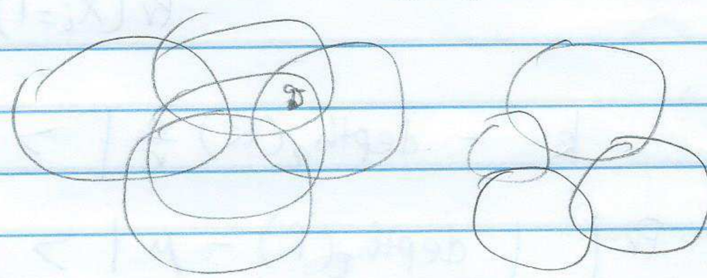
Approx. Counting Cont'd

factor $1+\varepsilon$ instead of additive error εn?

Problem Given set S of n disks in $\mathbb{R}^2$, build data structure s.t. given query pt q, can quickly count # disks containing q

denoted k ($\text{depth}_q(S)$)



special case: $k=0$? ("range emptiness")

Fact union of n disks has $O(n)$ vertices

$\Rightarrow$ $O(n)$ space, $O(\log n)$ query time by "point location"

larger k?

Known exact sol'ns: $O(n)$ space, $O(\log n + k)$ query time (range reporting) (*)

or $O(n)$ space, $\tilde{O}(\sqrt{n})$ query time

combine $\Rightarrow$ $\tilde{O}(\sqrt{k})$ time

Approx Method 1: Sampling with diff sizes (Aronov, Har-Peled '06)

guess $2^i \le k \le 2^{i+1}$

(1

take random sample R of size $r = b\frac{n}{2^i}$,

solve problem for R

Let $R = \{s_1, \dots, s_r\}$ Fix pt q.

Let $X = \text{depth}_q(R)$.

Then $X = \sum_{i=1}^{r} X_i$ where $X_i = \begin{cases} 1 & \text{if } q \in s_i \\ 0 & \text{else} \end{cases}$

$Pr[X_i = 1] = \frac{k}{n}$, $\mu = \frac{kr}{n} = \Theta(b)$

$$Pr\left[\left| k - \text{depth}_q(R)\frac{n}{r}\right| > \varepsilon k\right]$$

$$= Pr\left[\left|\text{depth}_q(R) - \mu\right| > \underbrace{\varepsilon\frac{kr}{n}}_{\varepsilon\mu}\right]$$

$$\underset{\text{Chernoff}}{\leq} \frac{1}{e^{\Theta(\varepsilon^2\mu^2/\mu)}} = \frac{1}{e^{\Theta(\varepsilon^2\mu)}} = \frac{1}{e^{\Theta(\varepsilon^2 b)}}$$

There are $O(n^2)$ "different" pts q

$$\Rightarrow Pr[R \text{ fails for some } q] \leq \frac{n^2}{e^{\Theta(\varepsilon^2 b)}} < \frac{1}{n^{100}}$$

by setting $b = \text{large const} * \frac{1}{\varepsilon^2}\log n$.

But how to solve problem for R?

$\text{depth}_q(R)$ is $\Theta(b) = \Theta(\frac{1}{\varepsilon^2}\log n)$

by $(*)$ $\Rightarrow O(\frac{n}{2^i}\log n)$ space, $O(\frac{1}{\varepsilon^2}\log n)$ query time

try all $i \Rightarrow O\left(\sum_{i=\cancel{0}\ \text{to}\ \log n}^{\log n}\frac{n}{2^i}\log n\right) = O(n\cancel{\log}n)$ space,

reduces to $O(\frac{1}{\varepsilon}\lg n\lg\lg n)$ $\rightarrow$ $O(\frac{1}{\varepsilon}\log^2 n)$ query time by binary search (Monte Carlo) (harder to derand. efficiently)

Method 2 (Afshani - C. '07)

Guess $2^i \le k \le 2^{i+1}$

Shallow Cutting Lemma (Matoušek '92) For any r,

can find $O(r)$ cells

s.t. each cell intersects $\le O(\frac{n}{r})$ disks

The cells cover all pts of depth $\le \frac{n}{r}$

"shallow"

Compute shallow cutting with $r = \frac{n}{2^{i+1}}$

for each cell γ,

solve problem for $S_\gamma = \{$ all disks intersecting $\gamma \}$ of size $O(2^i)$ (conflict list)

with additive error $O(\varepsilon 2^i)$

by ε-approximation!

Space $O\left(\sum_{i=0}^{\log n} \frac{n}{2^{i+1}} \cdot \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon} \right)$

$= O\left(n \cdot \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon} \right)$

query time

$O\left(\left(\log n + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon} \right) \cancel{\log n} \right)$ $\log\log n$

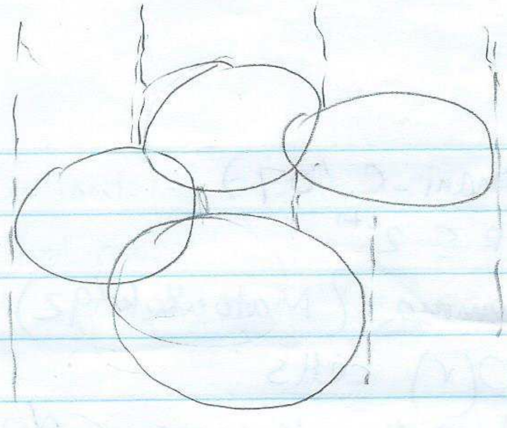
pt loc to find γ — by binary search

Pf of Weaker Lemma: (with extra logs)

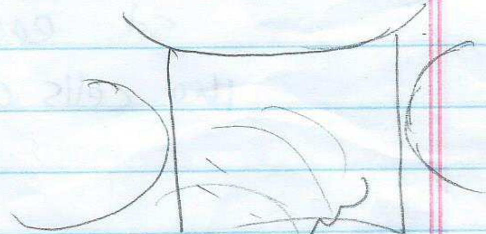
1. take rand sample R of size r
2. compute union $U(R)$ of R
3. return vertical decomposition of complement of $U(R)$

$VD(R)$

$O(r)$ pseudo-trapezoids



Fix a pseudo-trapezoid γ
intersecting $> b\frac{n}{r}$ disks



$$\Pr[\,\gamma \in VD(R)\,] \le \left(\frac{r}{n}\right)^4 \left(1-\frac{r}{n}\right)^{b\frac{n}{r}} = O\!\left(\left(\frac{r}{n}\right)^4 e^{-b}\right)$$

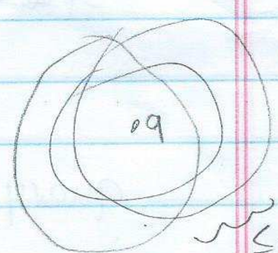
\# ~~t~~trapezoids $O(n^4)$

$$\Rightarrow \Pr[\text{not every cell of } VD(R) \text{ intersects} < c\tfrac{n}{r}\text{disks}] \le O\!\left(n^4 \cdot \frac{1}{e^b}\right) < \frac{1}{n^{100}}$$

by setting b = large const $*\ \ln n$

Fix pt q of depth $\le \frac{n}{2r}$.

$$\Pr[\,q \text{ not covered by } VD(R)\,] = \Pr[\,q \text{ inside } U(R)\,] \le \frac{n}{2r} \cdot \frac{r}{n} = \frac{1}{2}$$



Repeat b times $\Rightarrow$ $\Pr[q \text{ not covered}] \le \frac{1}{2^b}$

\# "diff" pts $q = O(n^2)$

$$\Rightarrow \Pr[\text{fail}] \le O\!\left(\frac{n^2}{2^b}\right) < \frac{1}{n^{100}}$$

$\square$

Rmk: Afshani-C. '07: $O(n)$ space, $O(\log(\frac{n}{k}))$ query time

Other objects: union complexity ...