

CMSC 754 - Computational Geometry

Lecture 18: Coresets and Kernels

Approximation by Sampling:

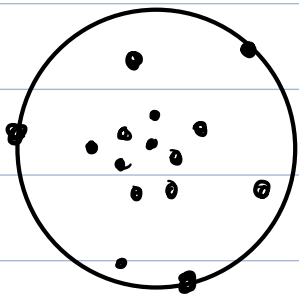
- Running time too slow?
- Maybe your data size is too large!
- Idea:
 - Extract a small subset, $P' \subseteq P$
 - Run solve problem exactly on P'
 - Prove that the answer on P' is "close to optimal" on P .

How to compute P' ?

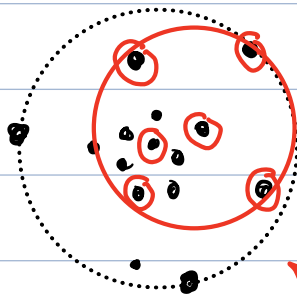
- Depends on your problem
- Random sampling is most common, but not necessarily best

Example: Minimum Enclosing Ball (MEB)

- Given a set $P = \{p_1, \dots, p_n\}$ in $\mathbb{R}^d$ compute the Euclidean ball of min. radius enclosing P .

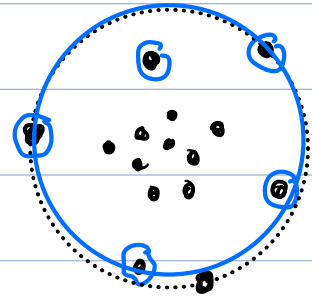


$MEB(P)$



$MEB(P')$

$P' = \text{random}$



$MEB(P'')$

$P'' = \text{coreset}$

Problem with random sampling:

- $MEB(P)$ depends on points near periphery
- Random sample extracts many irrelevant points.
- Smarter: Use a sampling method that gives priority to peripheral points

Coreset: Let P be input set.

$f^*(P) \rightarrow \mathbb{R}$ is our objective function
(eg. $f^*(P) = \text{radius of MEB}$)

Given $\epsilon > 0$, an ϵ -coreset is a subset $Q \subseteq P$ s.t.

$$1 - \epsilon \leq \frac{f^*(Q)}{f^*(P)} \leq 1 + \epsilon$$

The opt. soln. for Q is close to opt. for P

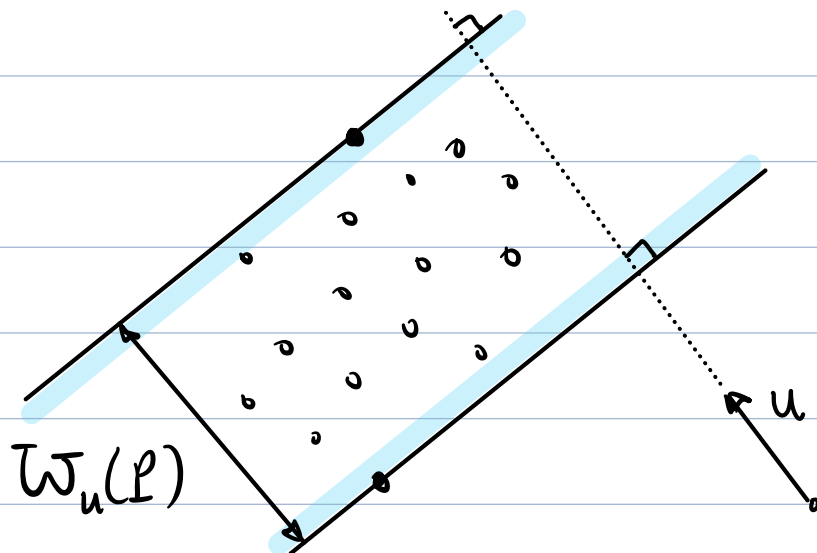
Questions:

- For what optimization problems do (small) coresets exist?
- (As a function of $n + \epsilon$) how small is the coreset?
- How fast can we compute a coreset?

Coreset for Directional Width: (also called ϵ -kernel)

- Given a pt set $P \subseteq \mathbb{R}^d$
- Given a unit vector $\vec{u}$
- Directional width of P in direction $\vec{u}$ is:

$$W_u(P) = \max_{p \in P} (\vec{p} \cdot \vec{u}) - \min_{p \in P} (\vec{p} \cdot \vec{u})$$



Given $\varepsilon > 0$, an ε -coreset for direc. width (also called ε -kernel) is a subset $R \subseteq P$ s.t.

$\forall$ unit vect. $\vec{u}$:

$$(1-\varepsilon) \bar{w}_u(P) \leq \bar{w}_u(R) \leq \bar{w}_u(P)$$

Trivially true
since $R \subseteq P$

Getting this is
the objective

Aside: When computing approx. lower bounds we sometimes write:

$$(1-\varepsilon) \cdot \text{exact} \leq \text{approx}$$

and other times:

$$\frac{\text{exact}}{1+\varepsilon} \leq \text{approx}$$

Does the form matter?

Not really. If $0 < \varepsilon < 1$, then

$$1-\varepsilon \leq \frac{1}{1+\varepsilon} \leq 1-\frac{\varepsilon}{2}$$

- Only constant factors are affected

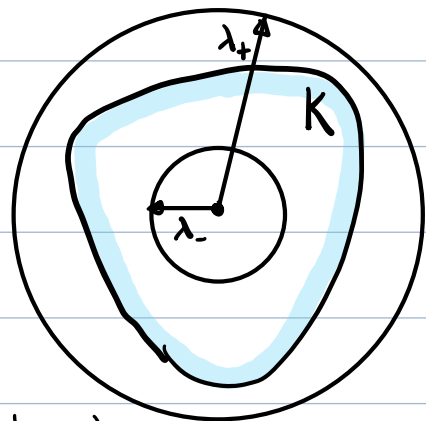
Useful Facts:

Chain Property: If X is an ε -kernel for Y
and Y is an ε' -kernel for Z
then X is an $(\varepsilon + \varepsilon')$ -kernel for Z

Union Property: If X is an ε -kernel for P
 X' is an ε -kernel for P'
then $X \cup X'$ is an ε -kernel for $P \cup P'$

Canonical Position: We like fat things...

Fat: Given $0 \leq \alpha \leq 1$, a convex body K is α -fat if K can be sandwiched between two concentric balls of radii $\lambda_- \leq \lambda_+$ where $\alpha = \lambda_- / \lambda_+$



Canonical Position: Convex body K is in α -canonical form if it is sandwiched between balls of radius $\lambda_- = \frac{1}{2}\alpha + \lambda_+ = \frac{1}{2}$ centered at the origin.

Why $1/2$? $\Rightarrow K$'s diameter ≤ 1

A point set P is $\left\{ \begin{array}{l} \alpha\text{-fat} \\ \alpha\text{-canonical form} \end{array} \right\}$ if $\text{conv}(P)$ is.

We can convert any pt set into canonical form.

Affine Transformation: Is a linear transformation (scaling + rotation + shearing) + translation

Lemma: Given any n -element pt. set $P \subseteq \mathbb{R}^d$, there exists an affine transformation T that maps P into $(1/d)$ -canonical form

- $R \subseteq P$ is an ε -kernel for P
iff $T(R)$ is an ε -kernel for $T(P)$
- T can be computed in $O(n)$ time

Proof makes use of important fact: (1948)

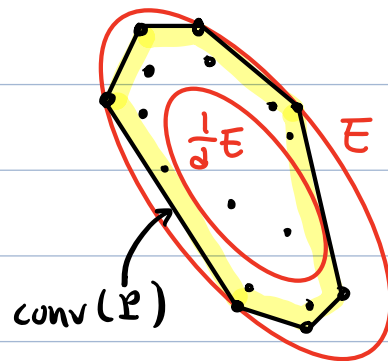
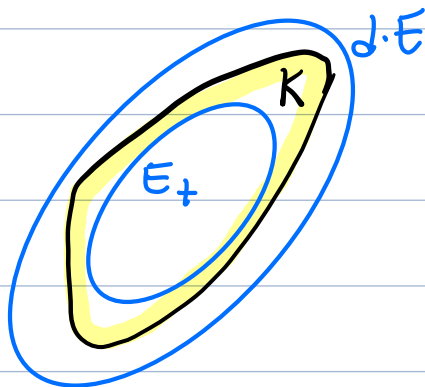
John's Theorem: Given any convex body $K \subseteq \mathbb{R}^d$, let E be max volume ellipsoid contained in K , then

$$E \subseteq K \subseteq d \cdot E$$

where $d \cdot E$ is a factor- d scaling E about its center.

Equiv: Given pt. set P , let E be min vol. ellipsoid containing P , then

$$\frac{1}{d} E \subseteq \text{conv}(P) \subseteq E$$

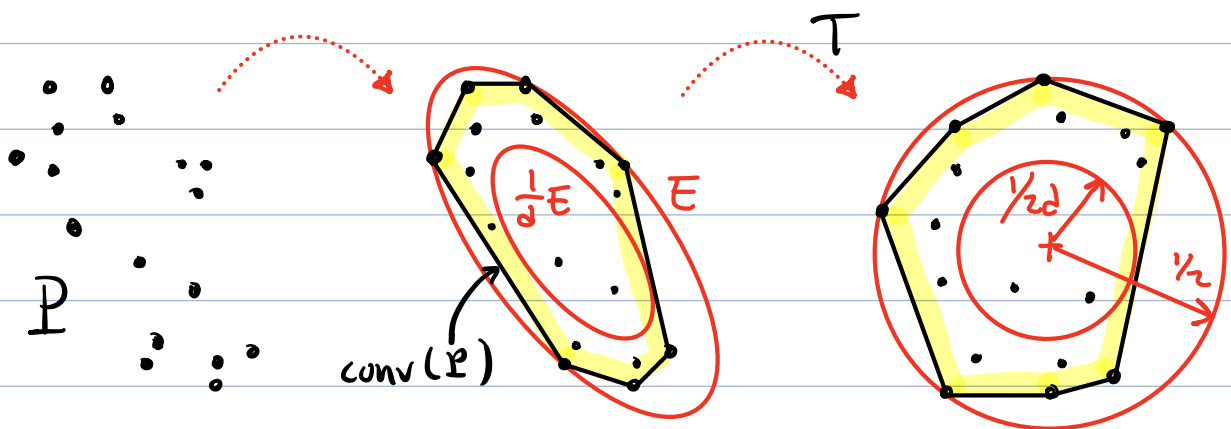


- The ellipsoid is called the **John Ellipsoid** or **Löwner-John Ellipsoid**
- Can compute it in $O(n)$ time (randomized incremental)

Fact: Given any ellipsoid E , there exists an affine transformation that maps E to a unit ball, centered at origin.

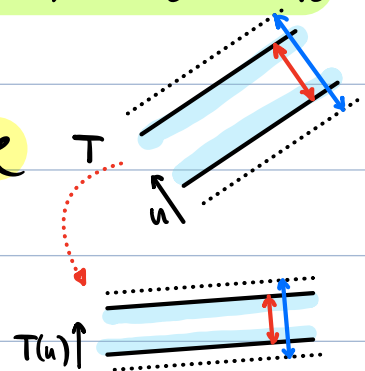
Proof (of canonical form lemma):

- ① Compute P 's outer John ellipsoid E
- ② Find affine transformation mapping E to unit ball centered at origin
- ③ Scale by $1/2 \rightarrow$ output resulting transformation T



Why are directional width approximations preserved?

- Affine transformations preserve ratios of parallel lengths (Details omitted)



Quick + Dirty Kernel: Simple but not optimal size
 $- \mathcal{O}(1/\epsilon^d)$

Given $P \subseteq \mathbb{R}^d$ + $\epsilon > 0$:

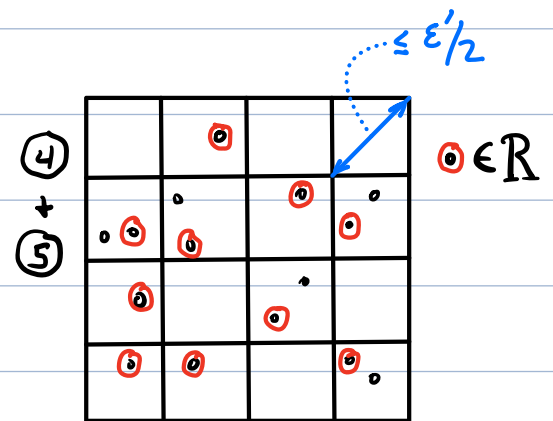
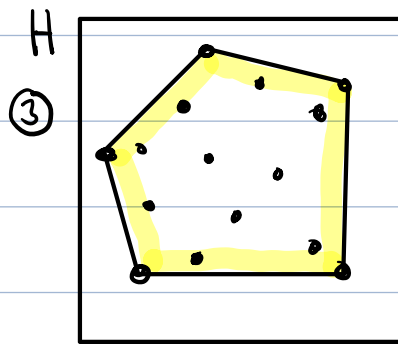
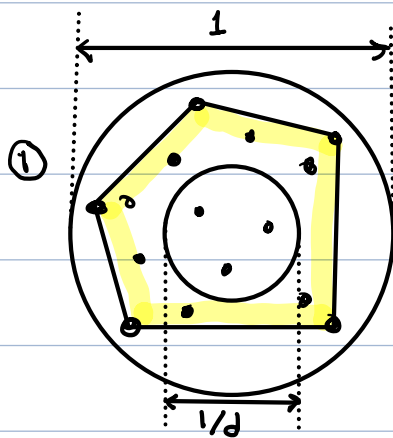
① Map P to $1/2$ -canonical position

Note: $\forall u, 1/2 \leq W_u(P) \leq 1$

$\Rightarrow$ absolute error of $\epsilon/d \Rightarrow$ rel. error $\leq \epsilon$

② Let $\epsilon' = \epsilon/d$

③ Let $H = [-1/2, +1/2]^d$ be unit hypercube containing P



④ Subdivide H into square grid of diameter $\leq \epsilon'/2$ (equiv., side length = $\epsilon'/2\sqrt{d}$)

Note: No. of grid cells is $\left(\frac{1}{\epsilon'/2\sqrt{d}}\right)^d = \mathcal{O}(1/\epsilon^d)$

⑤ $R \leftarrow$ take one pt of P from each occupied cell

Note: $|R| = \mathcal{O}(1/\epsilon^d)$. Computable in $\mathcal{O}(n)$ time

Running time: $O(n + 1/\epsilon^d)$

- Canonical position - $O(n)$

- Place pts in grid cells - $O(n)$

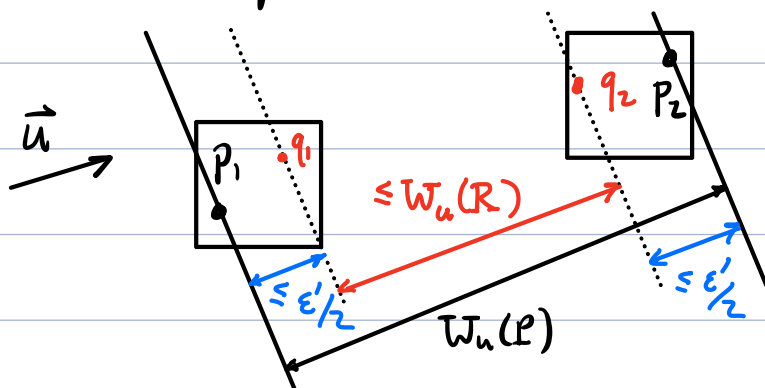
[integer division + hashing]

- Output R - $O(1/\epsilon^d)$

Correctness:

- Given any direction $\vec{u}$, let $p_1, p_2 \in P$ be pts that define $W_u(P)$

- Let $q_1, q_2 \in R$ be corresponding representatives from p_1 + p_2 's cells



- Since cell diameter $\leq \epsilon'/2$, it follows that

$$W_u(P) \leq \epsilon'/2 + W_u(R) + \epsilon'/2$$

$$= \epsilon' + W_u(R) = \epsilon/d + W_u(R)$$

- By canonical form, $W_u(P) \geq 1/d$

$$W_u(P) \leq \epsilon \cdot W_u(P) + W_u(R)$$

$$\Rightarrow (1 - \epsilon) W_u(P) \leq W_u(R) \leq W_u(P)$$

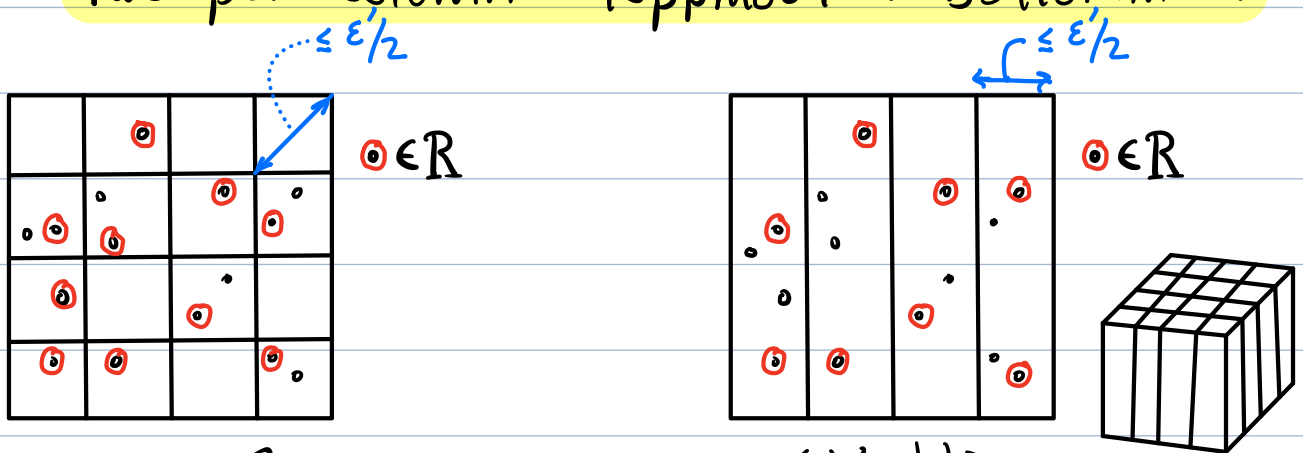
$\Rightarrow R$ is an ϵ -kernel

since $R \subseteq P$



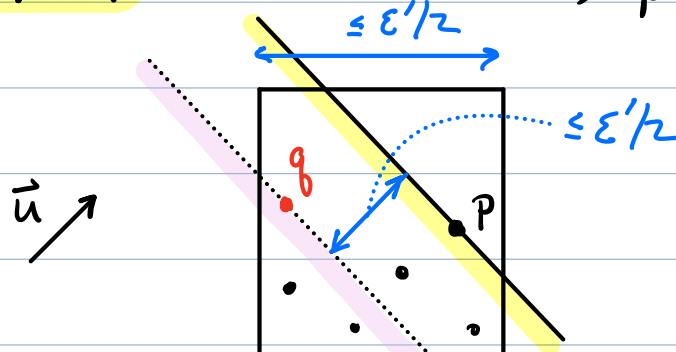
Small Improvement: ~~$O(1/\epsilon^d)$~~ $\rightarrow O(1/\epsilon^{d-1})$

- Quick + dirty's grid includes many internal points $\rightarrow$ wasteful
- Rather than take:
 - one representative per cell, instead
 - two per column - toppmost + bottommost



- How many? Top grid has $O(1/\epsilon^{d-1})$ cells
 $|R| = 2 \cdot O(1/\epsilon^{d-1}) = O(1/\epsilon^{d-1})$

- Correctness? Let p be extreme pt in direction $\vec{u}$ + let $q \in R$ be topmost (or bottommost) pt in column



Directional distance betw. q + p is $\leq \epsilon/2$
 ... remaining details omitted

Big Improvement - ε -kernel of size $O(1/\varepsilon^{\frac{d-1}{2}})$

[Optimal in the worst case]

Construction based on idea discovered
(independently) by Dudley + Bronstejn +
Ivanov (~1974)

① Map P to $1/d$ -canonical position

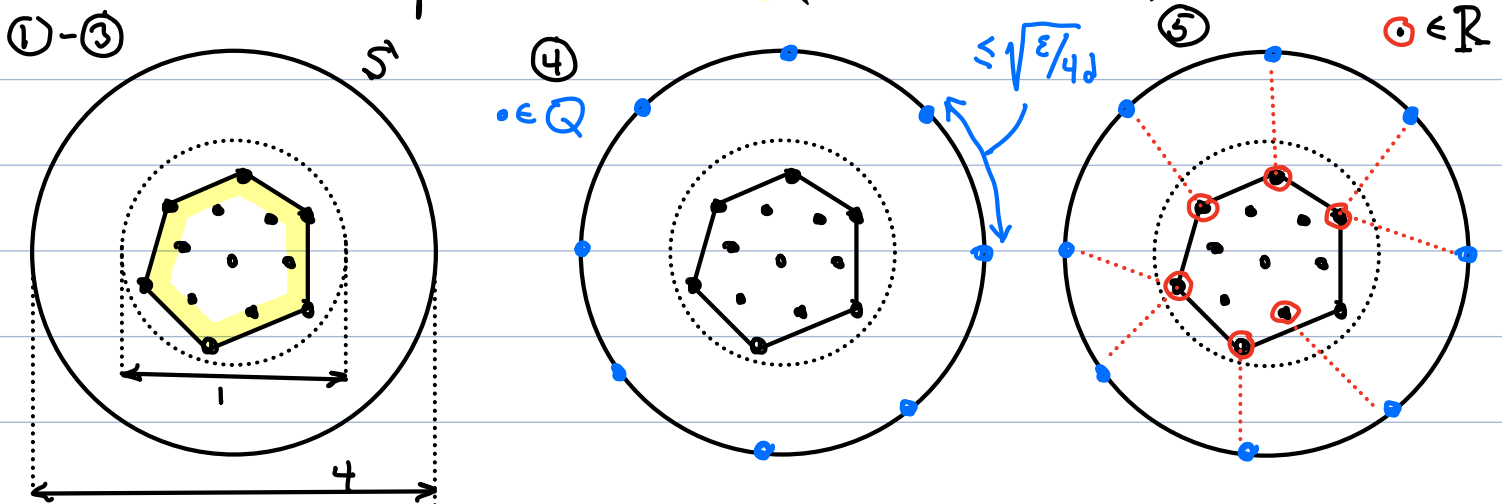
Note: $\forall u, 1/d \leq W_u(P) \leq 1$

$\Rightarrow$ absolute error of $\varepsilon/d \Rightarrow$ rel. error $\leq \varepsilon$

② Let $\varepsilon' = \varepsilon/d$

③ Let $S =$ sphere of radius 1 centered
at origin, let $\delta = \sqrt{\varepsilon'/4d}$

④ Let Q be a set of points on S s.t.
any point of S is within distance δ of
some pt of Q . (Q is " δ -dense")



⑤ For each $q \in Q$, let $nn(q) \in P$ be its closest pt.
Return: $R = \bigcup_{q \in Q} nn(q)$

Size: $|R| \leq |Q|$

- Claim that $|Q| = O((1/\sqrt{\epsilon})^{d-1}) = O(1/\epsilon^{\frac{d-1}{2}})$

- Intuition: Each $q \in Q$ covers a spherical cap of radius $\delta \approx \sqrt{\epsilon}$

- Such a cap has surface area
 $\approx \delta^{d-1} \approx \sqrt{\epsilon}^{d-1} \approx \epsilon^{(d-1)/2}$

- S has constant radius $\Rightarrow$ constant area

- No. caps needed to cover S
 $\approx \text{const} / \epsilon^{(d-1)/2} = O(1/\epsilon^{(d-1)/2})$

$\Rightarrow |R| = O(1/\epsilon^{(d-1)/2})$

Running Time:

- Canonical position: $O(n)$

- Computing δ -dense Q
 $O(|Q|) = O(1/\epsilon^{(d-1)/2})$

How? Enclose S in a hypercube

Cover hypercube with grid $\sim \delta$

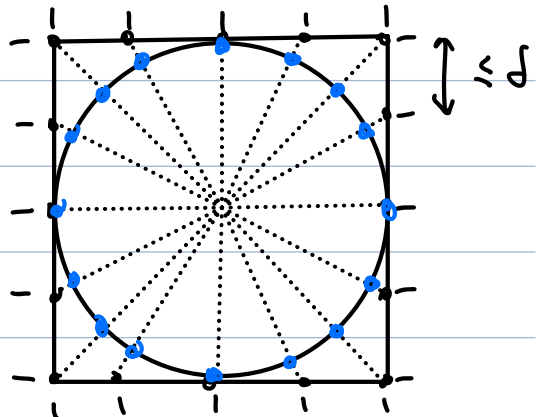
Project onto S

- Compute $nn(q)$

- Suffices to use

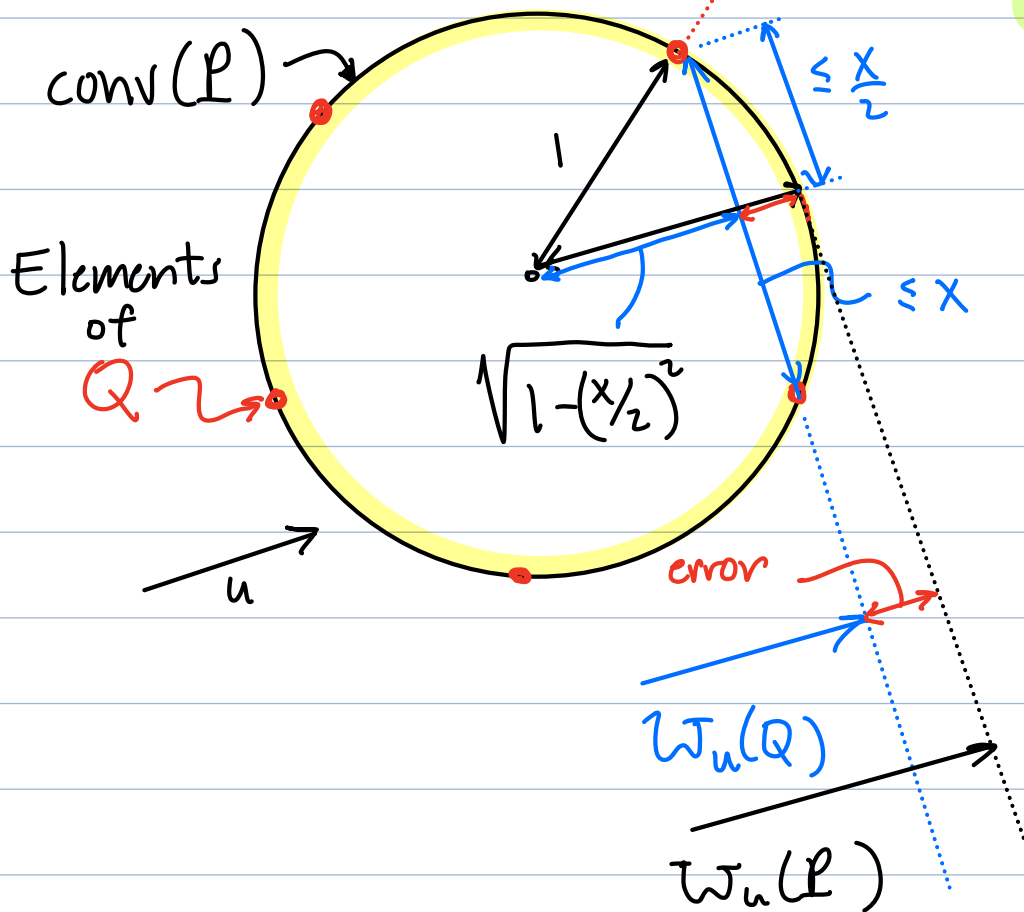
approx nn

- $O(\text{poly}(1/\epsilon) \cdot \log n)$



Correctness: (Complex - See latex notes)
 We'll consider a simpler question:

Why $\sqrt{\epsilon}$ spacing? Simple case: Suppose $\text{conv}(P)$ is a unit ball



To generate a large error select u to hit $\text{conv}(P)$ midway between pts of Q .

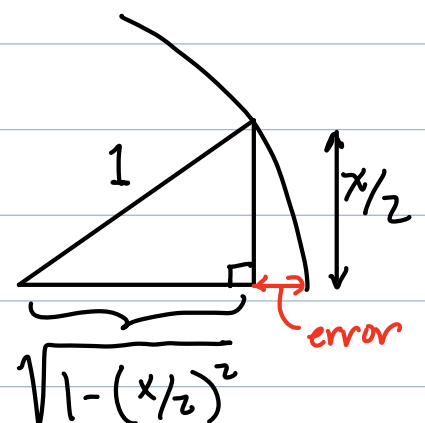
By Pythagorean Thm:

$$\text{error} \leq 1 - \sqrt{1 - (x/2)^2}$$

We want

$$\text{error} \leq \epsilon$$

That is, want x s.t.



$$1 - \sqrt{1 - (x/2)^2} \leq \varepsilon$$

Solving for x , we have:

$$\Leftrightarrow 1 - (x/2)^2 \geq (1 - \varepsilon)^2 = 1 - 2\varepsilon + \varepsilon^2$$

$$\text{if } \varepsilon \leq 1, \text{ then } \varepsilon^2 \leq \varepsilon \Rightarrow 1 - 2\varepsilon + \varepsilon^2 \leq 1 - \varepsilon$$

$$\Leftrightarrow 1 - (x/2)^2 \geq 1 - \varepsilon$$

$$\Leftrightarrow x/2 \leq \sqrt{\varepsilon}$$

$$x \leq 2\sqrt{\varepsilon}$$

- This explains why spacing $\sim \sqrt{\varepsilon}$ is the right thing to do

- Notice this is tight up to constant factors.