On Coresets and Shape Fitting in High Dimensions

Sariel Har-Peled

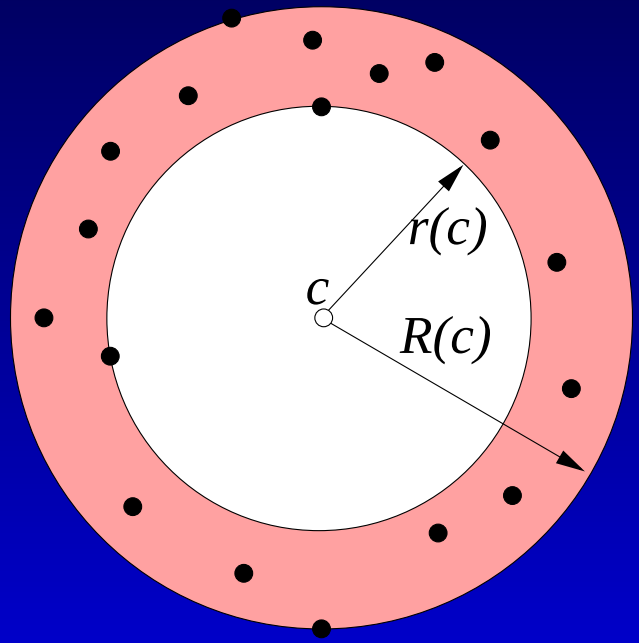
UIUC

Based on...

1. M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 250–257, 2002
2. S. Har-Peled and K. R. Varadarajan. Projective clustering in high dimensions using core-sets. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 312–318, 2002
3. S. Har-Peled and K.R. Varadarajan. High-dimensional shape fitting in linear time. In *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, pages 39–47, 2003

Problem

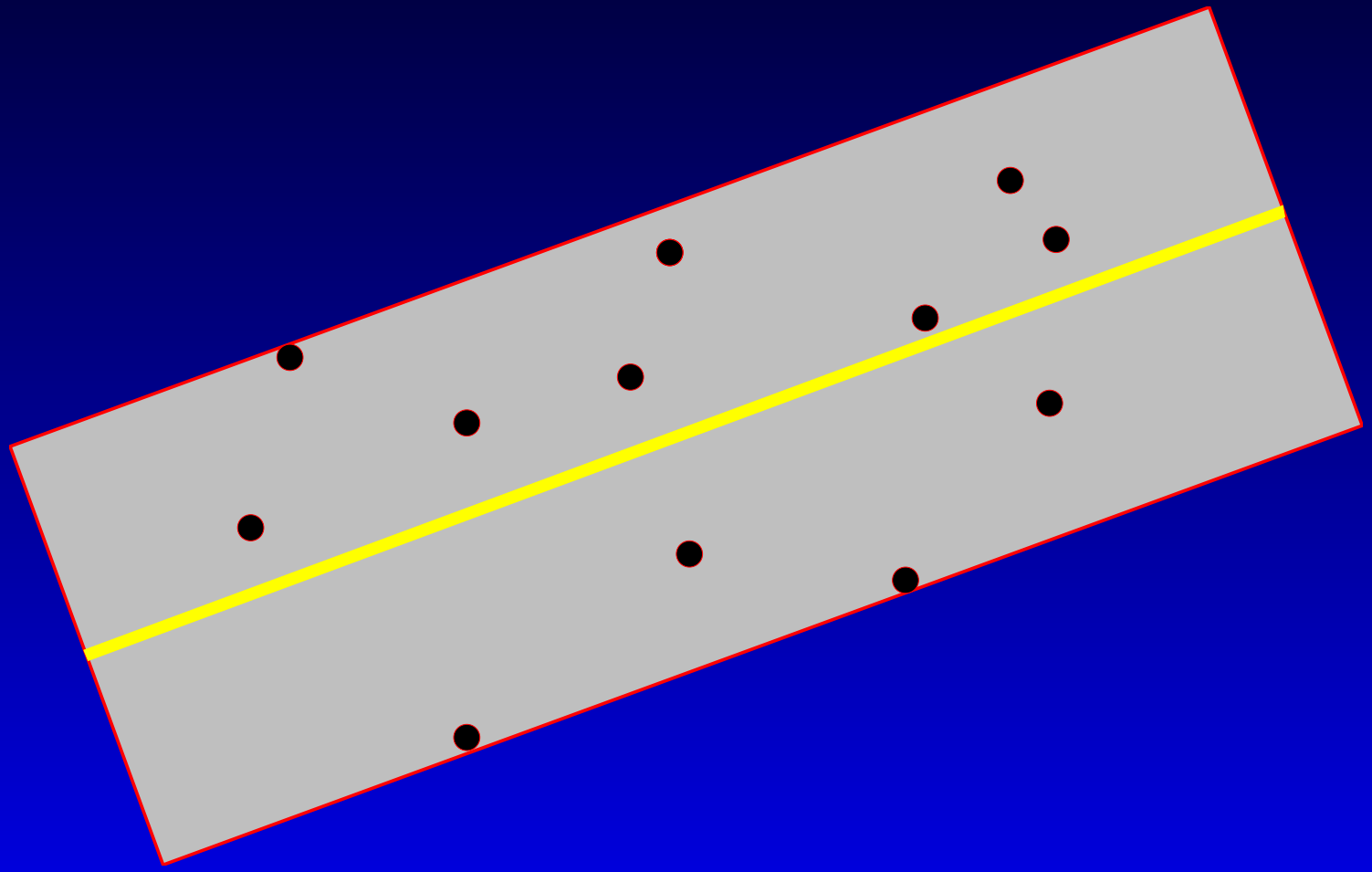
- P - a set of n points in $\mathbb{R}^d$
- $\mathcal{F}$ - a family of shapes
- **Target:** Find best shape in $\mathcal{F}$ matching P.



- **Motivation:** Clustering, learning, graphics...

Example: Find best line

- Finding min width strip/cylinder covering P.



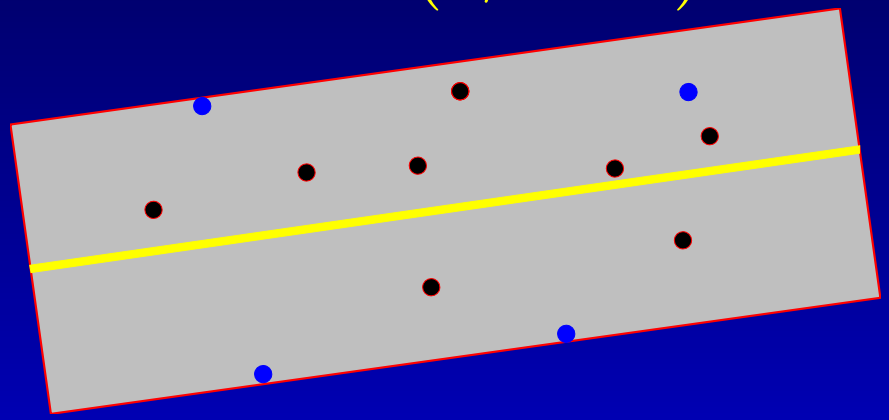
- Fitting $\Leftrightarrow$ covering problems.

Approx in low dim

- **[Agarwal, Har-Peled, Varadarajan,01-03]** Max shape fitting problems
  - ε-approximation
  - Running time: $O\left(n + 1/\varepsilon^{O(1)}\right)$
  - Extracts a coreset of size $1/\varepsilon^{O(1)}$).
- **Problems:** Min ball, min cylinder, min vol bounding box, min width spherical shell, etc.

What is a (low dim) coreset?

- $S \subseteq P$
- S is a ε-*coreset* of P, if for any cover C of S:
  - $P \subseteq (1+\varepsilon)C$
  - S is "small" (i.e., $O\left(1/\varepsilon^{O(d)}\right)$.



- Approximation algorithm:
  - Extract coreset.
  - Find optimal solution on coreset.

Low dim coresets

- Can do:
  - Maintain approx under ins/del (polylog update time).
  - Maintain approx for streams (ins only) with small space.
- Coresets exists for shape fitting:
  - $\#$ of outliers small (work with Yusu Wang).
  - For k-median and k-mean (in progress).
- **Problem:** Coreset size exponential in dim.
- **Q:** Coreset in high dim?

High dim coresets

- Exponential lower bound.
- Weaker coresets:
  - $S \subseteq P$
  - B smallest enclosing ball of S.
  - $P \subseteq (1+\varepsilon)B$.
- **Q:** Exist?
- **Q:** How to compute?
- **Target:** Size independent of dim.

Intuition: Why should high dim coresets exist?

- High dim - glut of information
  - JL Lemma - dimension reduction.
- “Not all points are created equal, some are more equal than others”.
- Input dim vs. target dim.
  - Min ball - target dim 0
  - min cylinder - target dim 1.

Min Ball

- P - a set of points.
- $p \in P$, $q \leftarrow$ farthest neighbor of p in P.
- R - radius of min enclosing ball of P.
- $S_0 \leftarrow \{p, q\}$
- $B_0 \leftarrow$ `MinBallQP`(S_0)
  (`MinBallQP` - using quadratic programming)
- $r_0 = r(B_0)$.
- **Claim:** $R \leq 2r_0$.

`MinBall` Algorithm

- While $P \not\subseteq (1+\varepsilon)B_i$ do
  - $q_i \Leftarrow Furthest_P(B_i)$
  - $S_{i+1} \leftarrow S_i \cup \{q_i\}$.
  - $B_{i+1} \leftarrow \texttt{MinBallQP}(S_{i+1})$
  - $i \leftarrow i+1$
- Return B_i.

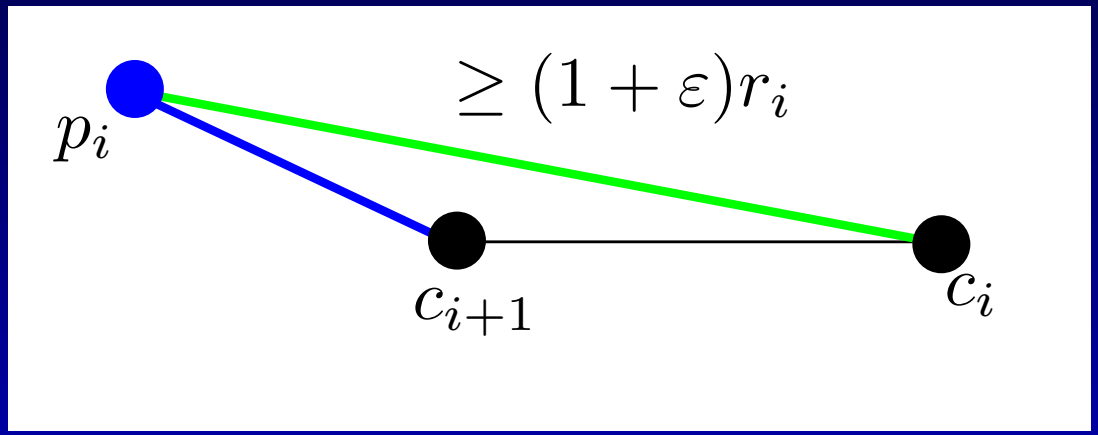
MinBall # of iterations

- $r_{i+1} \leftarrow r(B_{i+1})$.
- If $r_{i+1} \geq (1+\varepsilon^2) r_i$ we made progress.
- $l_{i+1} = \|c_{i+1} c_i\|$ - distance between centers of B_{i+1} and B_i.

MinBall # of iters - short case

If $l_{i+1} < \varepsilon r_i/2$ then

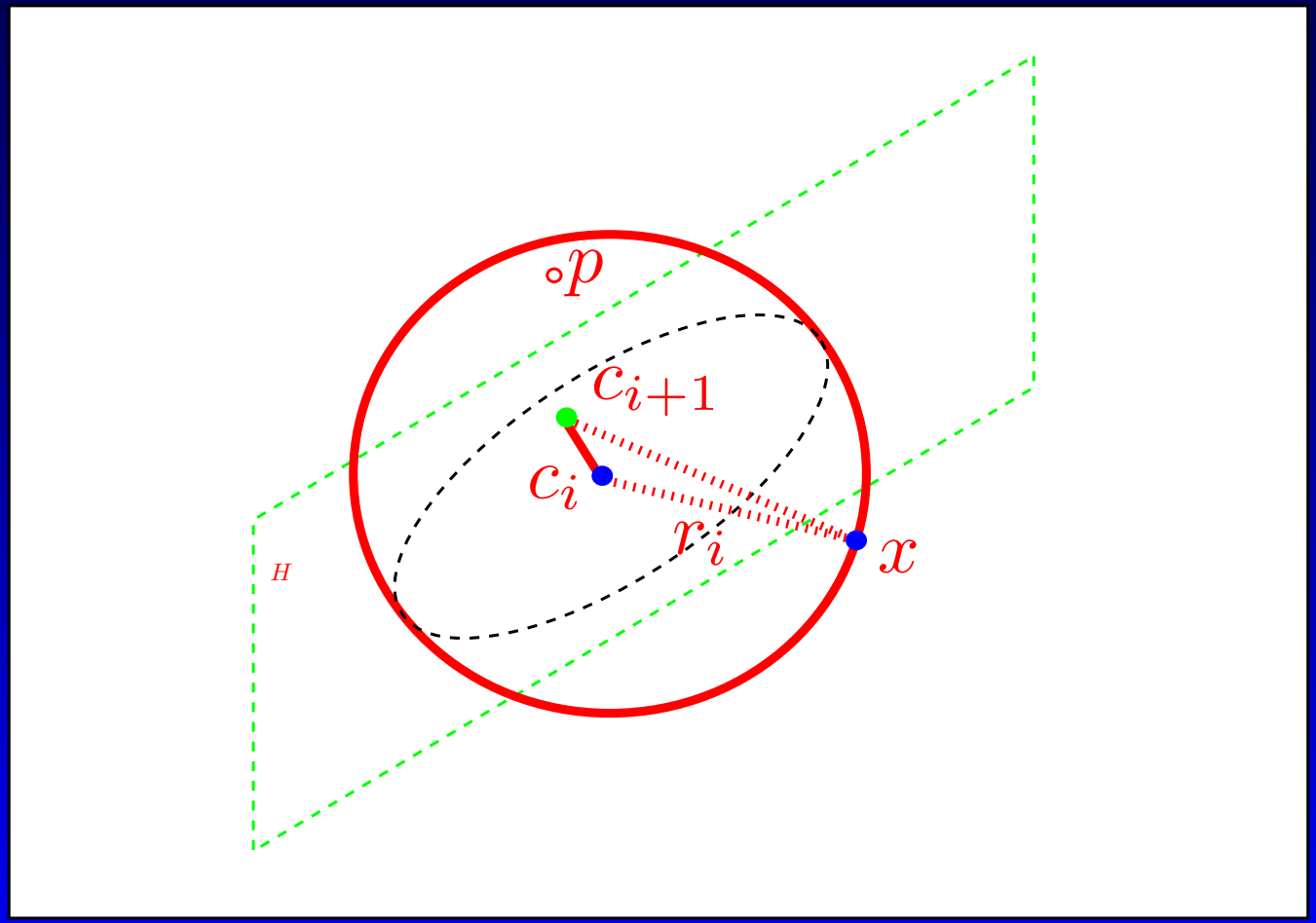
$$r_{i+1} \geq (1+\varepsilon)r_i - l_{i+1} \geq (1+\varepsilon/2)r_i.$$



MinBall # of iters - long case

- If $l_{i+1} > \varepsilon r_i$ then

$$r_{i+1} \geq \sqrt{r_i^2 + (\varepsilon r_i)^2} \geq (1 + \varepsilon^2) r_i.$$



`MinBall` Result

Theorem:
`MinBall` algorithm terminates after $O(1/\varepsilon^2)$ iterations.

[Bădoiu, Clarkson, 02]

Number of iterations $O(1/\varepsilon)$.

`MinBall` Algorithm

- $O(1/\varepsilon)$ iterations. Every iteration:
  - Scan points - linear time.
  - Quadratic programming for `MinBallQP`: $O(1/\varepsilon^{10})$ time.
- Overall running time $O\left(nd/\varepsilon + 1/\varepsilon^{10}\right)$.

k-center clustering

- **Target:** Cover points with k balls.
- Want to run k parallel copies of `MinBall`
- At every iteration:
  - $\mathcal{B}_i$ - set of k current balls.
  - If $P \subseteq (1+\varepsilon)\mathcal{B}_i$ - done.
  - q_i - point furthest away from $\mathcal{B}_i$.

- **Q:** To which of k balls to add q_i?
  A: Ask an oracle.

k-center clustering cont'

Theorem: Compute $(1+\varepsilon)opt$ k-center clustering in $O(ndk/\varepsilon + k/\varepsilon^{10})$ time. Requires $O(k/\varepsilon)$ oracle queries.

Removing the oracle:

Theorem: Compute $(1+\varepsilon)opt$ k-center clustering in $k^{O(k/\varepsilon)}(nd/\varepsilon + k/\varepsilon^{10})$ time.

Min Ball with Outliers

- P - set of n points.
- A large subset of P are outliers.
- **Q:** Find smallest ball that contains m points of P.
- $Q \subseteq P$ - required set.
- $S \subseteq Q$ - coreset of MinBall of Q of size $O(1/\varepsilon)$.

Min Ball with Outliers

- **Algorithm:**
  - Try all subsets $\subseteq P$ of size $O(1/\varepsilon)$:
    - Compute $B \leftarrow$ `MinBall`(S)
    - Compute how many points of P covered by B
  - Return best ball computed
- Running time $n^{O(1/\varepsilon)}$.

1-Median

- A random sample span a subspace that contains a good center.
  Yield a fast algorithm for approximate 1-median...
- Can be extended to k-median. Either:
  - All clusters are “big”.
    All clusters can be sampled using random sampling. And solved on subspace.
  - Sample large cluster, remove it, and recurse on remaining points.

1-Median - Result

Theorem:
For any point-set $P \subset \mathbb{R}^d$, $\varepsilon < 1$, k, a $(1+\varepsilon)$-approximate k-median for P in time

$$2^{(k/\varepsilon)^{O(1)}} d^{O(1)} n \log^{O(k)} n$$

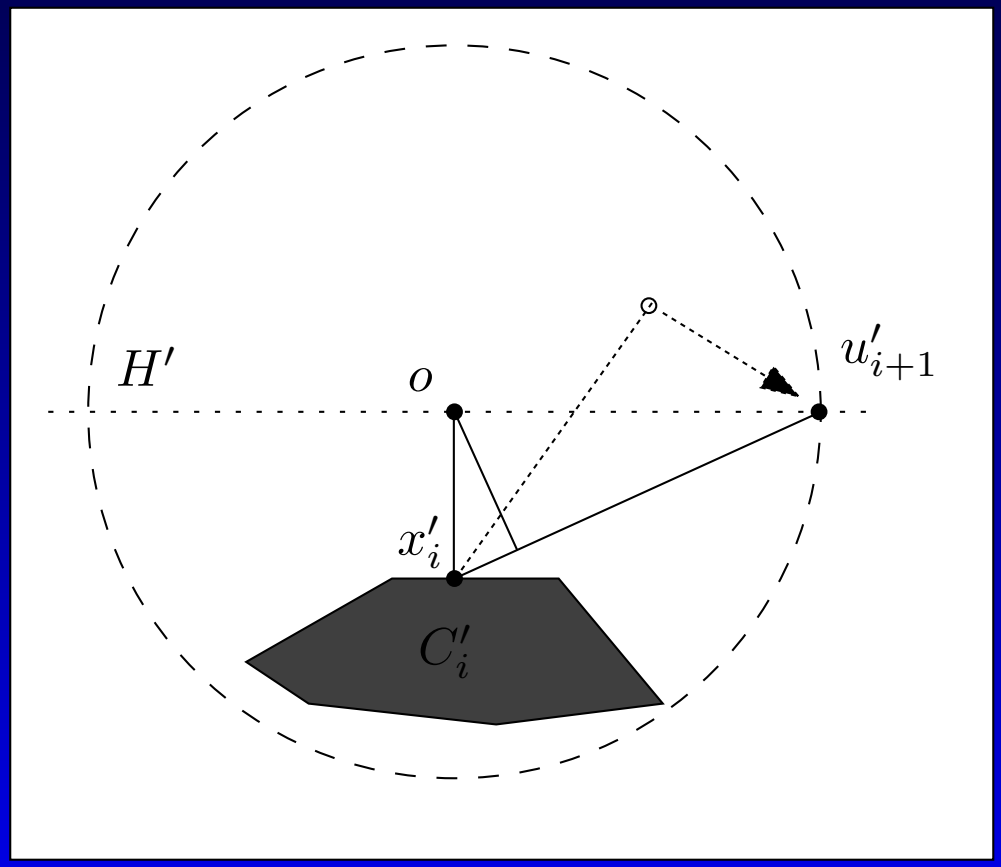
Min Radius Cylinder

- Find a line close to the optimal center line.
- Idea: Find two points close to center line - use spanned line as center.
- **Q:** How to find points?

Min Cyld - Finding one point close to cline

Observation:

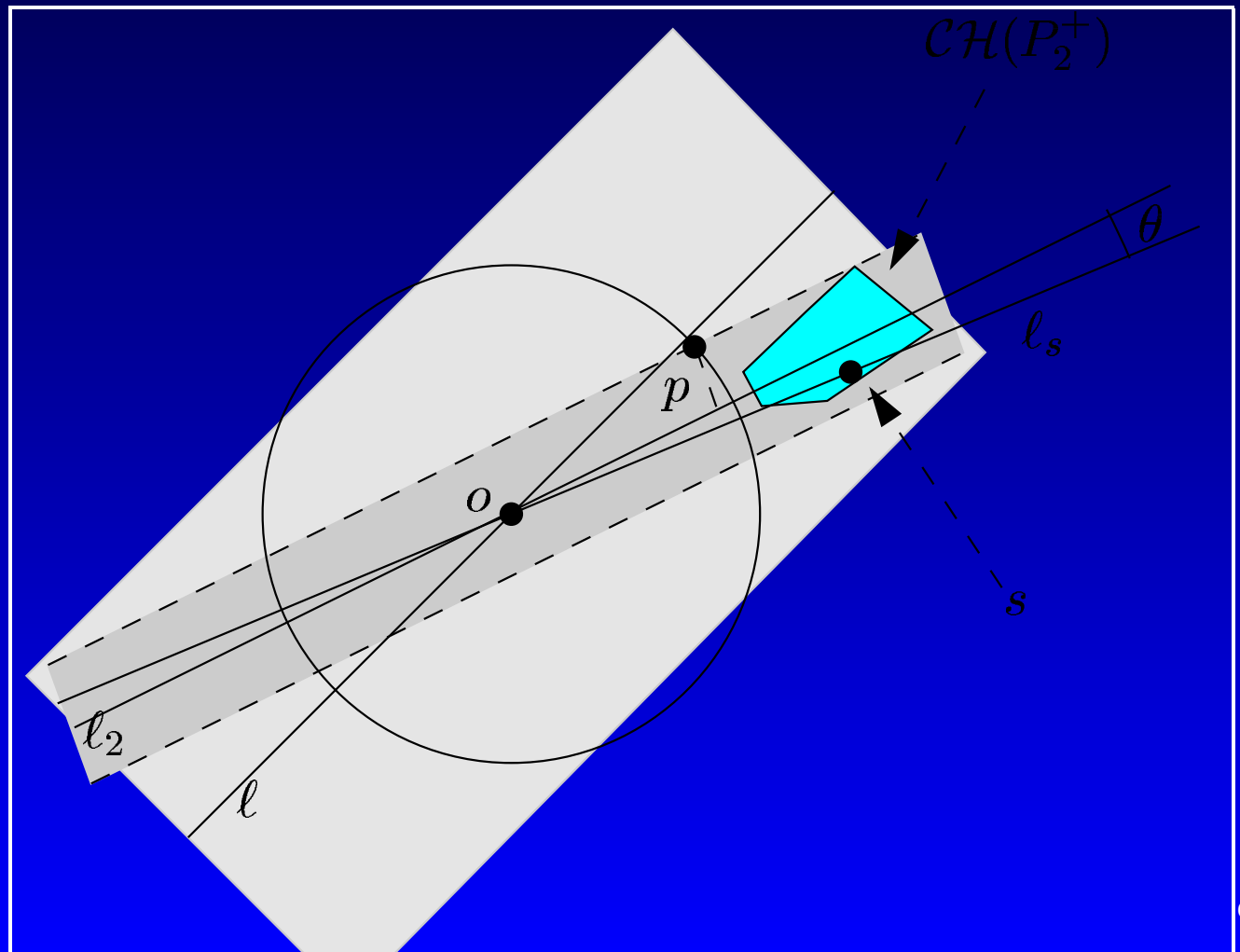
One point is easy using similar argument to MinBall coreset...



Min Radius Cylinder - cont'

o - point on center line of cylinder (via coreset)
Two cases to analyze:

- Short case: $\mathrm{diam}(P) \leq r_{\mathrm{opt}}(P)/\varepsilon$.



Min Radius Cylinder - cont'

- Long case: $\mathrm{diam}(P) > r_{\mathrm{opt}}(P)/\varepsilon$
  Remove points inside $\mathrm{ball}(o, \varepsilon\,\mathrm{diam}(P))$, and use similar argument to short case.

Min Radius Cylinder - cont'

Theorem: Given a pointset P in $\mathbb{R}^d$, one can find a coreset S of size $O(1/\varepsilon^5 \log 1/\varepsilon)$. Such that $\mathrm{span}(S)$ contains $(1+\varepsilon)$-approximate line center.

- Enumerate all such coresets: n^ρ

$$\rho = O\left(\frac{1}{\varepsilon^5}\log\frac{1}{\varepsilon}\right).$$

- Compute for each such coreset its line.
- Return best line out of all of them.

Min Radius Cylinder - cont'

1. **Theorem:** Given a pointset P in $\mathbb{R}^d$, one can compute a cylinder covering P of width $\leq (1+\varepsilon) r_{\text{opt}}(P)$, in $dn^{1/\varepsilon^5 \log 1/\varepsilon}$ time.
2. **[Bădoiu, Clarkson, 02]**
   Coreset of cylinder of size $O(1/\varepsilon^4)$.

3. Find min-cylinder with outliers: $n^{O\left(1/\varepsilon^4\right)}$.
4. Find cover by j min-cylinders: $n^{O\left(j/\varepsilon^4\right)}$.

Min k-flat

1. $v_1, \ldots, v_{k-1}, v_k$ vectors spanning optimal k-flat.
2. Collapse space by projecting along $v_1, v_2, \ldots, v_{k-1}$.
3. Optimal k-flat - optimal cylinder in collapsed space.
4. By coreset find approx cylinder line ℓ.
5. Collapse original space along ℓ.
6. Repeat argument on collapsed space.

$\exists$ a coreset of size $O(k/\varepsilon^4)$ for min-width k-flat.

Min k-flat

1. Coreset of cylinder of size $O(k/\varepsilon^4)$. (dim independent!)
2. Find min-width k-flat + outliers: $n^{O\left(k/\varepsilon^4\right)}$.
3. Find cover by j min-width k-flats:

$$n^{O\left(jk/\varepsilon^4\right)}.$$

Running time not exciting, can we do better?

Min Width Cylinder Lower Bound

By extending a lower bound of Megiddo (Meg90):

- Unless $\mathbb{P} = \mathbb{NP}$.
- There is no algorithm:
  - Input: P - set of n points in $\mathbb{R}^d$,$0 < \varepsilon \leq 1$.
  - Output: r, such that $r_{\text{opt}}(P) \leq r \leq (1+\varepsilon) r_{\text{opt}}(P)$.
  - runs in time polynomial in n, d, and $1/\varepsilon$.

Min Width Cyl' in Linear Time

- **Idea:** Extend the algorithm for `MinBall`.
  - Start from two good points in coreset.
  - While cover for coreset not $(1+\varepsilon)$-cover
    - Add furthest pnt from curr line to coreset.
    - Recompute cover of coreset by a cylinder.
- If algorithm do $O(1/\varepsilon^{O(1)})$ iters, then...
- Running time: $O\left(nd/\varepsilon^{O(1)} + 1/\varepsilon^{O(1/\varepsilon)}\right)$.
- **Conjecture:** This algorithm works.

Min Width Cylinder

Quantifying Progress

- `MinBall` was analyzable because argument showed progress in every step.
- ℓ_{opt} optimal line center.
- Observation: If a line ℓ is close to ℓ_{opt} in "interesting region", then ℓ good approximation.
- I_{opt} - interval of ℓ_{opt} containing projection of P.
- **Lemma:** If $\forall x \in I_{\text{opt}}$ we have:
  $\text{dist}(x, \ell) \leq \varepsilon r_{\text{opt}}$
  then ℓ required approximation.

Min Width Cylinder Algorithm

- ℓ_0 - line passing through a point $x \in P$, and its furthest neighbor in P.
- $R \leftarrow$ Guess optimal radius.
- While $\mathrm{dist}(\ell_i, P) \geq (1+\varepsilon)R$ do
  - $q_i \leftarrow$ furthest point in P from ℓ_i.
  - $H_i \leftarrow \mathrm{span}(\ell_i, q_i)$.
  - $\ell_{i+1} \leftarrow$ Projection of ℓ_{opt} to H_i.
  - $i \leftarrow i+1$
- Return ℓ_i.

Min Width Cylinder - Cont'

- For a line ℓ_i, define distance function:
  - For $x \in I_{\text{opt}}$, $f_i(x) = \text{dist}(x, \ell_i)$.
- Observations:
  - $f_i(x) \leq f_{i-1}(x)$.
  - $f_0(x) \leq 5r_{\text{opt}}$.

  for $x \in I_{\text{opt}}$.
- If could just show that:
  - $f_i(x) \leq (1 - \varepsilon) f_{i-1}(x)$

  then bound $\#$ iters by $1/\varepsilon^{O(1)}$.

Min Width Cylinder - Cont'

- At every iteration, a portion of I_{opt} is "hit".
- $T \subseteq I_{\text{opt}}$, $|T| = 1/\varepsilon^{O(1)}$.
- There exists a point $t_i \in T$ s.t.
  - $f_{i-1}(t_i) \leq (1-\varepsilon) f_i(t_i)$.
- After $1/\varepsilon^{O(1)}$ iterations

$$\forall t \in T \qquad f_i(t) \leq \varepsilon r_{\text{opt}}.$$

By convexity

$$\forall x \in I_{\text{opt}} \qquad f_i(x) \leq \varepsilon r_{\text{opt}}.$$

And we found the required approximation.

Why is a point being hit?

- p_i - furthest point in P from ℓ_i.
- q_i - projection of p_i on optimal line ℓ_{opt}.
- u_i - projection of q_i to ℓ_i.

We can show that

$$f_{i+1}(q_i) \leq (1 - \varepsilon/2) f_i(q_i).$$

Why a point is being hit? Cont'

We know that:

- $\|p_i q_i\| \leq r_{\text{opt}}$.
- $f_i(q_i) = \|p_i u_i\| \geq (1+\varepsilon) r_{\text{opt}}$.
- $p_i u_i \subseteq H_i$.

one can show

$$\begin{aligned} f_{i+1}(q_i) &= \text{dist}(q_i, H_i) \\ &\leq \text{dist}(q_i, p_i u_i) \\ &\leq (1-\varepsilon/2)\text{dist}(q_i, u_i) \\ &= (1-\varepsilon/2) f_i(q_i). \end{aligned}$$

Without a little help from an Oracle

Life without an oracle.

- Oracle project opt. line to curr. plane.
- Second rate (drunk?) oracle would suffice.
- Oracle project line approx.
- small # of candidate lines.
- ...and a second rate oracle can be replaced.
- Generate all possible oracle guesses.

Linear Time Min Cylinder

- Running time of algorithm with oracle: $O(nd/\varepsilon^3 \log(1/\varepsilon) + noise)$.
- Oracle provides $O\left(1/\varepsilon^3 \log^2(1/\varepsilon)\right)$ random bits.
- Brute force enumeration, gives an algorithm with

$$nd \cdot \exp\left(O\left(\frac{1}{\varepsilon^3}\log^2\frac{1}{\varepsilon}\right)\right)$$

which outputs a cylinder of width $\leq (1+\varepsilon) r_{\text{opt}}$.

Linear Time Min Width k-Flat

- A tedious extension of Cylinder alg.
- Algorithm with

$$n \cdot d \cdot \exp\left(e^{O(k^2)}/\varepsilon^{2k+3}\right)$$

  running time.
- output a k flat of width $\leq (1+\varepsilon)^k r_{\text{opt}}$.

Open problems

- Big discrepancy between algorithm with outliers and without outliers.
- A near linear time algorithm for the case of k-cylinders.
- An efficient approximate to the k-slab problem in 3d.