CSE520 Computational Geometry
Lecture 22
Geometric Approximation Algorithms II

Antoine Vigneron

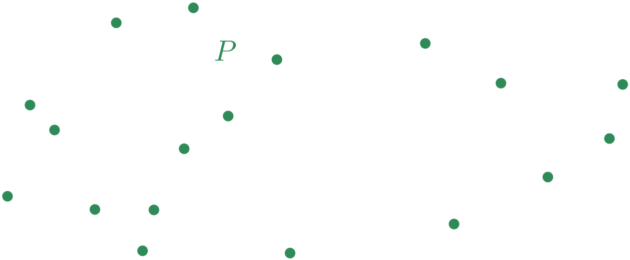
Ulsan National Institute of Science and Technology

June 15, 2020

Course Organization

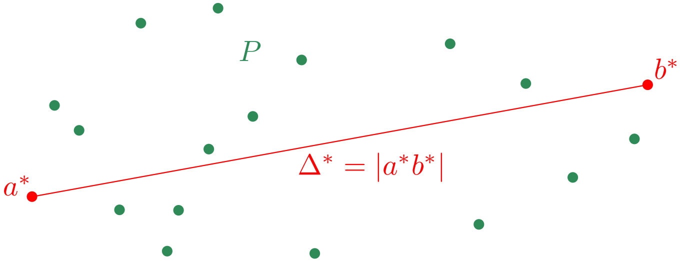
- Today, I will show how to improve the algorithm from Lecture 21 for approximating the diameter of a point set.
- References:
- Sariel Har Peled's book.
- Paper by T. Chan, *Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus*, Section 2.

The Diameter Problem



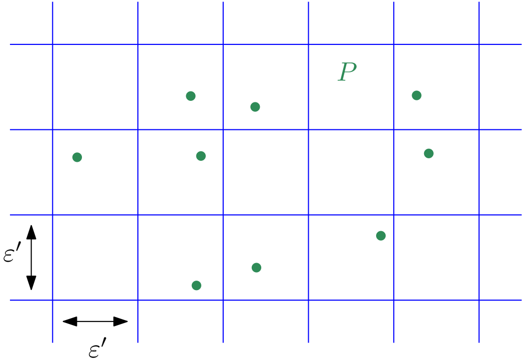
- Input: a set P of n points in $\mathbb{R}^d$.

The Diameter Problem



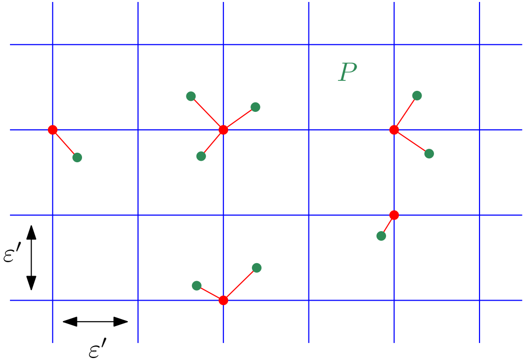
- Input: a set P of n points in $\mathbb{R}^d$.
- Output: the maximum distance Δ^* between any two points of P.
- $\Delta^* = \text{diam}(P)$ is the *diameter* of P.

Rounding to a Grid



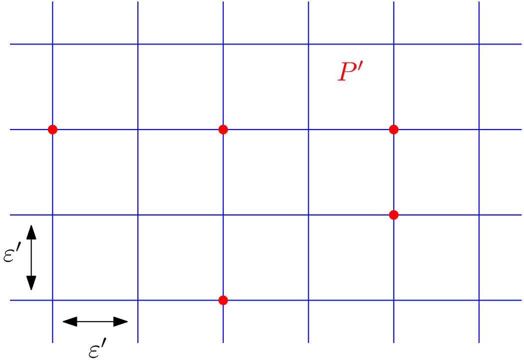
- Consider a regular grid over $\mathbb{R}^d$.
- The side length of the grid is ε', to be specified later.
- Intuition: we will choose $\varepsilon' \approx \varepsilon\Delta^*$, which is the error we allow.

Rounding to a Grid



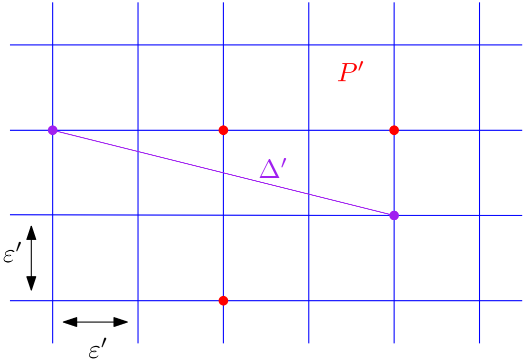
- Replace each point of P with the nearest grid point.
- This operation is called *rounding*.

Rounding to a Grid



- The grid points we obtain form the set P'.

Rounding to a Grid

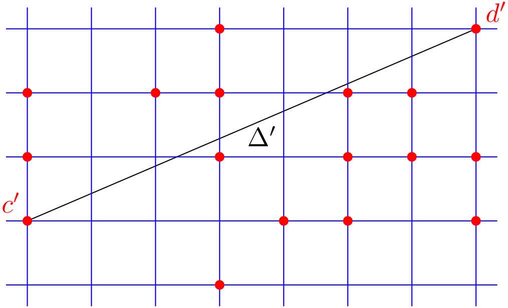


- Compute $\Delta' = \text{diam}(P')$ by brute force.

Rounding to a Grid

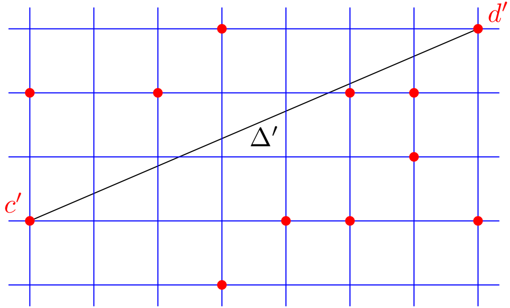
- Let Δ_0 be the 2-approximation of Δ^* that we computed in linear time.
- In the previous lecture, we saw that if we choose $\varepsilon' = \dfrac{\varepsilon \Delta_0}{4\sqrt{d}}$, then $\Delta = \Delta' - \varepsilon'\sqrt{d}$ satisfies $\Delta \leqslant \Delta^* \leqslant (1+\varepsilon)\Delta^*$, and $|P'| = O(1/\varepsilon^d)$.
- P' can be computed in linear time, so Δ can be computed in $O(1/\varepsilon^{2d})$ time by brute force.
- We now explain how to improve this result by a simple observation.

Grid Cleaning: Example in $\mathbb{R}^2$



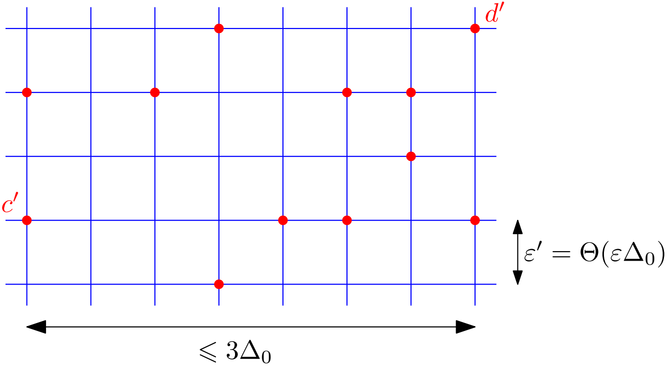
- c' is the lowest point on a vertical line
- d' is the highest point on a vertical line

Grid Cleaning: Example in $\mathbb{R}^2$



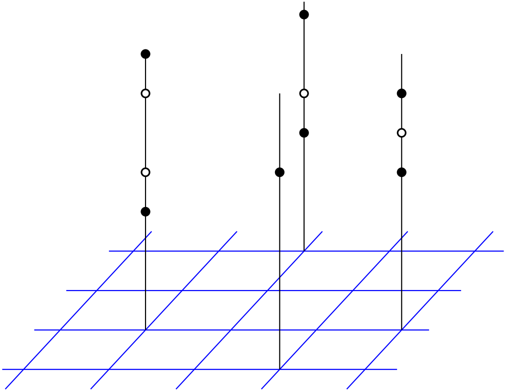
- We only keep the highest and the lowest point on each vertical line.
- Then run the brute-force algorithm.

Analysis



- When $d = 2$, only $O(1/\varepsilon)$ points remain after grid cleaning, because there are $O(1/\varepsilon)$ vertical lines containing a point of P'.
- So we are left with $O(1/\varepsilon)$ points instead of $O(1/\varepsilon^2)$.
- The algorithm runs in $O(n + 1/\varepsilon^2)$ time instead of $O(n + 1/\varepsilon^4)$.

Analysis



- In higher dimension, consider rounded points that coincide in their first $(d-1)$ coordinates.
- Keep only highest and lowest. Then only $O((1/\varepsilon)^{d-1})$ points remain.
- Compute their diameter by brute force.

Analysis

- Grid cleaning can be done in $O(1/\varepsilon^{d-1})$ time:
- WLOG, suppose that the smallest value of the ith coordinate of the points in P' is 0 for all i.
- Construct a $(d-1)$-dimensional array $L[0 \dots E][0 \dots E] \dots [0 \dots E]$ where $E = \Theta(1/\varepsilon)$.
- $L[k_1][k_2] \dots [k_{d-1}]$ records the lowest point on the vertical line through the point $(k_1\varepsilon', k_2\varepsilon', \dots, k_{d-1}\varepsilon')$.
- This array has $O(1/\varepsilon^{d-1})$ cells.
- Same with the highest point.
- So rounding + grid cleaning yields a running time $O(n + 1/\varepsilon^{2d-2})$, instead of $O(n + 1/\varepsilon^{2d})$.

Projecting on Lines

- We measure angles in radian.
- That is, an angle is in $[0, 2\pi]$.

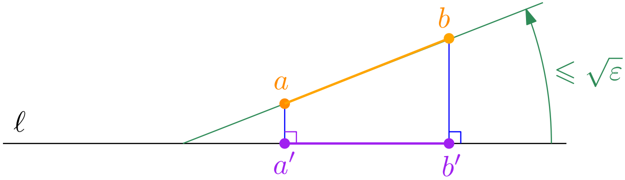
Property

For any α,

$$1 - \frac{\alpha^2}{2} \leqslant \cos\alpha \leqslant 1.$$

- Idea: We get a relative error ε by choosing α to be roughly $\sqrt{\varepsilon}$.

Projecting on Lines



$$|a'b'| \leqslant |ab| \leqslant (1+\varepsilon)|a'b'|$$

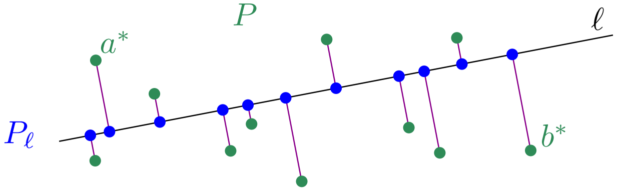
Projecting on Lines

- Assume that the angle between line ab and line ℓ is at most $\sqrt{\varepsilon}$.
- a' (resp. b') is the orthogonal projection of a (resp. b) into ℓ.
- Then $|a'b'| \leqslant |ab|$ and

$$\begin{aligned} |ab| &\leqslant \frac{|a'b'|}{\cos\sqrt{\varepsilon}} \\ &\leqslant |a'b'| \times \frac{1}{1-\varepsilon/2} \\ &= |a'b'| \times \left(1 + \frac{\varepsilon}{2} + \frac{\varepsilon^2}{4} + \dots\right) \\ &\leqslant |a'b'|(1+\varepsilon) && \text{since } \varepsilon < 1 \end{aligned}$$

- In other words, $|a'b'|$ is a $(1+\varepsilon)$–factor approximation of $|ab|$.

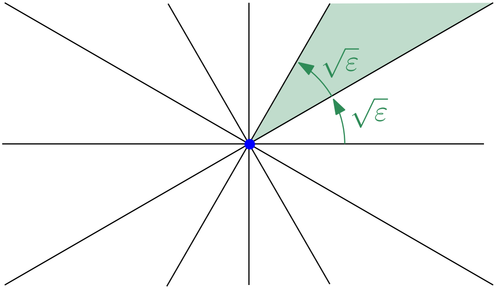
Approach



- P_ℓ is obtained by projecting P onto a line ℓ.
- Compute $\text{diam}(P_\ell)$.
- Can be done in $O(n)$ time: Find maximum and minimum along ℓ.

Approach

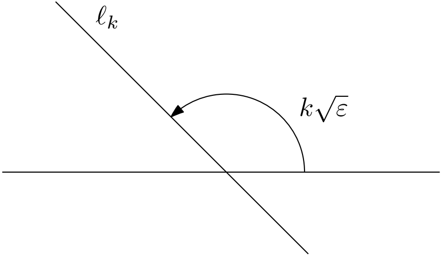
- If the angle between ℓ and a^*b^* is less than $\sqrt{\varepsilon}$, then $\mathrm{diam}(P_\ell)$ is a $(1+\varepsilon)$-factor approximation of Δ^*.
- How can we find a line that makes an angle $\sqrt{\varepsilon}$ with a^*b^*?



$O(1/\sqrt{\varepsilon})$ cones with angular diameter $\sqrt{\varepsilon}$

- Take several lines. In the plane: angle $k\sqrt{\varepsilon}$ for each $k \in [0, \pi/\sqrt{\varepsilon}]$.

Algorithm in $\mathbb{R}^2$



- For each integer $k \in [0, \pi/\sqrt{\varepsilon}]$, we denote by ℓ_k a line that makes angle $k\sqrt{\varepsilon}$ with horizontal. Project P onto ℓ_k, obtaining P_{ℓ_k}.
- Then we will prove that $\Delta = \max_k (\mathrm{diam}(P_{\ell_k}))$ is a $(1+\varepsilon)$-factor approximation of $\mathrm{diam}(P)$.

Algorithm in $\mathbb{R}^2$: Analysis

- Projecting onto a particular ℓ_k takes time $O(n)$.
- Computing $\text{diam}(P_{\ell_k})$ takes time $O(n)$.
- There are $O(1/\sqrt{\varepsilon})$ such lines.
- Overall running time: $O(n/\sqrt{\varepsilon})$.

Algorithm in $\mathbb{R}^2$: Proof

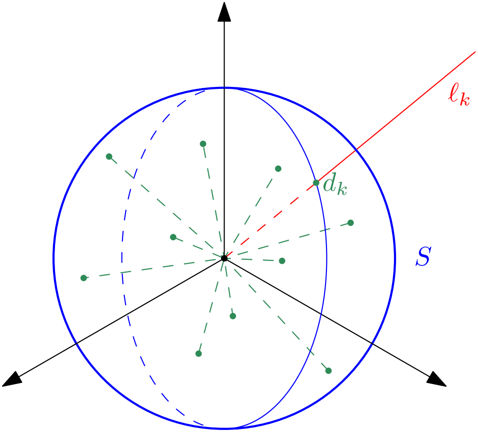
- Let θ be the angle of a^*b^* with horizontal.
- There exists k such that $k\sqrt{\varepsilon} \leqslant \theta < (k+1)\sqrt{\varepsilon}$.
- The angle between a^*b^* and ℓ_k is at most $\sqrt{\varepsilon}$.
- So $\mathrm{diam}(P_{\ell_k})$ is at least $\Delta^*/(1+\varepsilon)$.
- On the other hand, the algorithm only looks at distances between two projected points, which are always smaller than Δ^*.
- So we have
$$\frac{1}{1+\varepsilon}\Delta^* \leqslant \max_k \left(\mathrm{diam}(P_{\ell_k})\right) \leqslant \Delta^*$$
which means that
$$\frac{1}{1+\varepsilon}\Delta^* \leqslant \Delta \leqslant \Delta^*.$$

Algorithm in $\mathbb{R}^2$: Proof

- It implies that $(1-\varepsilon)\Delta^* \leqslant \Delta \leqslant \Delta^*$.
- We say that these inequalities mean that Δ is a $(1-\varepsilon)$-approximation of Δ^*.

- According to the definition from the previous lecture, we wanted to prove that $\Delta^* \leqslant \Delta \leqslant (1+\varepsilon)\Delta^*$.
- There is not a big difference: In both cases, the relative error is ε.
- For instance, when $\varepsilon = 0.01$, in both cases, we make a 1% error.

Generalization in $\mathbb{R}^d$



Generalization in $\mathbb{R}^d$

- Problem: Find a set of directions that approximates well the set of all directions.

- Reformulation:
- Let S be the unit sphere in $\mathbb{R}^d$.
- Find $D \subset S$ with small cardinality such that $\forall x \in S$ there is a point $d_k \in D$ such that $|d_k x| \leqslant \sqrt{\varepsilon}$.
- Each point $d_k \in D$ is associated with the line ℓ_k through the origin and d_k.
- d_k handles a cone of direction with angular radius $\sqrt{\varepsilon}$.

- Such a set D with cardinality $O(1/\varepsilon^{(d-1)/2})$ can be computed efficiently.
- So the algorithm runs in time $O(n/\varepsilon^{(d-1)/2})$.

Combining the two Techniques

- Running times:
- Grid + cleaning: $O(n + 1/\varepsilon^{2d-2})$.
- Projections: $O(n/\varepsilon^{(d-1)/2})$.

- Improvement:
- First round to P' and do grid cleaning.
- We are left with $O(1/\varepsilon^{d-1})$ points.
- Project on lines.
- Overall running time: $O(n + 1/\varepsilon^{3(d-1)/2})$

- Technical problem: the relative error is now bounded by $\approx 2\varepsilon$. How can we solve it?