High-Dimensional Geometric Streaming in Polynomial Space

Taisuke Yasuda
CMU

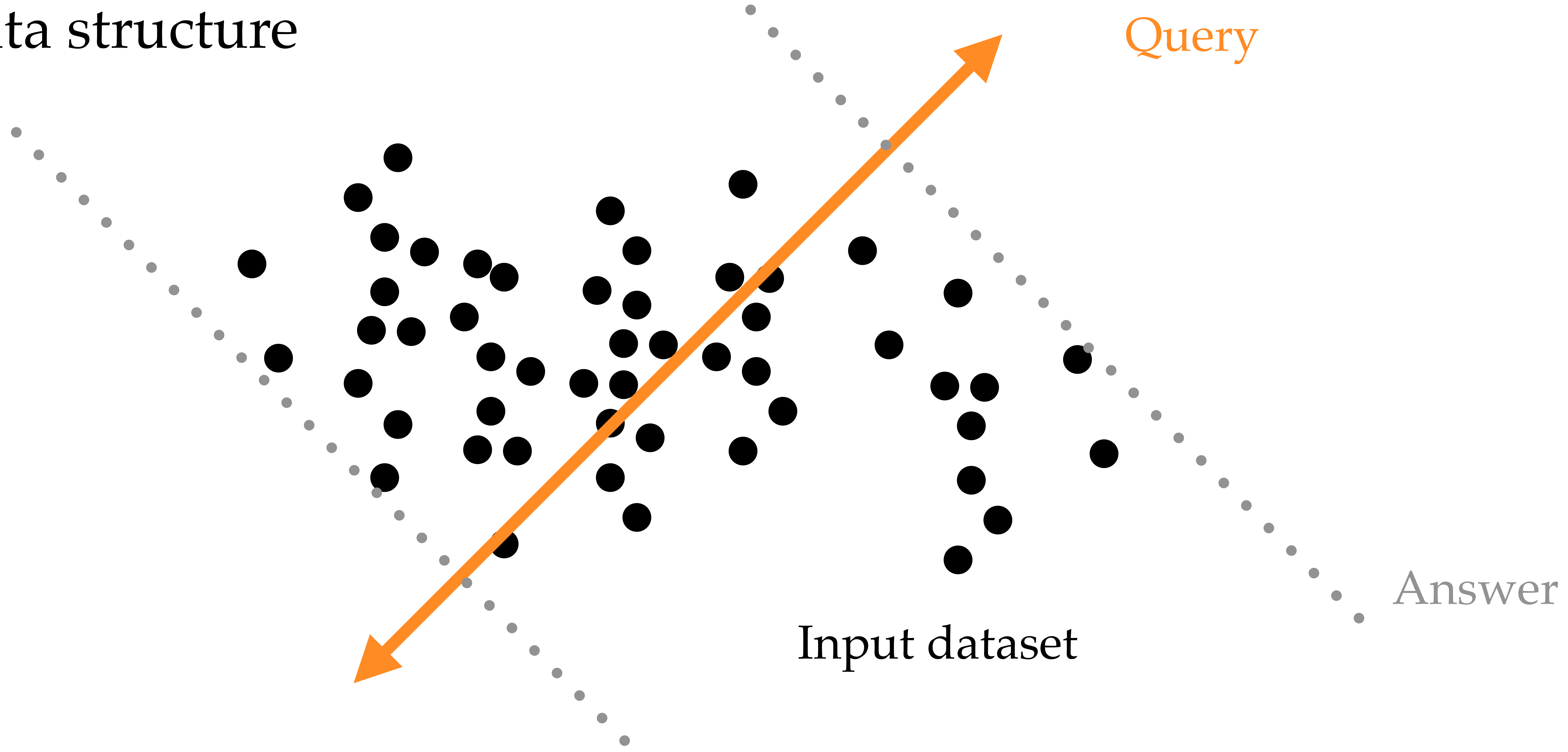
based on work with

David P. Woodruff
CMU

Width Estimation of a Dataset

Setup

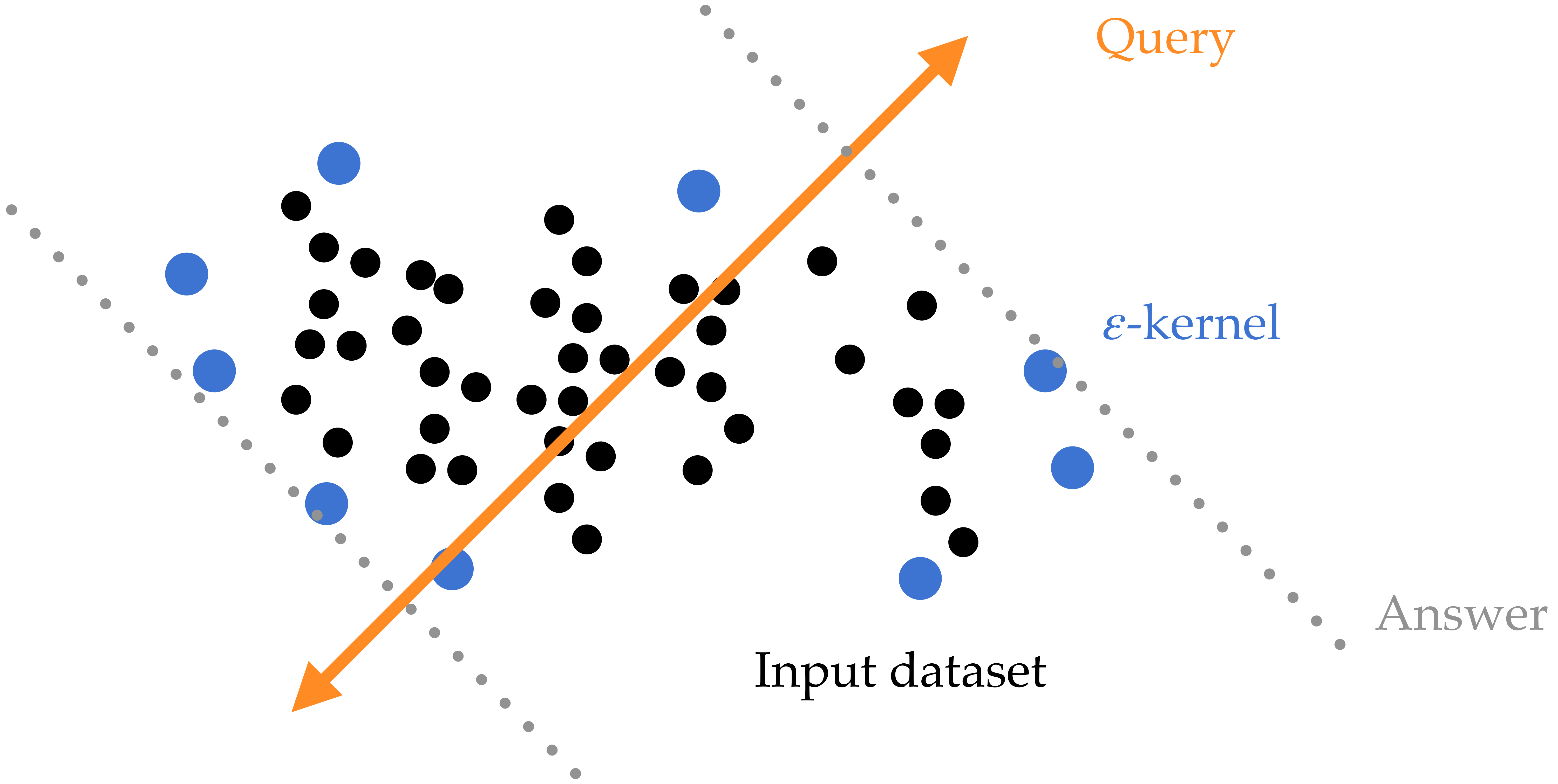
- Input: dataset with n points in d dimensions
- Question: how wide is my dataset in a given direction $x \in \mathbb{R}^d$?
- Goal: space-efficient data structure



Width Estimation of a Dataset

Algorithms: Low Dimensions

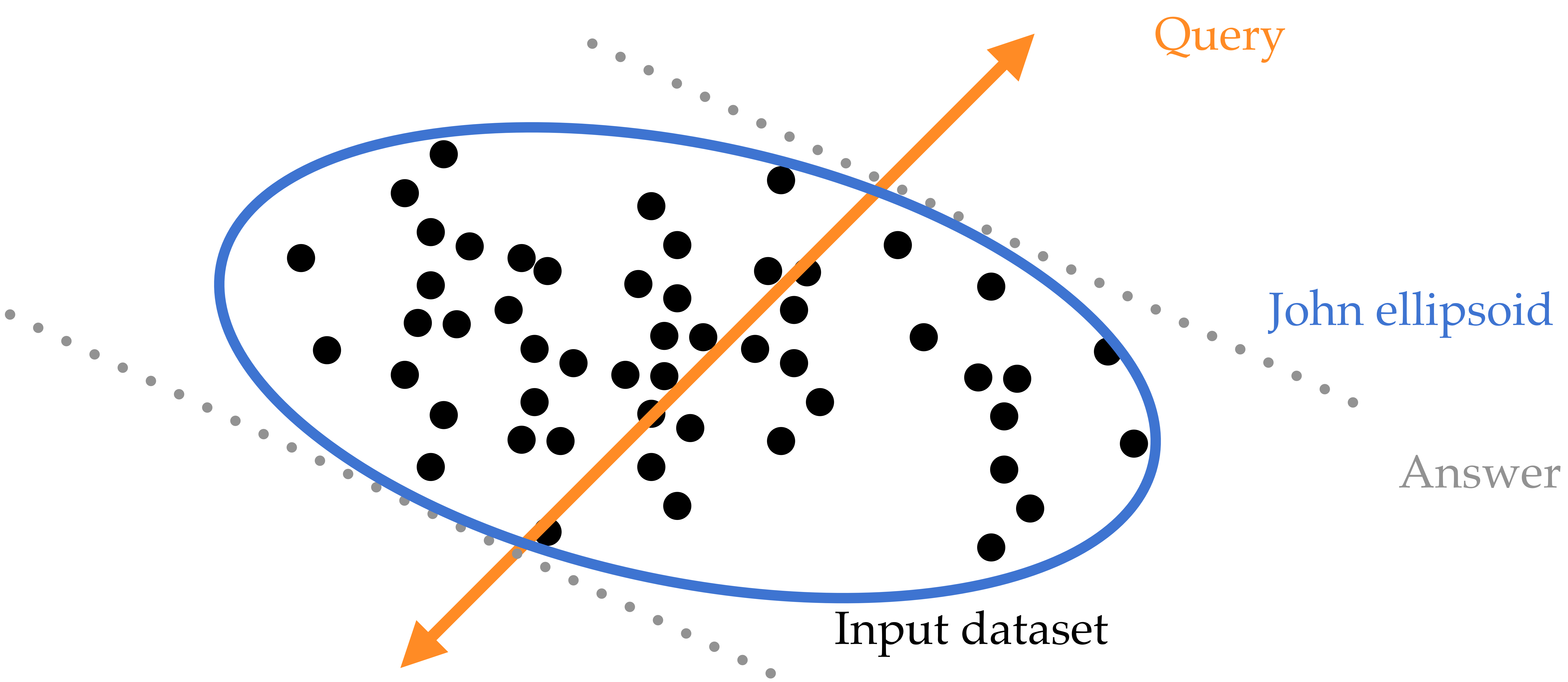
- In low dimensions: ε-kernels (Agarwal—Har-Peled—Varadarajan, 2004)
  - Subset of points approximating the width of every direction, up to $(1+\varepsilon)$ factor
  - Size: $1/\varepsilon^{\Theta(d)}$
    - Independent of n
    - **Exponential in d!**



Width Estimation of a Dataset

Algorithms: High Dimensions

- In high dimensions: **CANNOT** get better than $\sqrt{d}$ approximation in poly(d) space!
- Matching algorithm: John ellipsoids (minimum-volume enclosing ellipsoids)

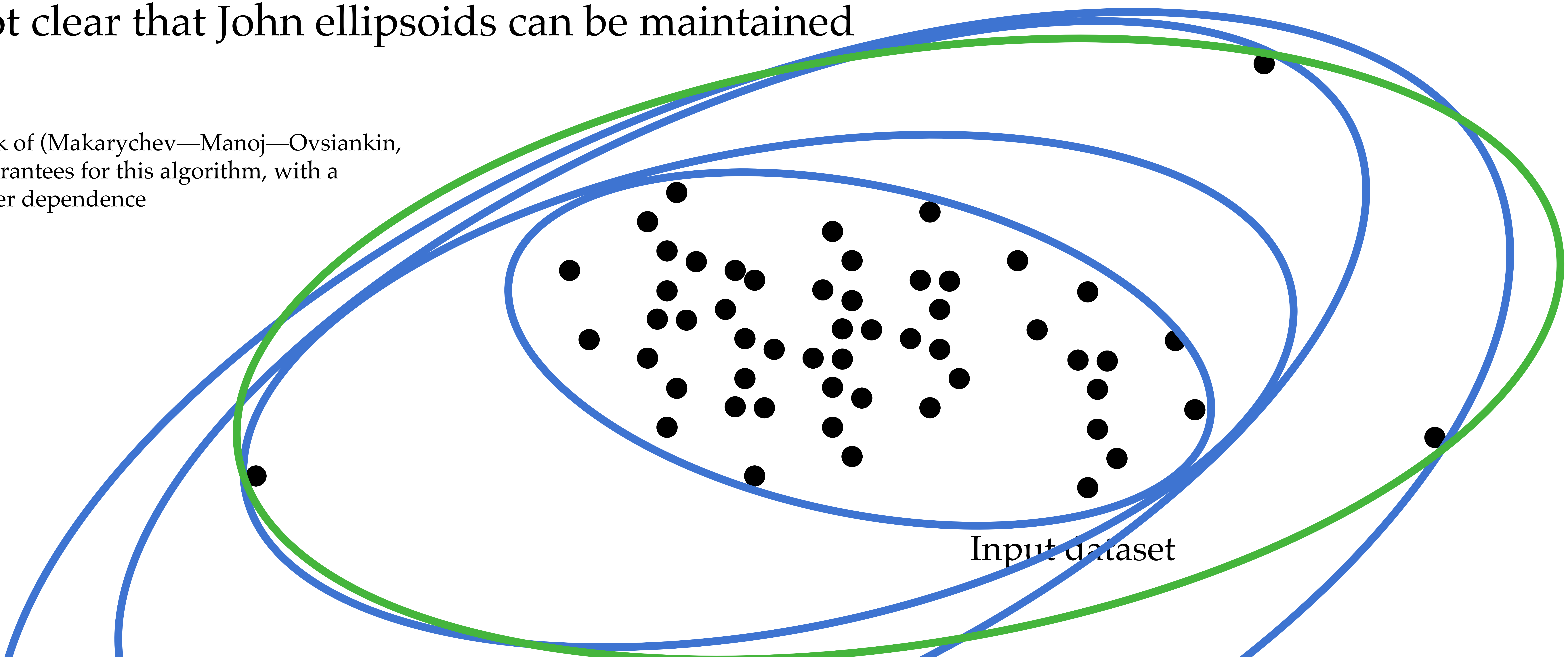


Width Estimation of a Dataset

Challenge: Streaming Algorithms!

- Streaming setting: must support insertion of new points
- Not clear that John ellipsoids can be maintained

Concurrent work of (Makarychev—Manoj—Ovsiankin, 2022) shows guarantees for this algorithm, with a condition number dependence



Width Estimation of a Dataset

Formal Statement

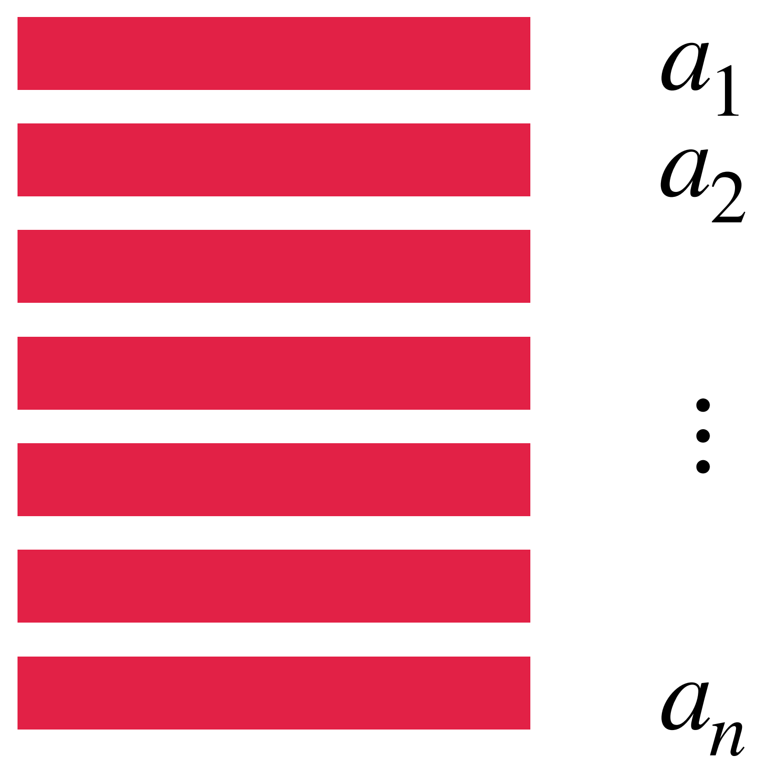
Streaming Width Estimation Problem.

- **Input**: stream $a_1, a_2, \ldots, a_n \in \mathbb{Z}^d$ with entries bounded by n^{100}
- **Output**: data structure $Q : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ s.t. for every $x \in \mathbb{R}^d$,

  $$\mathrm{width}(x) \leq Q(x) \leq \Delta \cdot \mathrm{width}(x), \qquad \mathrm{width}(x) := \max_{i=1}^{n} |\langle a_i, x \rangle| = \|Ax\|_\infty$$

- **Goal**: $\Delta = \mathrm{poly}(d, \log n)$, using $\mathrm{poly}(d, \log n)$ bits of space

$A =$



Theorem. There is a deterministic streaming algorithm which solves the **streaming width estimation problem** with $\Delta = O(\sqrt{d \log n})$ distortion, using $O(d^2 \log^2 n)$ bits of space.

Width Estimation of a Dataset

Applications

- First poly($d, \log n$) space and poly($d, \log n$) distortion streaming algorithms for…
  - Robust width estimation
  - Convex hull estimation
  - John ellipsoid estimation
  - ℓ_p subspace embeddings
  - Volume maximization
  - Minimum-width spherical shell
  - Linear programming
  - …

Proof Sketch

Proof Sketch

High Level Plan: Subset Selection

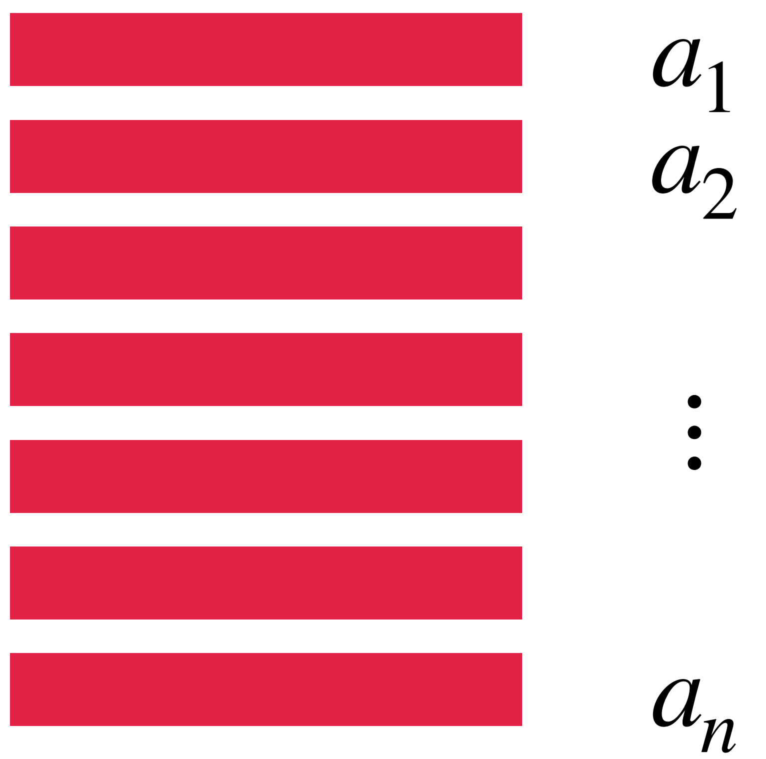
- Our approach: *select a subset of input points* $S \subseteq [n]$ s.t.

  $\|Ax\|_\infty \le \Delta \cdot \|A_S x\|_\infty$

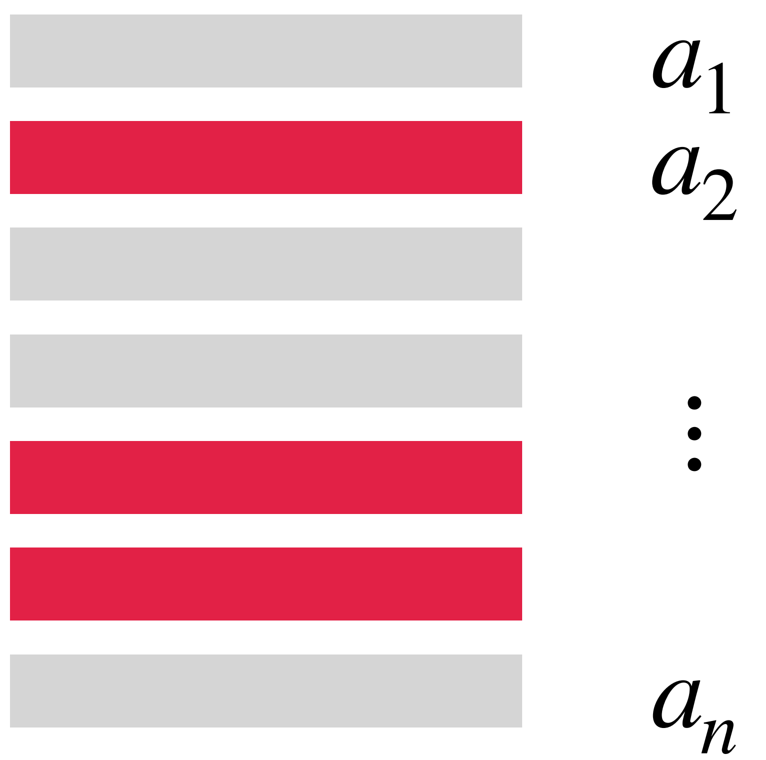
  "The width of A in the x direction is at most Δ times that of A_S"

  for every $x \in \mathbb{R}^d$
- This implies the result, since $\|A_S x\|_\infty \le \|Ax\|_\infty$ for every $x \in \mathbb{R}^d$
  - $Q(x) = \|A_S x\|_\infty$
- Question: *how do we know when to include a_i in our subset?*

$A =$



$A_S =$



Proof Sketch

First Attempt

- Update rule: $S' \leftarrow S \cup \{i\}$ if there exists $x \in \mathbb{R}^d$ s.t.

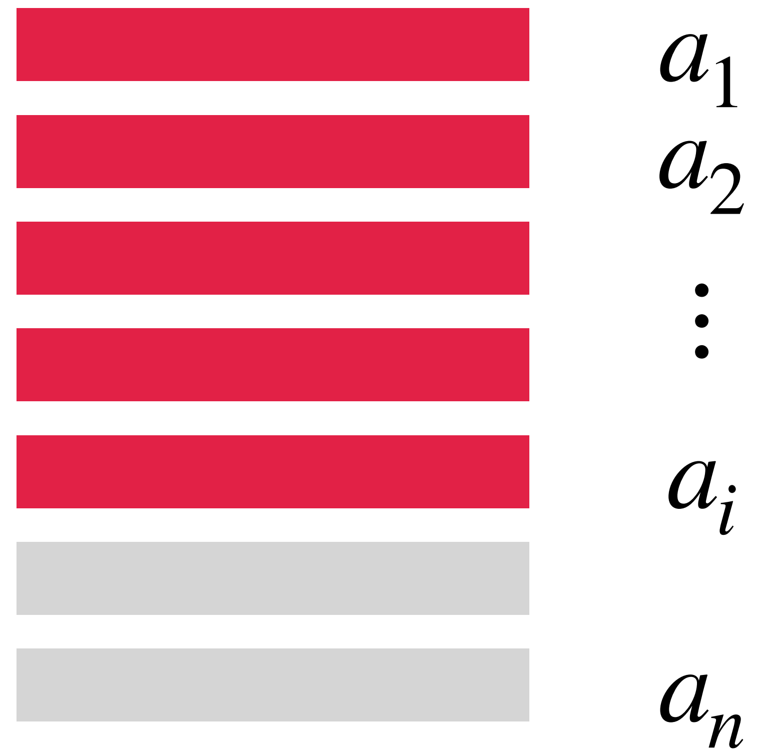
  $|\langle a_i, x\rangle| > \Delta \cdot \|A_S x\|_\infty$

  "Include i if S does not capture row i up to Δ factor"

- Correctness: by definition
- Space complexity: ?????
- Problem: ℓ_∞ has very little structure to work with…

Proof Sketch

$A_i =$



Secret Sauce: Online Leverage Scores

- Leverage scores: largest fraction of ℓ_2 norm occupied by ith row of A

$$\tau_i(A) = \sup_{x \in \mathbb{R}^d} \frac{|\langle a_i, x \rangle|^2}{\|Ax\|_2^2}$$

Key fact: $\sum_{i=1}^{n} \tau_i(A) = d$

- Online leverage scores (Cohen—Musco—Pachocki, 2016): ith leverage score of A_i

$$\tau_i^{\mathsf{OL}}(A) = \tau_i(A_i) = \sup_{x \in \mathbb{R}^d} \frac{|\langle a_i, x \rangle|^2}{\|A_i x\|_2^2}$$

Lemma [CMP16]. $\sum_{i=1}^{n} \tau_i^{\mathsf{OL}}(A) \leq O(d \log \kappa^{\mathsf{OL}})$

Lemma [WY22]. $\sum_{i=1}^{n} \tau_i^{\mathsf{OL}}(A) \leq O(d \log n)$

Proof Sketch

Revised Attempt

- Update rule: $S' \leftarrow S \cup \{i\}$ if there exists $x \in \mathbb{R}^d$ s.t.

$$\cancel{|\langle a_i, x\rangle| > \Delta \quad \|A_S x\|_\infty} \qquad |\langle a_i, x\rangle| > \|A_S x\|_2$$

- Observation: $\|A_S x\|_2 \le \sqrt{|S|}\,\|A_S x\|_\infty$
  - If $|S| = \mathrm{poly}(d, \log n)$, then we can replace $\|A_S x\|_\infty$ by $\|A_S x\|_2$!
- Then,

$$1 \le \frac{\overbrace{|\langle a_i, x\rangle|^2 + |\langle a_i, x\rangle|^2}^{2 \cdot |\langle a_i, x\rangle|^2}}{\underbrace{\|A_S x\|_2^2 + |\langle a_i, x\rangle|^2}_{\|A_{S'} x\|_2^2}} \le 2 \cdot \sup_{x \in \mathbb{R}^d} \frac{\langle a_i, x\rangle^2}{\|A_{S'} x\|_2^2} \implies \frac{1}{2} \le \tau_i^{\mathsf{OL}}(A_{S'})$$

This is the ith **online leverage score of $A_{S'}$**!

Proof Sketch

Revised Attempt

- We have shown:
  - Every row in A_S has online leverage score at least $1/2$
  - The online leverage scores of A_S must sum to at most $O(d \log n)$
- $\implies |S| = O(d \log n)$
- $\implies \Delta \leq \sqrt{|S|} = O(\sqrt{d \log n})$

Lemma [WY22]. $\sum_{i=1}^{n} \tau_i^{\mathsf{OL}}(A) \leq O(d \log n)$

Theorem. There is a deterministic streaming algorithm which solves the streaming width estimation problem with $\Delta = O(\sqrt{d \log n})$, using $O(d^2 \log^2 n)$ bits of space.

Conclusion

- We obtain the first polynomial space algorithm for maintaining a width estimation data structure in a stream
- As a corollary, we obtain the first polynomial space algorithm for a variety of problems in streaming computational geometry
- Our techniques draw a novel connection between online numerical linear algebra and computational geometry, which may be of independent interest

arXiv link