Streaming Facility Location in High Dimension via Geometric Hashing

Pavel Veselý (Charles University, Prague)

Artur Czumaj
(Warwick)

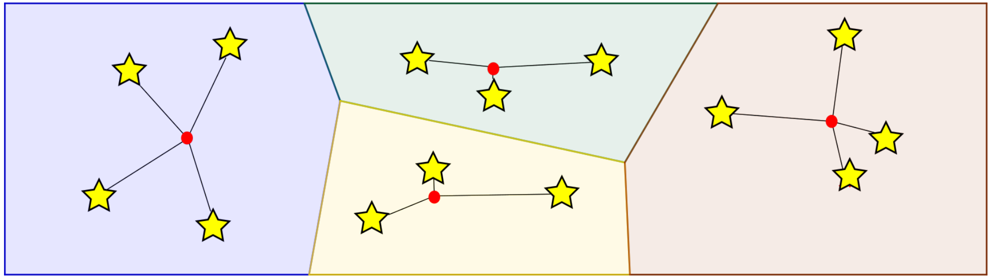
Shaofeng Jiang
(Peking)

Robert Krauthgamer
(Weizmann)

Mingwei Yang
(Peking)

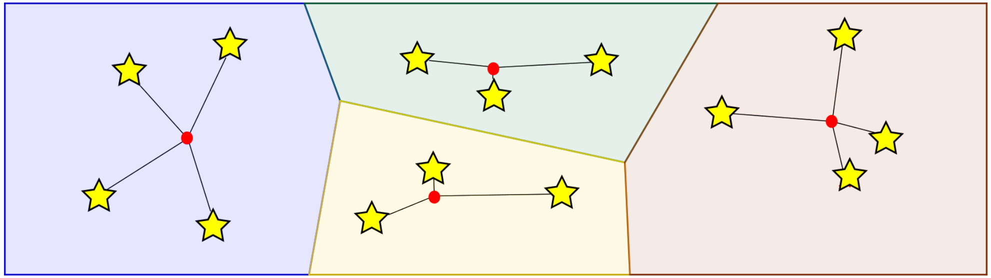
Geometric streams

- Input: **sequence of points**☆ from $\mathbb{R}^d$
- Processed in a few passes **using small memory**
- Goal: **estimate** a statistic of the point set
  - e.g. diameter, **cost** of clustering, MST, matching, ...
  - solution can take space $\Omega(n)$



Geometric streams

- Input: **sequence of points** ☆ from $\mathbb{R}^d$
- Processed in a few passes **using small memory**
- Goal: **estimate** a statistic of the point set
  - e.g. diameter, **cost** of clustering, MST, matching, ...
  - solution can take space $\Omega(n)$

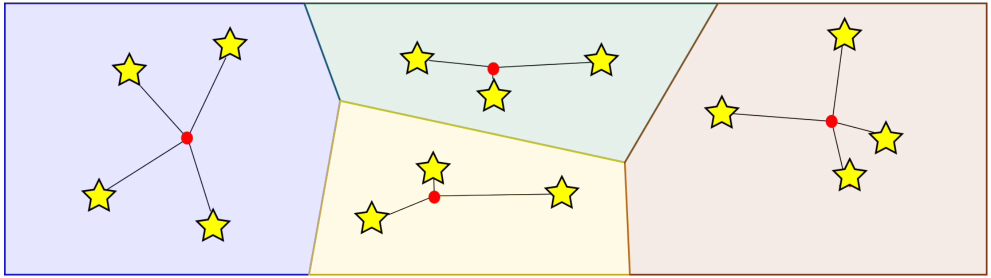


Dynamic geometric streams: classical model [Indyk STOC '04]

- insertions & deletions
- points from $[\Delta]^d$ for integer $\Delta > 0$
- space ideally poly$(d \cdot \log \Delta)$
  - will ignore poly$(\log(\Delta + n))$ factors in space

Geometric streams

- Input: **sequence of points** ☆ from $\mathbb{R}^d$
- Processed in a few passes **using small memory**
- Goal: **estimate** a statistic of the point set
  - e.g. diameter, **cost** of clustering, MST, matching, ...
  - solution can take space $\Omega(n)$



Dynamic geometric streams: classical model [Indyk STOC '04]

- insertions & deletions
- points from $[\Delta]^d$ for integer $\Delta > 0$
- space ideally $\text{poly}(d \cdot \log \Delta)$
  - will ignore $\text{poly}(\log(\Delta + n))$ factors in space

Often: "Algo. for insertion-only $\Rightarrow$ Algo. for dynamic geometric streams"

"Counterexample": diameter with $\text{poly}(d)$ space [Indyk'03], [Agarwal,Sharathkumar'15]

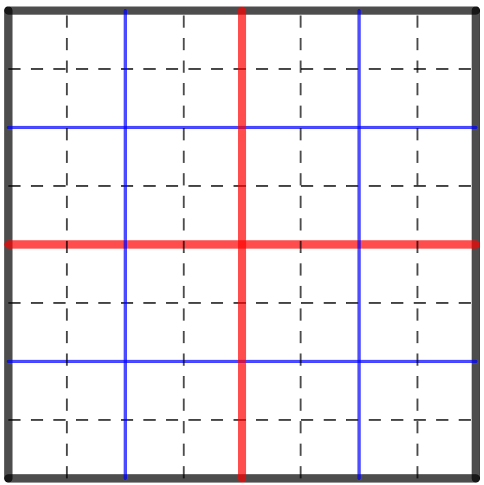
Geometric streams: Main dichotomy

Low Dimension: space $\exp(d)$ | **High Dimension: space $\text{poly}(d)$**

Geometric streams: Main dichotomy

Low Dimension: space $\exp(d)$

- $O(1)$ or even $(1+\varepsilon)$-approximation e.g. for:
  - MST, TSP, and Steiner tree [Frahling,Indyk,Sohler '05]
  - k-median, k-means, Max-Cut, ... [Frahling&Sohler '05]
  - Facility Location [Czumaj et al. '13]
  - Steiner Forest [Czumaj,Jiang,Krauthgamer,**V.** '22]
- typical space decompositions: grids/quadtree

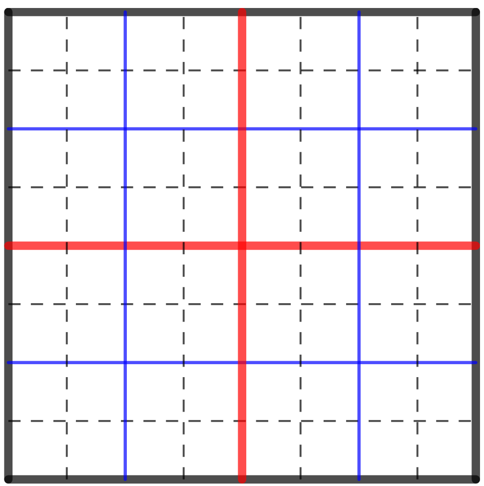


High Dimension: space $\mathrm{poly}(d)$

Geometric streams: Main dichotomy

Low Dimension: space $\exp(d)$

- $O(1)$ or even $(1+\varepsilon)$-approximation e.g. for:
  - MST, TSP, and Steiner tree [Frahling,Indyk,Sohler '05]
  - k-median, k-means, Max-Cut, ... [Frahling&Sohler '05]
  - Facility Location [Czumaj et al. '13]
  - Steiner Forest [Czumaj,Jiang,Krauthgamer,**V.** '22]
- typical space decompositions: grids/quadtree



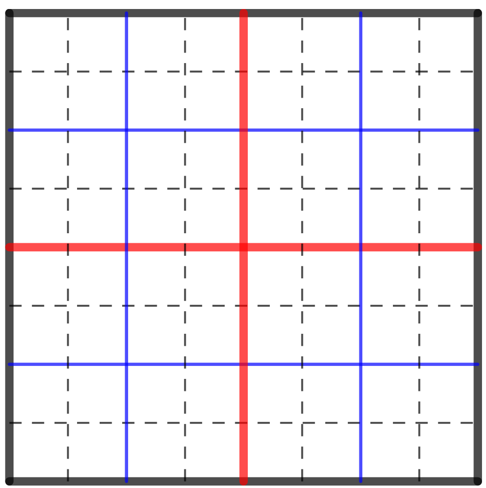
High Dimension: space $\text{poly}(d)$

- Important case: $d = \Theta(\log n)$ (JL lemma)
- only $O(\log n)$-approximation (or worse)
  - ratio $O(d \cdot \log \Delta)$ by tree embeddings [Indyk '04]
  - ratio $O(\log n)$ for MST and EMD [Chen, Jayaram, Levi, Waingarten '22]
- lack of techniques for $O(1)$-approx.
  - tree embedding distorts distances by $\Omega(\log n)$

Geometric streams: Main dichotomy

Low Dimension: space exp (d)

- $O(1)$ or even $(1+\varepsilon)$-approximation e.g. for:
  - MST, TSP, and Steiner tree [Frahling,Indyk,Sohler '05]
  - k-median, k-means, Max-Cut, ... [Frahling&Sohler '05]
  - Facility Location [Czumaj et al. '13]
  - Steiner Forest [Czumaj,Jiang,Krauthgamer,**V.** '22]
- typical space decompositions: grids/quadtree



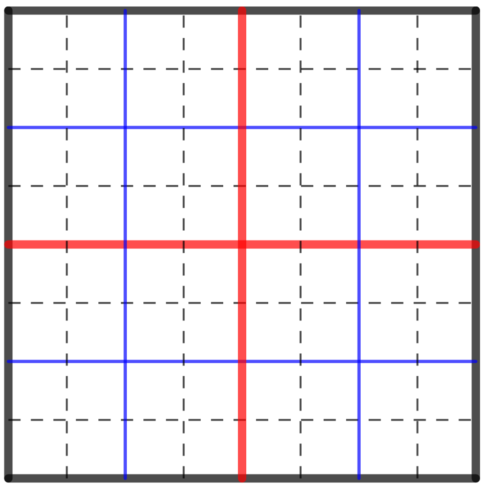
High Dimension: space poly (d)

- Important case: $d = \Theta(\log n)$ (JL lemma)
- only $O(\log n)$-approximation (or worse)
  - ratio $O(d \cdot \log \Delta)$ by tree embeddings [Indyk '04]
  - ratio $O(\log n)$ for MST and EMD [Chen, Jayaram, Levi, Waingarten '22]
- lack of techniques for $O(1)$-approx.
  - tree embedding distorts distances by $\Omega(\log n)$
- exception: ratio $(1+\varepsilon)$ for k-median and k-means
  - low space only for small k
  - [Braverman,Frahling,Lang,Sohler,Yang '17], [Song,Yang,Zhong '18]

Geometric streams: Main dichotomy

Low Dimension: space $\exp(d)$

- $O(1)$ or even $(1+\varepsilon)$-approximation e.g. for:
  - MST, TSP, and Steiner tree [Frahling,Indyk,Sohler '05]
  - k-median, k-means, Max-Cut, . . . [Frahling&Sohler '05]
  - Facility Location [Czumaj et al. '13]
  - Steiner Forest [Czumaj,Jiang,Krauthgamer,**V.** '22]
- typical space decompositions: grids/quadtree



High Dimension: space $\text{poly}(d)$

- Important case: $d = \Theta(\log n)$ (JL lemma)
- only $O(\log n)$-approximation (or worse)
  - ratio $O(d \cdot \log \Delta)$ by tree embeddings [Indyk '04]
  - ratio $O(\log n)$ for MST and EMD [Chen, Jayaram, Levi, Waingarten '22]
- lack of techniques for $O(1)$-approx.
  - tree embedding distorts distances by $\Omega(\log n)$
- exception: ratio $(1+\varepsilon)$ for k-median and k-means
  - low space only for small k
  - [Braverman,Frahling,Lang,Sohler,Yang '17], [Song,Yang,Zhong '18]
- Insertion-only setting:
  - Diameter et al.: ratio $O(1)$ [Agarwal,Sharathkumar'15]
  - Width in any direction [Woodruff,Yasuda'22]

Euclidean Uniform Facility Location

Input: pointset $X \subset \mathbb{R}^d$, opening cost $\mathfrak{f} > 0$

Goal: open a set of facilities F to minimize

$$\mathrm{cost}(X, F) := \underbrace{\sum_{p \in X} \mathrm{dist}(p, F)}_{\text{connection cost}} + \underbrace{\mathfrak{f} \cdot |F|}_{\text{opening cost}}$$

$$\mathrm{dist}(p, q) := \|p - q\|_2 \quad \text{and} \quad \mathrm{dist}(p, F) := \min_{q \in F} \mathrm{dist}(p, q)$$

Image credits: NASA Hubble, CC BY 2.0, via Wikimedia Commons

Euclidean Uniform Facility Location

Input: pointset $X \subset \mathbb{R}^d$, opening cost $\mathfrak{f} > 0$

Goal: open a set of facilities F to minimize

$$\mathrm{cost}(X, F) := \underbrace{\sum_{p \in X} \mathrm{dist}(p, F)}_{\text{connection cost}} + \underbrace{\mathfrak{f} \cdot |F|}_{\text{opening cost}}$$

$$\mathrm{dist}(p, q) := \|p - q\|_2 \quad \text{and} \quad \mathrm{dist}(p, F) := \min_{q \in F} \mathrm{dist}(p, q)$$

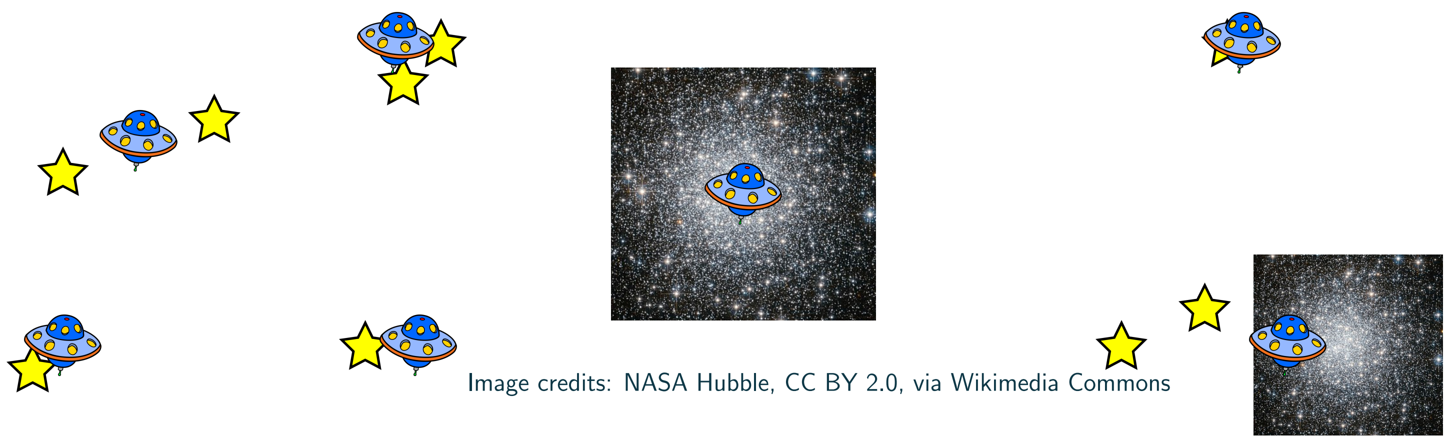


Image credits: NASA Hubble, CC BY 2.0, via Wikimedia Commons

Euclidean Uniform Facility Location

Input: pointset $X \subset \mathbb{R}^d$, opening cost $\mathfrak{f} > 0$

Goal: open a set of facilities F to minimize

$$\mathrm{cost}(X, F) := \underbrace{\sum_{p \in X} \mathrm{dist}(p, F)}_{\text{connection cost}} + \underbrace{\mathfrak{f} \cdot |F|}_{\text{opening cost}}$$

$$\mathrm{dist}(p, q) := \|p - q\|_2 \quad \text{and} \quad \mathrm{dist}(p, F) := \min_{q \in F} \mathrm{dist}(p, q)$$

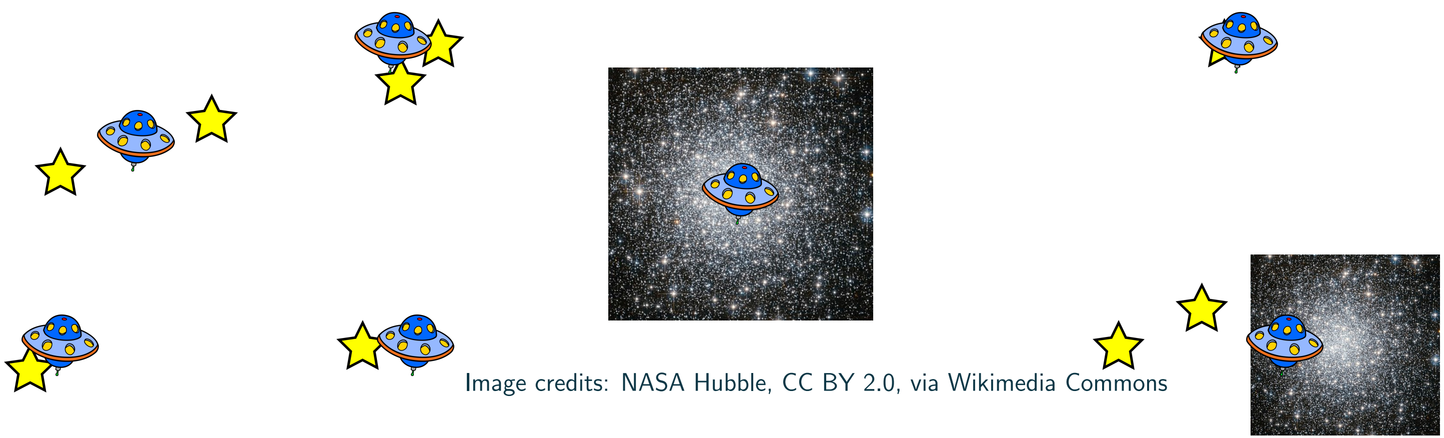


Image credits: NASA Hubble, CC BY 2.0, via Wikimedia Commons

This talk: unit facility cost $\mathfrak{f} = 1$

Facility Location in Geometric Streams

	# of passes	ratio	space	reference & notes
Previous work:	1	$O(d \cdot \log^2 \Delta)$	poly(d)	[Indyk '04]
	1	exp(d)	exp(d)	[Lammersen,Sohler '08]
	1	$1 + \varepsilon$	exp(d)	only for $d = 2$ [Czumaj,Lammersen,Monemizadeh,Sohler '13]

Facility Location in Geometric Streams

	# of passes	ratio	space	reference & notes
Previous work:	1	$O(d \cdot \log^2 \Delta)$	poly(d)	[Indyk '04]
	1	exp(d)	exp(d)	[Lammersen,Sohler '08]
	1	$1 + \varepsilon$	exp(d)	only for $d = 2$ [Czumaj,Lammersen,Monemizadeh,Sohler '13]
We wanted:	1	$O(1)$	poly(d)	

Facility Location in Geometric Streams

	# of passes	ratio	space	reference & notes
Previous work:	1	$O(d \cdot \log^2 \Delta)$	$\text{poly}(d)$	[Indyk '04]
	1	$\exp(d)$	$\exp(d)$	[Lammersen,Sohler '08]
	1	$1 + \varepsilon$	$\exp(d)$	only for $d = 2$ [Czumaj,Lammersen,Monemizadeh,Sohler '13]
We wanted:	1	$O(1)$	$\text{poly}(d)$	(still conjectured. . .)
We got:	**2**	$O(1)$	$\text{poly}(d)$	$*$
	$\mathbf{1}^\dagger$	$O(1)$	$\text{poly}(d)$	$*$; $\mathbf{1}^\dagger$ **= random-order streams**

Facility Location in Geometric Streams

	# of passes	ratio	space	reference & notes
Previous work:	1	$O(d \cdot \log^2 \Delta)$	poly(d)	[Indyk '04]
	1	exp(d)	exp(d)	[Lammersen,Sohler '08]
	1	$1+\varepsilon$	exp(d)	only for $d = 2$ [Czumaj,Lammersen,Monemizadeh,Sohler '13]
We wanted:	1	$O(1)$	poly(d)	(still conjectured. . .)
We got:	**2**	$O(1)$	poly(d)	$*$
	$\mathbf{1^\dagger}$	$O(1)$	poly(d)	$*$; $\mathbf{1^\dagger}$ **= random-order streams**
	1	$O(d^{1.5})$	poly(d)	$*$

- for $d = \Theta(\log n)$ (from JL lemma): improvement from ratio $\Theta(\log^3 n)$ [Indyk '04] to $\Theta(\log^{1.5} n)$

Facility Location in Geometric Streams

	# of passes	ratio	space	reference & notes
Previous work:	1	$O(d \cdot \log^2 \Delta)$	poly(d)	[Indyk '04]
	1	exp(d)	exp(d)	[Lammersen,Sohler '08]
	1	$1+\varepsilon$	exp(d)	only for $d = 2$ [Czumaj,Lammersen,Monemizadeh,Sohler '13]
We wanted:	1	$O(1)$	poly(d)	(still conjectured. . .)
We got:	**2**	$O(1)$	poly(d)	*
	1†	$O(1)$	poly(d)	*; **1**† **= random-order streams**
	1	$O(d^{1.5})$	poly(d)	*
Lower bound:	1	< 1.085	$\Omega\left(2^{\text{poly}(d)}\right)$	* follows from Boolean Hidden Matching

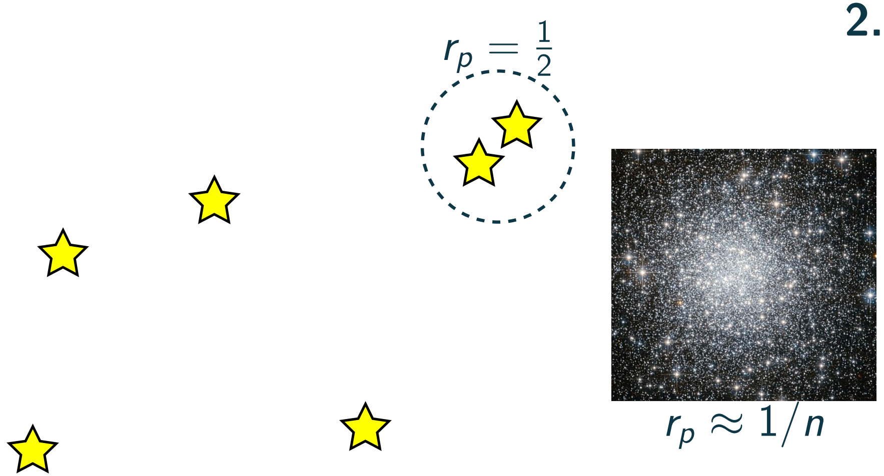
- for $d = \Theta(\log n)$ (from JL lemma): improvement from ratio $\Theta(\log^3 n)$ [Indyk '04] to $\Theta(\log^{1.5} n)$

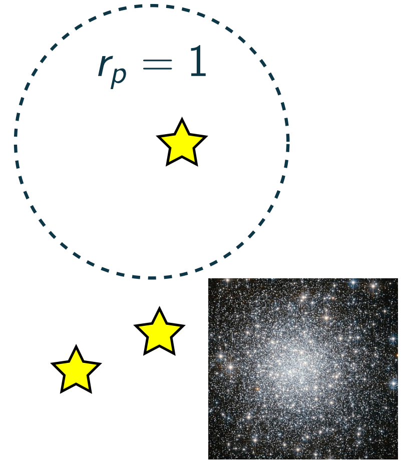
Estimator from Mettu-Plaxton algorithm [Mettu, Plaxton '03], [Bădoiu, Czumaj, Indyk, and Sohler '05]

For every point p, we define $\frac{1}{n} \leq r_p \leq 1$ such that:

1. $\sum_p r_p = \Theta(\mathrm{OPT})$

2. $|X \cap B(p, r_p)| \approx 1/r_p$ (X = input point set)





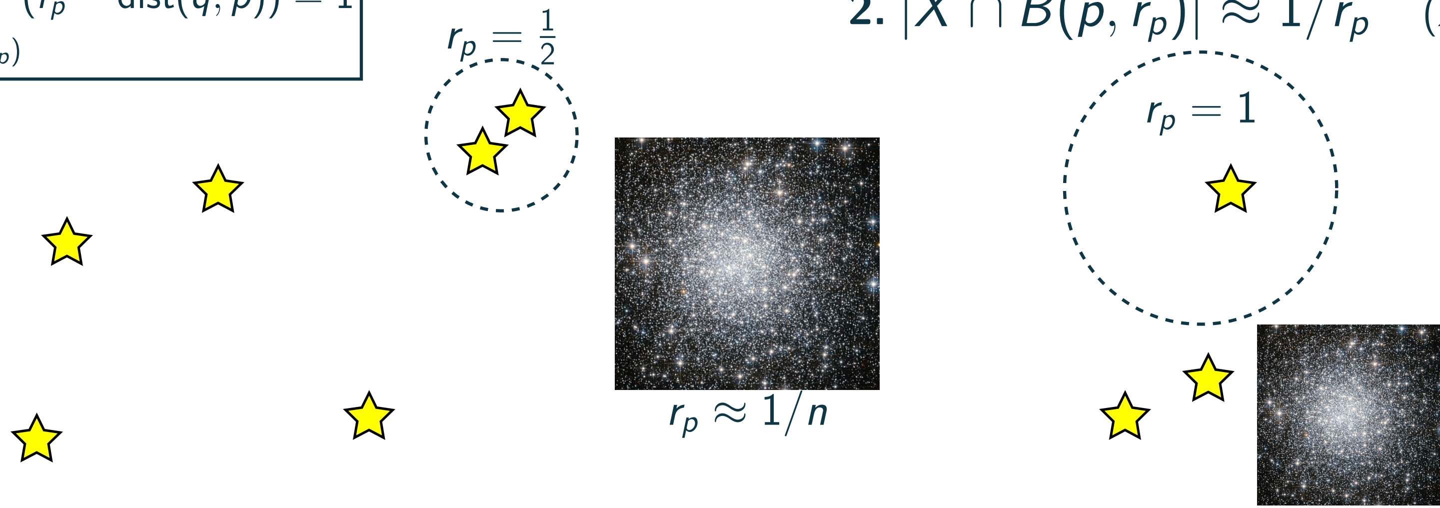
Estimator from Mettu-Plaxton algorithm [Mettu, Plaxton '03], [Bădoiu, Czumaj, Indyk, and Sohler '05]

For every point p, we define $\frac{1}{n} \leq r_p \leq 1$ such that:

$$\sum_{q \in X \cap B(p, r_p)} (r_p - \mathrm{dist}(q, p)) = 1$$

1. $\sum_p r_p = \Theta(\mathrm{OPT})$

2. $|X \cap B(p, r_p)| \approx 1/r_p \quad (X = \text{input point set})$



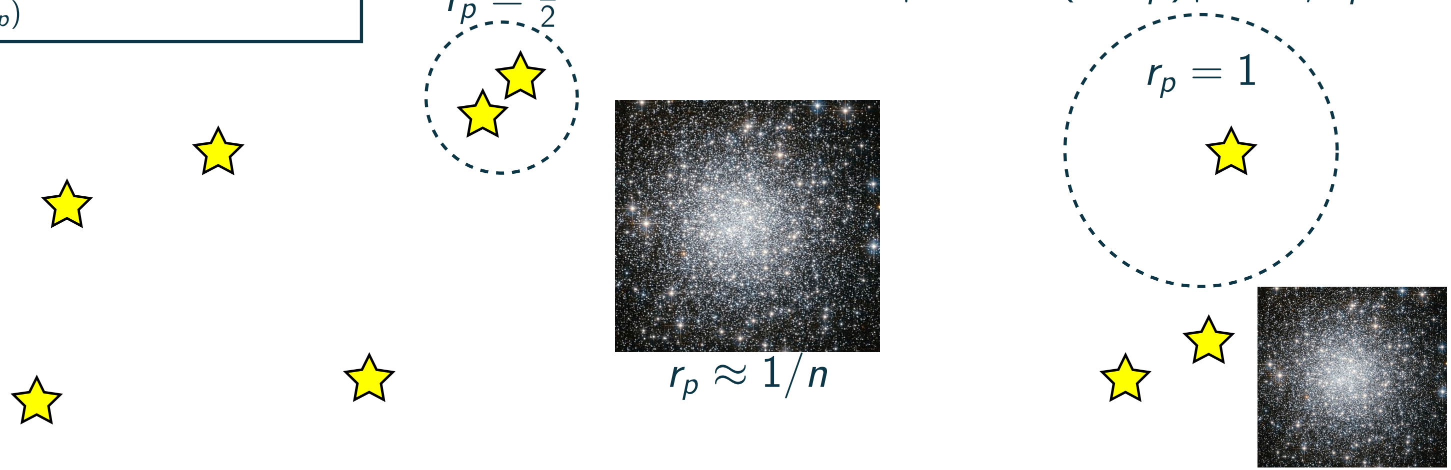
Estimator from Mettu-Plaxton algorithm [Mettu, Plaxton '03], [Bădoiu, Czumaj, Indyk, and Sohler '05]

For every point p, we define $\frac{1}{n} \leq r_p \leq 1$ such that:

$$\sum_{q \in X \cap B(p, r_p)} (r_p - \text{dist}(q, p)) = 1$$

1. $\sum_p r_p = \Theta(\text{OPT})$

2. $|X \cap B(p, r_p)| \approx 1/r_p$ $\quad (X = \text{input point set})$



- property 2 $\Rightarrow$ streaming $O(1)$-approximation of r_p for p **given in advance**
- **but** $\Omega(n)$ space needed for any finite approx. when p **given as a query** (from INDEX)

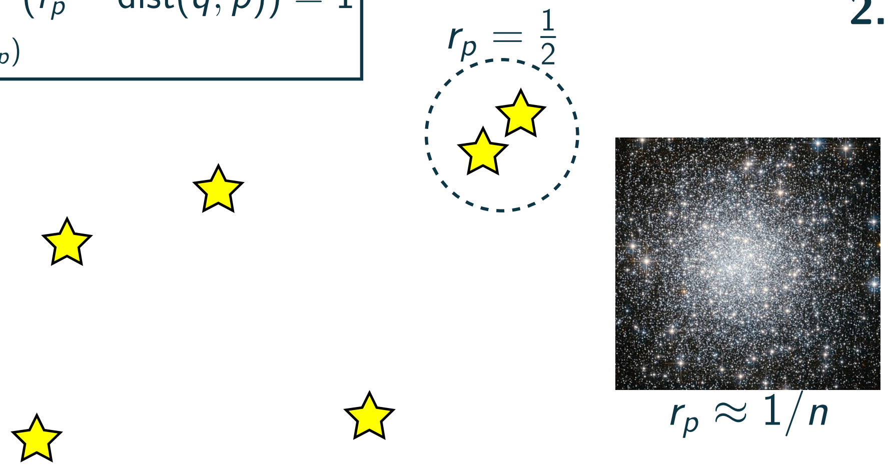
Estimator from Mettu-Plaxton algorithm [Mettu, Plaxton '03], [Bădoiu, Czumaj, Indyk, and Sohler '05]

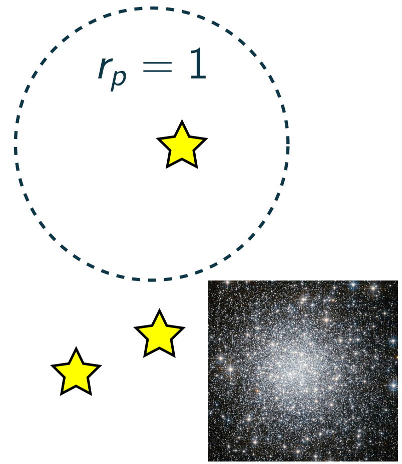
For every point p, we define $\frac{1}{n} \le r_p \le 1$ such that:

$$\sum_{q \in X \cap B(p, r_p)} (r_p - \mathrm{dist}(q,p)) = 1$$

1. $\sum_p r_p = \Theta(\mathrm{OPT})$

2. $|X \cap B(p, r_p)| \approx 1/r_p \quad (X = \text{input point set})$





- property 2 $\Rightarrow$ streaming $O(1)$-approximation of r_p for p **given in advance**
- **but** $\Omega(n)$ space needed for any finite approx. when p **given as a query** (from INDEX)

$\Rightarrow$ Naïve two-pass algo. for Facility Location

- 1st pass: sample a few points **uniformly**
- 2nd pass: estimate r_p's for sampled points

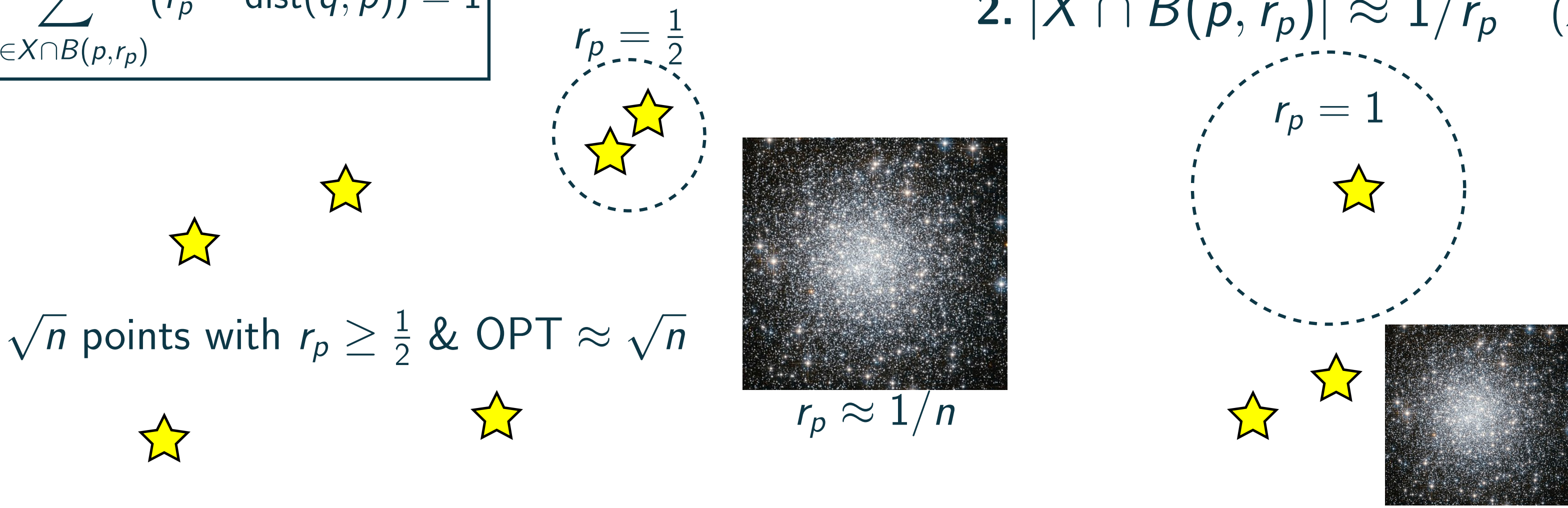
Estimator from Mettu-Plaxton algorithm [Mettu, Plaxton '03], [Bădoiu, Czumaj, Indyk, and Sohler '05]

For every point p, we define $\frac{1}{n} \le r_p \le 1$ such that:

$$\sum_{q \in X \cap B(p, r_p)} (r_p - \text{dist}(q, p)) = 1$$

1. $\sum_p r_p = \Theta(\text{OPT})$

2. $|X \cap B(p, r_p)| \approx 1/r_p$ (X = input point set)



- property 2 $\Rightarrow$ streaming $O(1)$-approximation of r_p for p **given in advance**
- **but $\Omega(n)$** space needed for any finite approx. when p **given as a query** (from INDEX)

$\Rightarrow$ Naïve two-pass algo. for Facility Location

- 1st pass: sample a few points **uniformly**
- 2nd pass: estimate r_p's for sampled points

Uniform sampling has too large variance ☹

Geometric Importance Sampling

Goal: sample proportionally to r_p in one pass $\Rightarrow$ $O(1)$-approximation in two passes

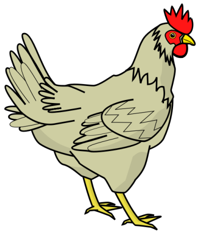
- for $p^* =$ sampled point, $r_{p^*} / \Pr[p^*]$ unbiased estimator of $\sum_p r_p = \Theta(\mathrm{OPT})$

Geometric Importance Sampling

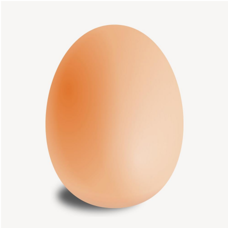
Goal: sample proportionally to r_p in one pass $\Rightarrow$ $O(1)$-approximation in two passes

- for $p^* =$ sampled point, $r_{p^*} / \Pr[p^*]$ unbiased estimator of $\sum_p r_p = \Theta(\mathrm{OPT})$

Want to sample w/ prob. $\sim r_p$ but cannot estimate r_p for queried p in one pass



vs.

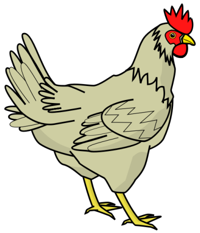


Geometric Importance Sampling

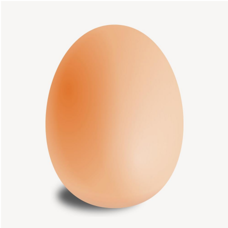
Goal: sample proportionally to r_p in one pass $\Rightarrow O(1)$-approximation in two passes

- for $p^* =$ sampled point, $r_{p^*} / \Pr[p^*]$ unbiased estimator of $\sum_p r_p = \Theta(\mathrm{OPT})$

Want to sample w/ prob. $\sim r_p$ but cannot estimate r_p for queried p in one pass



vs.



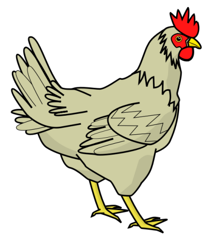
$\Rightarrow$ need to sample w.r.t. geometry

Geometric Importance Sampling

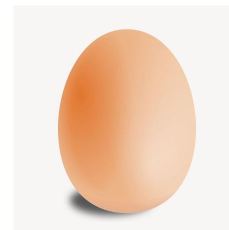
Goal: sample proportionally to r_p in one pass $\Rightarrow O(1)$-approximation in two passes

- for p^* = sampled point, $r_{p^*} / \Pr[p^*]$ unbiased estimator of $\sum_p r_p = \Theta(\mathrm{OPT})$

Want to sample w/ prob. $\sim r_p$ but cannot estimate r_p for queried p in one pass



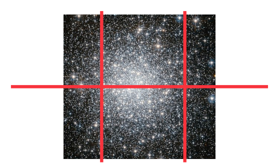
vs.



$\Rightarrow$ need to sample w.r.t. geometry

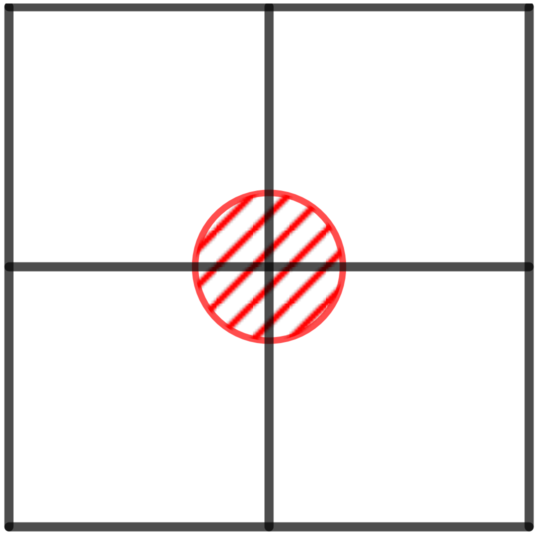
Goal: map/hash $\varphi : \mathbb{R}^d \to \mathbb{R}^d$, then sample uniformly from the support of $\varphi(X)$

- $\varphi^{-1}(p)$ = bucket of points p
- desired properties:
  - "large" r_p (say $r_p \approx 1$) $\Rightarrow$ few points in the bucket of p
  - dense clusters with points of "small" r_p (say $r_p = o(1)$) mapped to few buckets



Cosistent Geometric Hashing

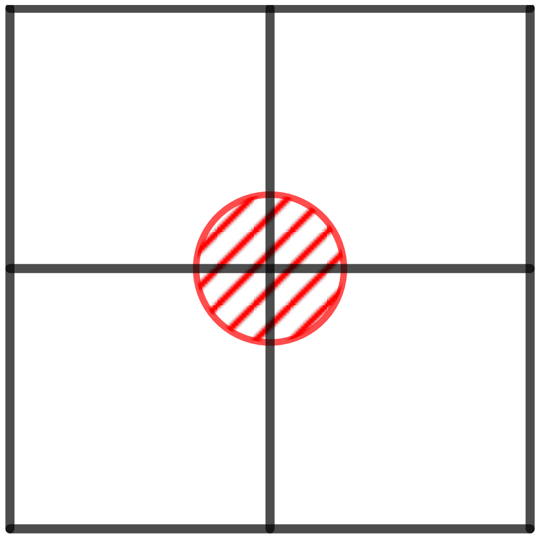
Grids/quadtrees not good:



- cluster intersects 2^d buckets

Cosistent Geometric Hashing

Grids/quadtrees not good:



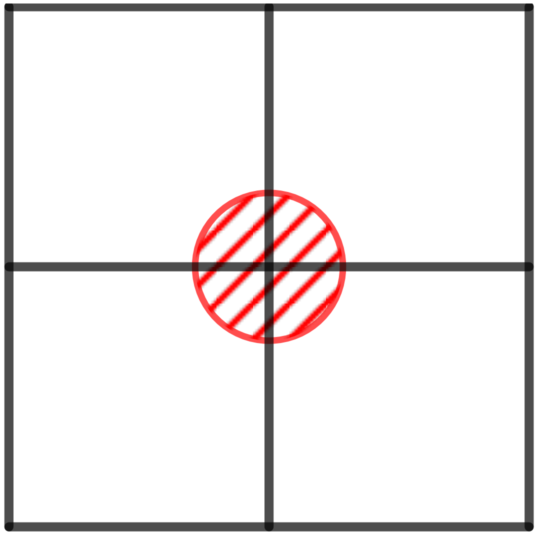
- cluster intersects 2^d buckets

Goal: space decomposition such that:

1. bounded diameter buckets
2. **ball of small-enough diameter intersects poly (d) buckets**

Cosistent Geometric Hashing

Grids/quadtrees not good:



- cluster intersects 2^d buckets

Goal: space decomposition such that:

1. bounded diameter buckets
2. **ball of small-enough diameter intersects poly(d) buckets**

Def.: $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ is Γ-gap Λ-consistent hash if

1. Bounded diameter: every bucket $\varphi^{-1}(y)$ has diameter ≤ 1
2. **Consistency**: $\forall S \subseteq \mathbb{R}^d$ with $\text{Diam}(S) \leq \mathbf{1/\Gamma}$: $\quad |\varphi(S)| \leq \mathbf{\Lambda}$

- need $\Gamma, \Lambda = \text{poly}(d)$
  - Γ determines the approx. ratio of our 1-pass algo.

$\sim$ sparse partitions from [Jia-Lin-Noubir-Rajaraman-Sundaram'05], [Filtser'20]

- we require computing $\varphi(p)$ in poly(d) time & space
- we need data-oblivious φ

Construction of Consistent Geometric Hashing

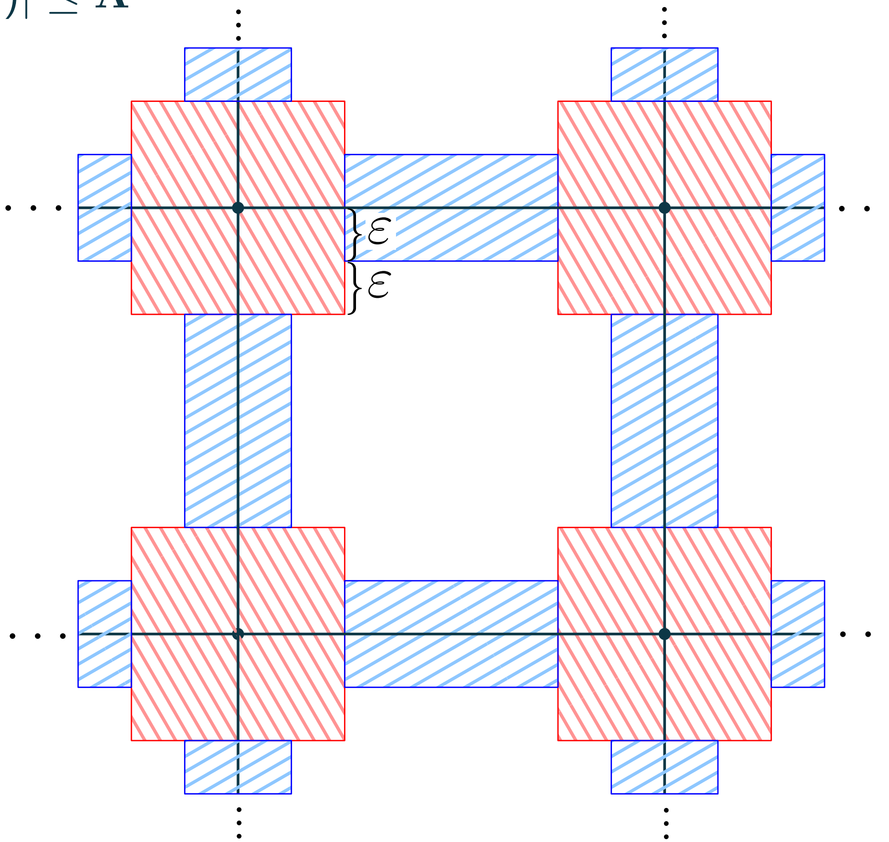
Def.: $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ is $\boldsymbol{\Gamma}$-gap $\boldsymbol{\Lambda}$-consistent hash if

1. Bounded diameter: every bucket $\varphi^{-1}(y)$ has diameter ≤ 1

2. Consistency: $\forall S \subseteq \mathbb{R}^d$ with $\text{Diam}(S) \leq \mathbf{1}/\boldsymbol{\Gamma}$: $|\varphi(S)| \leq \boldsymbol{\Lambda}$

We get $\boldsymbol{\Gamma} = O(\boldsymbol{d}^{1.5})$ and $\Lambda = d + 1$

- Start with grid & remove ℓ_∞ neighborhoods of faces



Algorithmic Framework Overview

Recall: $\sum_p r_p = \Theta(\mathrm{OPT})$

We focus on estimating # of points with $r_p \geq 1/2$

- Estimating # of points with $r_p \geq 1/2^i$ similar using subsampling

Two-pass algo:

- Hash points using consistent φ
- Sample a non-empty bucket b uniformly & a point from $\varphi^{-1}(b)$
  - using two-level ℓ_0 samplers
- 2nd pass: estimate r_p for each sampled point

Algorithmic Framework Overview

Recall: $\sum_p r_p = \Theta(\mathrm{OPT})$

We focus on estimating # of points with $r_p \geq 1/2$

- Estimating # of points with $r_p \geq 1/2^i$ similar using subsampling

Two-pass algo:

- Hash points using consistent φ
- Sample a non-empty bucket b uniformly & a point from $\varphi^{-1}(b)$
  - using two-level ℓ_0 samplers
- 2nd pass: estimate r_p for each sampled point

Bottom line: sampling p with probability $\geq \dfrac{r_p}{\mathrm{poly}(d \cdot \log \Delta)}$

Algorithmic Framework Overview

Recall: $\sum_p r_p = \Theta(\mathrm{OPT})$

We focus on estimating # of points with $r_p \geq 1/2$

- Estimating # of points with $r_p \geq 1/2^i$ similar using subsampling

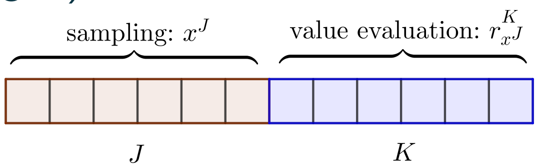
Two-pass algo:

- Hash points using consistent φ
- Sample a non-empty bucket b uniformly & a point from $\varphi^{-1}(b)$
  - using two-level ℓ_0 samplers
- 2nd pass: estimate r_p for each sampled point

Bottom line: sampling p with probability $\geq \frac{r_p}{\mathrm{poly}(d \cdot \log \Delta)}$

Random-order streams:

- 1st half of stream for sampling
- 2nd half for estimating r_p's of sampled points



Algorithmic Framework Overview

Recall: $\sum_p r_p = \Theta(\mathrm{OPT})$

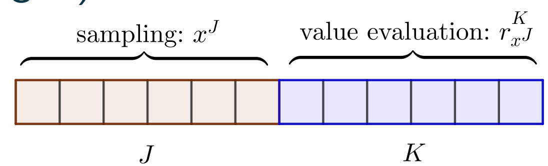
We focus on estimating # of points with $r_p \geq 1/2$

- Estimating # of points with $r_p \geq 1/2^i$ similar using subsampling

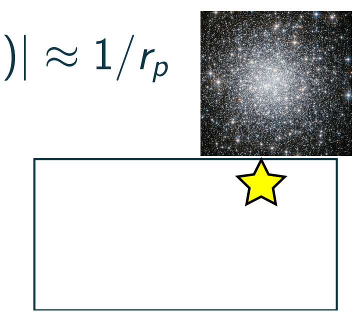
Two-pass algo:
- Hash points using consistent φ
- Sample a non-empty bucket b uniformly & a point from $\varphi^{-1}(b)$
  - using two-level ℓ_0 samplers
- 2nd pass: estimate r_p for each sampled point

$$\text{Bottom line: sampling } p \text{ with probability} \geq \frac{r_p}{\mathrm{poly}(d \cdot \log \Delta)}$$

Random-order streams:
- 1st half of stream for sampling
- 2nd half for estimating r_p's of sampled points



One-pass algo:  if "few" points around $p \Rightarrow r_p$ "large" — recall: $|X \cap B(p, r_p)| \approx 1/r_p$



Algorithmic Framework Overview

Recall: $\sum_p r_p = \Theta(\mathrm{OPT})$

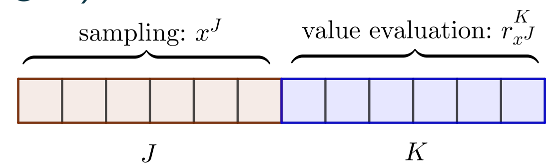
We focus on estimating # of points with $r_p \geq 1/2$

- Estimating # of points with $r_p \geq 1/2^i$ similar using subsampling

Two-pass algo:
- Hash points using consistent φ
- Sample a non-empty bucket b uniformly & a point from $\varphi^{-1}(b)$
  - using two-level ℓ_0 samplers
- 2nd pass: estimate r_p for each sampled point

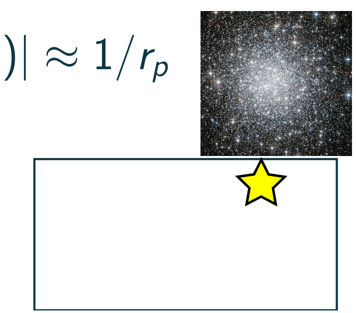
Bottom line: sampling p with probability $\geq \dfrac{r_p}{\mathrm{poly}(d \cdot \log \Delta)}$

Random-order streams:
- 1st half of stream for sampling
- 2nd half for estimating r_p's of sampled points



One-pass algo:  if "few" points around $p \Rightarrow r_p$ "large" — recall: $|X \cap B(p, r_p)| \approx 1/r_p$

- Count points in close neighborhood of each bucket
  - Similar idea as in [Frahling-Indyk-Sohler'05]
- We can distinguish $r_p \geq \frac{1}{2}$ and $r_p \leq 1/\Gamma$ using Γ-gap hash



Conclusions & Open Problem

	# of passes	ratio	space	notes
We wanted:	1	$O(1)$	poly(d)	(conjecture)
We got:	**2**	$O(1)$	poly(d)	$*$; **also 1-pass random-order**
	1	$O(d^{1.5})$	poly(d)	$*$
Lower bound:	1	< 1.085	$\Omega\left(2^{\text{poly}(d)}\right)$	$*$

Conclusions & Open Problem

	# of passes	ratio	space	notes
We wanted:	1	$O(1)$	poly(d)	(conjecture)
We got:	**2**	$O(1)$	poly(d)	$*$; **also 1-pass random-order**
	1	$O(d^{1.5})$	poly(d)	$*$
Lower bound:	1	< 1.085	$\Omega\left(2^{\text{poly}(d)}\right)$	$*$

Open problems:

- Prove/disprove what we wanted
- In general: need new techniques for high-dimensional spaces

Conclusions & Open Problem

	# of passes	ratio	space	notes
We wanted:	1	$O(1)$	poly(d)	(conjecture)
We got:	**2**	$O(1)$	poly(d)	**$*$; also 1-pass random-order**
	1	$O(d^{1.5})$	poly(d)	$*$
Lower bound:	1	< 1.085	$\Omega\left(2^{\mathrm{poly}(d)}\right)$	$*$

Open problems:

- Prove/disprove what we wanted
  - In general: need new techniques for high-dimensional spaces
- Consistent geometric hashing with better gap $\Gamma \Rightarrow$ one-pass $O(\Gamma)$-approx.
  - $\Gamma = O(d/\log d)$ seems possible [Filtser]

Conclusions & Open Problem

	# of passes	ratio	space	notes
We wanted:	1	$O(1)$	poly(d)	(conjecture)
We got:	**2**	$O(1)$	poly(d)	$*$; **also 1-pass random-order**
	1	$O(d^{1.5})$	poly(d)	$*$
Lower bound:	1	< 1.085	$\Omega\left(2^{\mathrm{poly}(d)}\right)$	$*$

Open problems:

- Prove/disprove what we wanted
  - In general: need new techniques for high-dimensional spaces
- Consistent geometric hashing with better gap $\Gamma \Rightarrow$ one-pass $O(\Gamma)$-approx.
  - $\Gamma = O(d/\log d)$ seems possible [Filtser]
  - Lower bound: $\Gamma = \Omega(d/\log d)$ (for poly(d) space) [Filtser '20]

Conclusions & Open Problem

	# of passes	ratio	space	notes
We wanted:	1	$O(1)$	poly(d)	(conjecture)
We got:	**2**	$O(1)$	poly(d)	$*$; **also 1-pass random-order**
	1	$O(d^{1.5})$	poly(d)	$*$
Lower bound:	1	< 1.085	$\Omega\left(2^{\text{poly}(d)}\right)$	$*$

Open problems:

- Prove/disprove what we wanted
  - In general: need new techniques for high-dimensional spaces
- Consistent geometric hashing with better gap $\Gamma \Rightarrow$ one-pass $O(\Gamma)$-approx.
  - $\Gamma = O(d/\log d)$ seems possible [Filtser]
  - Lower bound: $\Gamma = \Omega(d/\log d)$ (for poly(d) space) [Filtser '20]
- Multiple passes
  - Lower bound for two passes or random-order streams?
  - How many passes do we need for $1+\varepsilon$ approx. in poly($d \cdot \log n$) space

Conclusions & Open Problem

	# of passes	ratio	space	notes
We wanted:	1	$O(1)$	$\text{poly}(d)$	(conjecture)
We got:	**2**	$O(1)$	$\text{poly}(d)$	$*$; **also 1-pass random-order**
	1	$O(d^{1.5})$	$\text{poly}(d)$	$*$
Lower bound:	1	< 1.085	$\Omega\left(2^{\text{poly}(d)}\right)$	$*$

Open problems:

- Prove/disprove what we wanted
  - In general: need new techniques for high-dimensional spaces
- Consistent geometric hashing with better gap $\Gamma \Rightarrow$ one-pass $O(\Gamma)$-approx.
  - $\Gamma = O(d/\log d)$ seems possible [Filtser]
  - Lower bound: $\Gamma = \Omega(d/\log d)$ (for $\text{poly}(d)$ space) [Filtser '20]
- Multiple passes
  - Lower bound for two passes or random-order streams?
  - How many passes do we need for $1+\varepsilon$ approx. in $\text{poly}(d \cdot \log n)$ space
- Other applications of consistent geometric hashing / sparse partitions

Thank You!