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 \( \star \) 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
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 $\text{poly}(d)$

- $O(1)$ or even $(1 + \epsilon)$-approximation e.g. for:
  - MST, TSP, and Steiner tree
    - \cite{Frahling, Indyk, Sohler '05}
  - $k$-median, $k$-means, Max-Cut, ...
    - \cite{Frahling & Sohler '05}
  - Facility Location
    - \cite{Czumaj et al. '13}
  - Steiner Forest
    - \cite{Czumaj, Jiang, Krauthgamer, '22}

- Typical space decompositions: grids/quadtree

- Important case: $d = \Theta(\log n)$ (JL lemma)
  - only $O(\log n)$-approximation (or worse)
  - ratio $O(d \cdot \log \Delta)$ by tree embeddings
    - \cite{Indyk '04}
  - ratio $O(\log n)$ for MST and EMD
    - \cite{Chen, Jayaram, Levi, Waingarten '22}

- Lack of techniques for $O(1)$-approx.

- Tree embedding distorts distances by $\Omega(\log n)$
  - exception: ratio $(1 + \epsilon)$ for $k$-median and $k$-means
  - low space only for small $k$
    - \cite{Braverman, Frahling, Lang, Sohler, Yang '17}, \cite{Song, Yang, Zhong '18}

- Insertion-only setting:
  - Diameter et al.: ratio $O(1)$
    - \cite{Agarwal, Sharathkumar '15}
  - Width in any direction
    - \cite{Woodruff, Yasuda '22}
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, Veselý, Yang ’22]

- typical space decompositions: grids/quadtree

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

- 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]
**Geometric streams:**

**Low Dimension:** $\text{space exp (} d \text{)}$
- $O(1)$ 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:** $\text{space poly (} d \text{)}$
- 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 $\Omega(\log n)$

**Main dichotomy**
**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]
Geometric streams: Main dichotomy

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

High Dimension: \( \text{space } \text{poly}(d) \)
- Important case: \( d = \Theta(\log n) \) (JL lemma)
- only \( \O(\log n) \)-approximation (or worse)
  - ratio \( \O(d \cdot \log \Delta) \) by tree embeddings \([\text{Indyk '04}]\)
  - ratio \( \O(\log n) \) for MST and EMD \([\text{Chen, Jayaram, Levi, Waingarten '22}]\)
- 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 \(k\)
    - \([\text{Braverman, Frahling, Lang, Sohler, Yang '17}], [\text{Song, Yang, Zhong '18}]\)
- Insertion-only setting:
  - Diameter et al.: ratio \( O(1) \) \([\text{Agarwal, Sharathkumar '15}]\)
  - Width in any direction \([\text{Woodruff, Yasuda '22}]\)
Euclidean Uniform Facility Location

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

Goal: open a set of facilities $F$ to minimize

$$\text{cost}(X, F) := \sum_{p \in X} \text{dist}(p, F) + f \cdot |F|$$

where

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

This talk: unit facility cost $f = 1$

Image credits: NASA Hubble, CC BY 2.0, via Wikimedia Commons
Euclidean Uniform Facility Location

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

Goal: open a set of facilities $F$ to minimize

$$
cost(X, F) := \sum_{p \in X} \text{dist}(p, F) + f \cdot |F|
$$

$$
\text{connection cost}
$$

$$
\text{opening cost}
$$

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

Image credits: NASA Hubble, CC BY 2.0, via Wikimedia Commons
Euclidean Uniform Facility Location

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

Goal: open a set of facilities $F$ to minimize

$$\text{cost}(X, F) := \sum_{p \in X} \text{dist}(p, F) + f \cdot |F|$$

connection cost

opening cost

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

This talk: unit facility cost $f = 1$
Facility Location in Geometric Streams

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>reference &amp; notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous work:</td>
<td>1 (O(d \cdot \log^2 \Delta))</td>
<td>poly(d)</td>
<td>[Indyk ’04]</td>
</tr>
<tr>
<td>1</td>
<td>(\exp(d))</td>
<td>(\exp(d))</td>
<td>[Lammersen,Sohler ’08]</td>
</tr>
<tr>
<td>1</td>
<td>(1 + \varepsilon)</td>
<td>(\exp(d))</td>
<td>only for (d = 2)</td>
</tr>
<tr>
<td>1</td>
<td>(1 + \varepsilon)</td>
<td>(\exp(d))</td>
<td>[Czumaj,Lammersen,Monemizadeh,Sohler ’13]</td>
</tr>
</tbody>
</table>
Facility Location in Geometric Streams

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>reference &amp; notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous work:</td>
<td>1</td>
<td>$O(d \cdot \log^2 \Delta)$</td>
<td>poly($d$)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$\exp(d)$</td>
<td>$\exp(d)$</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$1 + \varepsilon$</td>
<td>$\exp(d)$</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$O(1)$</td>
<td>poly($d$)</td>
</tr>
</tbody>
</table>

We wanted: 1 $O(1)$ poly($d$)
## Facility Location in Geometric Streams

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>reference &amp; notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous work:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$O(d \cdot \log^2 \Delta)$</td>
<td>poly($d$)</td>
<td>[Indyk '04]</td>
</tr>
<tr>
<td>1</td>
<td>$\exp(d)$</td>
<td>$\exp(d)$</td>
<td>[Lammersen,Soehler '08]</td>
</tr>
<tr>
<td>1</td>
<td>$1 + \varepsilon$</td>
<td>$\exp(d)$</td>
<td>only for $d = 2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Czumaj,Lammersen,Monemizadeh,Sohler '13]</td>
</tr>
<tr>
<td>We wanted:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$O(1)$</td>
<td>poly($d$)</td>
<td>(still conjectured...)</td>
</tr>
<tr>
<td>We got:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$O(1)$</td>
<td>poly($d$)</td>
<td>*</td>
</tr>
<tr>
<td>1$^\dagger$</td>
<td>$O(1)$</td>
<td>poly($d$)</td>
<td>*; $1^\dagger = \text{random-order streams}$</td>
</tr>
</tbody>
</table>
## Facility Location in Geometric Streams

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>reference &amp; notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous work:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$O(d \cdot \log^2 \Delta)$</td>
<td>$\text{poly}(d)$</td>
<td>[Indyk '04]</td>
</tr>
<tr>
<td>1</td>
<td>$\exp(d)$</td>
<td>$\exp(d)$</td>
<td>[Lammersen,Sohler '08]</td>
</tr>
<tr>
<td>1</td>
<td>$1 + \varepsilon$</td>
<td>$\exp(d)$</td>
<td>only for $d = 2$</td>
</tr>
<tr>
<td>We wanted:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$O(1)$</td>
<td>$\text{poly}(d)$</td>
<td>(still conjectured...)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>We got:</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$O(1)$</td>
<td>$\text{poly}(d)$</td>
<td>*</td>
</tr>
<tr>
<td>1$^\dagger$</td>
<td>$O(1)$</td>
<td>$\text{poly}(d)$</td>
<td>*; $1^\dagger =$ random-order streams</td>
</tr>
<tr>
<td>1</td>
<td>$O(d^{1.5})$</td>
<td>$\text{poly}(d)$</td>
<td>*</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>reference &amp; notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous work:</td>
<td>1  (O(d \cdot \log^2 \Delta))</td>
<td>poly((d))</td>
<td>[Indyk '04]</td>
</tr>
<tr>
<td></td>
<td>1  (\exp(d))</td>
<td>(\exp(d))</td>
<td>[Lammersen,Sohler '08]</td>
</tr>
<tr>
<td></td>
<td>1  (1 + \varepsilon)</td>
<td>(\exp(d))</td>
<td>only for (d = 2)</td>
</tr>
</tbody>
</table>

We wanted: 1 \(O(1)\) poly(\(d\)) (still conjectured...)

We got: 2 \(O(1)\) poly(\(d\)) *

1\(^\dagger\) \(O(1)\) poly(\(d\)) *; 1\(^\dagger\) = random-order streams

1 \(O(d^{1.5})\) poly(\(d\)) *

Lower bound: 1 \(< 1.085\) \(\Omega(2^{\text{poly}(d)})\) * 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 \cite{Mettu, Plaxton '03}, \cite{Badoiu, Czumaj, Indyk, and Sohler '05}

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

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

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

1. \( \sum_p r_p = \Theta(\text{OPT}) \)
2. \( |X \cap B(p, r_p)| \approx \frac{1}{r_p} \) (\( X = \) input point set)

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

\( r_p = \frac{1}{2} \)

\( r_p \approx \frac{1}{n} \)

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

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

- 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 $\frac{1}{n} \leq r_p \leq 1$ such that:

1. $\sum r_p = \Theta(OPT)$
2. $|X \cap B(p, r_p)| \approx \frac{1}{r_p}$ (where $X = \text{input point set}$)

- 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)

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

- 1st pass: sample a few points uniformly
- 2nd pass: estimate $r_p$'s for sampled points
For every point \( p \), we define \( \frac{1}{n} \leq r_p \leq 1 \) such that:

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

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

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

\( r_p \approx \frac{1}{n} \)

\( \sqrt{n} \) points with \( r_p \geq \frac{1}{2} \) & \( \text{OPT} \approx \sqrt{n} \)

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

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

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

Goal: map/hash $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^d$, then sample uniformly from the support of $\phi(X)$
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 \( \sum_p r_p = \Theta(\text{OPT}) \)

Want to sample w/ prob. \( \sim r_p \) but cannot estimate \( r_p \) for queried \( p \) in one pass
**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 $\sum_p r_p = \Theta(OPT)$

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

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

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

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

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

- $\varphi^{-1}(p) = \text{bucket of points } p$
- 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

Goal: space decomposition such that:

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

Def.:

$$\phi: \mathbb{R}^d \rightarrow \mathbb{R}^d$$ is $\Gamma$-gap $\Lambda$-consistent hash if

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

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

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

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

Czumaj, Jiang, Krauthgamer, Veselý, Yang

Streaming Facility Location in High Dimension
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.:**

$\phi: \mathbb{R}^d \rightarrow \mathbb{R}^d$ is $\Gamma$-gap $\Lambda$-consistent hash if

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

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

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

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

Czumaj, Jiang, Krauthgamer, Veselý, Yang

Streaming Facility Location in High Dimension
Cosistent Geometric Hashing

Goal: space decomposition such that:
1. bounded diameter buckets
2. ball of small-enough diameter intersects poly \((d)\) buckets

Def.: \(\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^d\) is \(\Gamma\)-gap \(\Lambda\)-consistent hash if

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

- need \(\Gamma, \Lambda = \text{poly}(d)\)
  - \(\Gamma\) 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 \(\text{poly}(d)\) time & space
- we need data-oblivious \(\varphi\)
Construction of Consistent Geometric Hashing

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

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

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

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

- Start with grid & remove $\ell_{\infty}$ neighborhoods of faces

\[\text{Czumaj, Jiang, Krauthgamer, Veselý, Yang} \quad \text{Streaming Facility Location in High Dimension} \quad 8 / 11\]
Algorithmic Framework Overview

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

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 $\varphi$
    
    - Sample a non-empty bucket $b$ uniformly & a point from $\varphi^{-1}(b)$
      - using two-level $\ell_0$ samplers
    
    - 2nd pass: estimate $r_p$ for each sampled point

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

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

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 \( \varphi \)
  
- Sample a non-empty bucket \( b \) uniformly & a point from \( \varphi^{-1}(b) \)
    
  - using two-level \( \ell_0 \) samplers

- 2nd pass: estimate \( r_p \) for each sampled point

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

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

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 $\varphi$
  - Sample a non-empty bucket $b$ uniformly & a point from $\varphi^{-1}(b)$
    - using two-level $\ell_0$ samplers
  - 2nd pass: estimate $r_p$ for each sampled point

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

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

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

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 $\varphi$
  - Sample a non-empty bucket $b$ uniformly & a point from $\varphi^{-1}(b)$
    - using two-level $\ell_0$ samplers
  - 2nd pass: estimate $r_p$ for each sampled point

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

**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: \( \sum_p r_p = \Theta(\text{OPT}) \)

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 \( \varphi \)
  - Sample a non-empty bucket \( b \) uniformly & a point from \( \varphi^{-1}(b) \)
    - using two-level \( \ell_0 \) samplers
  - 2nd pass: estimate \( r_p \) for each sampled point

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

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 \( \Gamma \)-gap hash
Conclusions & Open Problem

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>We wanted:</strong></td>
<td>1</td>
<td>$O(1)$</td>
<td>$\text{poly}(d)$ (conjecture)</td>
</tr>
<tr>
<td><strong>We got:</strong></td>
<td>2</td>
<td>$O(1)$</td>
<td>$\text{poly}(d)$ $\ast$; also 1-pass random-order</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$O(d^{1.5})$</td>
<td>$\text{poly}(d)$ $\ast$</td>
</tr>
<tr>
<td><strong>Lower bound:</strong></td>
<td>1</td>
<td>$&lt; 1.085$</td>
<td>$\Omega \left(2^{\text{poly}(d)}\right)$ $\ast$</td>
</tr>
</tbody>
</table>
Conclusions & Open Problem

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>We wanted:</td>
<td>1</td>
<td>$O(1)$</td>
<td>poly$(d)$ (conjecture)</td>
</tr>
<tr>
<td>We got:</td>
<td>2</td>
<td>$O(1)$</td>
<td>poly$(d)$ $*$; also 1-pass random-order</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$O(d^{1.5})$</td>
<td>poly$(d)$ $*$</td>
</tr>
<tr>
<td>Lower bound:</td>
<td>1</td>
<td>$&lt; 1.085$</td>
<td>$\Omega(2^{\text{poly}(d)})$ $*$</td>
</tr>
</tbody>
</table>

Open problems:  
- Prove/disprove what we wanted
  - In general: need new techniques for high-dimensional spaces
Conclusions & Open Problem

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>We wanted:</strong></td>
<td>1</td>
<td>$O(1)$</td>
<td>poly($d$)</td>
</tr>
<tr>
<td><strong>We got:</strong></td>
<td>2</td>
<td>$O(1)$</td>
<td>poly($d$)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$O(d^{1.5})$</td>
<td>poly($d$)</td>
</tr>
<tr>
<td><strong>Lower bound:</strong></td>
<td>1</td>
<td>$&lt; 1.085$</td>
<td>$\Omega(2^{\text{poly}(d)})$</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>We wanted:</strong></td>
<td>1</td>
<td>$O(1)$</td>
<td>poly($d$) (conjecture)</td>
</tr>
<tr>
<td><strong>We got:</strong></td>
<td>2</td>
<td>$O(1)$</td>
<td>poly($d$) *; also 1-pass random-order</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$O(d^{1.5})$</td>
<td>poly($d$) *</td>
</tr>
<tr>
<td><strong>Lower bound:</strong></td>
<td>1</td>
<td>$&lt; 1.085$</td>
<td>$\Omega\left(2^{\text{poly}(d)}\right)$ *</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>We wanted:</strong></td>
<td>1</td>
<td>$O(1)$</td>
<td>$\text{poly}(d)$</td>
</tr>
<tr>
<td><strong>We got:</strong></td>
<td>2</td>
<td>$O(1)$</td>
<td>$\text{poly}(d)$</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$O(d^{1.5})$</td>
<td>$\text{poly}(d)$</td>
</tr>
<tr>
<td><strong>Lower bound:</strong></td>
<td>1</td>
<td>$&lt; 1.085$</td>
<td>$\Omega\left(2^{\text{poly}(d)}\right)$</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th># of passes</th>
<th>ratio</th>
<th>space</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>We wanted:</td>
<td>1</td>
<td>$O(1)$ poly($d$)</td>
<td>(conjecture)</td>
</tr>
<tr>
<td>We got:</td>
<td>2</td>
<td>$O(1)$ poly($d$)</td>
<td>$\ast$; also 1-pass random-order</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$O(d^{1.5})$ poly($d$)</td>
<td>$\ast$</td>
</tr>
<tr>
<td>Lower bound:</td>
<td>1</td>
<td>$&lt; 1.085$ $\Omega \left(2^{\text{poly}(d)}\right)$</td>
<td>$\ast$</td>
</tr>
</tbody>
</table>

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
- Other applications of consistent geometric hashing / sparse partitions
Thank You!