Streaming Algorithms for Robust Distinct Elements

Qin Zhang

Indiana University Bloomington

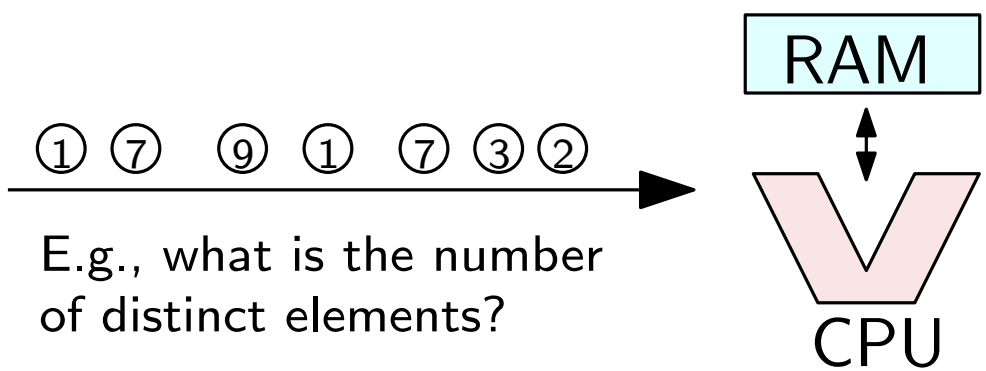
Joint work with Di Chen @ HKUST, in SIGMOD'16

Workshop on Multi-dimensional Proximity Problems

Jan. 13, 2015

The Streaming Model

– *high-speed* online data
– *limited* storage



- $o(m)$ space
- $\log^{O(1)}(m)$ update time
- $o(m)$ query time

m: stream length.

Linear sketches – a standard technique

Problem: For a data vector $x \in \mathbb{R}^d$, want to compute $f(x)$

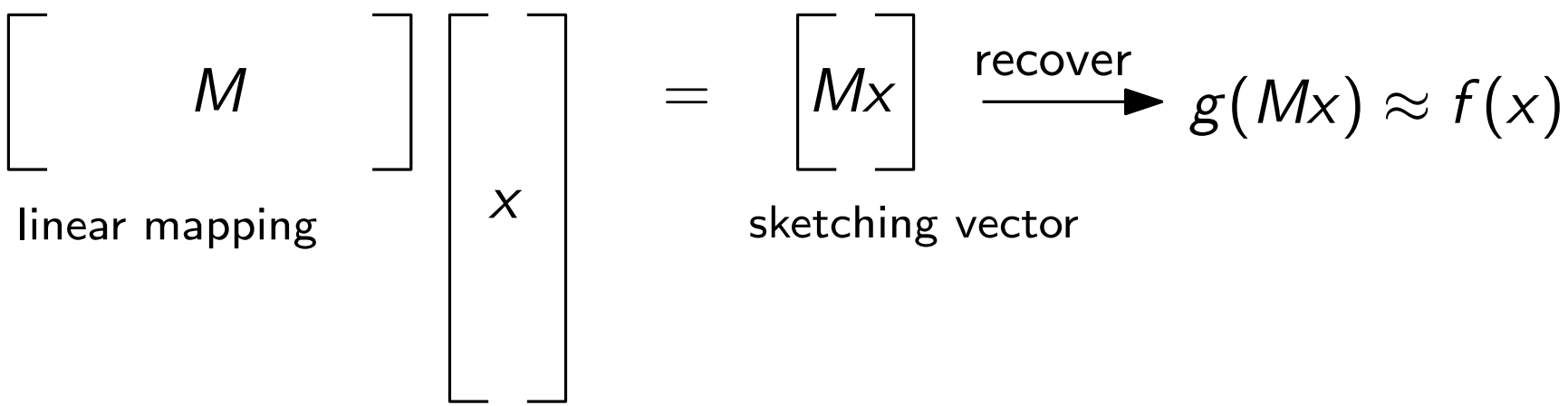
Can do this using linear sketches

$$\underbrace{\begin{bmatrix} & M & \end{bmatrix}}_{\text{linear mapping}} \begin{bmatrix} \\ x \\ \\ \end{bmatrix} = \underbrace{\begin{bmatrix} Mx \end{bmatrix}}_{\text{sketching vector}} \xrightarrow{\text{recover}} g(Mx) \approx f(x)$$

Linear sketches – a standard technique

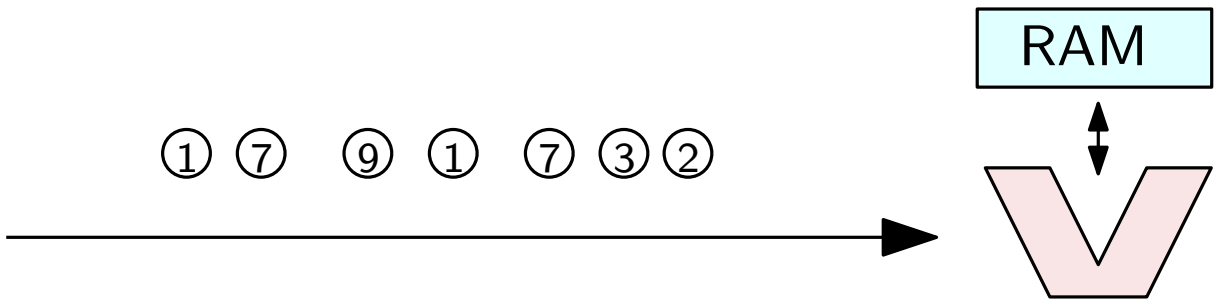
Problem: For a data vector $x \in \mathbb{R}^d$, want to compute $f(x)$

Can do this using linear sketches



Simple and useful: used in many statistical/graph/algebraic problems in streaming, compressive sensing, ...

Linear sketches in the streaming model



View each incoming element i as updating
$x \leftarrow x + \mathbf{e}_i$

Can update the sketching vector incrementally

$$\begin{bmatrix} M(x+\mathbf{e}_i) \end{bmatrix} = \begin{bmatrix} Mx \end{bmatrix} + \begin{bmatrix} M\mathbf{e}_i \end{bmatrix}$$

$$= \begin{bmatrix} Mx \end{bmatrix} + \begin{bmatrix} M^i \end{bmatrix}$$

sublinear space.:
size of sketch Mx

Real-world data is often noisy

- 

Music, Images, ...
After compressions, resize, reformat, etc.

Real-world data is often noisy

- 

  Music, Images, ...
  After compressions, resize, reformat, etc.

- 

  "multi-d workshop maryland"

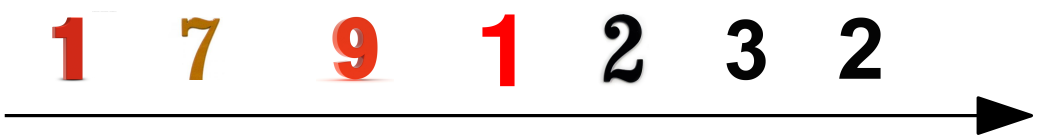
  "proximity problems workshop"

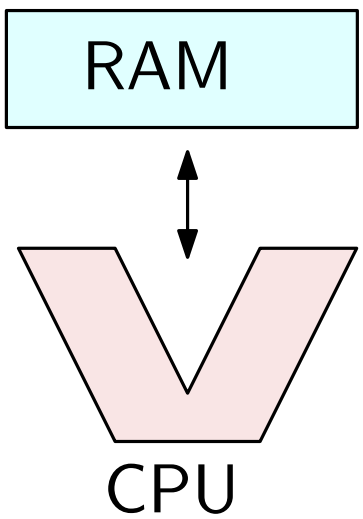
  "Rasmus Pagh workshop maryland"

  (unfortunately when I typed Dave's name the workshop didn't show up)

Queries of the same meaning sent to Google

Robust streaming algorithms





We (have to) consider similar items as one element. Then how to compute $f(x)$?

Linear sketches do not work

Linear sketches do not work. Why?

Items representing the same entity may be mapped into different coordinates of the sketching vector

Magic hash functions?

- Does there exist a magic hash function that can
  (1) map (only) items in same group into same bucket
  and
  (2) can be described succinctly?

  Answer: NO. #mappings is exponentially large.

Magic hash functions?

- Does there exist a magic hash function that can
  (1) map (only) items in same group into same bucket
  and
  (2) can be described succinctly?

  Answer: NO. #mappings is exponentially large.

- Locality sensitive hashing may help (will talk more later)

Clustering?

- Clustering will help?

  Answer: NO. #clusters can be linear

Clustering?

- Clustering will help?

  Answer: NO. #clusters can be linear

- Related to Entity Resolution: Identify and group different manifestations of the same real world object.

  Very important in data cleaning / integration. Have been studied for 40 years in DB, also in AI, NT.

  E.g. [Gill& Goldacre'03, Koudas et al.'06, Elmagarmid et al.'07, Herzog et al.'07, Dong& Naumann'09, Willinger et al.'09, Christen'12] for introductions, and [Getoor and Machanavajjhala'12] for a toturial.

Use at least linear space in the RAM model, detect items rep. the same entity, output all distinct entities.

This talk

- **Problem**: compute # robust distinct elements (F_0)
  (Useful in: traffic monitoring, query optimization, ...)

  Given a threshold α, partition the input item set S into a minimum *set of groups $\mathcal{G} = \{G_1, \ldots, G_n\}$ so that $\forall p, q \in G_i, d(p, q) \leq \alpha$.*

- **Data**: points in the Euclidean space and beyond
- **Model of Computation**: the streaming model

Well-shaped dataset

- For a fixed α, we say the point set S is (α, β)-sparse (with separation ratio β/α) if for $\forall u, v \in S$:

  $$\text{either } d(u, v) \leq \alpha \text{ or } d(u, v) \geq \beta.$$

  We call the dataset is well-shaped if $\max_\beta \beta \geq 2\alpha$.

Well-shaped dataset

- For a fixed α, we say the point set S is (α, β)-sparse (with separation ratio β/α) if for $\forall u, v \in S$:

  $$\text{either } d(u, v) \leq \alpha \text{ or } d(u, v) \geq \beta.$$

  We call the dataset is well-shaped if $\max_\beta \beta \geq 2\alpha$.

- A natural partition exists for a well-shaped dataset

Well-shaped dataset

- For a fixed α, we say the point set S is (α, β)-sparse (with separation ratio β/α) if for $\forall u, v \in S$:

  $$\text{either } d(u, v) \leq \alpha \text{ or } d(u, v) \geq \beta.$$

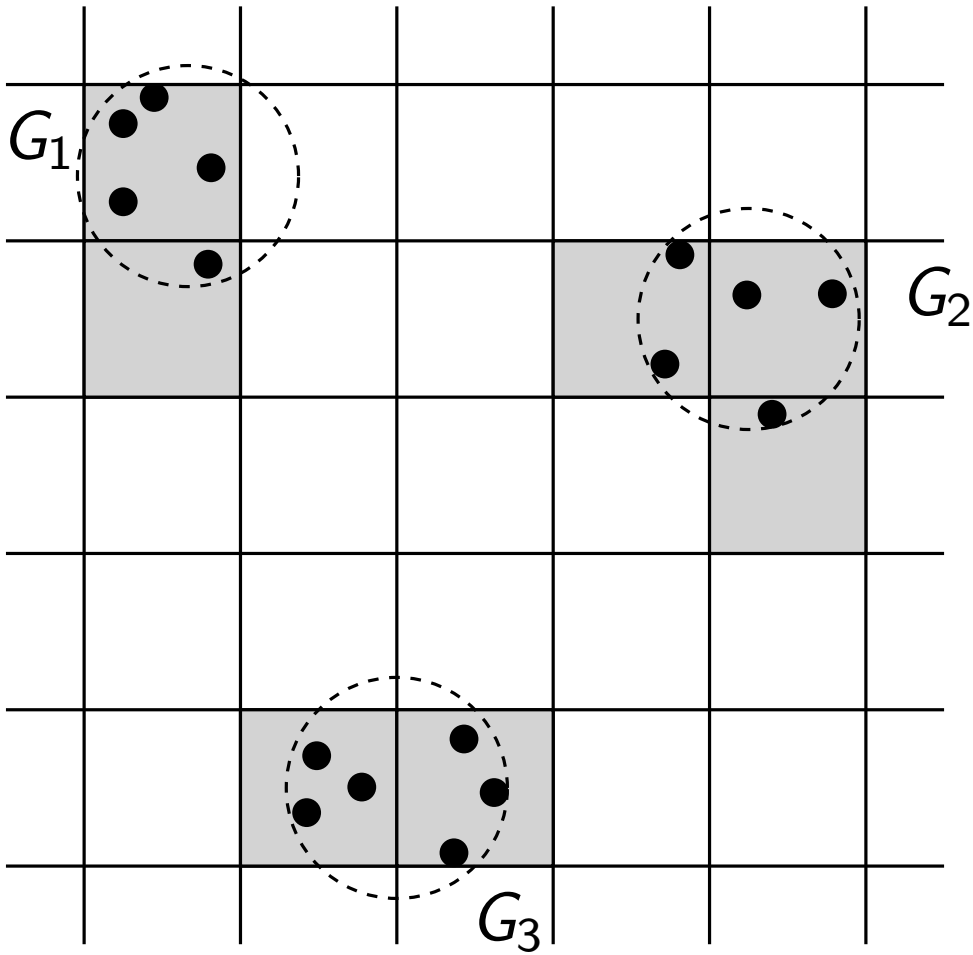
  We call the dataset is well-shaped if $\max_\beta \beta \geq 2\alpha$.

- A natural partition exists for a well-shaped dataset

- For general datasets, we introduce F_0-ambiguity:

  The F_0-ambiguity of S is the *minimum* δ s.t. there exists $T \subseteq S$ such that
  - $S \setminus T$ is well-shaped
  - $F_0(S \setminus T) \geq (1 - \delta) F_0(S)$

Algorithm for well-shaped datasets in 2D



A random grid $\mathbb{G}$ of side length $\alpha/\sqrt{2}$

Simple sampling (needs two passes)

Algorithm Simple Sampling

1. Sample $\eta \in \tilde{O}(1/\epsilon^2)$ *non-empty* cells $\mathcal{C}$
2. Use another pass to compute for each sampled cell C,

$$w(C) = 1/w(G_C),$$

   where G_C is the (only) group intersecting C, and $w(G_C)$ is #cells G_C intersects
3. Output $\frac{z}{\eta} \cdot \sum_{C \in \mathcal{C}} w(C)$, where z is the #non-empty cells in $\mathbb{G}$

Theorem

Simple-Sampling gives a $(1+\epsilon)$-approximation of F_0 with probability $2/3$ using $\tilde{O}(1/\epsilon^2)$ bits and 2 passes.

A dilemma and bucket sampling

- Cannot sample cell early: Most sampled cell will be empty thus useless for the estimation.
- Cannot sample late: Cannot obtain the "neighborhood" information to compute $w(C)$ for a sampled C

What to do?

A dilemma and bucket sampling

- Cannot sample cell early: Most sampled cell will be empty thus useless for the estimation.
- Cannot sample late: Cannot obtain the "neighborhood" information to compute $w(C)$ for a sampled C

What to do?

We propose **bucket sampling**: sample a *collection* of cells, but only maintain the neighborhood information for non-empty sampled cells.

Maintain the collection using a hash function: That is, all cells with $h(C) = 1$

Bucket sampling (cont.)

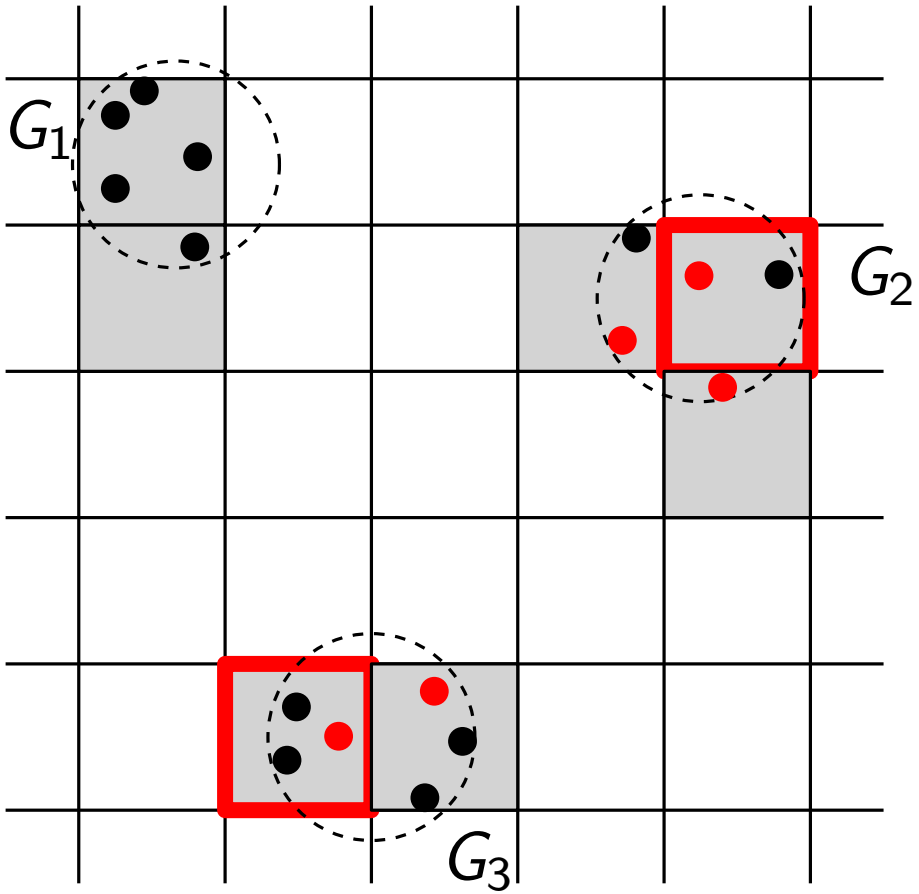
Algorithm 2 Store Point Centers for Sampled Cells $\mathcal{C}'$

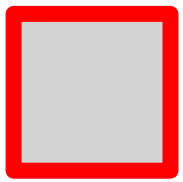
```
1: procedure StoreCenter(p)
2:     if ∃C ∈ 𝔾 s.t. h(C) = 1 ∧ d(p, C) ≤ 1 then
3:         if (∄q ∈ Γ s.t. cell(p) = cell(q)) then
4:             insert p to Γ                    ▷ Keep a new center
5:         end if
6:     end if
7: end procedure
```

Γ: the set of points we store, to recover $w(C)$ for each sampled cell C at the end

h: updated so that at at any point $|\{C \mid h(C) = 1\}| = O(1/\epsilon^2)$

Bucket sampling (cont.)





sampled cell

stored centers in Γ

Theorem

For a well-shaped dataset, exists an algorithm that gives a $(1+\epsilon)$-approximation of robust F_0 w.pr. $2/3$ using $\tilde{O}(1/\epsilon^2)$ bits and $O(1)$ processing time per item

Dealing with ambiguity

For general datasets, we introduce F_0-ambiguity:

The F_0-ambiguity of S is the *minimum* δ s.t. there exists $T \subseteq S$ such that

- $S \setminus T$ is well-shaped
- $F_0(S \setminus T) \geq (1 - \delta) F_0(S)$

Unfortunately approximate δ is hard – we cannot differetiate whether $\delta = 0$ or $1/2$ without an $\Omega(m)$ space, by reducing it to the `Diameter` problem

Dealing with ambiguity

For general datasets, we introduce F_0-ambiguity:

The F_0-ambiguity of S is the *minimum* δ s.t. there exists $T \subseteq S$ such that

- $S \setminus T$ is well-shaped
- $F_0(S \setminus T) \geq (1 - \delta) F_0(S)$

Unfortunately approximate δ is hard – we cannot differetiate whether $\delta = 0$ or $1/2$ without an $\Omega(m)$ space, by reducing it to the `Diameter` problem

However, we can still guarantee the following even *without* knowing the value δ

Dealing with ambiguity

For general datasets, we introduce F_0-ambiguity:

The F_0-ambiguity of S is the *minimum* δ s.t. there exists $T \subseteq S$ such that

- $S \setminus T$ is well-shaped
- $F_0(S \setminus T) \geq (1 - \delta) F_0(S)$

Unfortunately approximate δ is hard – we cannot differetiate whether $\delta = 0$ or $1/2$ without an $\Omega(m)$ space, by reducing it to the `Diameter` problem

Theorem

For a dataset with F_0-ambiguity δ, exists an algorithm that gives a $(1 + O(\epsilon + \delta))$ approximation of robust F_0 w.pr. $2/3$ using $\tilde{O}(1/\epsilon^2)$ bits and $O(1)$ processing time per item

Proof ideas (high level)

- We can think
  OPT $\approx$ natural grouping for $S \setminus T$ + balls of diameter α covering rest points in T.
- In $2D$ Euclidean space, a ball of diameter 2α can be covered by $O(1)$ balls of diameter α
- In OPT balls covering T are almost evenly spread w.r.t. the (natural) groups formed by $S \setminus T$
- SOL uses near-uniform group samplings $\Rightarrow$
  if δ is small, then outliers T will not affect much of our estimation of $F_0(S \setminus T)$, which is close to $F_0(S)$ by the definition of T.

We generalized to high-D, but need larger separation ratios to avoid exp. dependence on d

Theorem

For a dataset with separation ratio at least $d^{3/2}$ in d-dim, exists an algorithm that outputs a $(1+\epsilon)$-approximation to robust F_0 w.pr. $2/3$, using $O(d/\epsilon^2)$ space and amortized $O(d)$ processing time per item.

Random grid partition is a locality sensitive hashing (LSH). Can use bucket sampling with other LSHs.

Smart hash function

We say a hash function h is ρ-smart on an (α, β)-sparse ($\beta \geq 2\alpha$) dataset S and its natural minimum-cardinality group partition if it satisfies:

- Small "imaging radius". Each group is adjacent to ρ cells on average.
  – we say a group G is adjacent to a hash bucket C if there exists a pair of items $p, q \in S$ such that $p \in G, h(q) = C$ and $d(p, q) \leq \alpha$.
- No false-positive. Items from different groups will be hashed into disjoint buckets.

Smart hash function

We say a hash function h is ρ-smart on an (α, β)-sparse ($\beta \geq 2\alpha$) dataset S and its natural minimum-cardinality group partition if it satisfies:

- Small "imaging radius". Each group is adjacent to ρ cells on average.
  – we say a group G is adjacent to a hash bucket C if there exists a pair of items $p, q \in S$ such that $p \in G, h(q) = C$ and $d(p, q) \leq \alpha$.
- No false-positive. Items from different groups will be hashed into disjoint buckets.

This is what we really need in the analysis for `Grid + 2D Euclidean space`

Locality sensitive hashing

- We say a hash family $\mathcal{H}$ is (ℓ, u, p_1, p_2)-sensitive if for any two items p, q,
  1. if $d(p, q) \leq \ell$ then $Pr_{h \in_r \mathcal{H}}[h(p) = h(q)] \geq p_1$,
  2. if $d(p, q) \geq u$ then $Pr_{h \in_r \mathcal{H}}[h(p) = h(q)] \leq p_2$

Locality sensitive hashing

- We say a hash family $\mathcal{H}$ is (ℓ, u, p_1, p_2)-sensitive if for any two items p, q,
  1. if $d(p, q) \leq \ell$ then $Pr_{h \in_r \mathcal{H}}[h(p) = h(q)] \geq p_1$,
  2. if $d(p, q) \geq u$ then $Pr_{h \in_r \mathcal{H}}[h(p) = h(q)] \leq p_2$

- A hash function h is called η-concentrated on S if for any $G \in \mathcal{G}$,

$$|\{h(x) \mid \exists y \in G \ s.t. \ d(x, y) \leq \alpha\}| \leq \eta.$$

  We say an LSH family that is η-concentrated on S if for any $h \in \mathcal{H}$, h is η-concentrated on S.

The connections

S: an (α, β)-sparse ($\beta \geq 2\alpha$) dataset, $|S| = m$.

$\mathcal{H}$: a $(2\alpha, \beta, p_1, p_2)$-sensitive LSH family that is η-concentrated on S.

$\mathcal{F}$: a k-fold hash family of $\mathcal{H}$ and let $f \in_r \mathcal{F}$.

Then f is $100(\eta(1 - p_1) + p_1)^k$-smart on S w.pr. $(0.99 - m^2 p_2^k)$.

- **Gaussian LSH for Euclidean Metric**
  is $(\alpha, \beta, p(\alpha), p(\beta))$-sensitive and $O(1)$-concentrated; can be made $O(1)$-smart when $\beta/\alpha \geq \log m$

- **Random Projection LSH for Cosine Metric**
  is $(\alpha, \beta, 1 - \alpha/\pi, 1 - \beta/\pi)$-sensitive and $O(1)$-concentrated; can be made $O(1)$-smart when $\alpha \leq 1/\log m$ and $\Omega(1) \leq \beta < \pi$.

Experiments

Dataset: 4,000,000 images from `ImageNet`

I500k100x5d means the dataset consists of 500k images, each has 100 near-duplicates *on avarage*, and is mapped into a 5-dim Euclidean space (feature space)

Experiments on a desktop PC with 8GB of RAM and a 4-core 3.40GHz Intel i7 CPU

Correnctness (known α)

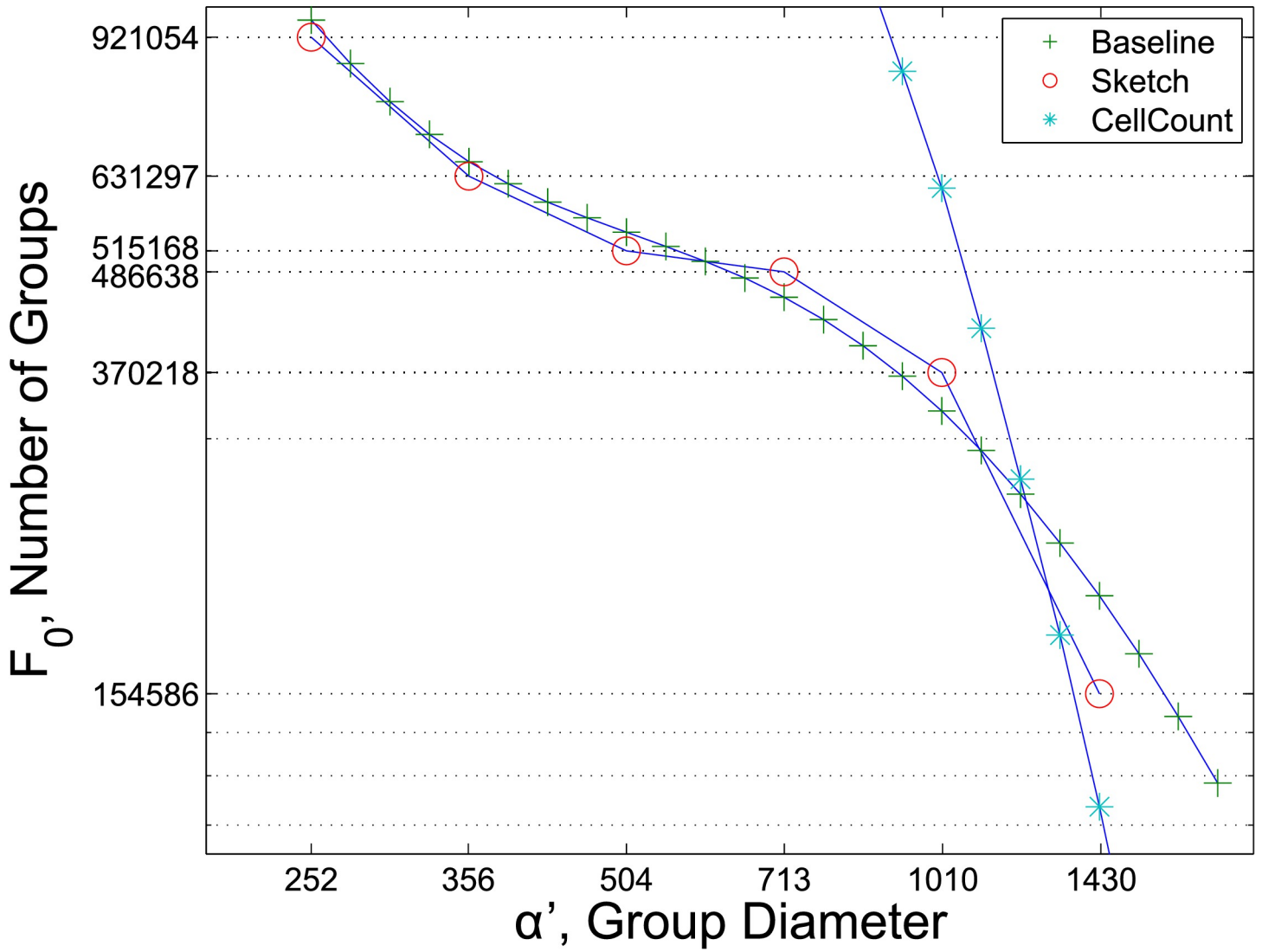
No. pts	9,000	18,000	36,000	72,000
I500k100x5d	22.8%	10.6%	8.3%	6.6%
I500k10x5d	15.8%	9.2%	6.7%	5.7%
I500k2x5d	5.2%	3.0%	2.8%	2.2%
I4m2x5d	6.0%	3.5%	3.3%	2.4%

Table 6: Vary duplication ratio; average error over 20 runs; median output of 6 sketches; known α.

No. pts	9,000	18,000	36,000	72,000	144,000
I4m2x5d	6.0%	3.5%	3.3%	2.4%	1.7%
I4m2x10d	5.8%	4.2%	3.4%	2.6%	1.5%
I4m2x20d	6.4%	4.4%	3.6%	2.0%	1.3%

Table 7: Vary dimensionality; average error over 20 runs; median output of 6 sketches; known α.

Correnctness (unknown α)



`Baseline` (greedy algo.)
$\Theta(n)$ space

`Sketch` (our algo.)
$\tilde{O}(1/\epsilon^2)$ space

`CellCount`: (streaming algo. for comparison)
$\tilde{O}(1/\epsilon^2)$ space

Dataset: I500k100x5d

Running time

Samples: / Space (pts):	200 / 1,500	400 / 3,000	800 / 6,000	1,600 / 12,000	Baseline
I500k100x5d	0.45	0.47	0.49	0.46	1.45
I500k10x5d	0.42	0.50	0.52	0.46	1.42
I100k100x5d	0.48	0.44	0.48	0.53	1.38
I10k100x5d	0.42	0.48	0.51	0.50	1.35

Table 5: Average processing time (seconds) per 10,000 pts

Open problems

- Theoretical analysis for high dimension Euclidean space is not complete yet.

Open problems

- Theoretical analysis for high dimension Euclidean space is not complete yet.
- Extending the analysis and experiments to other metrics

Open problems

- Theoretical analysis for high dimension Euclidean space is not complete yet.
- Extending the analysis and experiments to other metrics
- Other statistical aggregate problems

Thank you!
Questions?