Distinct Sampling on Streaming Data with Near-Duplicates

Jiecao Chen (IUB)

joint work with Qin Zhang (IUB)

Workshop on Data Summarisation, Mar/2018

Real-world data is often noisy

Real-world data is often noisy

images, videos, music ... after compressions, resize etc.

Real-world data is often noisy

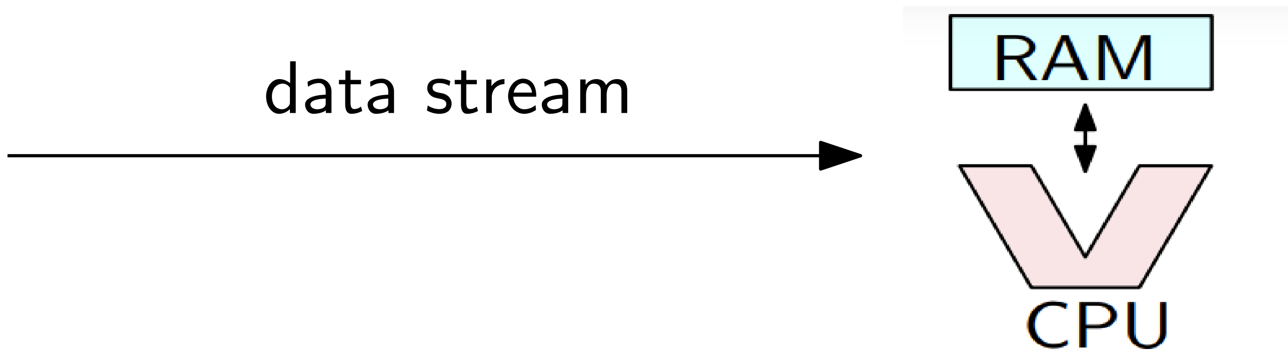
images, videos, music ... after compressions, resize etc.

"data summarization"
"summarization of data"
"the summarization of data"
"summarization data"

queries of the same meaning sent to Google

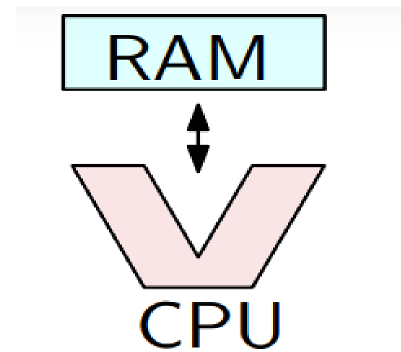
Models of Computation

Models of Computation



Models of Computation

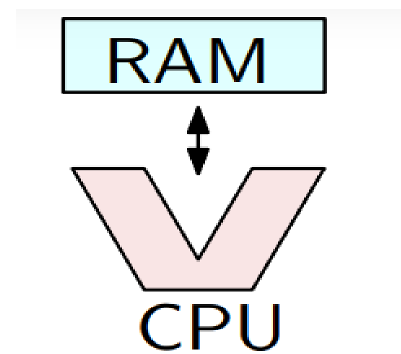
data stream



- high-speed online data
- want space/time efficient algorithms

Models of Computation

data stream



- high-speed online data
- want space/time efficient algorithms

sliding window: only consider recent w items.

Magic hash function

Magic hash function

Does there exist a magic hash function that can

- maps all similiar items to one id
- can be described succinctly?

Magic hash function

Does there exist a magic hash function that can

- maps all similiar items to one id
- can be described succinctly?

unlikely to have one ...

Magic hash function

Does there exist a magic hash function that can

- maps all similiar items to one id
- can be described succinctly?

unlikely to have one ...

So we could not apply existing streaming algorithms directly

need some new ideas

Robust ℓ_0-sampling

- **data:** data points in $\mathbb{R}^d$
- ℓ_0-**sampling:** each distinct element is sampled with prob. $\frac{1}{F_0}$

Robust ℓ_0-sampling

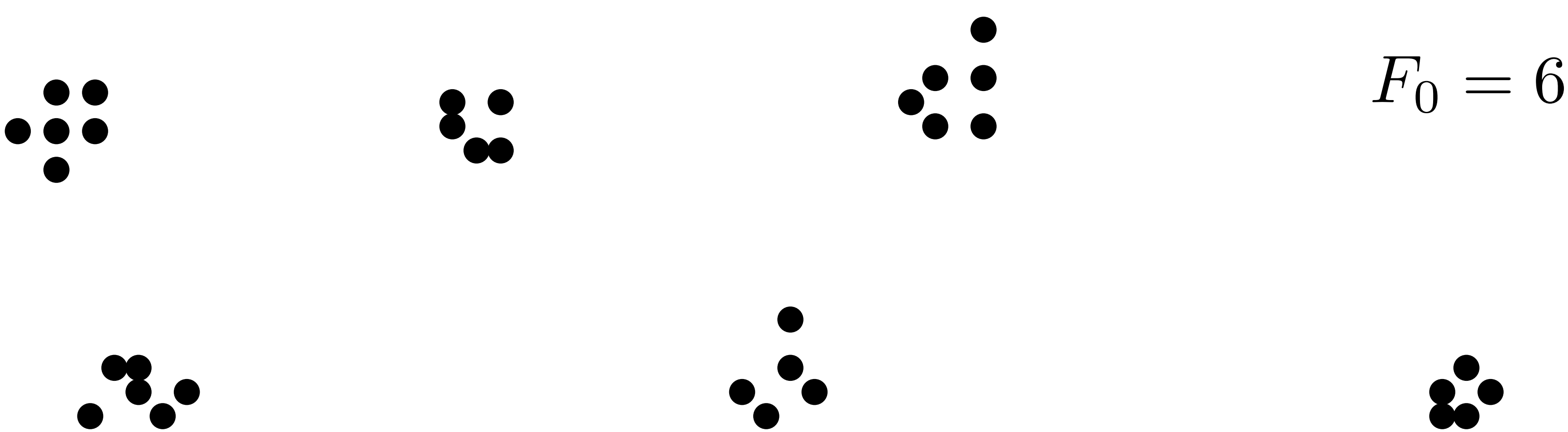
- **data:** data points in $\mathbb{R}^d$
- ℓ_0-**sampling:** each distinct element is sampled with prob. $\frac{1}{F_0}$

Robust ℓ_0-**sampling**: treat all close points as a group, each group is sampled with prob. $\frac{1}{F_0}$

Robust ℓ_0-sampling

- **data:** data points in $\mathbb{R}^d$
- ℓ_0-**sampling:** each distinct element is sampled with prob. $\frac{1}{F_0}$

Robust ℓ_0-sampling: treat all close points as a group, each group is sampled with prob. $\frac{1}{F_0}$



Formally ...

- $S \subset \mathbb{R}^d$ is (α, β)-sparse: either $d(u, v) \leq \alpha$ or $d(u, v) > \beta$ for all $u, v \in S$.

Formally ...

- $S \subset \mathbb{R}^d$ is (α, β)-sparse: either $d(u, v) \leq \alpha$ or $d(u, v) > \beta$ for all $u, v \in S$.
- when $\beta/\alpha > 2$,

$$G(v) = \{u \in S \mid d(u, v) \leq \alpha\}$$

  forms a group of v
- the dataset S is well-shaped
- a natural partition exists for a well-shaped dataset
- F_0 is the number of groups

Our goal:

- S is well-shaped, fed as a data stream
- $\mathcal{G} = \{G_1, G_2, \ldots, G_{F_0}\}$ is the natural partition
- **Goal:** outputs a point u such that,

$$\forall i \in [F_0], \Pr[u \in G_i] = 1/F_0$$

Our goal:

- S is well-shaped, fed as a data stream
- $\mathcal{G} = \{G_1, G_2, \ldots, G_{F_0}\}$ is the natural partition
- **Goal:** outputs a point u such that,

$$\forall i \in [F_0], \Pr[u \in G_i] = 1/F_0$$

focus on $\mathbb{R}^2$ case in this talk.
can be extended to general d

Our goal:

- S is well-shaped, fed as a data stream
- $\mathcal{G} = \{G_1, G_2, \ldots, G_{F_0}\}$ is the natural partition
- **Goal:** outputs a point u such that,

$$\forall i \in [F_0], \Pr[u \in G_i] = 1/F_0$$

focus on $\mathbb{R}^2$ case in this talk.
can be extended to general d

our algorithm also work with general datasets in $\mathbb{R}^{O(1)}$ (discuss later)

Infinite Window ($\mathbb{R}^2$)

Infinite Window ($\mathbb{R}^2$)

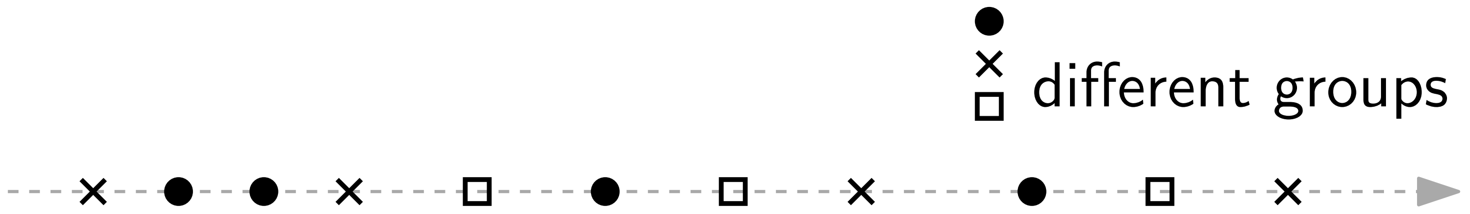
Basic idea

- over the data stream, mark one representative point for each group.

Infinite Window ($\mathbb{R}^2$)

Basic idea

- over the data stream, mark one representative point for each group.

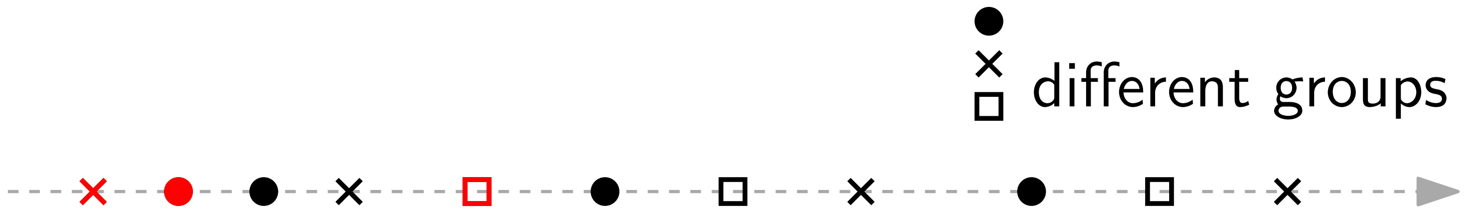


Infinite Window ($\mathbb{R}^2$)

Basic idea

- over the data stream, mark one representative point for each group.

e.g., pick the first arrived point of each group

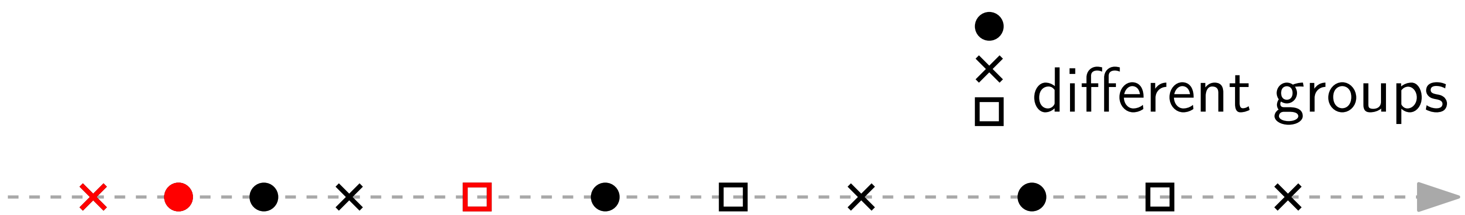


Infinite Window ($\mathbb{R}^2$)

Basic idea

- over the data stream, mark one representative point for each group.

e.g., pick the first arrived point of each group



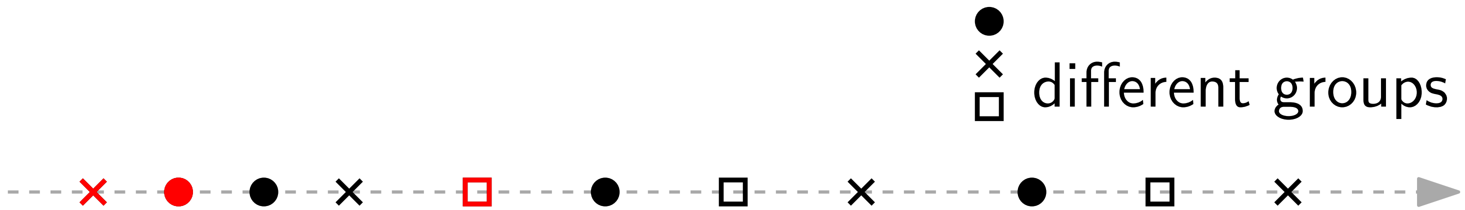
only sample from representative points

Infinite Window ($\mathbb{R}^2$)

Basic idea

- over the data stream, mark one representative point for each group.

e.g., pick the first arrived point of each group



only sample from representative points

Question: Can we identify (not necessarily store) the first arrived point of each group space-efficiently?

Question: Can we identify (not necessarily store) the first arrived point of each group space-efficiently?

Question: Can we identify (not necessarily store) the first arrived point of each group space-efficiently?

Unfortunately, $\Omega(F_0)$ space required to identify the representative points in noisy dataset

Solution: sample in advance

Question: Can we identify (not necessarily store) the first arrived point of each group space-efficiently?

Unfortunately, $\Omega(F_0)$ space required to identify the representative points in noisy dataset

Solution: sample in advance

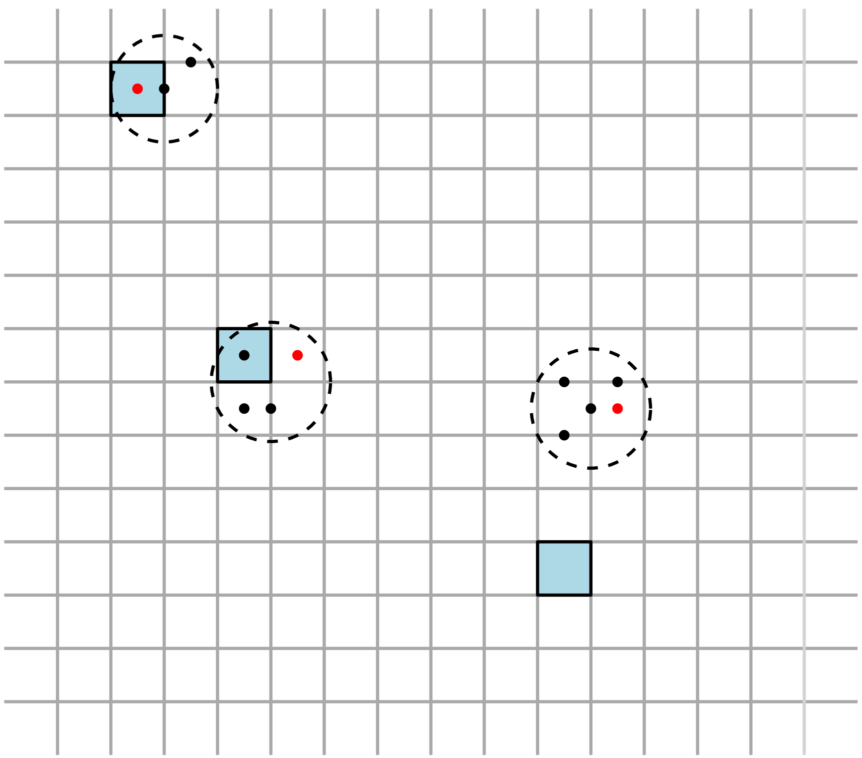
- how to sample in advance?
  - place a random grid (side length $\frac{\alpha}{2}$) in $\mathbb{R}^2$, sample cells before we see the data stream

Question: Can we identify (not necessarily store) the first arrived point of each group space-efficiently?

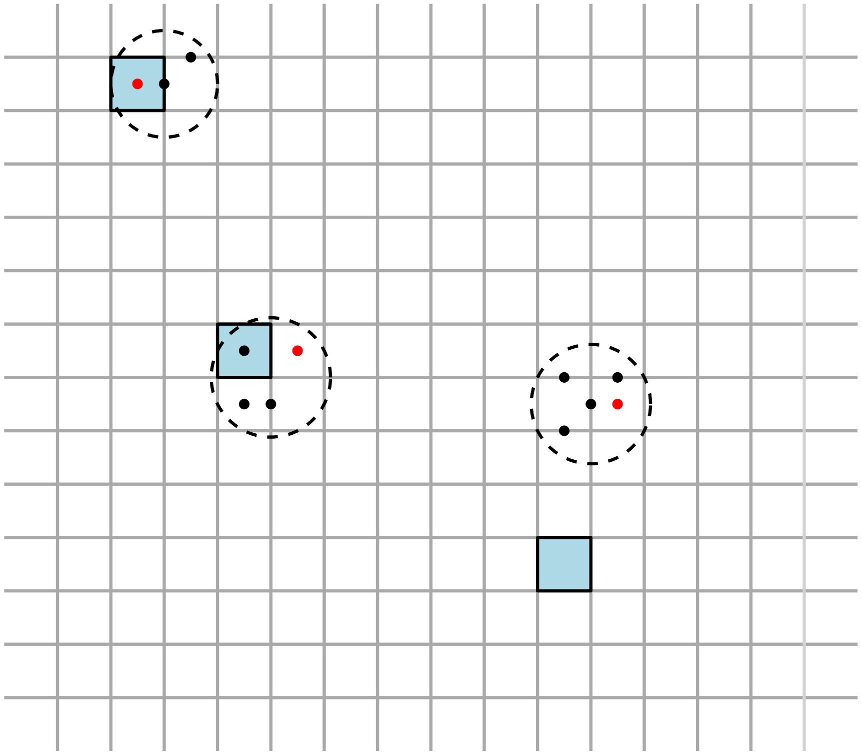
Unfortunately, $\Omega(F_0)$ space required to identify the representative points in noisy dataset

Solution: sample in advance

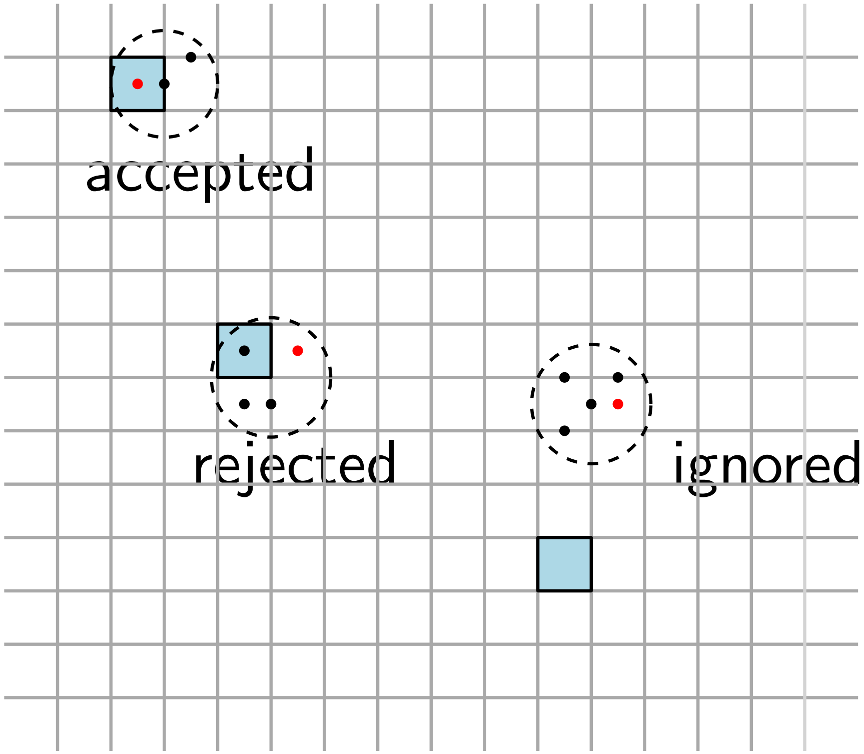
- how to sample in advance?
  - place a random grid (side length $\frac{\alpha}{2}$) in $\mathbb{R}^2$, sample cells before we see the data stream
- how to decide the sample rate?
  - decrease when see more groups



- blue cells: sampled cell
- red points: first arrived point of its group



- blue cells: sampled cell
- red points: first arrived point of its group



three types of groups:

- accepted: first arrived point falls into a sampled cell
- ignored: no point falls into a sampled cell
- rejected: has point falling into a sampled cell, but not the first arrived point

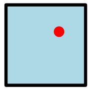
How to maintain accepted groups?

How to maintain accepted groups?

- keep all first points of accepted groups

How to maintain accepted groups?

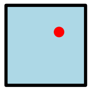
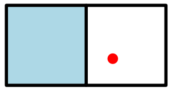
- keep all first points of accepted groups



keep it!

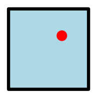
How to maintain accepted groups?

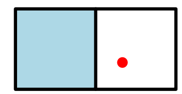
- keep all first points of accepted groups

   keep it!  discard?

How to maintain accepted groups?

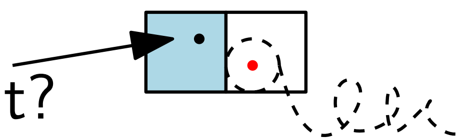
- keep all first points of accepted groups


keep it!


discard?

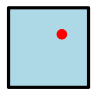
If we discard the first arrived point of a rejected group...

is this the first arrived point?

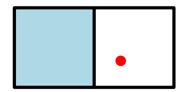

discard

How to maintain accepted groups?

- keep all first points of accepted groups



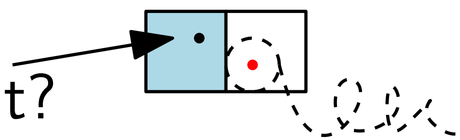
keep it!



discard?

If we discard the first arrived point of a rejected group...

is this the first arrived point?

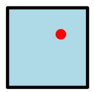
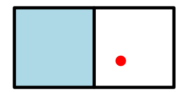


discard

we don't know!

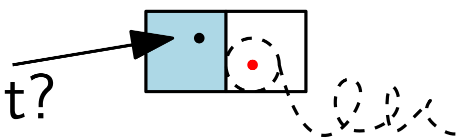
How to maintain accepted groups?

- keep all first points of accepted groups

 keep it!  discard?

If we discard the first arrived point of a rejected group...

is this the first arrived point?



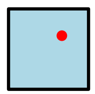
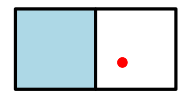
discard

we don't know!

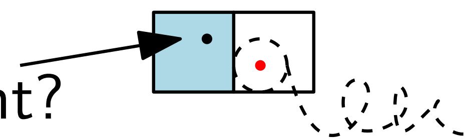
So we need to keep the first arrived point of a rejected group!

How to maintain accepted groups?

- keep all first points of accepted groups

 keep it!  discard?

If we discard the first arrived point of a rejected group...

is this the first arrived point?  discard

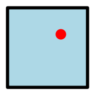
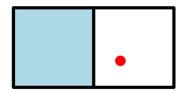
we don't know!

So we need to keep the first arrived point of a rejected group!

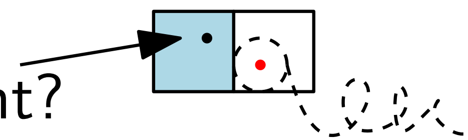
$$\mathrm{ADJ}(p) = \{\text{cells with distance smaller than } \alpha \text{ from } p\}$$

How to maintain accepted groups?

- keep all first points of accepted groups

 keep it!  discard?

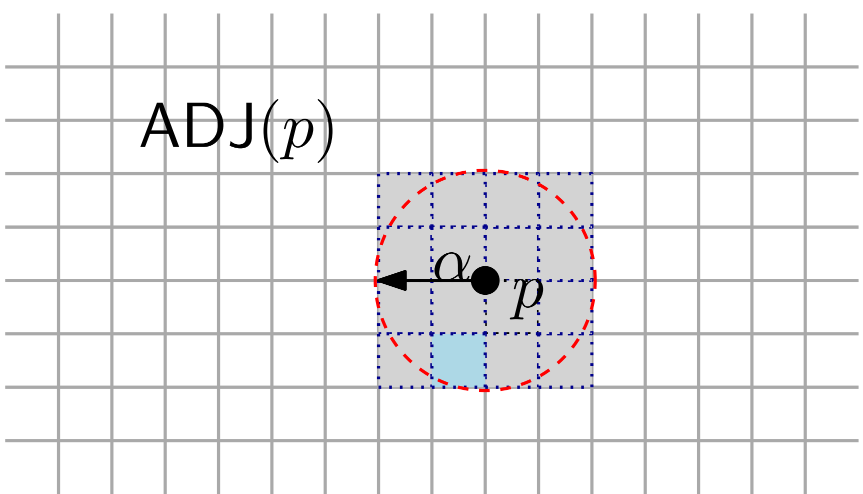
If we discard the first arrived point of a rejected group...

is this the first arrived point?  discard

we don't know!

So we need to keep the first arrived point of a rejected group!

$\mathsf{ADJ}(p) = \{$cells with distance smaller than α from $p\}$



if p is not in a sampled cell and $\mathsf{ADJ}(p)$ has cell sampled...

keep p!

Now we keep two sets,

- S^{acc}: first arrived points of accepted groups
- $S^{\mathsf{rej}} = \{$first point $p \notin S^{\mathsf{acc}}$ and ADJ(p) has sampled cell$\}$

Now we keep two sets,

- S^{acc}: first arrived points of accepted groups
- $S^{\text{rej}} = \{\text{first point } p \notin S^{\text{acc}} \text{ and ADJ}(p) \text{ has sampled cell}\}$

In $\mathbb{R}^2$, $|S^{\text{rej}}| = O(|S^{\text{acc}}|)$ w.h.p.

Now we keep two sets,

- S^{acc}: first arrived points of accepted groups
- $S^{\text{rej}} = \{\text{first point } p \notin S^{\text{acc}} \text{ and ADJ}(p) \text{ has sampled cell}\}$

In $\mathbb{R}^2$, $|S^{\text{rej}}| = O(|S^{\text{acc}}|)$ w.h.p.

how to decide the sample rate?

Now we keep two sets,

- S^{acc}: first arrived points of accepted groups
- $S^{\mathsf{rej}} = \{\text{first point } p \notin S^{\mathsf{acc}} \text{ and ADJ}(p) \text{ has sampled cell}\}$

In $\mathbb{R}^2$, $|S^{\mathsf{rej}}| = O(|S^{\mathsf{acc}}|)$ w.h.p.

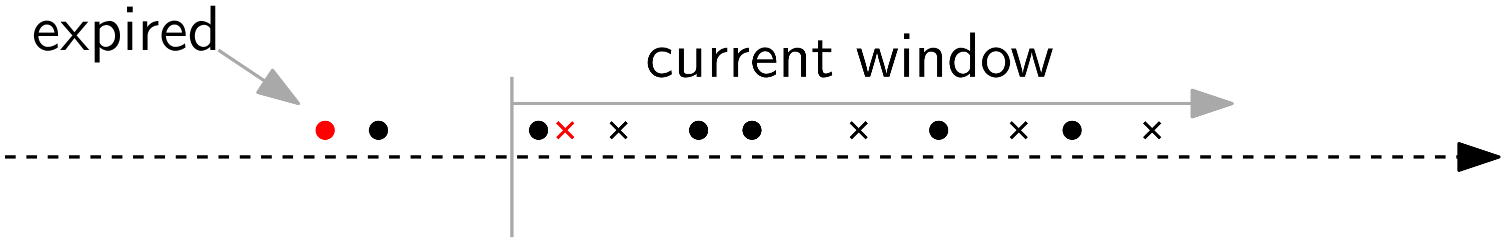
how to decide the sample rate?

- if $|S^{\mathsf{acc}}| > \kappa \log m$, re-sample each sampled cell with prob. $\frac{1}{2}$
- roughly, S^{acc} will drop half of its size
- S^{acc} is not empty w.h.p.
- space usage $O(\log m)$

Extend to Sliding Window is non-trivial ...

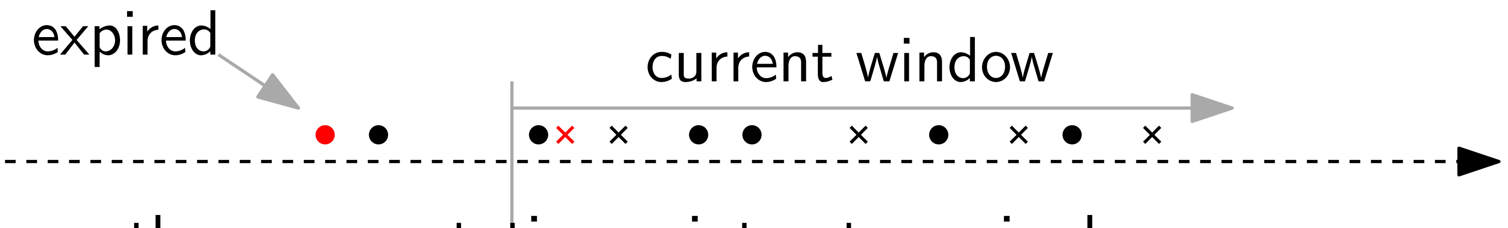
Extend to Sliding Window is non-trivial ...

What is the main difficulty?



Extend to Sliding Window is non-trivial ...

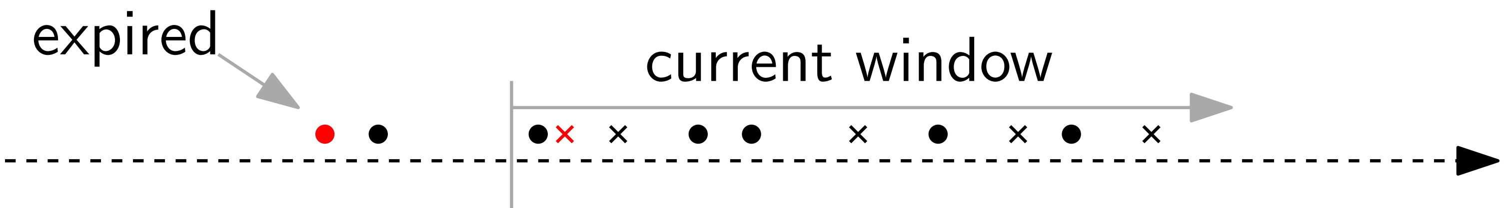
What is the main difficulty?



- the representative point gets expired
- but the group is still valid!

Extend to Sliding Window is non-trivial ...

What is the main difficulty?



- the representative point gets expired
- but the group is still valid!

the latest point in $G(u)$

in S^{acc}, maintain pairs,

(u, p)

the representative point

Any other issues?

Any other issues?

- pairs in S^{acc} get expired (unlike in IW)
- can not keep decreasing the sample rate

Any other issues?

- pairs in S^{acc} get expired (unlike in IW)
- can not keep decreasing the sample rate

Can we increase the sample rate?

Any other issues?

- pairs in S^{acc} get expired (unlike in IW)
- can not keep decreasing the sample rate

Can we increase the sample rate?

No, previous information is lost.

Any other issues?

- pairs in S^{acc} get expired (unlike in IW)
- can not keep decreasing the sample rate

Can we increase the sample rate?

No, previous information is lost.

Solution?

Any other issues?

- pairs in S^{acc} get expired (unlike in IW)
- can not keep decreasing the sample rate

Can we increase the sample rate?

No, previous information is lost.

Solution?

maintain a sampler (Level) for each possible sample rate:

$1,\ 1/2,\ 1/2^2,\ 1/2^3,\ 1/2^4 \ldots$

Any other issues?

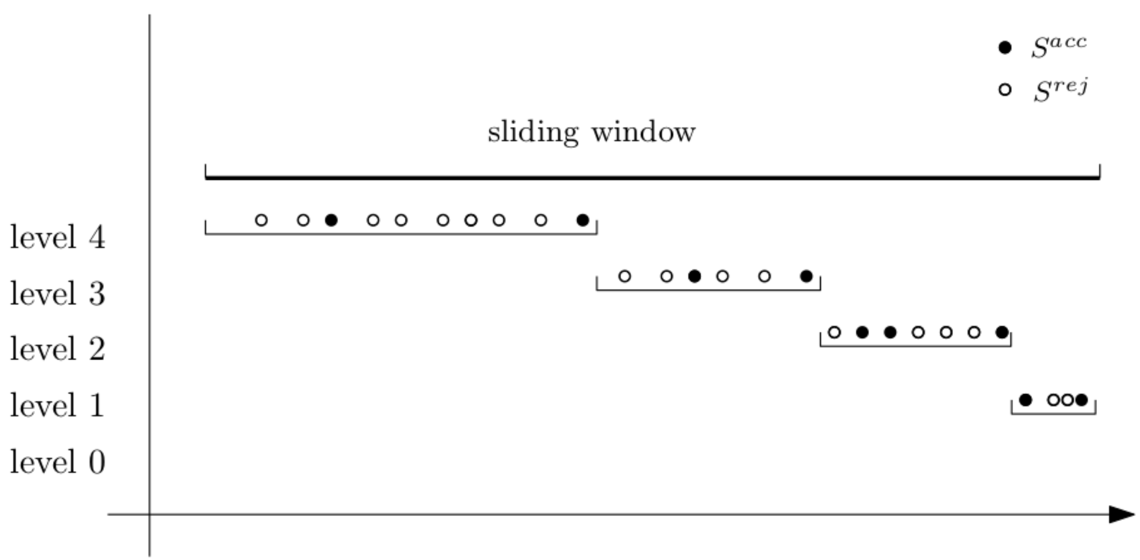
- pairs in S^{acc} get expired (unlike in IW)
- can not keep decreasing the sample rate

Can we increase the sample rate?

No, previous information is lost.

Solution?

maintain a sampler (Level) for each possible sample rate:
$1,\ 1/2,\ 1/2^2,\ 1/2^3,\ 1/2^4 \ldots$



- Level i samples cells with prob. $\frac{1}{2^i}$
- discard expired groups

Any other issues?

Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate

Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate

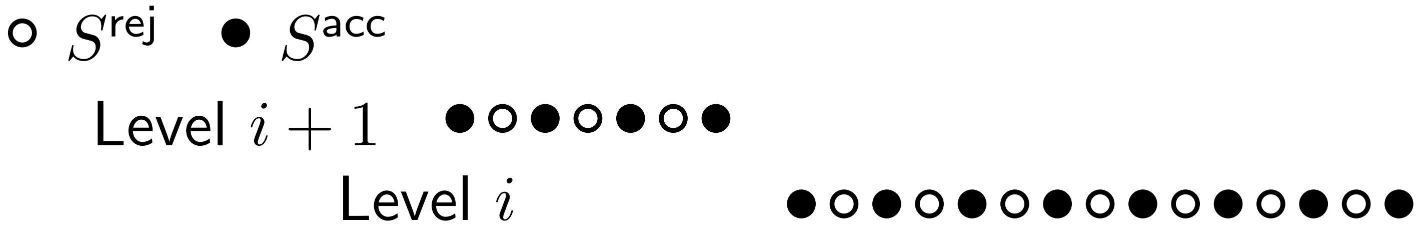
○ S^{rej} ● S^{acc}

Level $i+1$ ● ○ ● ○ ● ○ ●

Level i

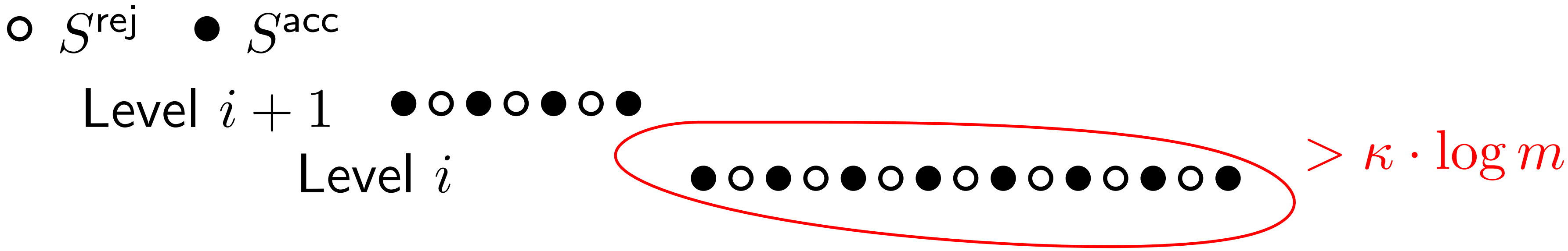
Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate



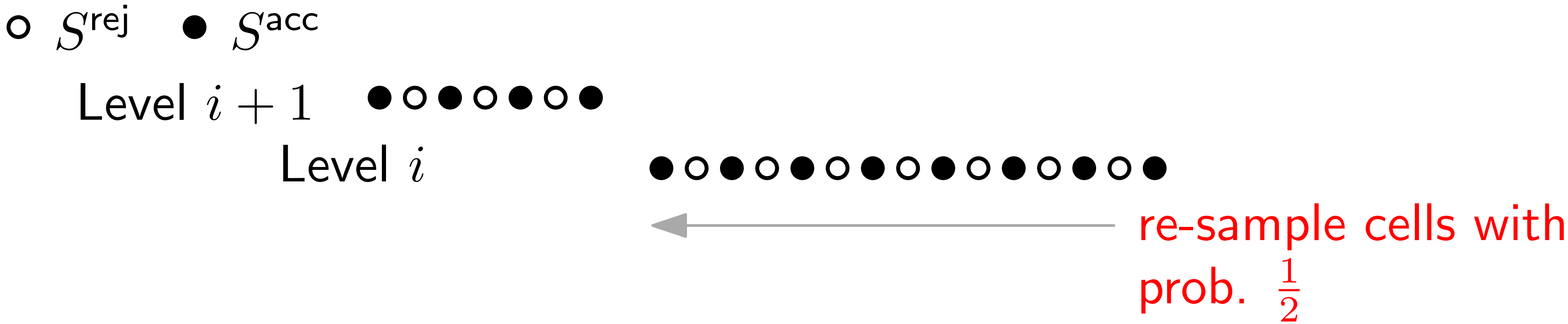
Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate



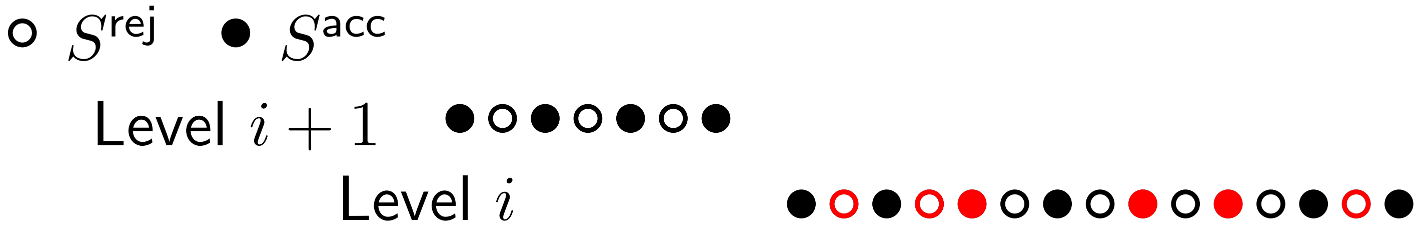
Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate



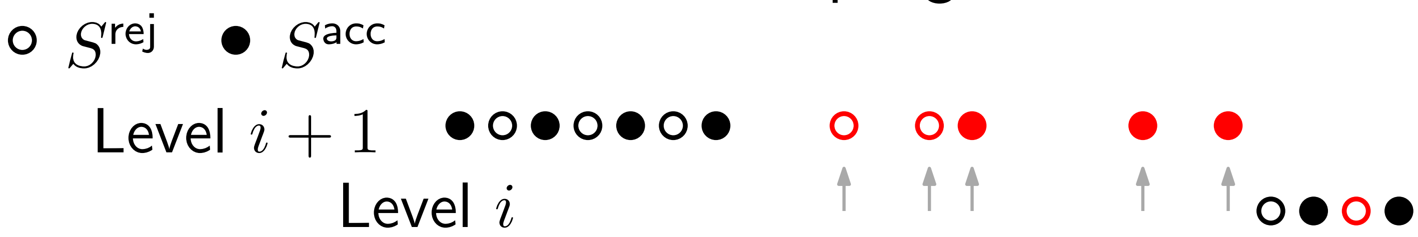
Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate



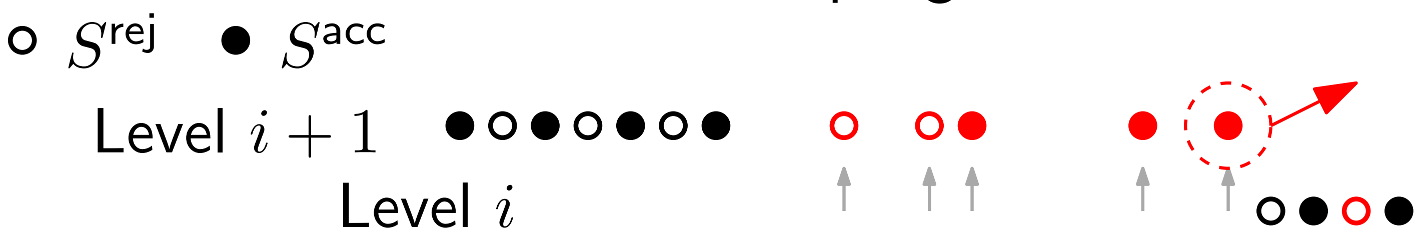
Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate



Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate

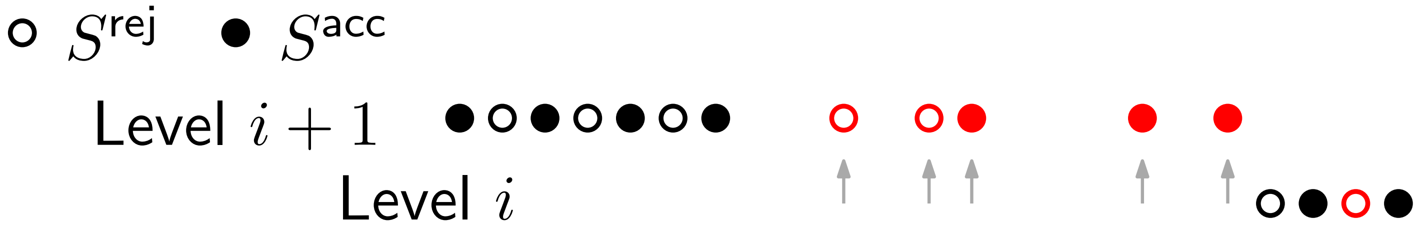


the last point must be in S^{acc}. Invariant during the split/merge process

- Level $i+1$ now may have $> \kappa \cdot \log m$
- do the re-sampling again in Level $i+1$
- this process may cascade to the top level
- each level actually samples from a disjoint subwindow

Any other issues?

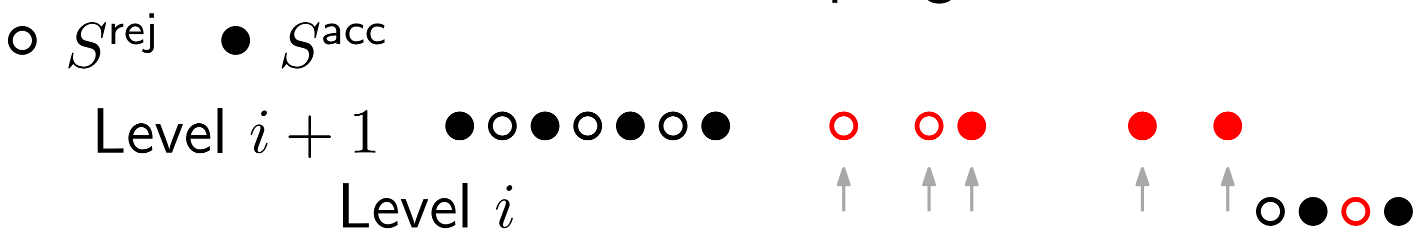
- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate



How to generate a sample at the point of query?

Any other issues?

- a level may sample $> \kappa \cdot \log m$
- can not do re-sampling because of the fixed rate



How to generate a sample at the point of query?

- Level T is the top non-empty level
- all groups in Level $T - i$ is re-sampled with prob. $\frac{1}{2^i}$
- union all sampled groups
- then return a sample uniformly at random

Theorems

For well-shaped datasets in $\mathbb{R}^{O(1)}$

- infinite window: use $O(\log m)$ space, $O(\log m)$ processing time
- sliding window: use $O(\log m \log w)$ space, $O(\log m \log w)$ amortized processing time

Theorems

For well-shaped datasets in $\mathbb{R}^{O(1)}$

- infinite window: use $O(\log m)$ space, $O(\log m)$ processing time
- sliding window: use $O(\log m \log w)$ space, $O(\log m \log w)$ amortized processing time

High Dimension $\mathbb{R}^d$?

If $\beta/\alpha > d^{1.5}$,

- infinite window: $O(d \log m)$ space and $O(d \log m)$ processing time
- sliding window: $O(d \log m \log w)$ space and $O(d \log m \log w)$ amertized processing time

Theorems

For well-shaped datasets in $\mathbb{R}^{O(1)}$

- infinite window: use $O(\log m)$ space, $O(\log m)$ processing time
- sliding window: use $O(\log m \log w)$ space, $O(\log m \log w)$ amortized processing time

High Dimension $\mathbb{R}^d$?

If $\beta/\alpha > d^{1.5}$,

- infinite window: $O(d \log m)$ space and $O(d \log m)$ processing time
- sliding window: $O(d \log m \log w)$ space and $O(d \log m \log w)$ amertized processing time

applying dimension reduction reduces d to $O(\log m)$

General datasets ...

General datasets ...

- even the partition of S is not well-defined
- What is our goal?

General datasets ...

even the partition of S is not well-defined

What is our goal?

Definition (F_0): minimum cardinality partition of S, $G_1, \ldots, G_{F_0}$, so that all points in the same group has distance $< \alpha$

General datasets ...

even the partition of S is not well-defined

What is our goal?

Definition (F_0): minimum cardinality partition of S, $G_1, \ldots, G_{F_0}$, so that all points in the same group has distance $< \alpha$

Goal: Robust ℓ_0 sampling returns q,

$$\forall p \in S, \;\; \Pr[q \in \mathsf{Ball}(p, \alpha) \cap S] = \Theta(1/F_0).$$

General datasets ...

even the partition of S is not well-defined

What is our goal?

Definition (F_0): minimum cardinality partition of S, $G_1, \ldots, G_{F_0}$, so that all points in the same group has distance $< \alpha$

Goal: Robust ℓ_0 sampling returns q,

$$\forall p \in S, \;\; \Pr[q \in \mathsf{Ball}(p, \alpha) \cap S] = \Theta(1/F_0).$$

consistent with the well-shaped case

General datasets ...

even the partition of S is not well-defined

What is our goal?

Definition (F_0): minimum cardinality partition of S, $G_1, \ldots, G_{F_0}$, so that all points in the same group has distance $< \alpha$

Goal: Robust ℓ_0 sampling returns q,

$$\forall p \in S, \;\; \Pr[q \in \mathsf{Ball}(p, \alpha) \cap S] = \Theta(1/F_0).$$

consistent with the well-shaped case

our algorithms in well-shaped dataset can achieve this goal in $\mathbb{R}^{O(1)}$

Open Questions

Open Questions

- How to weaken the requirement $\beta/\alpha > d^{1.5}$ in $\mathbb{R}^d$?

Open Questions

- How to weaken the requirement $\beta/\alpha > d^{1.5}$ in $\mathbb{R}^d$?

- Other formulations for robust ℓ_0-sampling on general data sets?

Open Questions

- How to weaken the requirement $\beta/\alpha > d^{1.5}$ in $\mathbb{R}^d$?

- Other formulations for robust ℓ_0-sampling on general data sets?

- How to extend to other metric spaces?

Questions?

Thank you!