Finding Monotone Patterns in Sublinear Time

Erik Waingarten (Columbia University)

Clément Canonne (Stanford University)
Omri Ben-Eliezer (Tel-Aviv University)
Shoham Letzter (ETH Zurich)

Testing Monotonicity of an Array: *Sortedness*

1	3	2	3	6	7	1	9	4	2	5	6	7

Given query access to an unknown $f : [n] \to \mathbb{R}$ and a parameter $\varepsilon > 0$:

- If f monotone, accept w.p. $> 2/3$;
- If f is ε-far from monotone, reject w.p. $> 2/3$;

Minimum query complexity in terms of n and ε?

Testing Monotonicity of an Array: *Sortedness*

1	3	2	3	6	7	1	9	4	2	5	6	7

Given query access to an unknown $f : [n] \to \mathbb{R}$ and a parameter $\varepsilon > 0$:

- If f monotone, accept w.p. $> 2/3$;
- If f is ε-far from monotone, reject w.p. $> 2/3$;

> Minimum query complexity in terms of n and ε?

- If f is ε-far from monotone, find evidence:
  - $i, j \in [n]$ where $i < j$ and $f(i) > f(j)$.

Testing Monotonicity of an Array: *Sortedness*

1	3	2	3	6	7	1	9	4	2	5	6	7

Given query access to an unknown $f: [n] \to \mathbb{R}$ and a parameter $\varepsilon > 0$:

- If f monotone, accept w.p. $> 2/3$;
- If f is ε-far from monotone, reject w.p. $> 2/3$;

> Minimum query complexity in terms of n and ε?

- If f is ε-far from monotone, find evidence:
  - $i, j \in [n]$ where $i < j$ and $f(i) > f(j)$. *one-sided* error.

ε-Far-From-Monotone Sequences

- f is ε-far from monotone: for any monotone $g\colon [n] \to \mathbb{R}$,

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{f(i) \neq g(i)\} \geq \varepsilon.$$

ε-Far-From-Monotone Sequences

- f is ε-far from monotone: for any monotone $g \colon [n] \to \mathbb{R}$,

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{f(i) \neq g(i)\} \geq \varepsilon.$$

Lemma

Let $f : [n] \to \mathbb{R}$ be ε-far from monotone. There exists set of disjoint pairs,

$$T = \left\{(i,j) \in [n]^2 : i < j \text{ and } f(i) > f(j)\right\}$$

of size $|T| \geq \varepsilon n/2$.

Monotonicity Testing

Theorem (Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99)

There exists a non-adaptive, one-sided algorithm for testing monotonicity of $f : [n] \to \mathbb{R}$ *making* $O((\log n)/\varepsilon)$ *queries.*

Monotonicity Testing

Theorem (Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99)

There exists a non-adaptive, one-sided algorithm for testing monotonicity of $f : [n] \to \mathbb{R}$ making $O((\log n)/\varepsilon)$ queries.

- $\Omega((\log n)/\varepsilon)$ queries needed for non-adaptive algorithms.

Monotonicity Testing

Theorem (Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99)

There exists a non-adaptive, one-sided algorithm for testing monotonicity of $f : [n] \to \mathbb{R}$ making $O((\log n)/\varepsilon)$ queries.

- $\Omega((\log n)/\varepsilon)$ queries needed for non-adaptive algorithms.
- $\Omega((\log n)/\varepsilon)$ queries needed for adaptive algorithms too! [Fischer 04].

Testing Forbidden Order Patterns

[Newman, Rabinovich, Rajendraprasad, Sohler 17]

Testing Forbidden Order Patterns

[Newman, Rabinovich, Rajendraprasad, Sohler 17]

Definition

Let $k \in \mathbb{N}$ and $\pi = (\pi_1, \ldots, \pi_k)$ be a permutation of $[k]$. Given $f : [n] \to \mathbb{R}$, the k-tuple $(i_1, \ldots, i_k)$ has *order pattern* π if:

$$f(i_\ell) < f(i_m) \text{ whenever } \pi_\ell < \pi_m.$$

Testing Forbidden Order Patterns

[Newman, Rabinovich, Rajendraprasad, Sohler 17]

Definition

Let $k \in \mathbb{N}$ and $\pi = (\pi_1, \ldots, \pi_k)$ be a permutation of $[k]$. Given $f : [n] \to \mathbb{R}$, the k-tuple $(i_1, \ldots, i_k)$ has *order pattern* π if:

$$f(i_\ell) < f(i_m) \text{ whenever } \pi_\ell < \pi_m.$$

If $f : [n] \to \mathbb{R}$ is ε-far from π-free, then there exists $T \subset [n]^k$ of disjoint violating k-tuples $(i_1, \ldots, i_k)$ with order pattern π of size at least $\varepsilon n/k$.

Testing Forbidden Order Patterns

[Newman, Rabinovich, Rajendraprasad, Sohler 17]

Definition

Let $k \in \mathbb{N}$ and $\pi = (\pi_1, \ldots, \pi_k)$ be a permutation of $[k]$. Given $f : [n] \to \mathbb{R}$, the k-tuple $(i_1, \ldots, i_k)$ has *order pattern* π if:

$$f(i_\ell) < f(i_m) \text{ whenever } \pi_\ell < \pi_m.$$

If $f : [n] \to \mathbb{R}$ is ε-far from π-free, then there exists $T \subset [n]^k$ of disjoint violating k-tuples $(i_1, \ldots, i_k)$ with order pattern π of size at least $\varepsilon n / k$.

For fixed k and π, query complexity of testing π-freeness?

Testing Forbidden Order Patterns

[Newman, Rabinovich, Rajendraprasad, Sohler 17]

Definition

Let $k \in \mathbb{N}$ and $\pi = (\pi_1, \ldots, \pi_k)$ be a permutation of $[k]$. Given $f : [n] \to \mathbb{R}$, the k-tuple $(i_1, \ldots, i_k)$ has *order pattern* π if:

$$f(i_\ell) < f(i_m) \text{ whenever } \pi_\ell < \pi_m.$$

If $f : [n] \to \mathbb{R}$ is ε-far from π-free, then there exists $T \subset [n]^k$ of disjoint violating k-tuples $(i_1, \ldots, i_k)$ with order pattern π of size at least $\varepsilon n / k$.

For fixed k and π, query complexity of testing π-freeness?

- Some sublinear in n upper bounds for general π.

Testing Forbidden Order Patterns

[Newman, Rabinovich, Rajendraprasad, Sohler 17]

Definition

Let $k \in \mathbb{N}$ and $\pi = (\pi_1, \ldots, \pi_k)$ be a permutation of $[k]$. Given $f : [n] \to \mathbb{R}$, the k-tuple $(i_1, \ldots, i_k)$ has *order pattern* π if:

$$f(i_\ell) < f(i_m) \text{ whenever } \pi_\ell < \pi_m.$$

If $f : [n] \to \mathbb{R}$ is ε-far from π-free, then there exists $T \subset [n]^k$ of disjoint violating k-tuples $(i_1, \ldots, i_k)$ with order pattern π of size at least $\varepsilon n/k$.

For fixed k and π, query complexity of testing π-freeness?

- Some sublinear in n upper bounds for general π.
- $\pi = (132)$ requires $\Omega(\sqrt{n})$ queries for non-adaptive, one-sided algorithms.

Testing Forbidden Order Patterns

[Newman, Rabinovich, Rajendraprasad, Sohler 17]

Definition

Let $k \in \mathbb{N}$ and $\pi = (\pi_1, \ldots, \pi_k)$ be a permutation of $[k]$. Given $f\colon [n] \to \mathbb{R}$, the k-tuple $(i_1, \ldots, i_k)$ has *order pattern* π if:

$$f(i_\ell) < f(i_m) \text{ whenever } \pi_\ell < \pi_m.$$

If $f\colon [n] \to \mathbb{R}$ is ε-far from π-free, then there exists $T \subset [n]^k$ of disjoint violating k-tuples $(i_1, \ldots, i_k)$ with order pattern π of size at least $\varepsilon n/k$.

For fixed k and π, query complexity of testing π-freeness?

- Some sublinear in n upper bounds for general π.
- $\pi = (132)$ requires $\Omega(\sqrt{n})$ queries for non-adaptive, one-sided algorithms.
- [Ben-Eliezer, Canonne 18] Many π have complexity a $n^{1-1/(k-\Theta(1))}$.

Finding Monotone Patterns: $\pi = (12 \ldots k)$

Finding Monotone Patterns: $\pi = (12 \ldots k)$

Given query access to $f\colon [n] \to \mathbb{R}$ and a parameter $\varepsilon > 0$ where:

Finding Monotone Patterns: $\pi = (12\dots k)$

Given query access to $f\colon [n] \to \mathbb{R}$ and a parameter $\varepsilon > 0$ where:

- There exists $T \subset [n]^k$ of disjoint violating k-tuples

$$T = \{(i_1, \dots, i_k) : i_1 < \cdots < i_k \text{ and } f(i_1) < \cdots < f(i_k)\}$$

of size $|T| \geq \varepsilon n / k$.

Finding Monotone Patterns: $\pi = (12 \ldots k)$

Given query access to $f\colon [n] \to \mathbb{R}$ and a parameter $\varepsilon > 0$ where:

- There exists $T \subset [n]^k$ of disjoint violating k-tuples

$$T = \{(i_1, \ldots, i_k) : i_1 < \cdots < i_k \text{ and } f(i_1) < \cdots < f(i_k)\}$$

  of size $|T| \geq \varepsilon n / k$.
- Find $i_1 < \cdots < i_k$ where $f(i_1) < \cdots < f(i_k)$.

Finding Monotone Patterns: $\pi = (12 \ldots k)$

Given query access to $f : [n] \to \mathbb{R}$ and a parameter $\varepsilon > 0$ where:

- There exists $T \subset [n]^k$ of disjoint violating k-tuples

  $$T = \{(i_1, \ldots, i_k) : i_1 < \cdots < i_k \text{ and } f(i_1) < \cdots < f(i_k)\}$$

  of size $|T| \geq \varepsilon n / k$.
- Find $i_1 < \cdots < i_k$ where $f(i_1) < \cdots < f(i_k)$.

Theorem (NRRS17)

There is a non-adaptive algorithm with query complexity $((\log n)/\varepsilon)^{O(k^2)}$.

Finding Monotone Patterns: $\pi = (12 \ldots k)$

Finding Monotone Patterns: $\pi = (12\ldots k)$

Theorem (Upper bound)

For $k \in \mathbb{N}$, there exists a non-adaptive algorithm with query complexity

$$(\log n)^{\lfloor \log_2 k \rfloor} \cdot \mathrm{poly}(1/\varepsilon).$$

Finding Monotone Patterns: $\pi = (12 \dots k)$

Theorem (Upper bound)

For $k \in \mathbb{N}$, there exists a non-adaptive algorithm with query complexity

$$(\log n)^{\lfloor \log_2 k \rfloor} \cdot \mathrm{poly}(1/\varepsilon).$$

Theorem (Lower bound)

Any non-adaptive algorithm needs to make $\Omega\left((\log n)^{\lfloor \log_2 k \rfloor}\right)$ queries.

The Case of $k = 2$

The Case of $k = 2$

Let $f : [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i, j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

The Case of $k = 2$

Let $f\colon [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i,j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

The Case of $k = 2$

Let $f \colon [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i,j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\ell \sim [n]$ uniformly and repeat the following for t iterations:

The Case of $k = 2$

Let $f\colon [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i,j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\boldsymbol{\ell} \sim [n]$ uniformly and repeat the following for t iterations:
  - Sample $\boldsymbol{s} \sim \{0, \ldots, \log_2 n\}$ and $\boldsymbol{i} \sim [\boldsymbol{\ell} - 2^{\boldsymbol{s}}, \boldsymbol{\ell}]$ and query $f(\boldsymbol{i})$.

The Case of $k = 2$

Let $f : [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i, j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\boldsymbol{\ell} \sim [n]$ uniformly and repeat the following for t iterations:
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{i} \sim [\boldsymbol{\ell} - 2^{\boldsymbol{s}}, \boldsymbol{\ell}]$ and query $f(\boldsymbol{i})$.
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{j} \sim [\boldsymbol{\ell}, \boldsymbol{\ell} + 2^{\boldsymbol{s}}]$ and query $f(\boldsymbol{j})$.

The Case of $k = 2$

Let $f\colon [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i,j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\boldsymbol{\ell} \sim [n]$ uniformly and repeat the following for t iterations:
  - Sample $\boldsymbol{s} \sim \{0, \ldots, \log_2 n\}$ and $\boldsymbol{i} \sim [\boldsymbol{\ell} - 2^{\boldsymbol{s}}, \boldsymbol{\ell}]$ and query $f(\boldsymbol{i})$.
  - Sample $\boldsymbol{s} \sim \{0, \ldots, \log_2 n\}$ and $\boldsymbol{j} \sim [\boldsymbol{\ell}, \boldsymbol{\ell} + 2^{\boldsymbol{s}}]$ and query $f(\boldsymbol{j})$.

$f =$ 

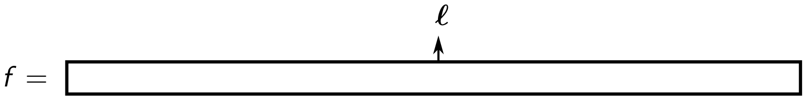
The Case of $k = 2$

Let $f \colon [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i,j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\boldsymbol{\ell} \sim [n]$ uniformly and repeat the following for t iterations:
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{i} \sim [\boldsymbol{\ell} - 2^{\boldsymbol{s}}, \boldsymbol{\ell}]$ and query $f(\boldsymbol{i})$.
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{j} \sim [\boldsymbol{\ell}, \boldsymbol{\ell} + 2^{\boldsymbol{s}}]$ and query $f(\boldsymbol{j})$.



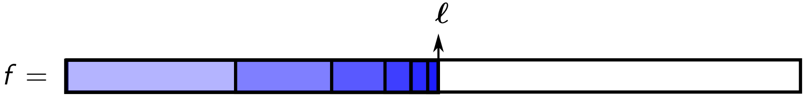
The Case of $k = 2$

Let $f\colon [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i,j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\boldsymbol{\ell} \sim [n]$ uniformly and repeat the following for t iterations:
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{i} \sim [\boldsymbol{\ell} - 2^{\boldsymbol{s}}, \boldsymbol{\ell}]$ and query $f(\boldsymbol{i})$.
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{j} \sim [\boldsymbol{\ell}, \boldsymbol{\ell} + 2^{\boldsymbol{s}}]$ and query $f(\boldsymbol{j})$.



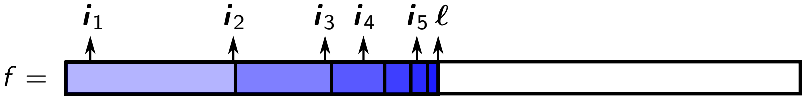
The Case of $k = 2$

Let $f : [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i, j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\boldsymbol{\ell} \sim [n]$ uniformly and repeat the following for t iterations:
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{i} \sim [\boldsymbol{\ell} - 2^{\boldsymbol{s}}, \boldsymbol{\ell}]$ and query $f(\boldsymbol{i})$.
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{j} \sim [\boldsymbol{\ell}, \boldsymbol{\ell} + 2^{\boldsymbol{s}}]$ and query $f(\boldsymbol{j})$.



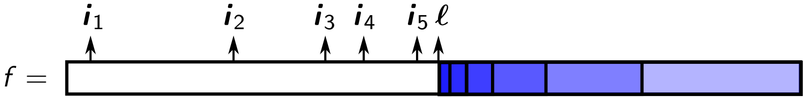
The Case of $k = 2$

Let $f\colon [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i,j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\boldsymbol{\ell} \sim [n]$ uniformly and repeat the following for t iterations:
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{i} \sim [\boldsymbol{\ell} - 2^{\boldsymbol{s}}, \boldsymbol{\ell}]$ and query $f(\boldsymbol{i})$.
  - Sample $\boldsymbol{s} \sim \{0, \dots, \log_2 n\}$ and $\boldsymbol{j} \sim [\boldsymbol{\ell}, \boldsymbol{\ell} + 2^{\boldsymbol{s}}]$ and query $f(\boldsymbol{j})$.



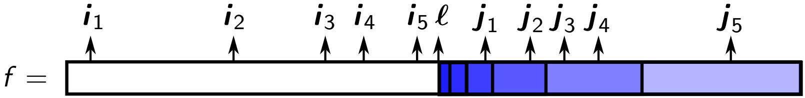
The Case of $k = 2$

Let $f\colon [n] \to \mathbb{R}$ and disjoint subset of pairs T of size $\varepsilon n/2$ with

$$T = \{(i,j) \in [n]^2 : i < j \text{ and } f(i) < f(j)\}.$$

[Ergün, Kannan, Kumar, Rubinfeld, Viswanathan 99] Find increasing pair:

- Sample $\boldsymbol{\ell} \sim [n]$ uniformly and repeat the following for t iterations:
  - Sample $\boldsymbol{s} \sim \{0, \ldots, \log_2 n\}$ and $\boldsymbol{i} \sim [\boldsymbol{\ell} - 2^{\boldsymbol{s}}, \boldsymbol{\ell}]$ and query $f(\boldsymbol{i})$.
  - Sample $\boldsymbol{s} \sim \{0, \ldots, \log_2 n\}$ and $\boldsymbol{j} \sim [\boldsymbol{\ell}, \boldsymbol{\ell} + 2^{\boldsymbol{s}}]$ and query $f(\boldsymbol{j})$.

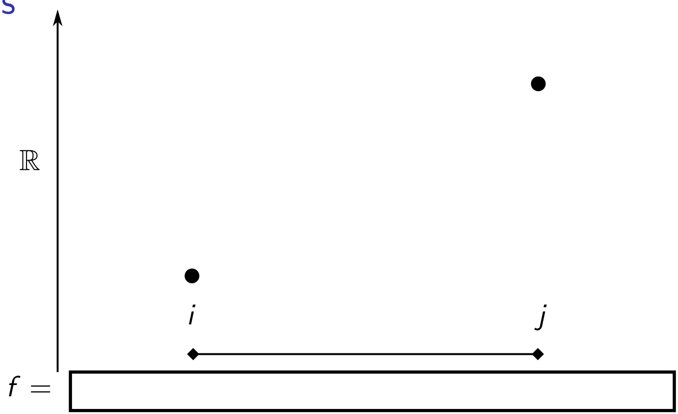


Analysis

Analysis

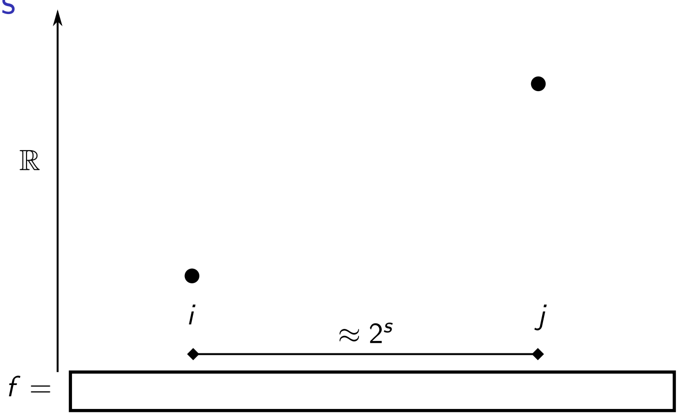
- Let T be set of monotone pairs.

Analysis



- Let T be set of monotone pairs.
- $(i, j) \in T$.

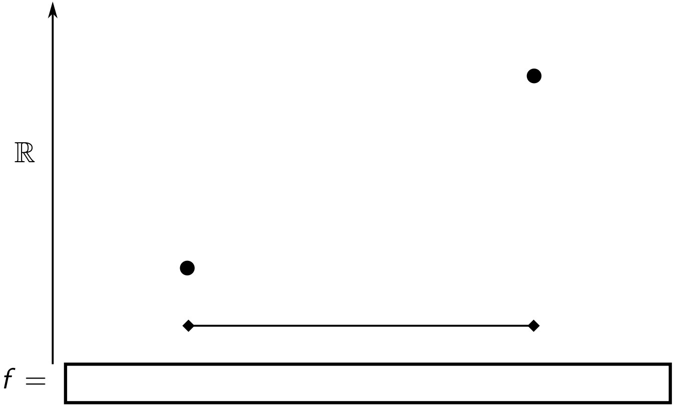
Analysis



- Let T be set of monotone pairs.
- $(i, j) \in T$. (i, j) *scale s* when $j - i \approx 2^s$.

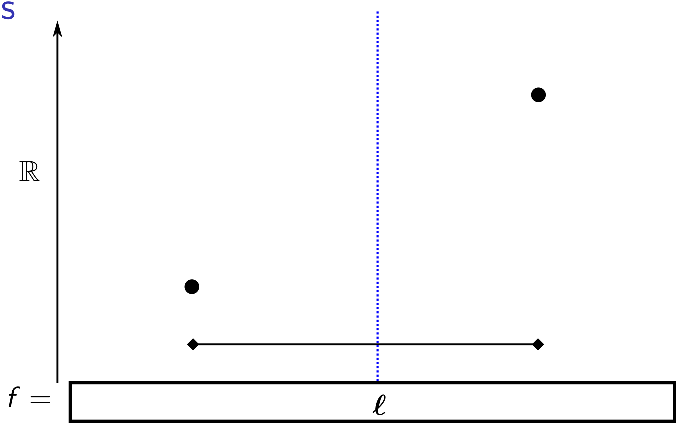
$$\Pr_{\ell \sim [n]} [i \leq \ell \leq j] \approx \frac{2^s}{n}.$$

Analysis



$$\mathop{\mathbb{E}}_{\ell \sim [n]} \left[\sum_{s=1}^{\log n} \frac{\#\ (i,j) \text{ scale } s \text{ cut}}{2^{s+1}} \right] \gtrsim \varepsilon.$$

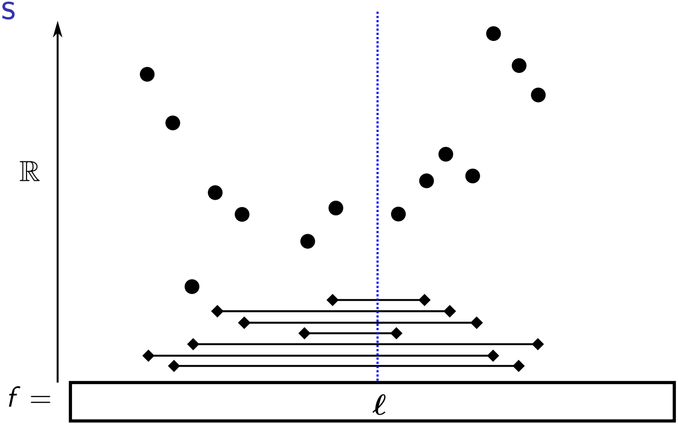
Analysis



- Sample ℓ such that:

$$\sum_{s=1}^{\log n} \frac{\#\ (i,j) \text{ scale } s \text{ cut}}{2^{s+1}} = \sum_{s=1}^{\log n} (\text{density of } T \text{ at } 2^{s+1}) \gtrsim \varepsilon.$$

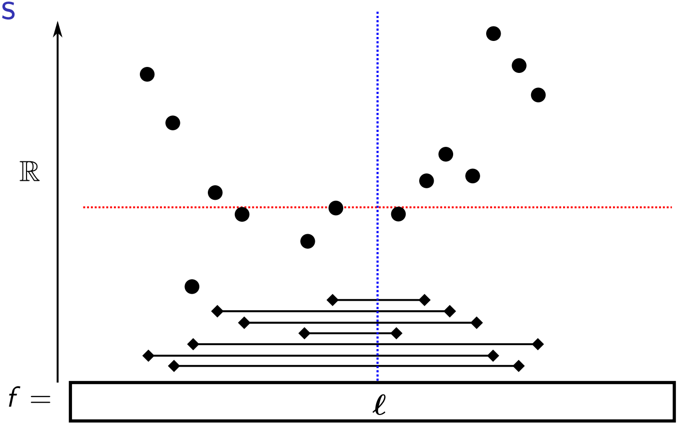
Analysis



- Sample ℓ such that:

$$\sum_{s=1}^{\log n} \frac{\#\ (i,j)\text{ scale } s \text{ cut}}{2^{s+1}} = \sum_{s=1}^{\log n} (\text{density of } T \text{ at } 2^{s+1}) \gtrsim \varepsilon.$$

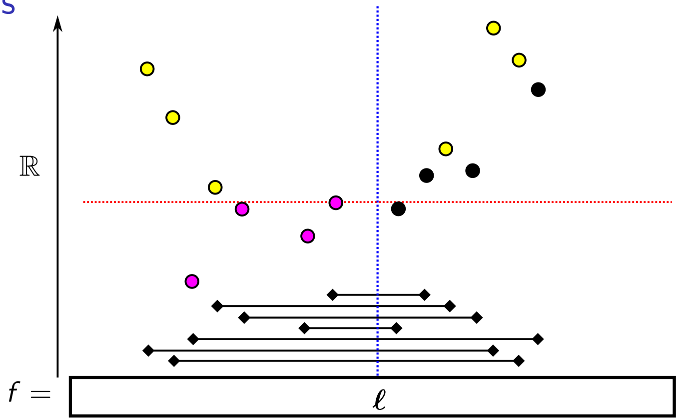
Analysis



- Sample ℓ such that:

$$\sum_{s=1}^{\log n} \frac{\#\ (i,j) \text{ scale } s \text{ cut}}{2^{s+1}} = \sum_{s=1}^{\log n} (\text{density of } T \text{ at } 2^{s+1}) \gtrsim \varepsilon.$$

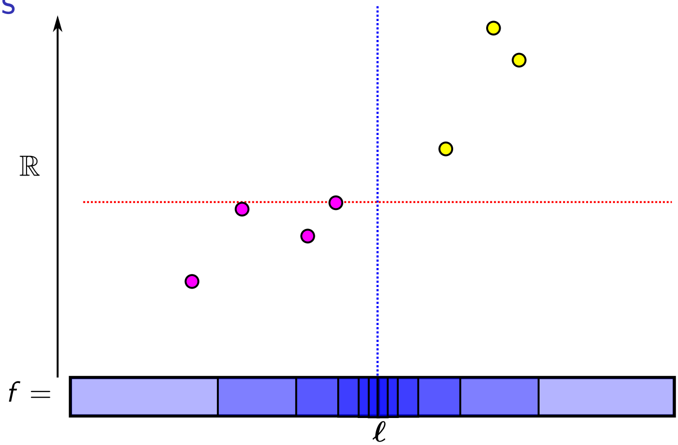
Analysis



- Sample ℓ such that:

$$\sum_{s=1}^{\log n} \frac{\#\ (i,j) \text{ scale } s \text{ cut}}{2^{s+1}} = \sum_{s=1}^{\log n} (\text{density of } T \text{ at } 2^{s+1}) \gtrsim \varepsilon.$$

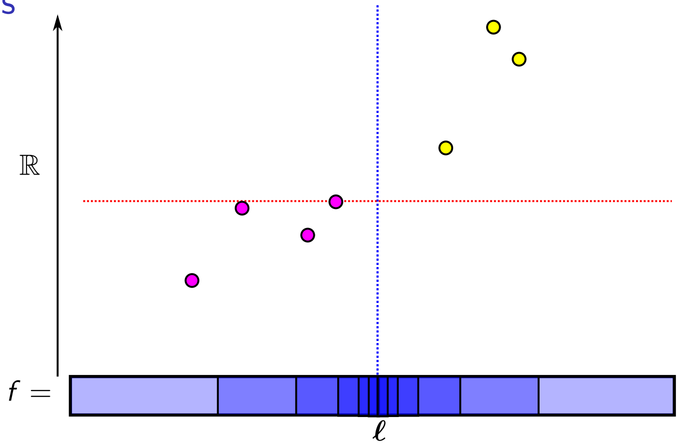
Analysis



- Sample ℓ such that:

$$\sum_{s=1}^{\log n} \frac{\#\ (i,j) \text{ scale } s \text{ cut}}{2^{s+1}} = \sum_{s=1}^{\log n} (\text{density of } T \text{ at } 2^{s+1}) \gtrsim \varepsilon.$$

Analysis



- Density δ_s^- and δ_s^+: $\sum_{s=1}^{\log n} \delta_s^- \geq \varepsilon$ and $\sum_{s=1}^{\log n} \delta_s^+ \geq \varepsilon$.

$$\Pr_{\substack{\boldsymbol{i}_1,\ldots,\boldsymbol{i}_t \\ \boldsymbol{j}_1,\ldots,\boldsymbol{j}_t}} [\text{avoid pair}] \leq \left(\frac{1}{\log n} \sum_{s=1}^{\log n} (1 - \delta_s^-) \right)^t + \left(\frac{1}{\log n} \sum_{s=1}^{\log n} (1 - \delta_s^+) \right)^t \leq \frac{1}{3}.$$

Monotone patterns for $k > 2$?

New algorithm for finding $(12 \ldots k)$ patterns:

Monotone patterns for $k > 2$?

New algorithm for finding $(12 \dots k)$ patterns:

- Round 1: Sample $O(1/\varepsilon)$ indices from $[n]$, include them in a set $\boldsymbol{A}$.

Monotone patterns for $k > 2$?

New algorithm for finding $(12 \ldots k)$ patterns:

- Round 1: Sample $O(1/\varepsilon)$ indices from $[n]$, include them in a set $\boldsymbol{A}$.
- Round r, $2 \leq r \leq \lfloor \log_2 k \rfloor + 1$: For each $i \in \boldsymbol{A}$ and $s \in \{1, \ldots, \log n\}$, sample $O(1/\varepsilon)$ indices from $[i - 2^s, i + 2^s]$ and include them in $\boldsymbol{A}$.

Monotone patterns for $k > 2$?

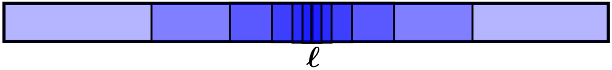
New algorithm for finding $(12\ldots k)$ patterns:

- Round 1: Sample $O(1/\varepsilon)$ indices from $[n]$, include them in a set $\boldsymbol{A}$.
- Round r, $2 \leq r \leq \lfloor \log_2 k \rfloor + 1$: For each $i \in \boldsymbol{A}$ and $s \in \{1, \ldots, \log n\}$, sample $O(1/\varepsilon)$ indices from $[i - 2^s, i + 2^s]$ and include them in $\boldsymbol{A}$.
- Query $f(i)$ for all $i \in \boldsymbol{A}$.

Monotone patterns for $k > 2$?

New algorithm for finding $(12 \dots k)$ patterns:

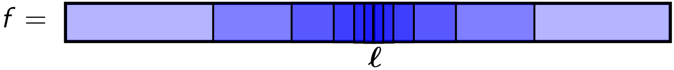
- Round 1: Sample $O(1/\varepsilon)$ indices from $[n]$, include them in a set $\boldsymbol{A}$.
- Round r, $2 \leq r \leq \lfloor \log_2 k \rfloor + 1$: For each $i \in \boldsymbol{A}$ and $s \in \{1, \dots, \log n\}$, sample $O(1/\varepsilon)$ indices from $[i - 2^s, i + 2^s]$ and include them in $\boldsymbol{A}$.
- Query $f(i)$ for all $i \in \boldsymbol{A}$.

$f =$ 

Monotone patterns for $k > 2$?

New algorithm for finding $(12\ldots k)$ patterns:

- Round 1: Sample $O(1/\varepsilon)$ indices from $[n]$, include them in a set $\boldsymbol{A}$.
- Round r, $2 \le r \le \lfloor \log_2 k \rfloor + 1$: For each $i \in \boldsymbol{A}$ and $s \in \{1, \ldots, \log n\}$, sample $O(1/\varepsilon)$ indices from $[i - 2^s, i + 2^s]$ and include them in $\boldsymbol{A}$.
- Query $f(i)$ for all $i \in \boldsymbol{A}$.

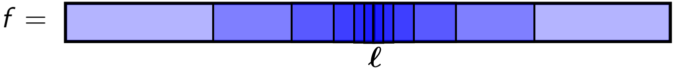


Query Complexity: $\quad O(1/\varepsilon) \cdot (O(\log n/\varepsilon))^{\lfloor \log_2 k \rfloor}$.

Monotone patterns for $k > 2$?

New algorithm for finding $(12\dots k)$ patterns:

- Round 1: Sample $O(1/\varepsilon)$ indices from $[n]$, include them in a set $\boldsymbol{A}$.
- Round r, $2 \le r \le \lfloor \log_2 k \rfloor + 1$: For each $i \in \boldsymbol{A}$ and $s \in \{1, \dots, \log n\}$, sample $O(1/\varepsilon)$ indices from $[i - 2^s, i + 2^s]$ and include them in $\boldsymbol{A}$.
- Query $f(i)$ for all $i \in \boldsymbol{A}$.



Query Complexity: $\quad O(1/\varepsilon) \cdot (O(\log n/\varepsilon))^{\lfloor \log_2 k \rfloor}$.

Theorem

Suppose $f : [n] \to \mathbb{R}$ contains $\varepsilon n/k$ disjoint $(12\dots k)$-patterns, algorithm finds one w.p $\ge 2/3$.

Open Problems

Open Problems

- More efficient tests for *two-sided* error testing for $(12\ldots k)$-freeness.

Open Problems

- More efficient tests for *two-sided* error testing for $(12 \ldots k)$-freeness.
- Adaptive algorithms for finding π-patterns?

Open Problems

- More efficient tests for *two-sided* error testing for $(12\ldots k)$-freeness.
- Adaptive algorithms for finding π-patterns?
  - Possibly $\mathrm{poly}(\log n)$ adaptive queries for any pattern π?

Open Problems

- More efficient tests for *two-sided* error testing for $(12\ldots k)$-freeness.
- Adaptive algorithms for finding π-patterns?
  - Possibly $\mathrm{poly}(\log n)$ adaptive queries for any pattern π?

Thanks!