Property Testing Tutorial

Sofya Raskhodnikova
Penn State University

Big Data

"Why Gramma, what big data you have!"

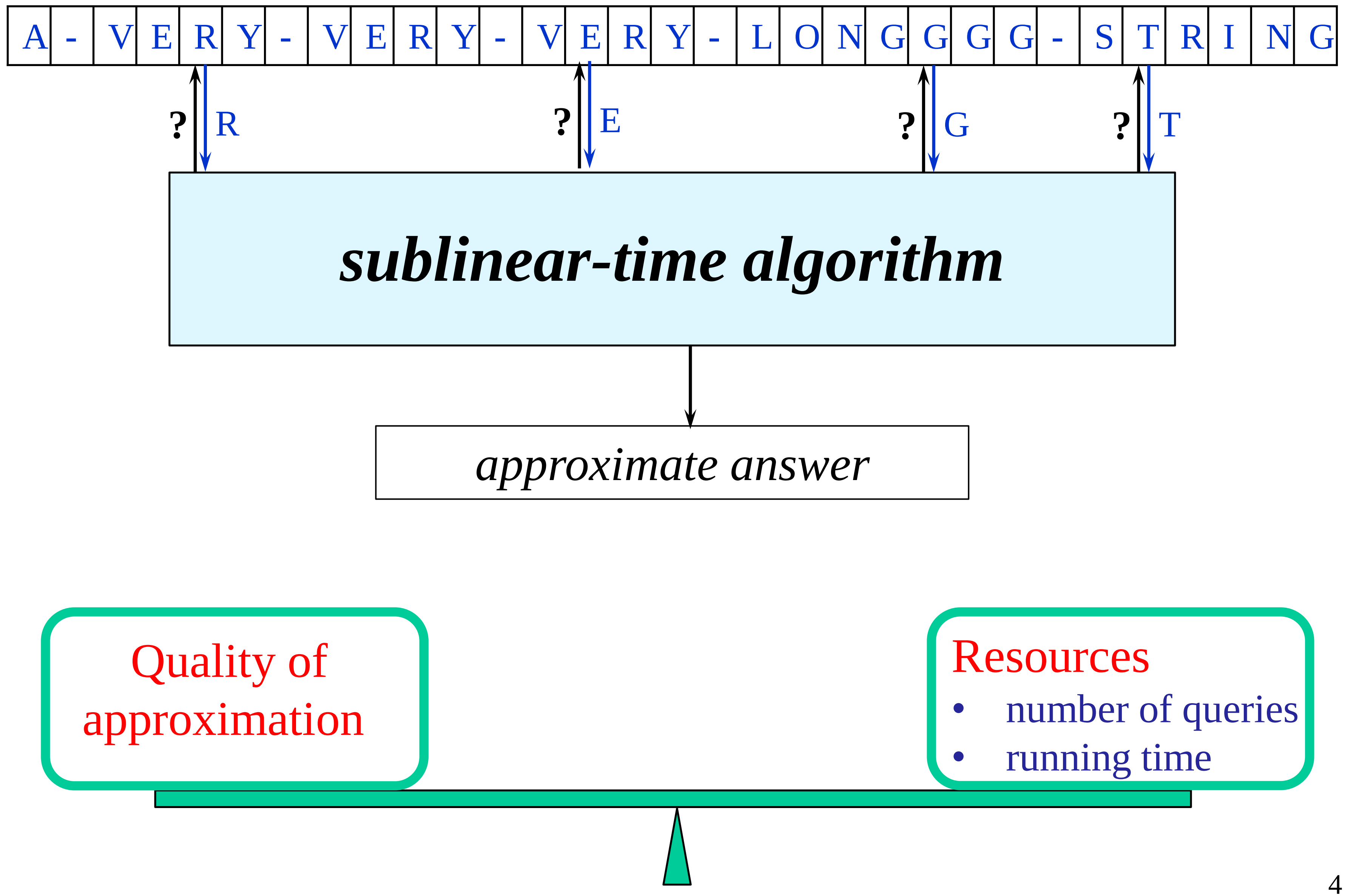
Do We Have To Read All the Data?

- What can an algorithm compute if it
  - reads only a **tiny** portion of the data?
  - runs in **sublinear** time?

- For most interesting problems sublinear-time algorithms must be
  - approximate
  - randomized

Image source: http://apandre.wordpress.com/2011/01/16/bigdata/

A Sublinear-Time Algorithm



Types of Approximation

- Classical approximation
  - need to compute a value
  - ➢ **output should be close to the desired value**
- Property testing
  - need to answer YES or NO
  - ➢ **intuition: only require correct answers on two sets of instances that are very different from each other**

Property Testing: YES/NO Questions

Does the input satisfy some property? (YES/NO)

"in the ballpark" vs. "out of the ballpark"

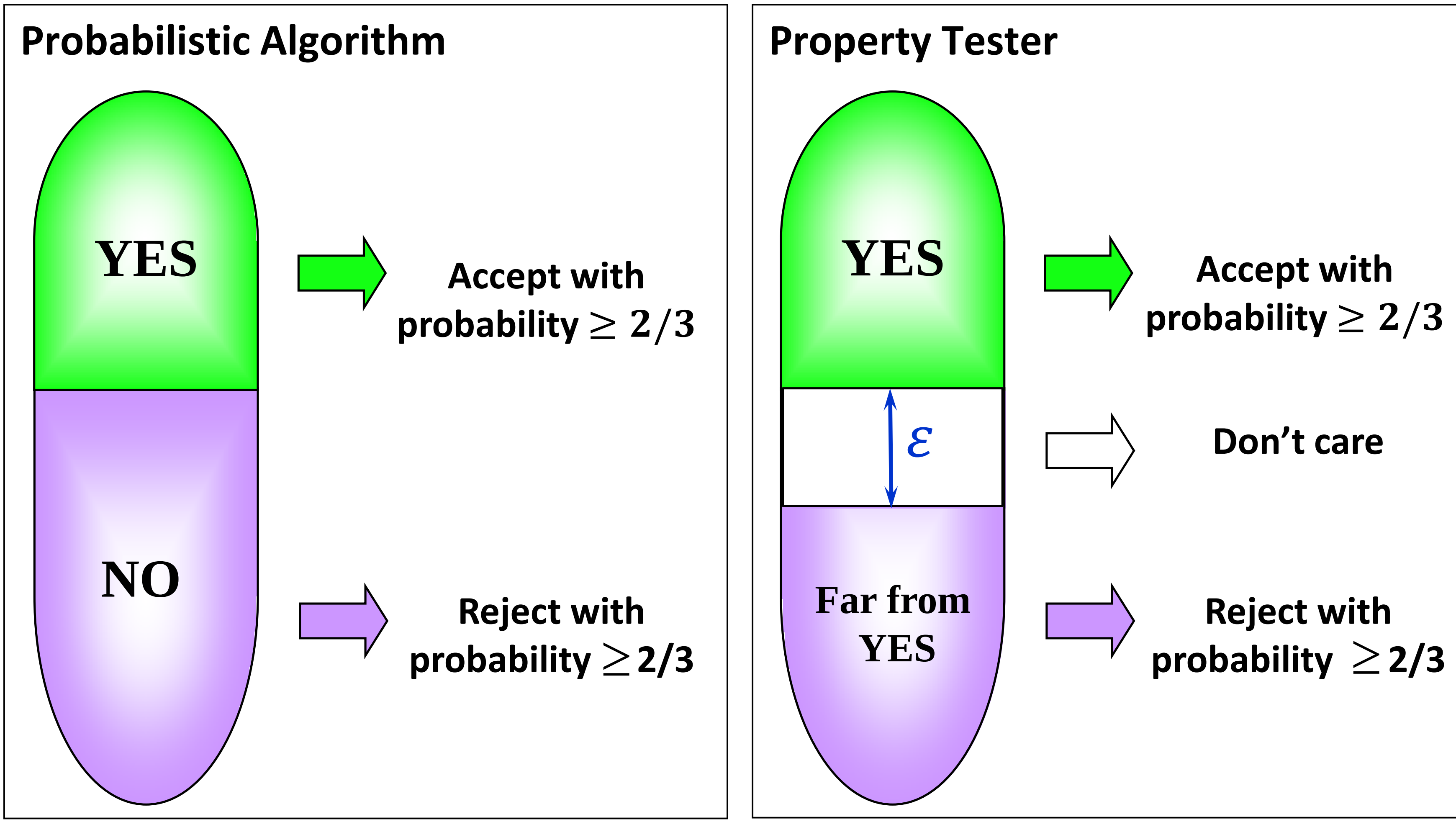
Does the input satisfy the property
or is it far from satisfying it?

- **for some applications, it is the right question (probabilistically checkable proofs (PCPs), precursor to learning)**
- **as good when the data is constantly changing**
- **fast sanity check to rule out inappropriate inputs**

(rejection-based image processing)

Property Testing: Definition

[Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]



ε-far = differs in many places (≥ ε fraction of places)

Property Testing: a Toy Example

0	0	0	1	…	0	1	0	0

Input: a string $w \in \{0,1\}^n$

Question: Is $w = 00 \dots 0$?

Requires reading entire input.

Approximate version: Is $w = 00 \dots 0$ or does it have $\geq \varepsilon n$ 1's ("errors")?

Test (n, w)

1. Sample $s = 2/\varepsilon$ positions uniformly and independently at random
2. If 1 is found, **reject**; otherwise, **accept**

Analysis: If $w = 00 \dots 0$, it is always accepted.

Used: $1 - x \leq e^{-x}$

If w is ε-far, Pr[error] = Pr[no 1's in the sample]$\leq (1-\varepsilon)^s \leq e^{-\varepsilon s} = e^{-2} < \frac{1}{3}$

Witness Lemma

If a test catches a witness with probability $\geq p$, then

$s = \frac{\ln k}{p}$ iterations of the test fail to catch a witness with probability $\leq 1/k$.

Property Testing

Simple Examples

Example 1: Testing if a List is Sorted

Input: a list of n numbers $x_1, x_2, \ldots, x_n$

- Question: Is the list sorted?

  Requires reading entire list: Ω(n) time

- Approximate version: Is the list sorted or ϵ-far from sorted?

  (An ϵ fraction of x_i's have to be changed to make it sorted.)

  [Ergün Kannan Kumar Rubinfeld Viswanathan]: O((log n)/ϵ) time

- Attempts:

  1. Test: Pick a random i and reject if $x_i > x_{i+1}$.

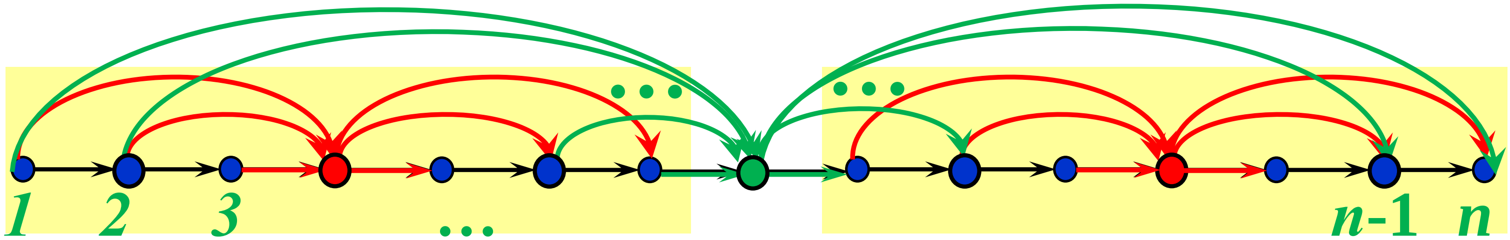
     Fails on: 1 1 1 1 1 1 1 0 0 0 0 0 0 0 $\leftarrow$ 1/2-far from sorted

  2. Test: Pick random $i < j$ and reject if $x_i > x_j$.

     Fails on: 1 0 2 1 3 2 4 3 5 4 6 5 7 6 $\leftarrow$ 1/2-far from sorted

Is a list sorted or ϵ-far from sorted?

Idea: Associate positions in the list with vertices of the directed line.



Construct a graph (2-spanner)

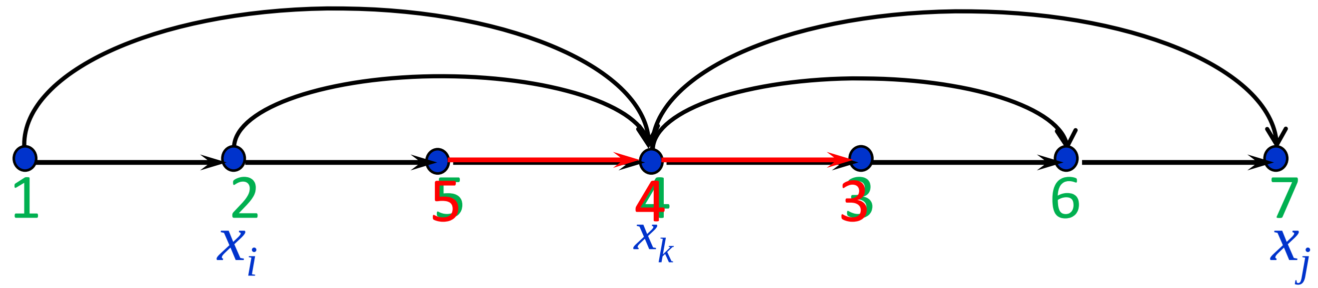
- by adding a few "shortcut" edges (i, j) for $i < j$
- where each pair of vertices is connected by a path of length at most 2

$\leq n \log n$ edges

Is a list sorted or ϵ-far from sorted?

Test [Dodis Goldreich Lehman **R** Ron Samorodnitsky, Bhattacharyya Grigorescu Jung **R** Woodruff]

Pick a random edge (i, j) from the 2-spanner and **reject** if $x_i > x_j$.



Analysis:

- Call an edge (i, j) **violated** if $x_i > x_j$, and **good** otherwise.
- If i is an endpoint of a **violated** edge, call x_i it **bad**. Otherwise, call it **good**.

Claim 1. All **good** numbers x_i are sorted.

Proof: Consider any two good numbers, x_i and x_j.

They are connected by a path of (at most) two **good** edges $(i, k), (k, j)$

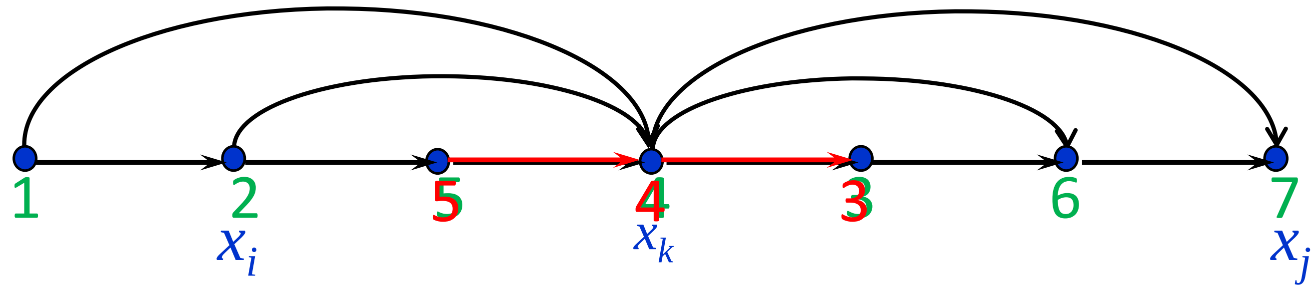
$\Rightarrow$ $x_i \leq x_k$ and $x_k \leq x_j$

$\Rightarrow$ $x_i \leq x_j$

Is a list sorted or ϵ-far from sorted?

Test [Dodis Goldreich Lehman **R** Ron Samorodnitsky, Bhattacharyya Grigorescu Jung **R** Woodruff]

Pick a random edge (i, j) from the 2-spanner and **reject** if $x_i > x_j$.



Analysis:

- Call an edge (i, j) **violated** if $x_i > x_j$, and **good** otherwise.
- If i is an endpoint of a **violated** edge, call x_i it **bad**. Otherwise, call it **good**.

Claim 1. All **good** numbers x_i are sorted.

Claim 2. An ϵ-far list **violates** $\geq \epsilon/(2 \log n)$ fraction of edges in 2-spanner.

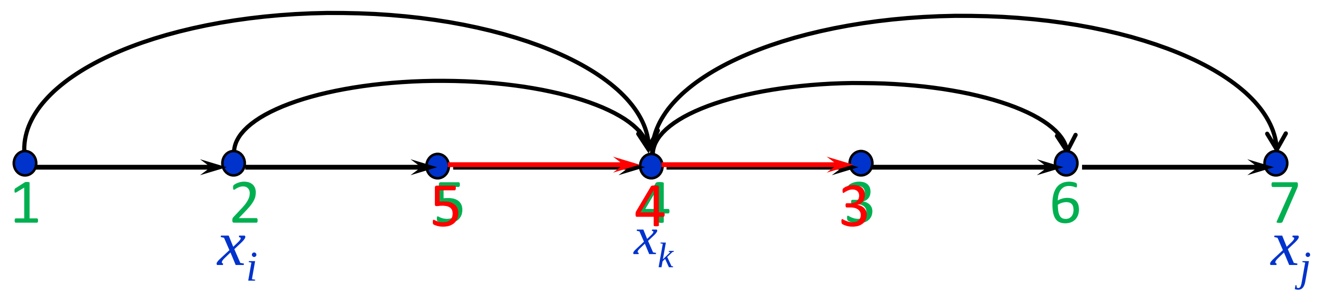
Proof: If a list is ϵ-far from sorted, it has $\geq \epsilon n$ **bad** numbers. (Claim 1)

- Each **violated** edge contributes 2 **bad** numbers.
- 2-spanner has $\geq \epsilon n/2$ **violated** edges out of $n \log n$.

Is a list sorted or ϵ-far from sorted?

Test [Dodis Goldreich Lehman **R** Ron Samorodnitsky, Bhattacharyya Grigorescu Jung **R** Woodruff]

Pick a random edge (i, j) from the 2-spanner and **reject** if $x_i > x_j$.



Analysis:

- Call an edge (i, j) **violated** if $x_i > x_j$, and **good** otherwise.

Claim 2. An ϵ-far list **violates** $\geq \epsilon/(2 \log n)$ fraction of edges in 2-spanner.

Algorithm

Sample $(4 \log n)/\epsilon$ edges (x_i, x_j) from the 2-spanner and **reject** if $x_i > x_j$.

Guarantee: All sorted lists are accepted. ✓

All lists that are ϵ-far from sorted are rejected with probability $\geq 2/3$. ✓

Time: $O((\log n)/\epsilon)$ ✓

Testing if a List is Sorted: Summary

We can determine if a list of n numbers is

sorted or ϵ-far from sorted

in $O\left(\frac{\log n}{\epsilon}\right)$ time.

- [Ergün Kannan Kumar Rubinfeld Viswanathan 98, Fischer 01, Blais R Yaroslavtsev 14]: This cannot be improved.

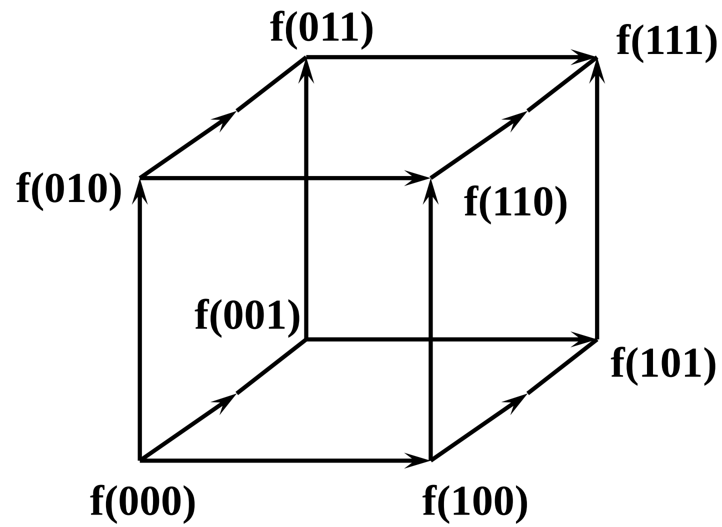
Basic Properties of Functions

Example 2

Boolean Functions $f : \{0,1\}^n \to \{0,1\}$

Graph representation:

n-dimensional hypercube



- vertices: bit strings of length n
- edges: (x, y) is an edge if y can be obtained from x by increasing one bit from 0 to 1

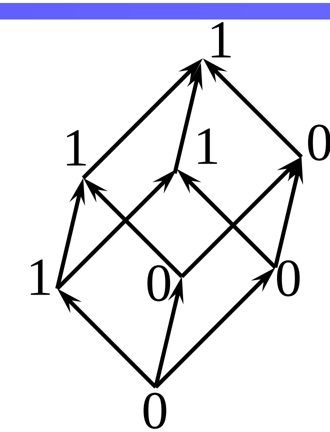
x	001001
y	011001

- each vertex x is labeled with $f(x)$

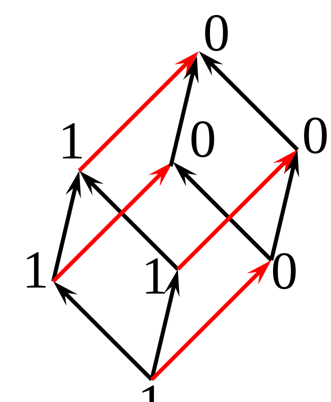
Testing Monotonicity of Functions

[Goldreich Goldwasser Lehman Ron Samorodnitsky,
Dodis Goldreich Lehman **R** Ron Samorodnitsky
Fischer Lehman Newman **R** Rubinfeld Samorodnitsky]

- A function $f : \{0,1\}^n \to \{0,1\}$ is monotone if increasing a bit of x does not decrease $f(x)$.
- Is f monotone or ε-far from monotone (f has to change on many points to become monontone)?
  - Edge $x \to y$ is violated by f if $f(x) > f(y)$.



monotone



$\frac{1}{2}$-far from monotone

Time:

- $O(n/\varepsilon)$, logarithmic in the size of the input, i.e., 2^n
- $\Omega(\sqrt{n}/\varepsilon)$ for nonadaptive, 1-sided error tests
- Recent: $\Theta(\sqrt{n}/\varepsilon^2)$ for nonadaptive tests [Khot Minzer Safra 15, Chen De Servidio Tang 15]

  $\tilde{\Omega}(n^{1/4})$ [Belov Blais 16]

Monotonicity Test [GGLRS, DGLRRS]

Idea: Show that functions that are far from monotone violate many edges.

EdgeTest (f, ε)

1. Pick $2n/\varepsilon$ edges (x, y) uniformly at random from the hypercube.
2. **Reject** if some (x, y) is violated (i.e. $f(x) > f(y)$). Otherwise, **accept**.

Analysis

- If f is monotone, EdgeTest always accepts.
- If f is ε-far from monotone, by Witness Lemma, it suffices to show that $\geq \varepsilon/n$ fraction of edges (i.e., $\frac{\varepsilon}{n} \cdot 2^{n-1}n = \varepsilon 2^{n-1}$ edges) are violated by f.
  - Let $V(f)$ denote the number of edges violated by f.

  Contrapositive: If $V(f) < \varepsilon\, 2^{n-1}$,

  f can be made monotone by changing $< \varepsilon\, 2^n$ values.

Repair Lemma

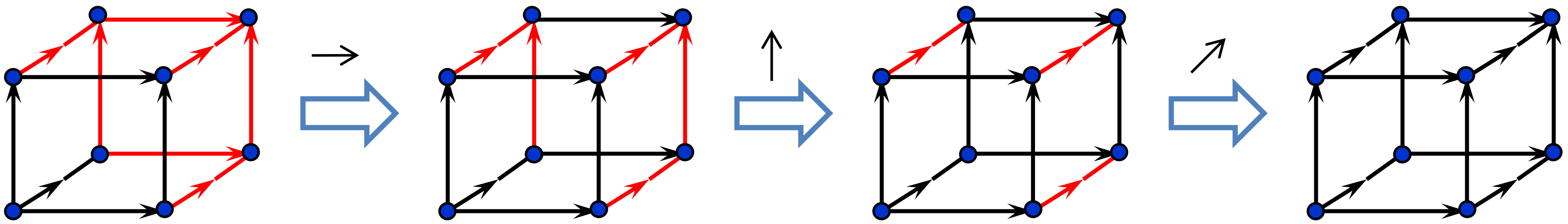
f can be made monotone by changing $\leq 2 \cdot V(f)$ values.

Repair Lemma: Proof Idea

Repair Lemma

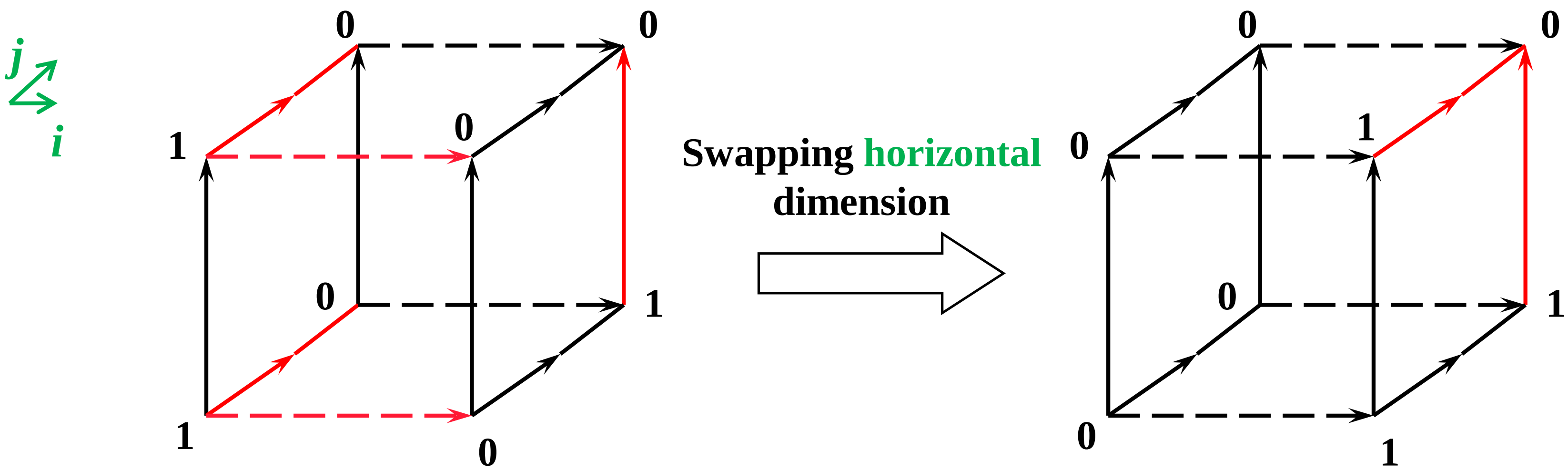
f can be made monotone by changing $\leq 2 \cdot V(f)$ values.

Proof idea: Transform f into a monotone function by repairing edges in one dimension at a time.



Repairing Violated Edges in One Dimension

Swap violated edges *1→0* in one dimension to *0→1*.



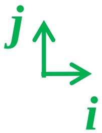
Let V_j = # of violated edges in dimension j

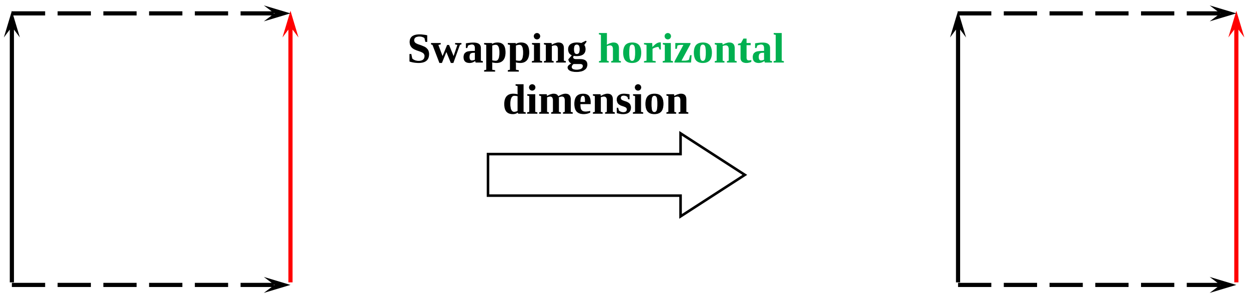
Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$

Enough to prove the claim for squares

Proof of The Claim for Squares

Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$

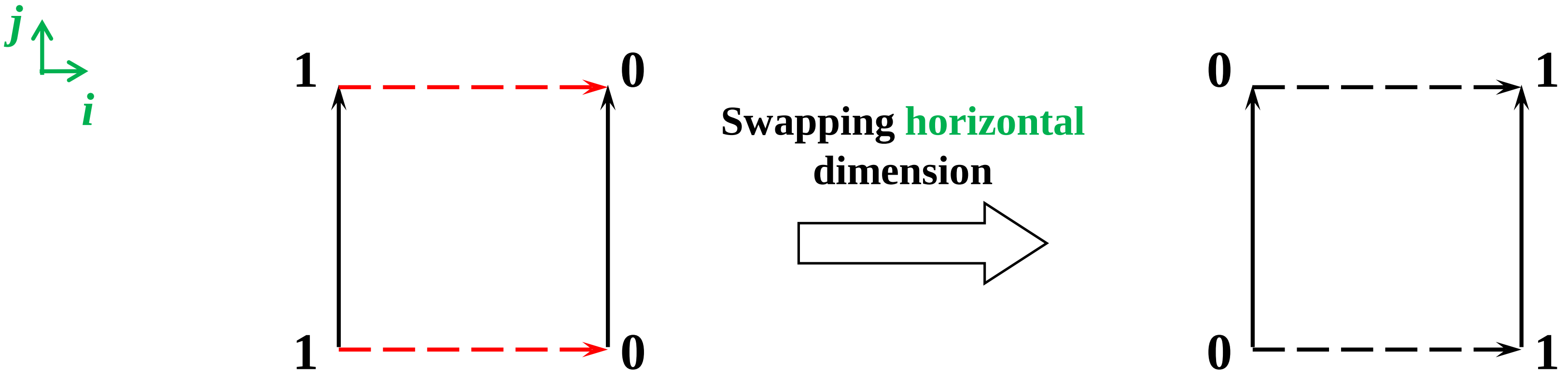




- If no horizontal edges are violated, no action is taken.

Proof of The Claim for Squares

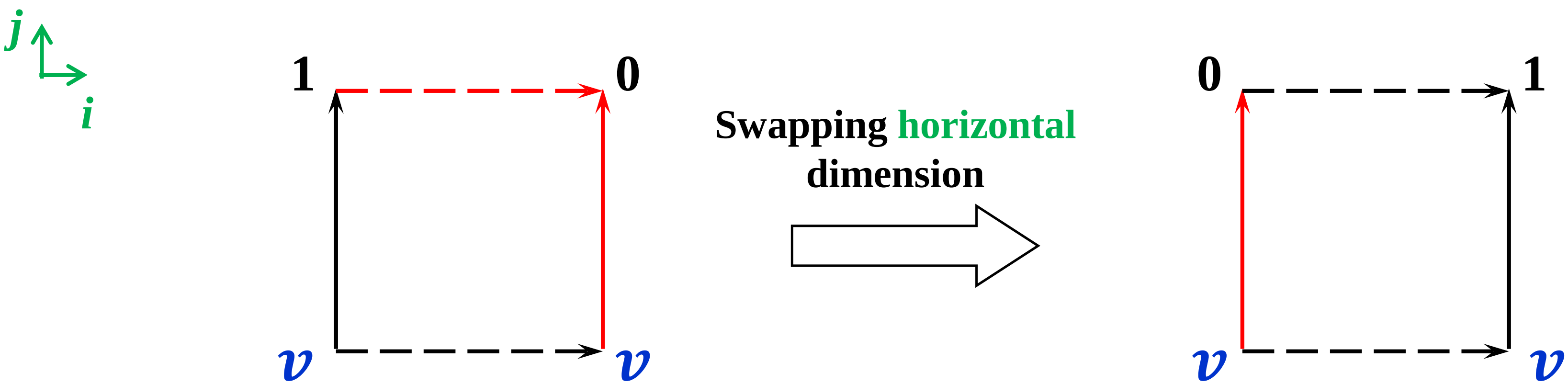
Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$



- If both horizontal edges are violated, both are swapped, so the number of vertical violated edges does not change.

Proof of The Claim for Squares

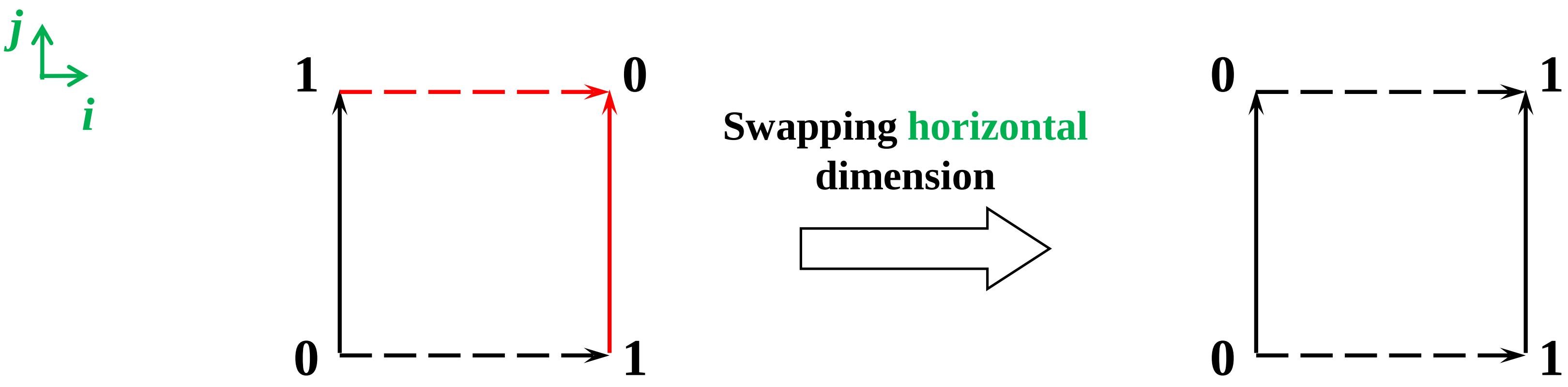
Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$



- Suppose one (say, top) horizontal edge is violated.
- If both bottom vertices have the same label, the vertical edges get swapped.

Proof of The Claim for Squares

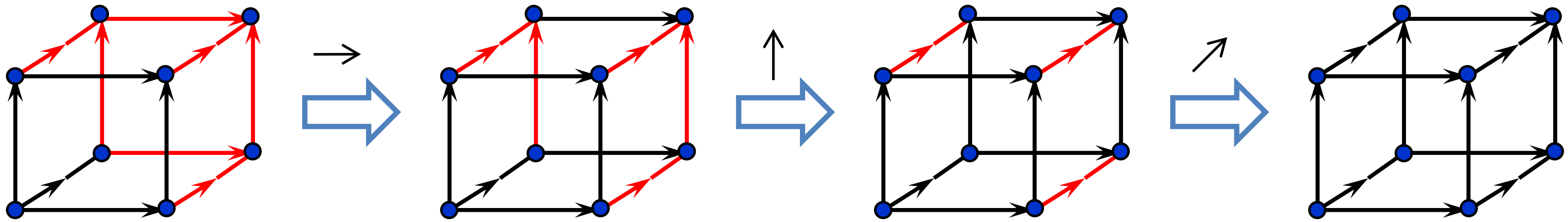
Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$



- Suppose one (say, top) horizontal edge is violated.
- If both bottom vertices have the same label, the vertical edges get swapped.
- Otherwise, the bottom vertices are labeled $0 \rightarrow 1$, and the vertical violation is repaired.

Proof of The Claim for Squares

Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$



After we perform swaps in all dimensions:

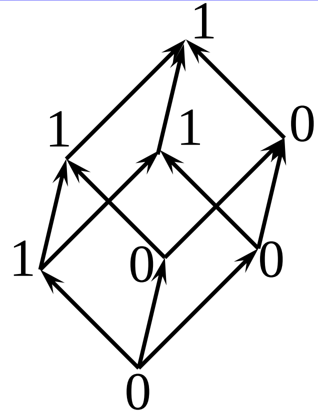
- f becomes monotone
- # of values changed:

  $2 \cdot V_1 + \ 2 \cdot$ (# violated edges in dim 2 after swapping dim 1)
  $+ 2 \cdot$ (# violated edges in dim 3 after swapping dim 1 and 2)
  $+ \ \ldots \leq 2 \cdot V_1 + 2 \cdot V_2 + \cdots 2 \cdot V_n = 2 \cdot V(f)$

Repair Lemma

f can be made monotone by changing $\leq 2 \cdot V(f)$ values.

Testing if a Functions $f:\{0,1\}^n \to \{0,1\}$ is monotone



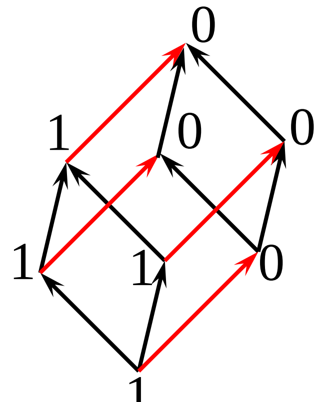
monotone

Monotone or

ε-far from monotone?

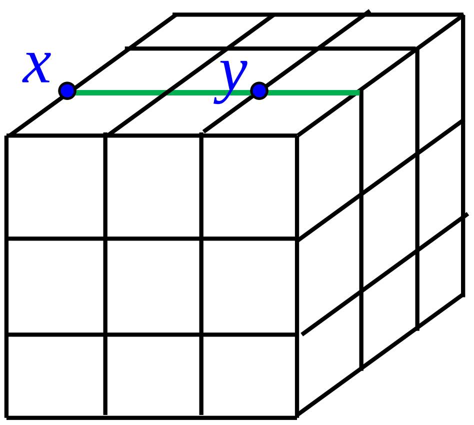
O(n/ε) time ✓

(logarithmic in the size of the input)



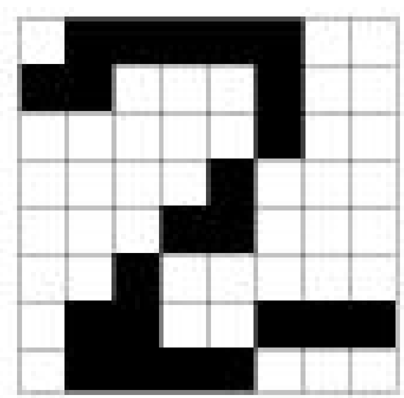
$\frac{1}{2}$-far from monotone

Generalizations



- [Dodis Goldreich Lehman **R** Ron Samorodnitsky, Chakrabarty Seshadhri]: generalization to testing monotonicity of discrete d-dimensional functions in polylogarithmic time.
- [Jha **R**, Chakrabarty Dixit Jha Sesh]: generalization to testing other properties of functions in polylog time.

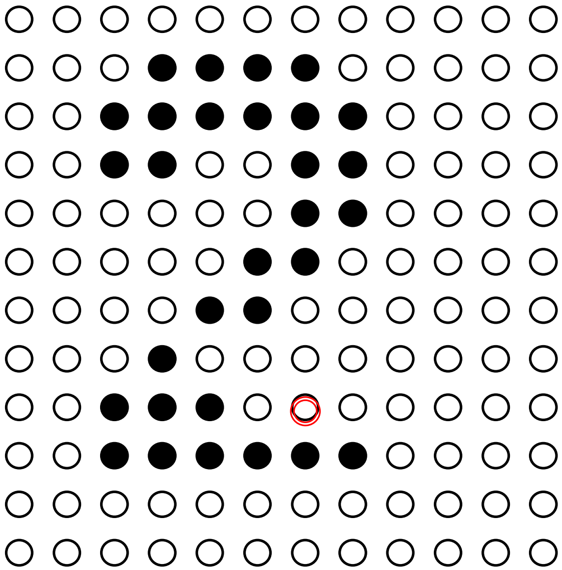
Example 3: Testing Properties of Images



Pixel Model

Input: $n \times n$ matrix of pixels

(0/1 values for black-and-white pictures)



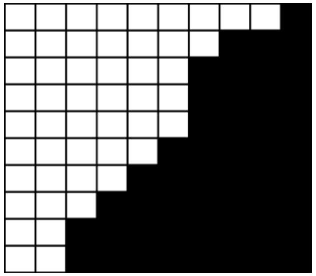
Query: point (i_1, i_2)

Answer: color of (i_1, i_2)

Testing if an Image is a Half-plane

A half-plane or

ε-far from a half-plane?

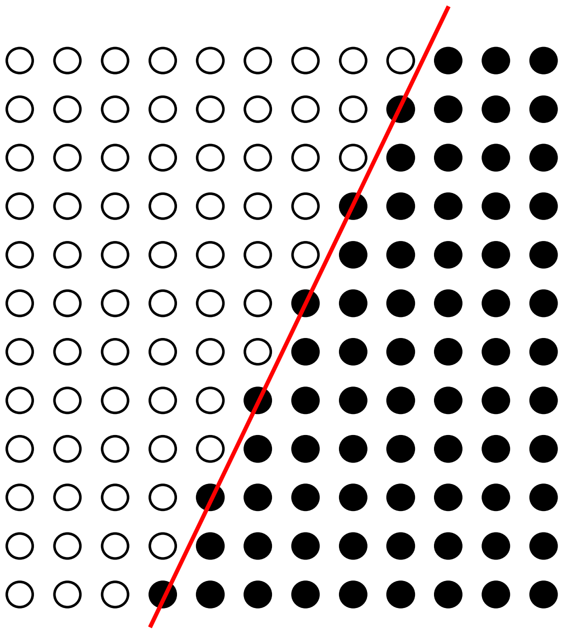


O($1/\varepsilon$) time [**R** 03]

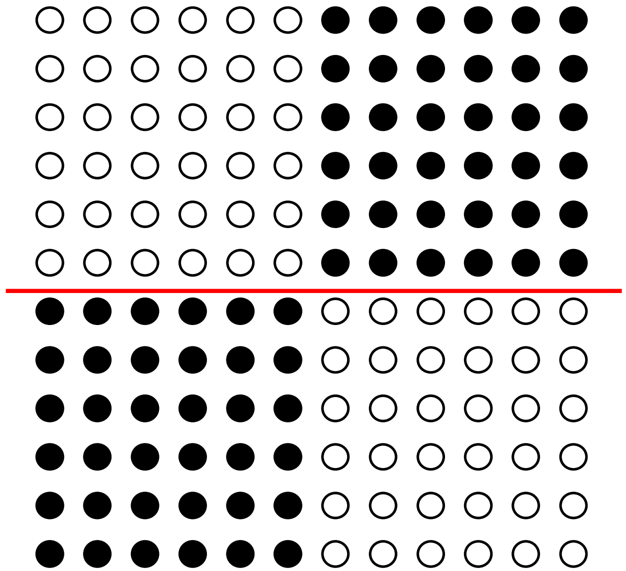
O($1/\varepsilon$) time with uniform samples

[Berman Murzabulatov **R** 16]

Half-plane Instances

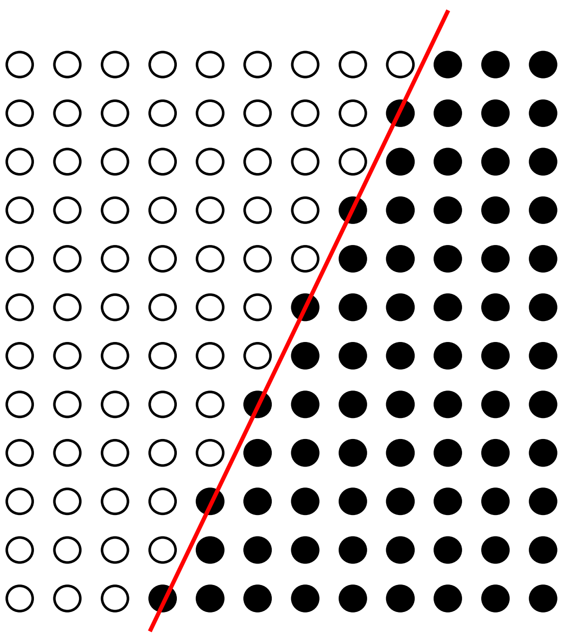


A half-plane

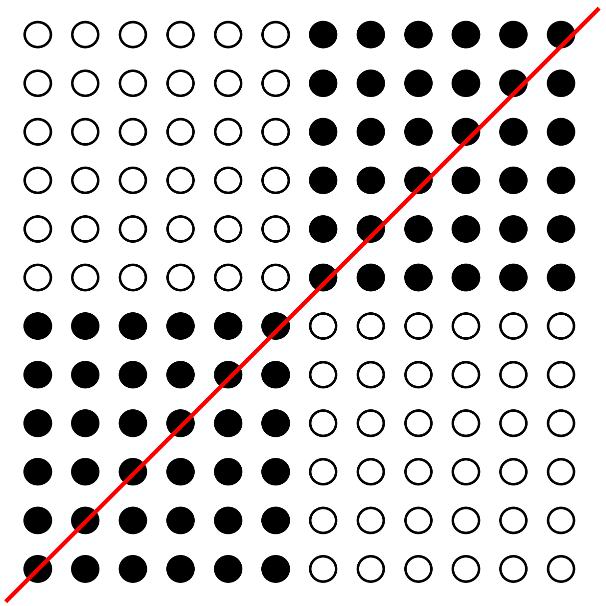


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

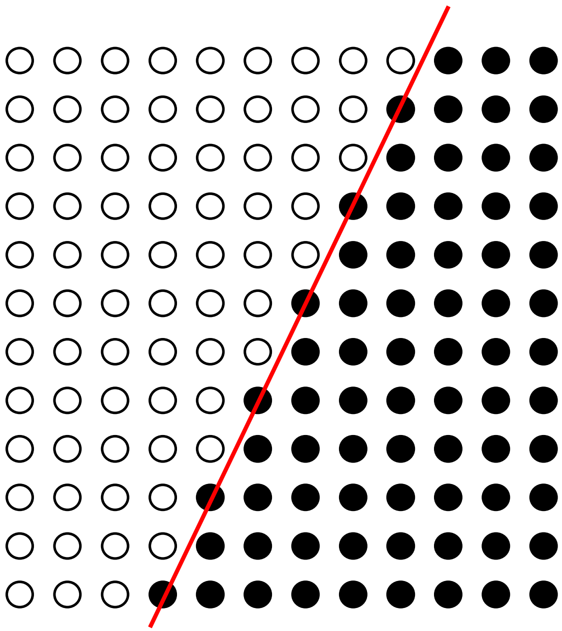


A half-plane

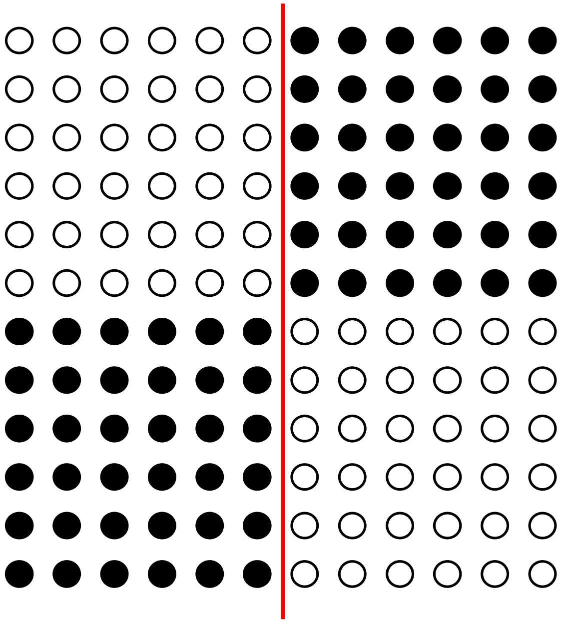


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

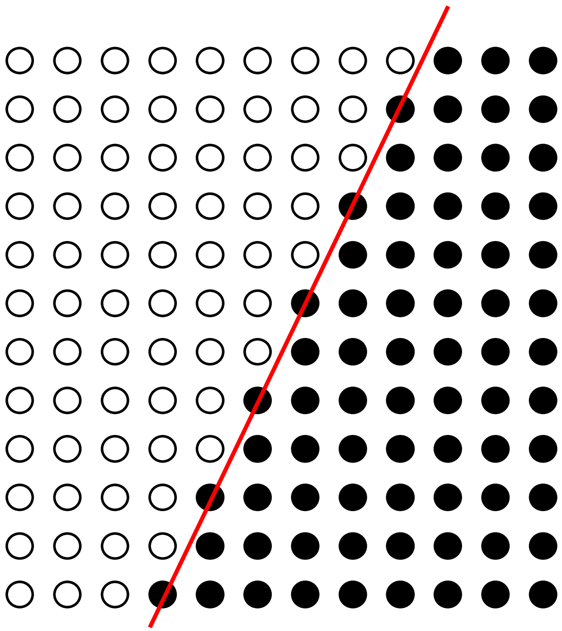


A half-plane

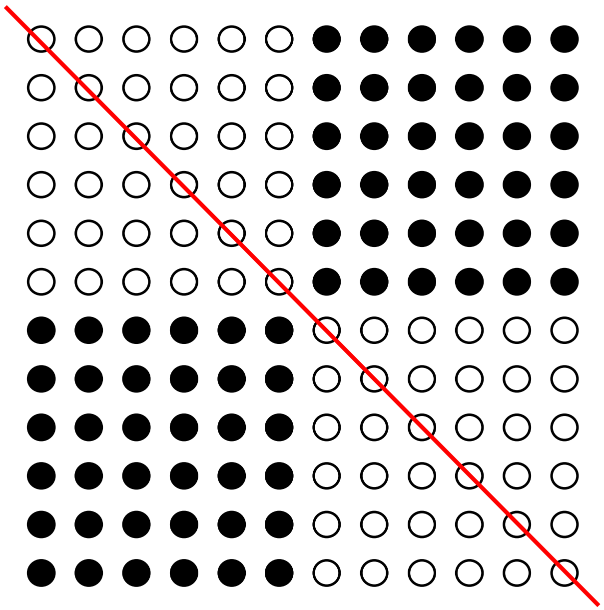


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

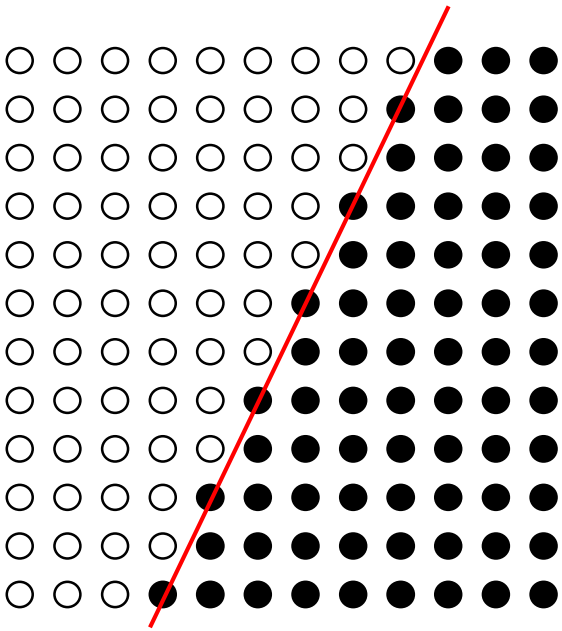


A half-plane

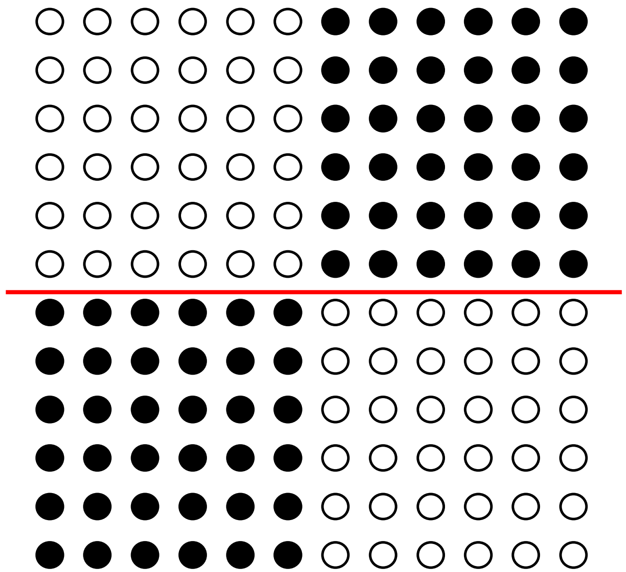


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

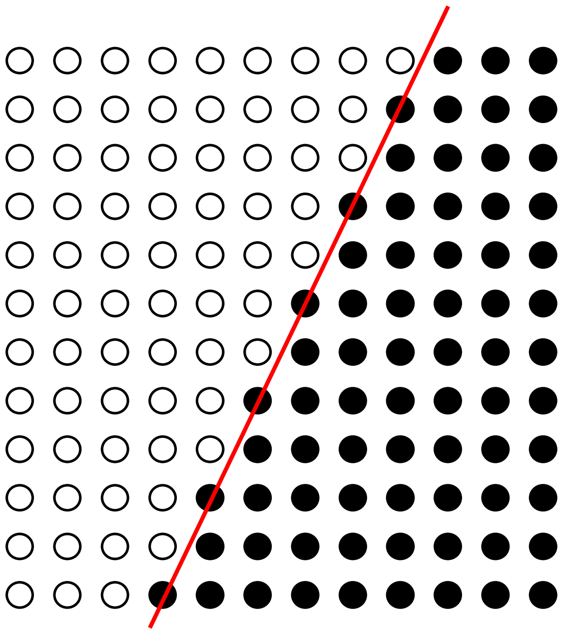


A half-plane

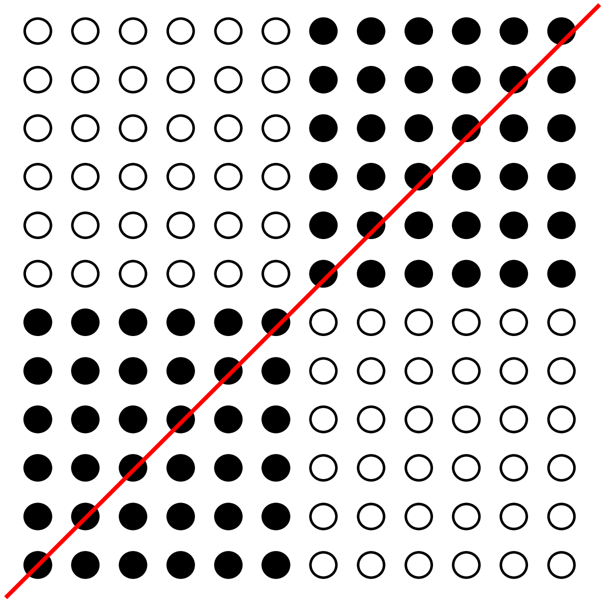


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

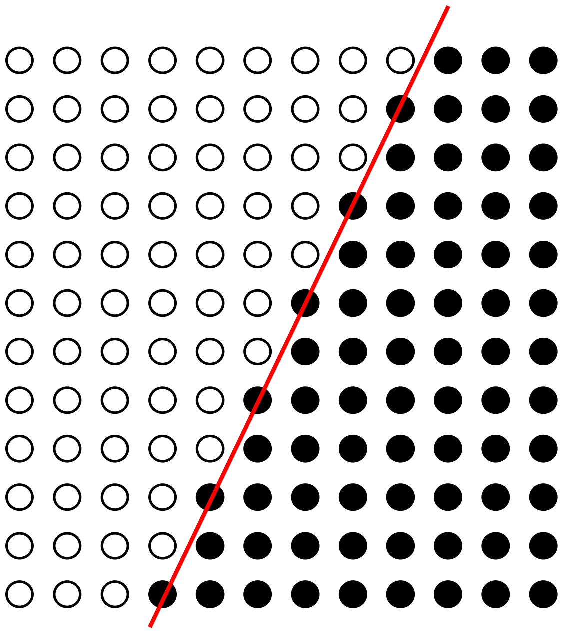


A half-plane

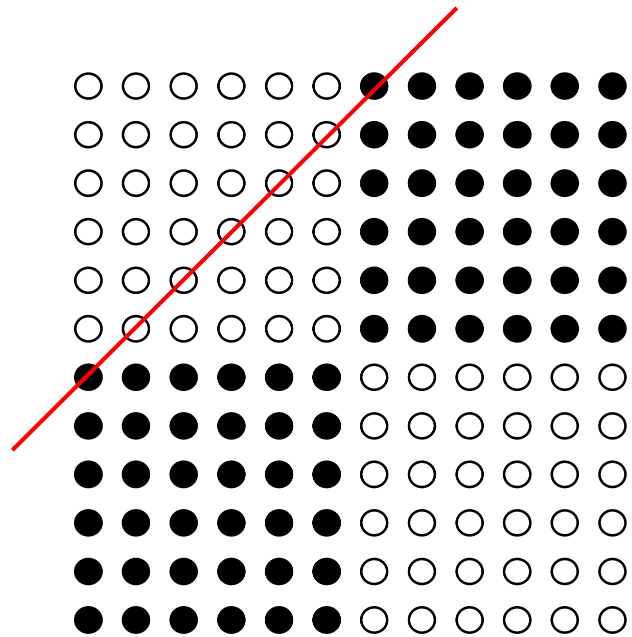


$\frac{1}{4}$-far from a half-plane

Half-plane Instances



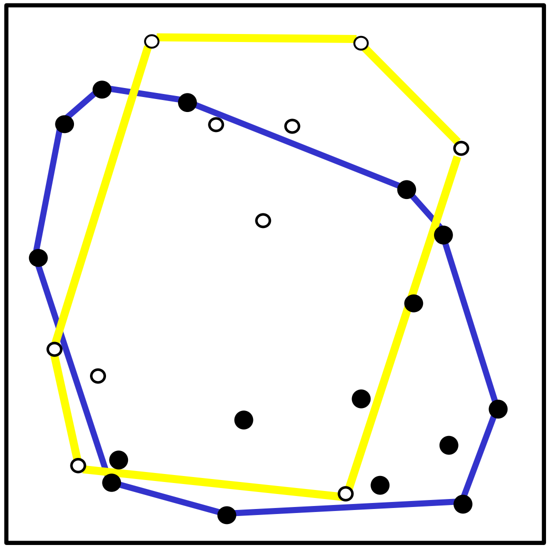
A half-plane



$\frac{1}{4}$-far from a half-plane

Half-Plane Tester

1. Sample $\boldsymbol{s} = \boldsymbol{\Theta}\left(\frac{\mathbf{1}}{\boldsymbol{\varepsilon}}\right)$ pixels uniformly and independently.
2. Find convex hull of black samples and convex hull of white samples.
3. If the two hulls intersect, **reject**; otherwise, **accept**.



- The tester always accepts half-plane images.

Correctness Theorem

If an image is ε-far from being a half-plane, it is rejected w.p. $\geq 2/3$.

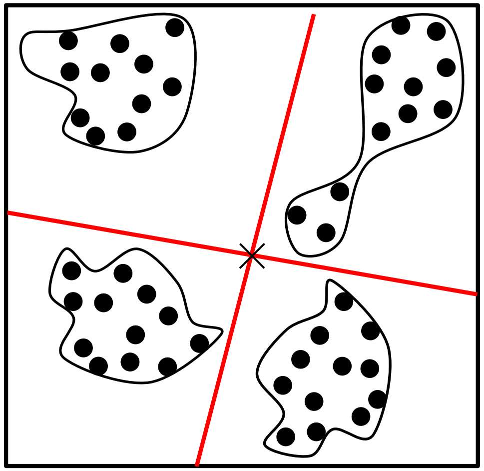
Analysis Idea: Central Points

Some points are likely to end up in the convex hull of black pixels.

- A point does not have to correspond to a pixel.

Definition

A point is **black-central** if it is the intersection of two lines such that each quadrant formed by the lines has $\geq \varepsilon n^2/4$ **black** pixels.



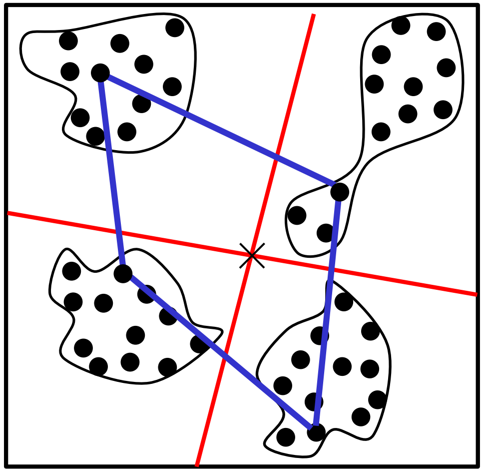
- A white-central point is defined analogously.

Analysis Idea: Central Points

Some points are likely to end up in the convex hull of black pixels.

Definition

A point is **black-central** if it is the intersection of two lines such that each quadrant formed by the lines has $\geq \varepsilon n^2/4$ **black** pixels.



- If we sample a black pixel (``witness'') from each quadrant, then the black-central point is in the convex hull of black pixels.

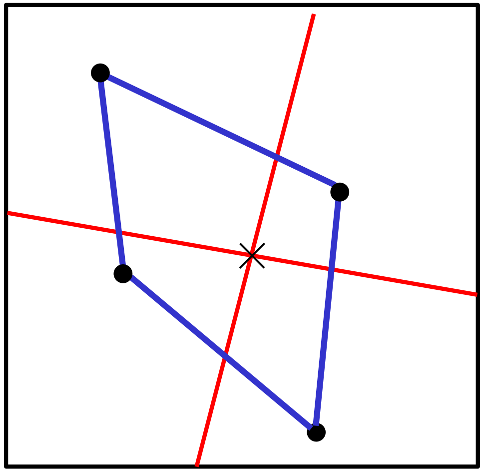
  We say ``we **captured** the black-central point''.

Analysis Idea: Central Points

Some points are likely to end up in the convex hull of black pixels.

Definition

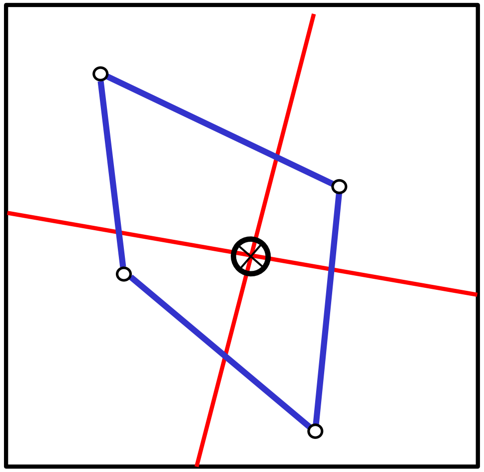
A point is **black-central** if it is the intersection of two lines such that each quadrant formed by the lines has $\geq \varepsilon n^2/4$ **black** pixels.



- By Witness Lemma, if we sample $\frac{\ln 100}{\varepsilon/4}$ random pixels, we fail to find a witness from a quadrant w.p. $\leq \frac{1}{100}$.
- By the union bound, we fail to capture a black-central w.p. $\leq \frac{4}{100}$

Analysis Idea: Central Points

Some points are likely to end up in the convex hull of black pixels.



Definition

A point is **black-central** if it is the intersection of two lines such that each quadrant formed by the lines has $\geq \varepsilon n^2/4$ **black** pixels.

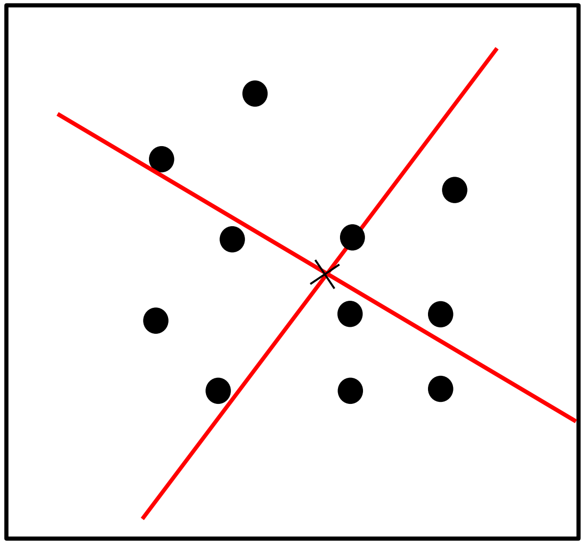
- Analogously, we fail to capture a white-central w.p. $\leq \frac{4}{100}$

Central Points Exist

The Ham Sandwich Theorem

In n dimensions, any n measurable sets can be simultaneously bisected (w.r.t. their measure) by an $(n-1)$-dimensional hyperplane.

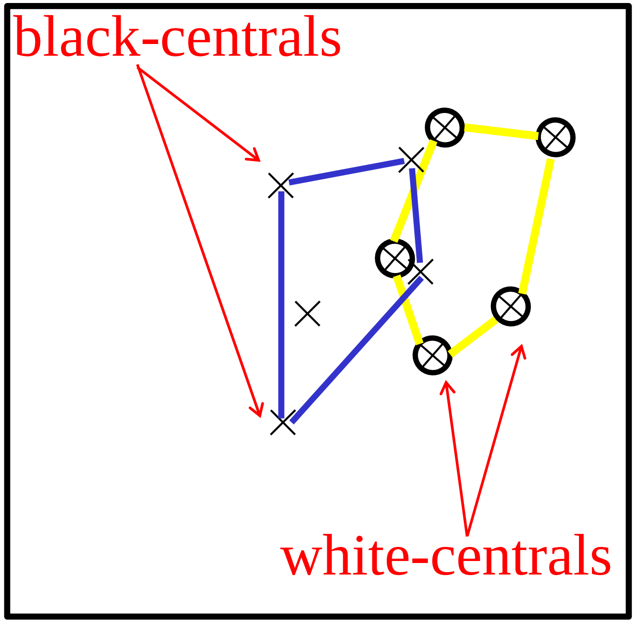
- If an image is ε-far from being a half-plane, it contains at least εn^2 pixels of each color.
- By continuity, there is a line that bisects all pixels of the same color into two sets.
- By the Ham Sandwich Theorem (for $n = 2$), there is another line that bisects both sets.



Hulls of Black- and White-Central Points

Main Lemma

If the image is ε-far from being a half-plane then the convex hull of black-centrals intersects the convex hull of white-centrals.

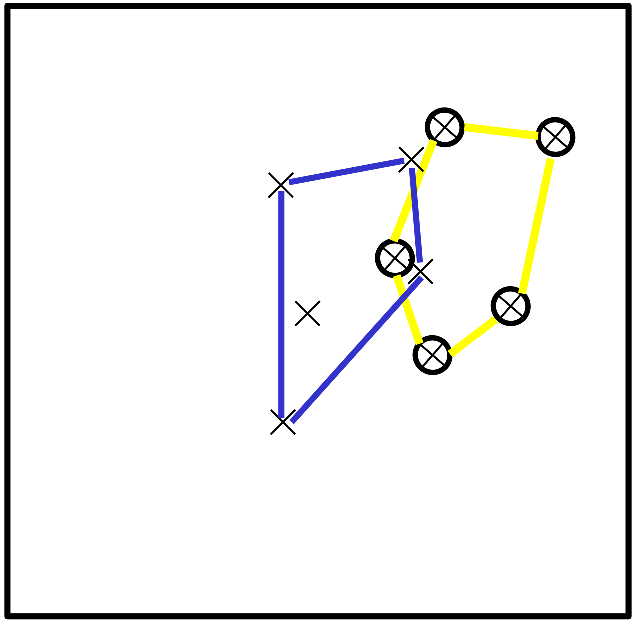


Hulls of Black- and White-Central Points

Main Lemma

If the image is ε-far from being a half-plane then the convex hull of black-centrals intersects the convex hull of white-centrals.

Proof: For the sake of contradiction, assume they do not intersect.

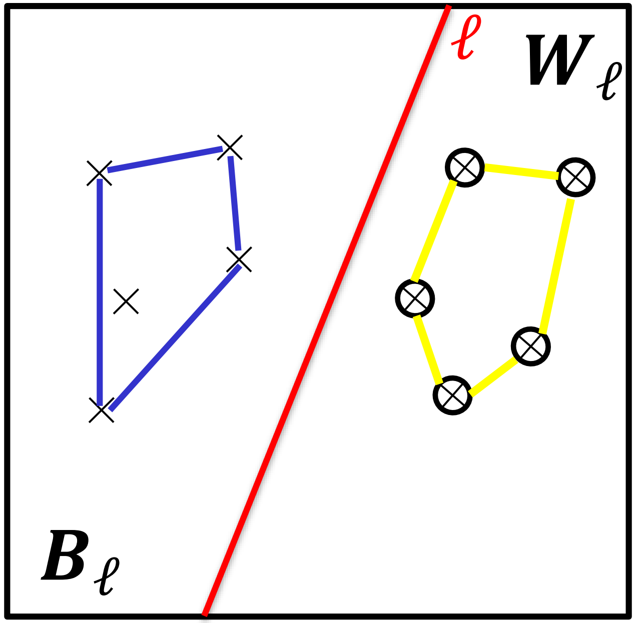


- Then some line ℓ separates white-central and black-central points.

Hulls of Black- and White-Central Points

Main Lemma

If the image is ε-far from being a half-plane then the convex hull of black-centrals intersects the convex hull of white-centrals.



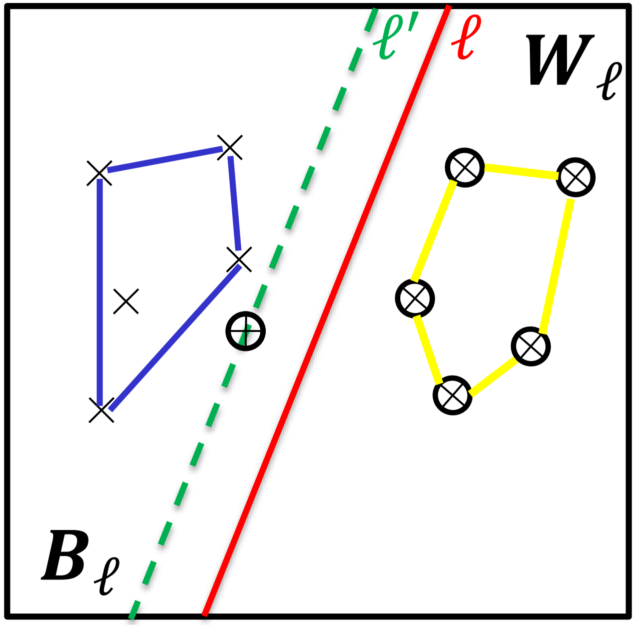
Proof: For the sake of contradiction, assume they do not intersect.

- Then some line ℓ separates white-central and black-central points.
- Let B_ℓ and W_ℓ be the closed half-planes formed by ℓ, with black-central and white-central points, respectively.

Hulls of Black- and White-Central Points

Main Lemma

If the image is ε-far from being a half-plane then the convex hull of black-centrals intersects the convex hull of white-centrals.

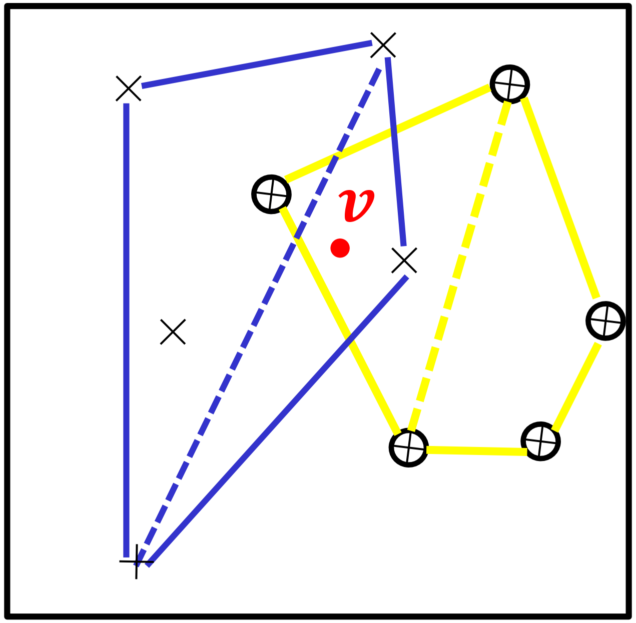


Proof: For the sake of contradiction, assume they do not intersect.

- There are $\geq \frac{\varepsilon n^2}{2}$ black pixels in W_ℓ or white pixels in B_ℓ. W.l.o.g. suppose the latter holds.
- Let ℓ' be the line parallel to ℓ and furthest from ℓ s.t. there $\geq \frac{\varepsilon n^2}{2}$ white pixels in closed half-plane to the left of ℓ'.
- There are $\geq \frac{\varepsilon n^2}{2}$ white pixels in closed half-plane to the right of ℓ'.
- By Ham Sandwich Theorem, there is a white-central point on ℓ'.

Contradiction!

Completing the Analysis



Main Lemma

If the image is ε-far from being a half-plane then the convex hull of black-centrals intersects the convex hull of white-centrals.

- Then some point v is in both hulls.
- Moreover, v is in the convex hull of
  - (at most) 3 black-central points;
  - (at most) 3 white-central points.

- If we capture all 6, then v is in the hull of black samples and in the hull of white samples.
- Recall: we fail to capture a central point w.p. $\leq \frac{4}{100}$
- By union bound, we fail to capture one or more of the 6 central points w.p. $\leq \frac{24}{100} < \frac{1}{3}$.

Summary: Half-plane Testing

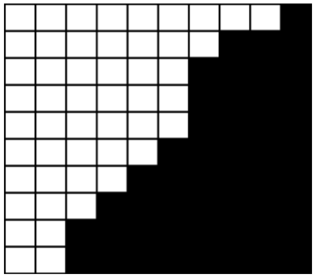
- $O(\mathbf{1}/\boldsymbol{\varepsilon})$ uniform samples are sufficient for testing the half-plane property with 1-sided error.
- It is easy to show that $\Omega(\mathbf{1}/\boldsymbol{\varepsilon})$ queries are necessary for even 2-sided error, adaptive testers.

Testing if an Image is a Half-plane

A half-plane or

ε-far from a half-plane?

O(1/ε) time ✓



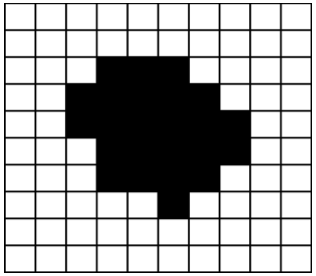
Other Results on Properties of Images

- Pixel Model

  Convexity [R 03, Berman Murzabulatov R 16]

  Convex or ε-far from convex?

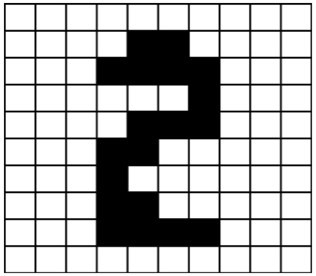
  $O(1/\varepsilon)$ time



  Connectedness [R 03, Berman Murzabulatov R 16]

  Connected or ε-far from connected?

  $O(1/\varepsilon^{3/2}\sqrt{\log 1/\varepsilon}\,)$ time

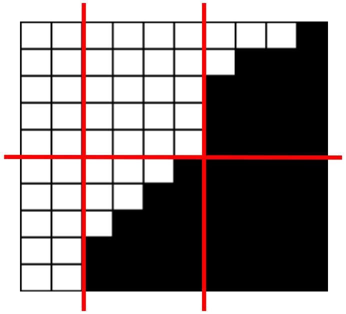


  Partitioning [Kleiner Keren Newman 10]

  Can be partitioned according to a template

  or is ε-far?

  time independent of image size



- Properties of sparse images [Ron Tsur 10]

Summary of Examples We Have Seen

- Properties of number sequences
  - sortedness
- Properties of functions
  - monotonicity
- Visual properties
  - half-plane

Property Testing: State of the Art

For many properties

of many types of objects

(images, number sequences, functions, graphs, codes, …),

there are testers that run in polylogarithmic time.

In some cases, the exact versions

of these problems are NP-complete!

Goal

Understanding sublinear algorithms and their limitations

- Algorithmic techniques

  (like dynamic programming for P)

- Lower bound techniques

  (like NP-completeness for NP)

Testing World ≠ Classical World

The approximate testing world is fascinating and different from the world of exact problems

Problem	Exact	Testing
Is a 3CNF satisfiable? (3SAT)	**NP-complete**	**easy** [Alon Shapira]
Does an assignment satisfy a fixed 3CNF?	**easy**	**hard** [Ben-Sasson Harsha **R**]

Limitations of Property Testing Algorithms

General lower bound methods:

- Yao's Minimax Principle
- Reductions from hard communication complexity problems [Blais Brody Matulef 11]

Yao's Minimax Principle

The following statements are equivalent.

Statement 1

For any **probabilistic** algorithm A of complexity q there exists an input x s.t.

$$\Pr_{\textit{coin tosses of } A}[\text{A(x) is wrong}] > 1/3.$$

Statement 2

There is a distribution *D* on the inputs,

s.t. for every **deterministic** algorithm of complexity q,

$$\Pr_{x \leftarrow D}[\text{A(x) is wrong}] > 1/3.$$

- Need for lower bounds

Yao's Minimax Principle (easy direction): Statement 2 $\Rightarrow$ Statement 1.

*Toy Example: Lower Bound for Testing 1**

Input: string of n bits

Question: Does the string contain only 1's or is it ε-far form the all-1 string?

Claim. *Any algorithm needs* $\mathbf{\Omega(1/\varepsilon)}$ *queries to answer this question w.p.* $\geq \mathbf{2/3}$.

Proof: By Yao's Minimax Principle, enough to prove Statement 2.

Distribution D on n-bit strings

- *Divide the input string into* $1/\varepsilon$ *blocks of size* εn.
- *Let* y_i *be the string where the* i*th block is 0's and remaining bits are 1.*
- *Distribution D gives the all-1 string w.p. 1/2 and* y_i *with w.p. 1/2, where* i *is chosen uniformly at random from* $1, \ldots, 1/\varepsilon$.

1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

εn $\quad$ εn $\quad$ εn $\quad$ εn

*A Lower Bound for Testing 1**

Claim. *Any ε-test for 1* needs $\Omega(1/\varepsilon)$ queries.*

Proof (continued): Now fix a deterministic tester A making $q < 1/3\varepsilon$ queries.

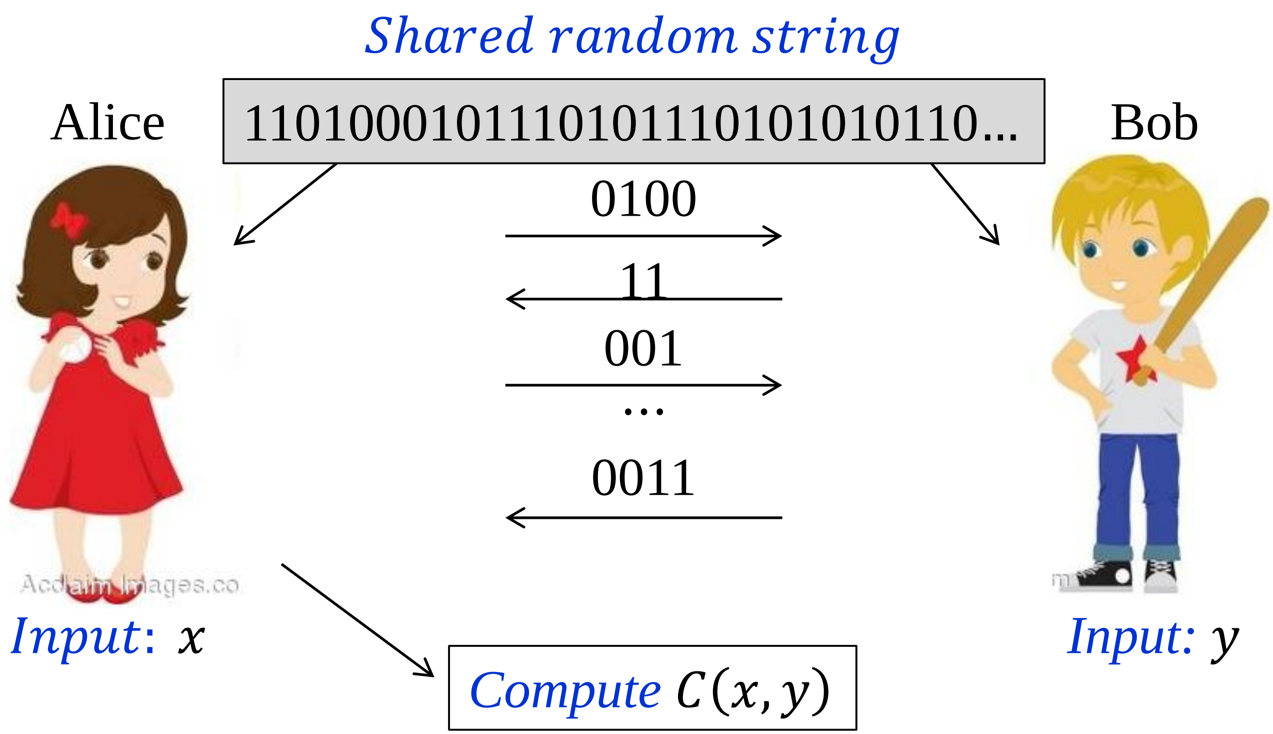
1. A must accept if all answers are 1. Otherwise, it would be wrong on all-1 string, that is, with probability 1/2 with respect to D.
2. Let $i_1, \ldots, i_q$ be the positions A queries when it sees only 1s. The test can choose its queries based on previous answers. However, since all these answers are 1 and since A is deterministic, the query positions are fixed.

- At least $1/\varepsilon - q > 2/3\varepsilon$ of the blocks do not hold any queried indices.
- Therefore, A accepts > 2/3 of the inputs y_i. Thus, it is wrong with probability $> 2/3\varepsilon \cdot \frac{\varepsilon}{2} = 1/3$

Context: [Alon Krivelevich Newman Szegedy 99]

Every regular language can be tested in $O(1/\varepsilon \text{ polylog } 1/\varepsilon)$ time

(Randomized) Communication Complexity



Goal: minimize the number of bits exchanged.

- Communication complexity of a protocol is the maximum number of bits exchanged by the protocol.
- Communication complexity of a function C, denoted $R(C)$, is the communication complexity of the best protocol for computing C.

Example: Set Disjointness $DISJ_k$

Alice 11010001011101011101010110... Bob

Input: $S \subseteq [n]$, $|S| = k$.

Input: $T \subseteq [n]$, $|T| = k$

Compute $DISJ_k(S, T)$

$$= \begin{cases} \boldsymbol{accept} & \text{if } S \cap T = \emptyset \\ \boldsymbol{reject} & \text{otherwise} \end{cases}$$

Theorem [Kalyanasundaram Schnitger 92]

$$R(\mathrm{DISJ}_k) \geq \Omega(k) \text{ for all } k \leq \frac{n}{2}.$$

Testing if a Function is a k-Parity

k-Parity Functions

A function $f : \{0,1\}^n \to \{0,1\}$ is a k-parity if

$$f(x) = \chi_S(x) = \sum_{i \in S} x_i \pmod 2.$$

for some set $S \subseteq [n]$ of size $|S| = k$.

Time to test:

$O(k \log k)$ [Chakraborty Garcia–Soriano Matsliah]

$\Omega(\min(k, n-k))$ [Blais Brody Matulef 11]

- Today: $\Omega(k)$ for $k \leq n/2$
- Today's bound implies $\Omega(\min(k, n-k))$

Reduction from $DISJ_{\boldsymbol{k/2}}$ to Testing k-Parity

- Let T be the best tester for the k-parity property for $\varepsilon = 1/2$
  - query complexity of T is q (testing k−parity).
- We will construct a communication protocol for $DISJ_{\boldsymbol{k/2}}$ that runs T and has communication complexity $2 \cdot q$(testing k−parity).

holds for CC of every protocol for $DISJ_{\boldsymbol{k}}$

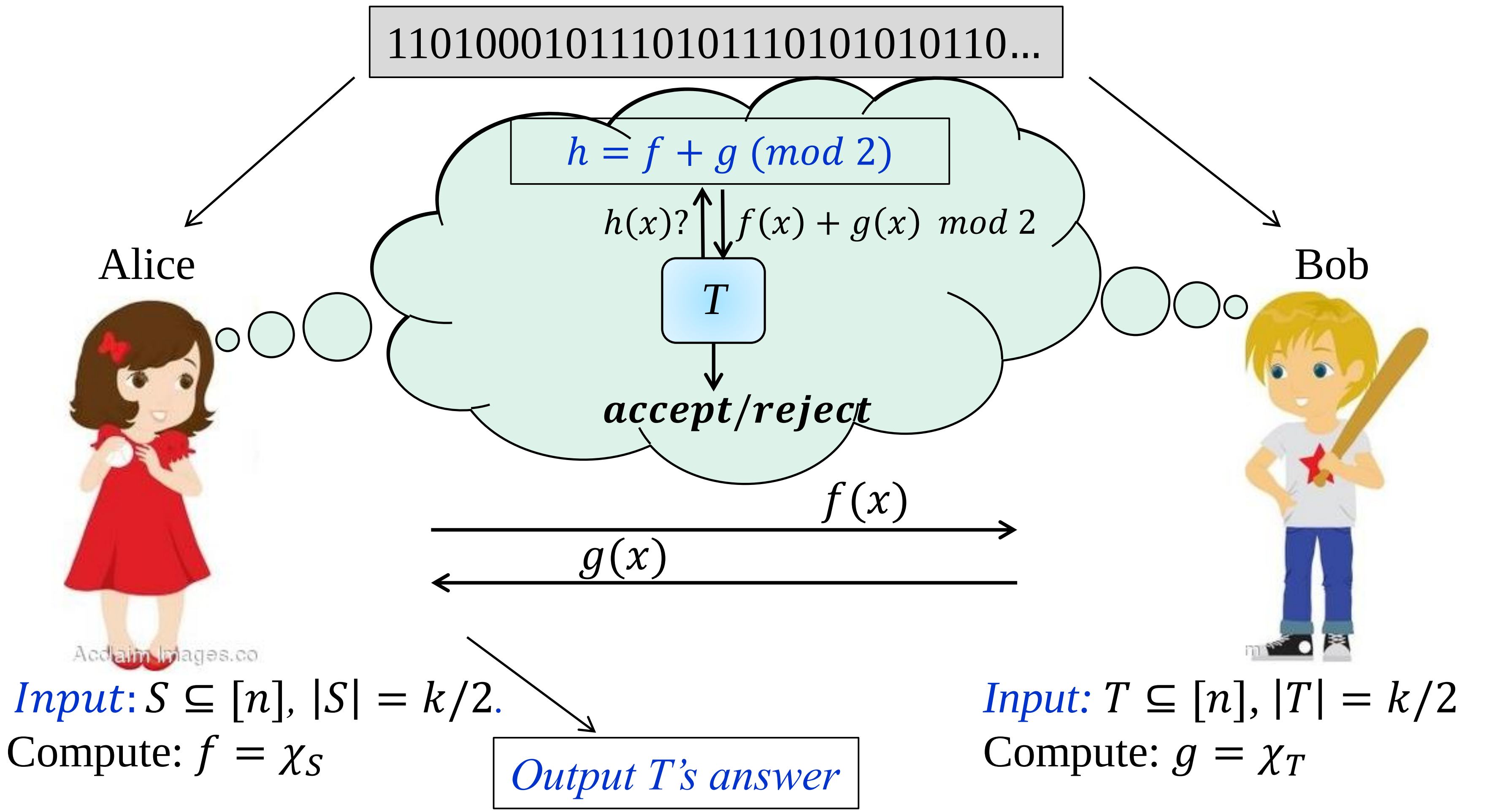
[Kalyanasundaram Schnitger 92]

- Then $2 \cdot q$(testing k−parity) $\geq R\left(\mathrm{DISJ}_{k/2}\right) \geq \Omega(k/2)$ for $k \leq n/2$

$$\Downarrow$$

q(testing k-parity) $\geq \Omega(k)$ for $k \leq n/2$

Reduction from $DISJ_{k/2}$ to Testing k-Parity



- T receives its random bits from the shared random string.

Analysis of the Reduction

Queries: Alice and Bob exchange 2 bits for every bit queried by T

Correctness:

- $h = f + g \ (mod\ 2) = \chi_S + \chi_T \ (mod\ 2) = \chi_{S\Delta T}$
- $|S\Delta T| = |S| + |T| - 2|S \cap T|$

- $|S\Delta T| = \begin{cases} k & \text{if } S\cap T = \emptyset \\ \le k-2 & \text{if } S\cap T \ne \emptyset \end{cases}$

$$h \text{ is } \begin{cases} k\text{−parity} & \text{if } S\cap T = \emptyset \\ k'\text{−parity where } k' \ne k & \text{if } S\cap T \ne \emptyset \end{cases}$$

1/2-far from every k-parity

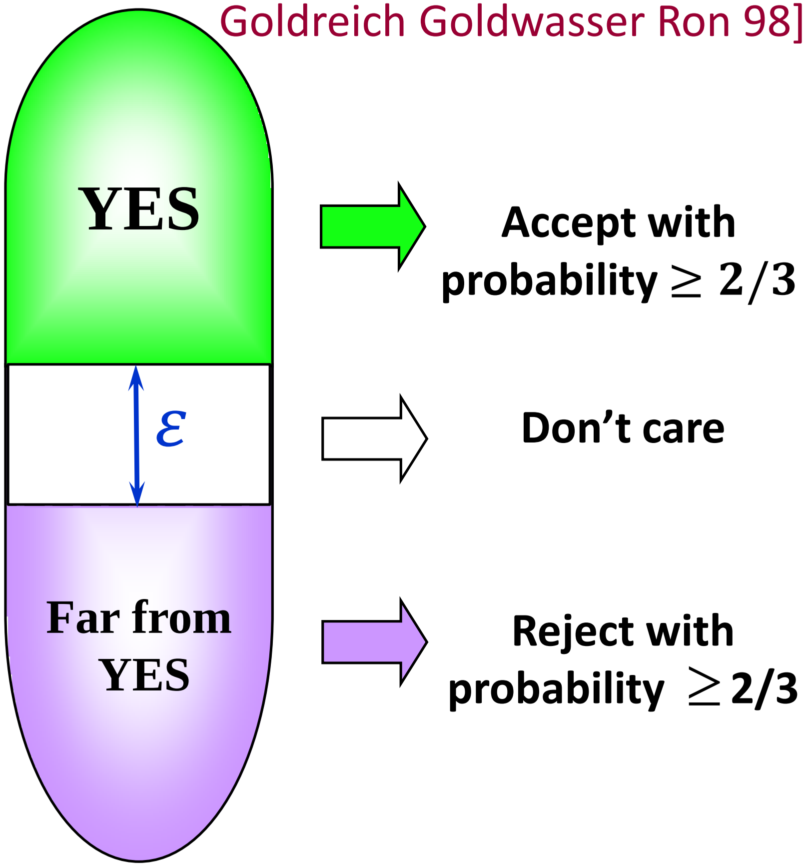
Summary: q(testing k-parity) $\ge \Omega(k)$ for $k \le n/2$

Current Directions

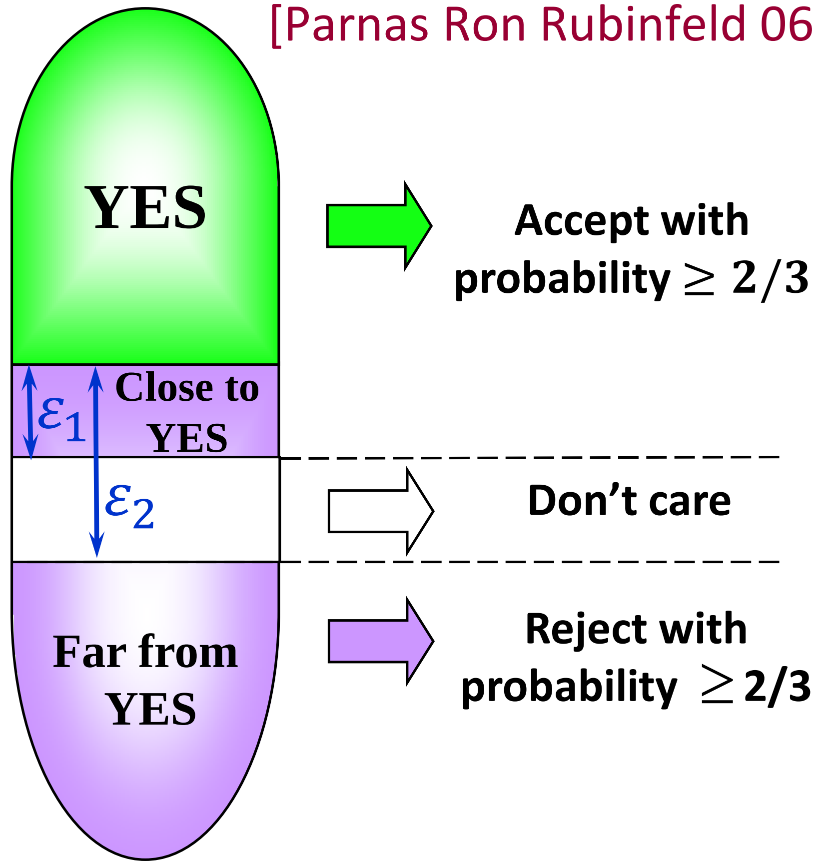
- Characterization
  - Which classes of properties are testable?
- Relationships with other computational tasks
  - e.g., learning, tolerant testing, local property reconstruction
- Simpler access to the input
  - e.g., nonadaptive and sample-based testers
- Relaxing the oracle assumption
  - distributional erasure-resilient testers
- New distance measures
  - L_p-testing

Tolerant Testing

Property Tester [Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]



Tolerant Property Tester [Parnas Ron Rubinfeld 06]

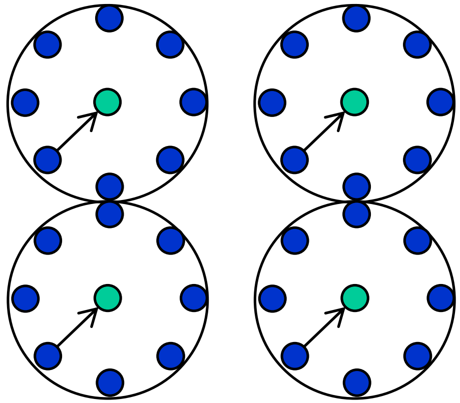


Equivalent to tolerant testing: estimating distance to the property.

Two objects are at distance ε = they differ in an ε fraction of places

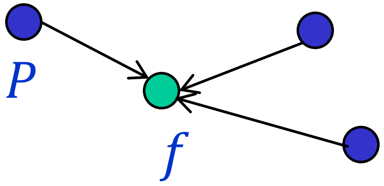
Sublinear-Time "Restoration" Models

- Local Decoding



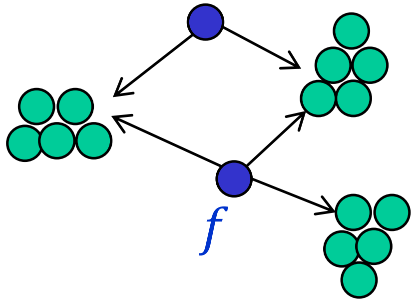
Input: A slightly corrupted codeword
Requirement: Recover individual bits of the closest codeword with a constant number of queries per recovered bit.

- Program Checking



Input: A program P computing f correctly on most inputs.
Requirement: Self-correct program P: for a given input x, compute $f(x)$ by making a few calls to P.

- Local Reconstruction

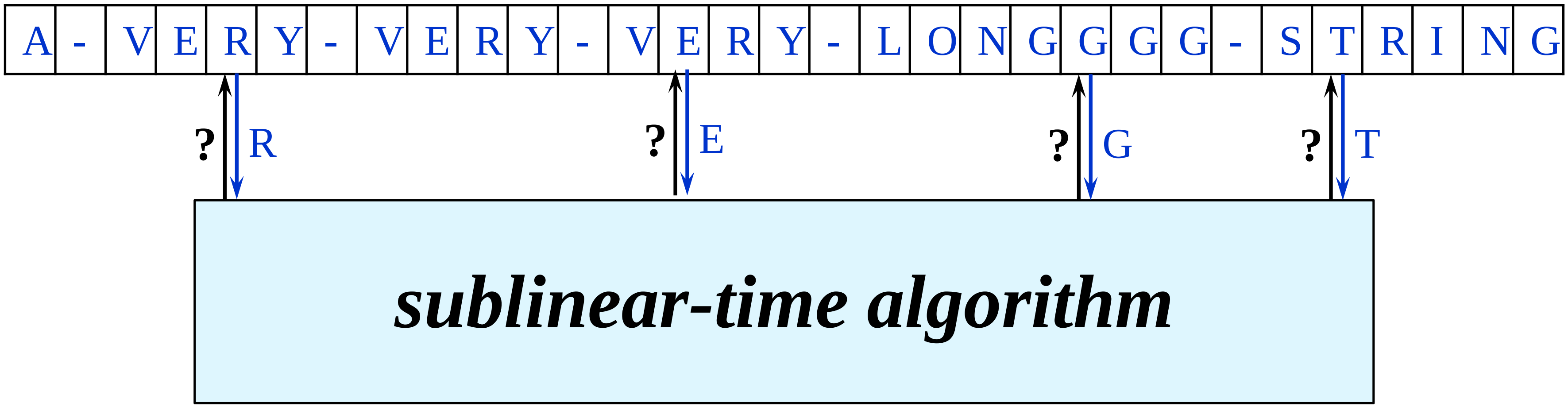


Input: Function f nearly satisfying some property P
Requirement: Reconstruct function f to ensure that the reconstructed function g satisfies P, changing f only when necessary. For each input x, compute $g(x)$ with a few queries to f.

New models for data access

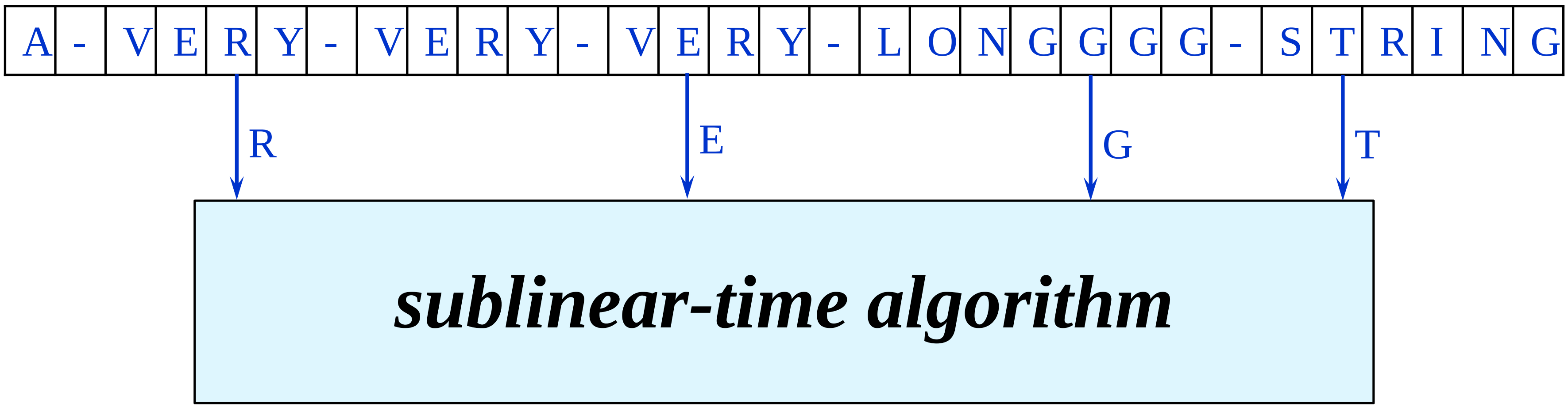
Relaxing the Oracle Assumption

A Sublinear-Time Algorithm



Is it always possible to be able to read any location of the input?

Distributional Assumptions on the Access



- **Sample-based testers** [Goldreich Goldwasser Ron, Goldreich Ron]

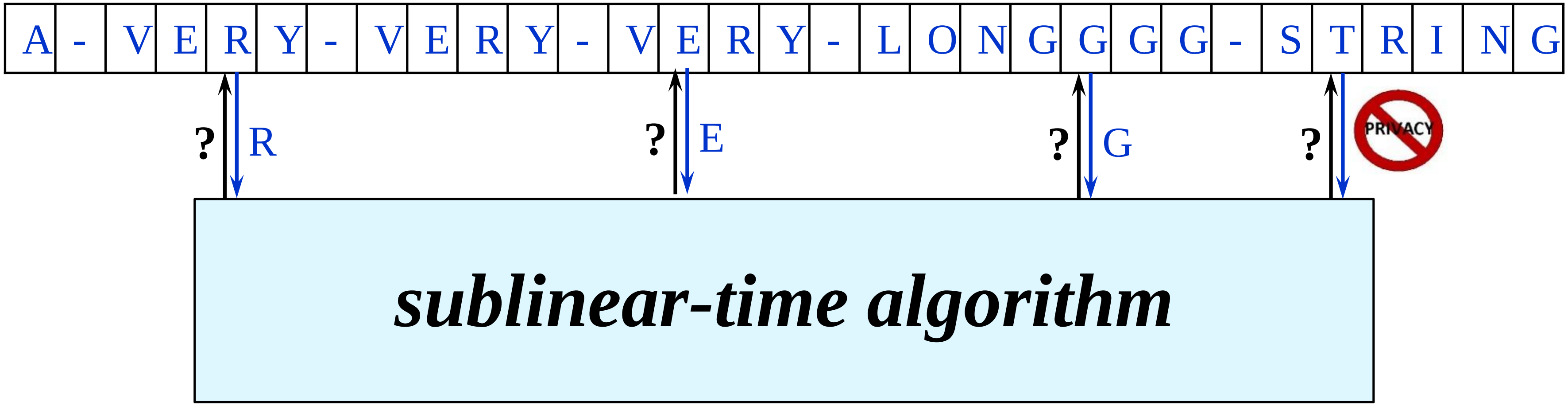
  can access only independent (labeled) samples from the domain
- **Blocks-sample-based testers** [Berman Murzabulatov **R**]

  can access uniformly random blocks of pixels from input *image*
- **Active testers** [Balcan Blais Blum Yang]

  get a small sample of domain points

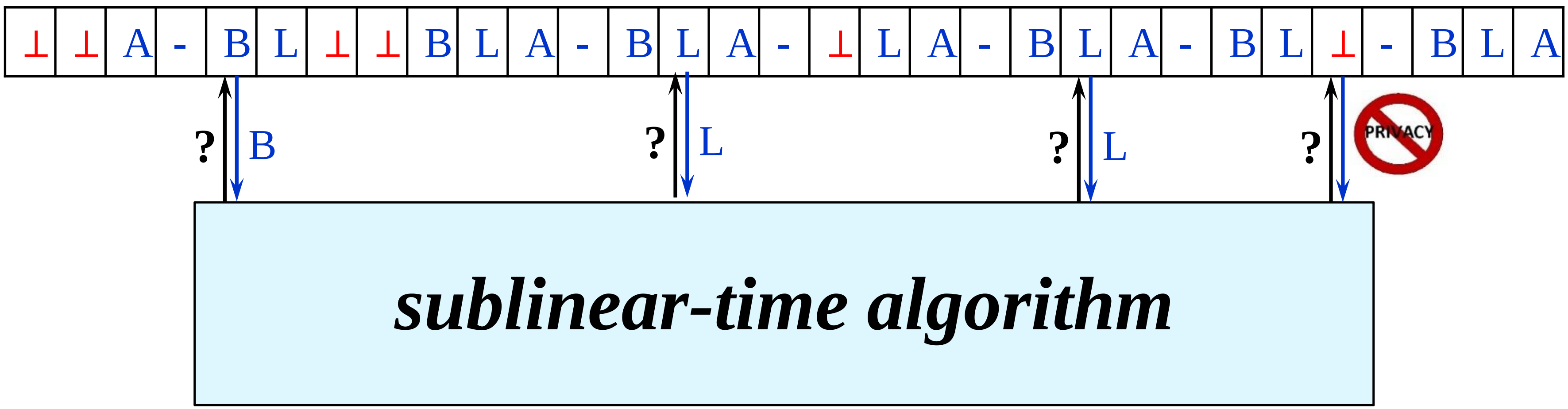
  can request labels only on points from the sample

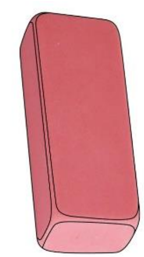
Testing with Faulty Oracles



- Erasure oracles
- Approximate oracles
- Malicious oracles

Erasure-Resilient Testing [Dixit R Thakurta Varma]





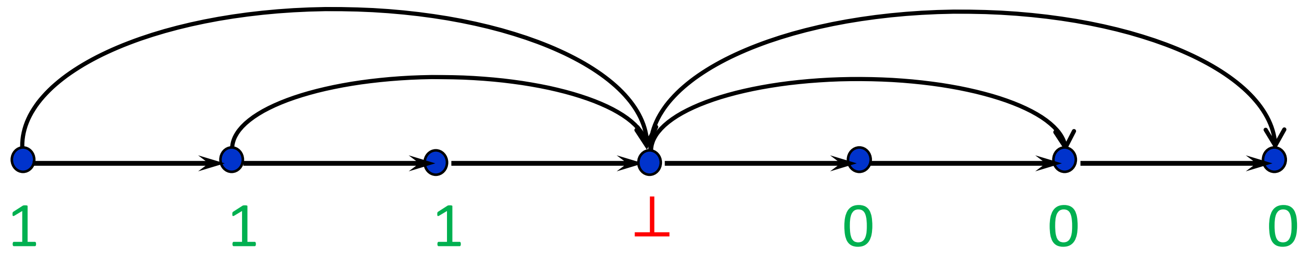
- α fraction of the input is erased adversarially
- Algorithm does not know in advance what's erased
- Is it possible that the input satisfied the property?

Automatic Erasure-Resilience?

Is every sublinear-time algorithm automatically erasure resilient?

2-Spanner-Based Test for Sortedness

Pick a random edge (i, j) from the 2-spanner and **reject** if $x_i > x_j$.



- This test detects a mistake only if midpoint is picked.
- If it is erased, it will fail.
- Not resilient even to 1 erasure!

Erasure-Resilient Property Testers

- We designed an tester for sortedness resilient to an α fraction of erasures that runs in time $\boldsymbol{O}\left(\frac{\log n}{(1-\alpha)\varepsilon}\right)$
- It is based on a binary search with random pivots
- Erasure-resilient algorithms for monotonicity, Lipschitz, convexity, bounded-derivative properties

Can all testable properties be tested in the presence of erasures?

- Separation between standard and erasure-resilient model

New measures of accuracy guaranties

L_p-testing [Berman **R** Yaroslavtsev 14]

Which Stocks Grew Steadily?

For some application with real-valued data
we should use L_1 or L_2 instead of Hamming distance

Source: http://finance.google.com

New L_p-Testing Model for Real-Valued Data

- **Generalizes standard testing**
- **Compatible with existing PAC-style learning models** (preprocessing for model selection)
- **Our L_p-testers beat lower bounds for standard testers**

➢ **E.g., L_1-testing sortedness takes time $\Theta\left(\frac{1}{\varepsilon}\right)$ instead of $\Theta\left(\frac{\log n}{\varepsilon}\right)$**

➢ **L_1-distance to nearest monotone function (L_1-isotonic regression) can be estimated within $\pm\varepsilon$ in time $\Theta\left(\frac{1}{\varepsilon^2}\right)$**

Open:

Can we perform other computational tasks on real data

(such as local reconstruction)

with accuracy measured w.r.t. L_p distances?

Current Directions

- Characterization
  - Which classes of properties are testable?
- Relationships with other computational tasks
  - e.g., learning, tolerant testing, local property reconstruction
- Simpler access to the input
  - e.g., nonadaptive and sample-based testers
- Relaxing the oracle assumption
  - distributional erasure-resilient testers
- New distance measures
  - L_p-testing

Conclusion

- Properties that admit sublinear-time testers are everywhere
- Algorithms are often simple
- Analysis requires creation of interesting combinatorial, geometric and algebraic tools
- Unexpected connections to other areas
- Many open questions