Quantum query complexity and the adversary bound
Part IV: Further topics

Alexander Belov
University of Latvia

22nd EWSCS, 5-10 March 2017, Palmse

Beyond non-adaptivity
  - Motivation
  - Adaptivity
  - Construction
  - Complexity
  - Graph collision
    - State of the Art
    - Motivation

k-distinctness

Group Testing

Beyond non-adaptivity

Motivation

Beyond non-adaptivity
Motivation
Adaptivity
Construction
Complexity
Graph collision
State of the Art
Motivation
k-distinctness
Group Testing

We want to solve the k-threshold problem ($k = O(1)$):

Given a string $x \in \{0,1\}^n$, detect if its Hamming weight is $\geq k$.

- From Lecture I, its quantum query complexity is $\Theta(\sqrt{n})$.
- It has the k-subset certificate structure.
- Hence, its learning graph complexity is $\Theta(n^{k/(k+1)})$.

We want to fix this issue.

Adaptivity

Beyond non-adaptivity
Motivation
Adaptivity
Construction
Complexity
Graph collision
State of the Art
Motivation
k-distinctness
Group Testing

- In Lectures II and III we considered non-adaptive learning graph.
- The weight of the arc from S to $S \cup \{j\}$ does not depend on anything.

- The weight can depend on the values of the variables in S.
- Adaptive learning graphs.

Construction (from Lecture II)

Beyond non-adaptivity
  Motivation
  Adaptivity
  Construction
  Complexity
  Graph collision
    State of the Art
    Motivation
k-distinctness
Group Testing

For each arc e from S to $S \cup \{j\}$, we define a block-diagonal matrix

$$X_j^e = \sum_{\alpha} Y_\alpha,$$

where the sum is over all assignments α on S.

Each Y_α is defined as $\psi\psi^*$, where (w_e is the weight of e):

$$\psi[\![z]\!] = \begin{cases} p_e(z)/\sqrt{w_e}, & f(z) = 1, \text{ and } z \text{ satisfies } \alpha; \\ \sqrt{w_e}, & f(z) = 0, \text{ and } z \text{ satisfies } \alpha; \\ 0, & \text{otherwise.} \end{cases}$$

Finally, we define

$$X_j = \sum_{e \text{ loads } j} X_j^e.$$

Construction (from Lecture II)

Beyond non-adaptivity
  Motivation
  Adaptivity
  Construction
  Complexity
  Graph collision
    State of the Art
    Motivation
k-distinctness
Group Testing

For each arc e from S to $S \cup \{j\}$, we define a block-diagonal matrix

$$X_j^e = \sum_{\alpha} Y_\alpha,$$

where the sum is over all assignments α on S.

Each Y_α is defined as $\psi\psi^*$, where ($w_{e,\alpha}$ is the weight of e on α):

$$\psi[\![z]\!] = \begin{cases} p_e(z)/\sqrt{w_{e,\alpha}}, & f(z) = 1, \text{ and } z \text{ satisfies } \alpha; \\ \sqrt{w_{e,\alpha}}, & f(z) = 0, \text{ and } z \text{ satisfies } \alpha; \\ 0, & \text{otherwise.} \end{cases}$$

Finally, we define

$$X_j = \sum_{e \text{ loads } j} X_j^e.$$

Complexity

Beyond non-adaptivity
Motivation
Adaptivity
Construction
Complexity
Graph collision
State of the Art
Motivation
k-distinctness
Group Testing

The complexity on input z is:

$$\begin{cases} \displaystyle\sum_{e:S\to S\cup\{j\}} w_{e,z|_S}, & \text{if } f(z) = 0; \\ \displaystyle\sum_{e:S\to S\cup\{j\}} \frac{p_e(z)^2}{w_{e,z|_S}}, & \text{if } f(z) = 1. \end{cases}$$

Example for the 2-threshold problem (whiteboard).

Graph collision

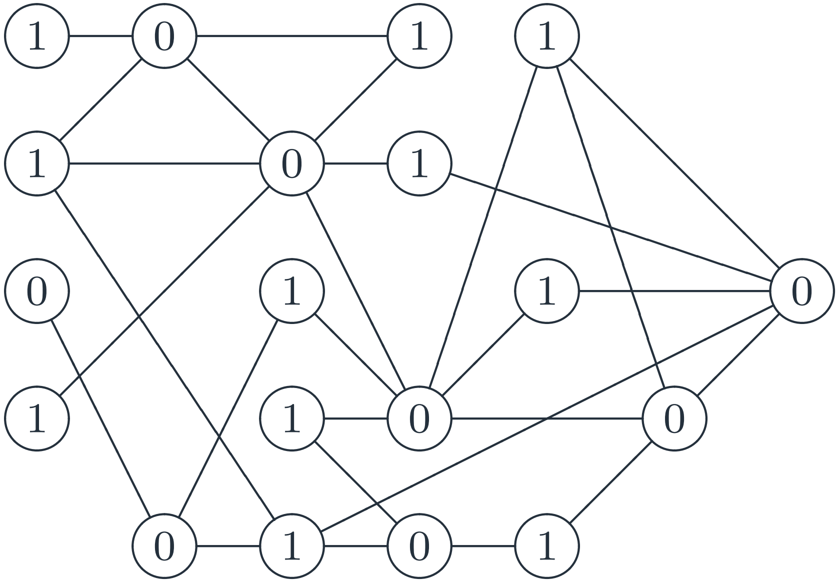
Beyond non-adaptivity
Motivation
Adaptivity
Construction
Complexity
Graph collision
State of the Art
Motivation
k-distinctness
Group Testing

Definition:

Graph G on n vertices: part of the problem, not input.

Given $x_i \in \{0,1\}$ for $i \in V$, detect if there exists an edge uv such that

$$x_u = x_v = 1.$$



Graph collision

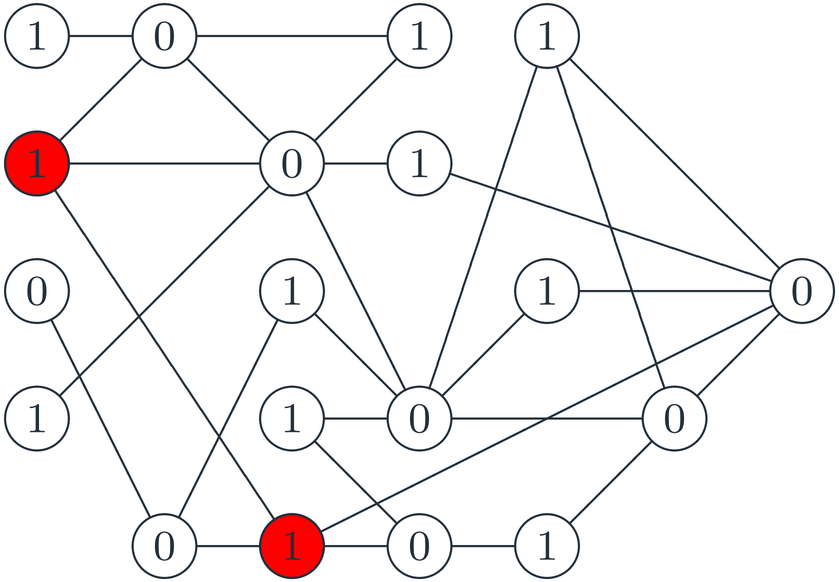
Beyond non-adaptivity
Motivation
Adaptivity
Construction
Complexity
Graph collision
State of the Art
Motivation
k-distinctness
Group Testing

Definition:

Graph G on n vertices: part of the problem, not input.

Given $x_i \in \{0, 1\}$ for $i \in V$, detect if there exists an edge uv such that

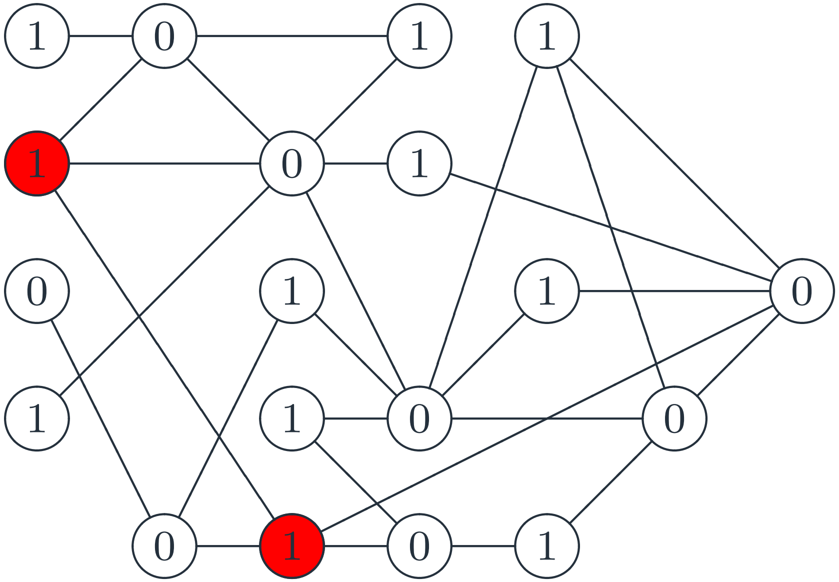
$$x_u = x_v = 1.$$



Graph collision: State of the Art

Beyond non-adaptivity
Motivation
Adaptivity
Construction
Complexity
Graph collision
State of the Art
Motivation
k-distinctness
Group Testing

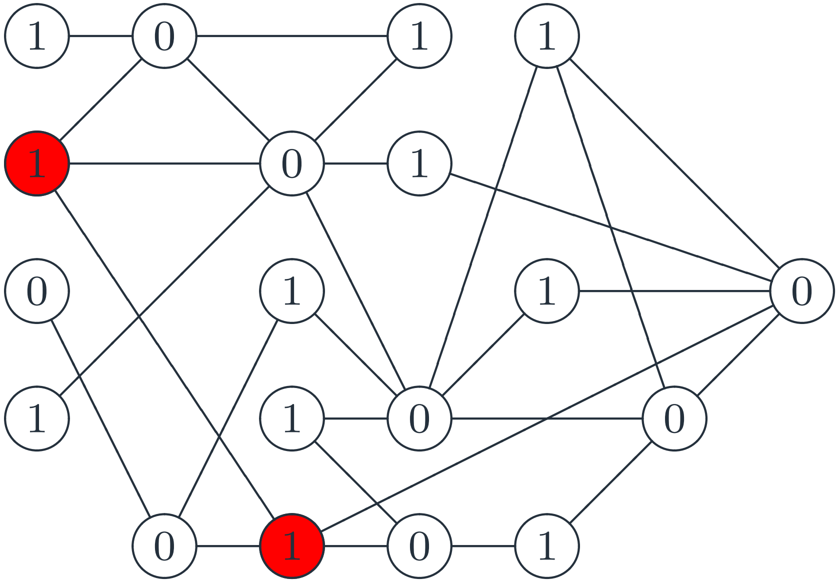
- 1-certificate complexity is 2.
- quantum query complexity is $O(n^{2/3})$.
- The best known lower bound is $\Omega(\sqrt{n})$.
- Easy for random graphs G.



Graph collision: Motivation

Beyond non-adaptivity
Motivation
Adaptivity
Construction
Complexity
Graph collision
State of the Art
Motivation
k-distinctness
Group Testing

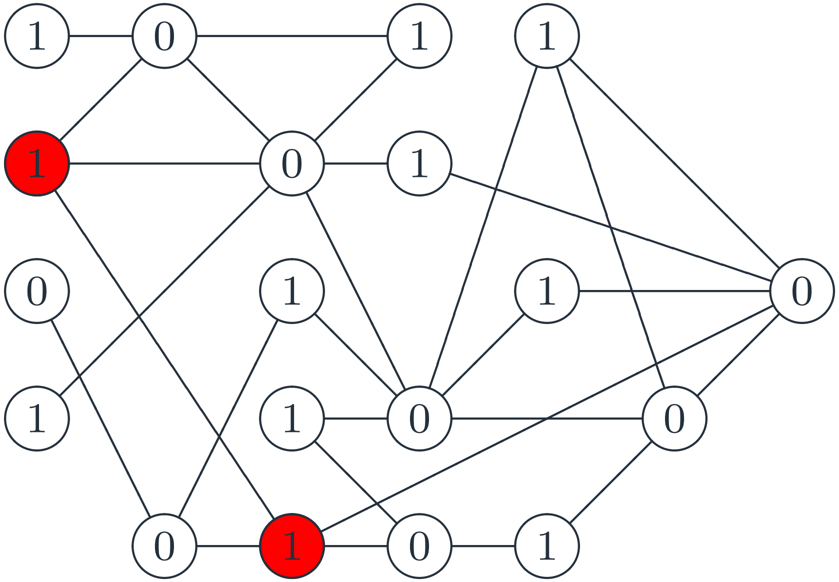
- Functions with 1-certificate complexity $k = O(1)$ solvable in $O(n^{k/(k+1)})$ quantum queries.
- This is tight, cf. k-sum problem.
- But the alphabet of the latter is large.



Graph collision: Motivation

- Functions with 1-certificate complexity $k = O(1)$ solvable in $O(n^{k/(k+1)})$ quantum queries.
- This is tight, cf. k-sum problem.
- But the alphabet of the latter is large.

What about Boolean functions with 1-certificate complexity $k = O(1)$?



Beyond non-adaptivity
k-distinctness
Definition
Learning graph
Complexity of the Final Stage
Distillation
Total complexity
Group Testing

k-distinctness

Definition

k-distinctness problem:
Given $x_1, \ldots, x_n \in [q]$, detect whether there exist $a_1, \ldots, a_k \in [n]$, all distinct, such that $x_{a_1} = x_{a_2} = \cdots = x_{a_k}$.

- Can be solved in $O(n^{k/(k+1)})$ quantum queries.
- Best known lower bound is $\Omega(n^{2/3})$.

We solve this problem in

$$O\left(n^{1-2^{k-2}/(2^k-1)}\right) = o(n^{3/4})$$

quantum queries.

Learning graph

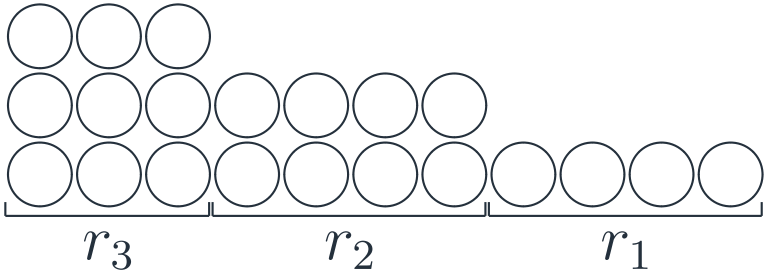
Let $a_1, \ldots, a_k$ be a 1-certificate in the input.

Like in Lecture II, the last k steps in the learning graph are:

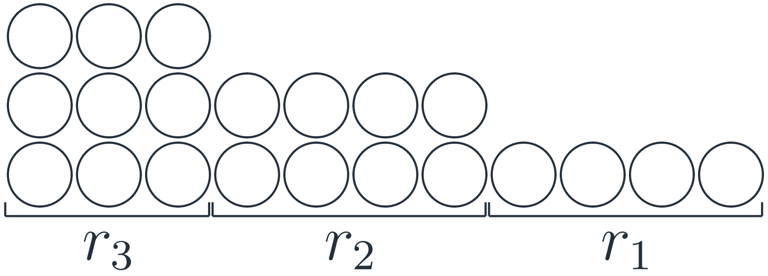
$\vdots$
Load a_1
Load a_2
$\vdots$
Load a_k

Let the vertices of the learning graphs ($\subseteq [n]$) contain

r_1 unique elements, r_2 pairs of equal elements, $\ldots$, r_{k-1} $(k-1)$-tuples of equal elements.



Complexity of the Final Stage



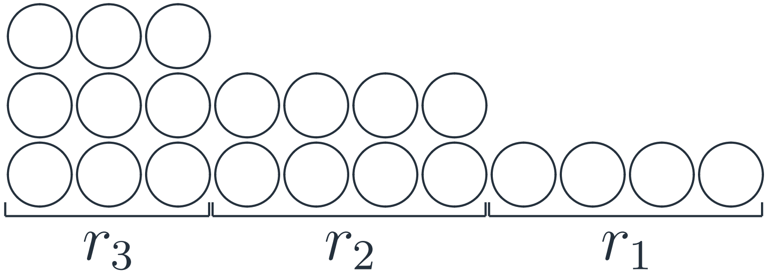
Claim. Complexity of loading $a_1, \ldots, a_k$ is

$$O(n/\sqrt{\min\{r_1, \ldots, r_{k-1}\}}).$$

Proof. When a_i is loaded, the $(i-1)$-tuple of equal elements $\{a_1, \ldots, a_{i-1}\}$ is hidden among $r_{i-1}+1$ such tuples.

- If $S \subseteq [n]$ of size r is chosen uniformly at random, r_{k-1} is very small: $r_{k-1} \approx n \cdot r^{k-1}/n^{k-1}$.
- We want to distill subsets with large r_{k-1}.

Distillation



1. Load $(r_1 + \cdots + r_{k-1})$ elements.

2. Load $r_2 + \cdots + r_{k-1}$ elements equal to already loaded elements.

3. Load $r_3 + \cdots + r_{k-1}$ elements equal to two loaded elements.

$\vdots$

$k-1$. Load r_{k-1} elements equal to $k-2$ already loaded elements.

Distillation

1. Load $(r_1 + \cdots + r_{k-1})$ elements.
2. Load $r_2 + \cdots + r_{k-1}$ elements equal to already loaded elements.
3. Load $r_3 + \cdots + r_{k-1}$ elements equal to two loaded elements.

$\vdots$

$k-1$. Load r_{k-1} elements equal to $k-2$ already loaded elements.

We may assume there are $\Omega(n)$ $(k-1)$-tuples of equal elements.
Assume also $r_1 > r_2 > \cdots > r_{k-1}$.
Then, complexity of the distillation stage is:

$$r_1 + r_2\sqrt{\frac{n}{r_1}} + r_3\sqrt{\frac{n}{r_2}} + \cdots + r_{k-1}\sqrt{\frac{n}{r_{k-2}}}.$$

Total complexity

Assume $r_1 > r_2 > \cdots > r_{k-1}$.

Complexity of the distillation stage is:

$$r_1 + r_2\sqrt{\frac{n}{r_1}} + r_3\sqrt{\frac{n}{r_2}} + \cdots + r_{k-1}\sqrt{\frac{n}{r_{k-2}}}.$$

Complexity of the final stage is

$$\frac{n}{\sqrt{r_{k-1}}}.$$

Total complexity is optimized to

$$O\left(n^{1-2^{k-2}/(2^k-1)}\right) = o(n^{3/4}).$$

Beyond non-adaptivity
k-distinctness
Group Testing
  Problem Formulation
  Examples
  Adversary Bound
  Probabilistic Language
  Hard Case
    Constraint
    Objective Value
    Analysis
  General Case
    Analysis
  Gapped version

Group Testing

Problem Formulation

Fixed: Symmetric Boolean function $h\colon \{0,1\}^k \to \{0,1\}$.

Given: Oracle access to $f_A\colon \{0,1\}^n \to \{0,1\}$ with $n \gg k$ defined by

$$f_A(x) = h(x_A)$$

for some k-subset A.

Task: Learn the function, i.e., find A.

- We identify $x \in \{0,1\}^n$ with the subset $S \subseteq [n]$.
- There are $\binom{n}{k}$ possible outcomes.
- Requires

$$\log \binom{n}{k} = \Omega\left(k \log \frac{n}{k}\right)$$

randomised queries.

Examples

Beyond non-adaptivity
k-distinctness
Group Testing
  Problem Formulation
  Examples
  Adversary Bound
  Probabilistic Language
  Hard Case
    Constraint
    Objective Value
    Analysis
  General Case
    Analysis
  Gapped version

Bernstein-Vazirani problem (1993)

Solve the case of $h = \text{XOR}$ in 1 quantum query exactly.

Examples

Beyond non-adaptivity

k-distinctness

Group Testing

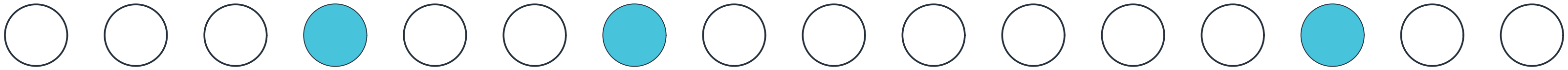
- Problem Formulation
- Examples
- Adversary Bound
- Probabilistic Language
- Hard Case
  - Constraint
  - Objective Value
  - Analysis
- General Case
  - Analysis
- Gapped version

Bernstein-Vazirani problem (1993)

Solve the case of $h =$ XOR in 1 quantum query exactly.

(Combinatorial) Group Testing problem, Dorfman (1943)

The case of $h =$ OR.



Examples

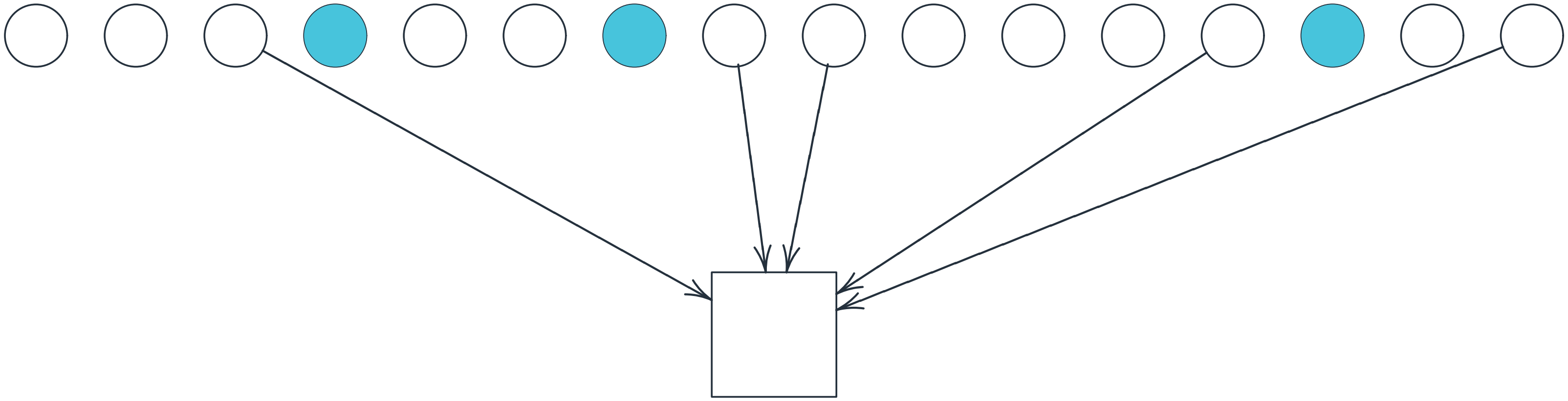
Beyond non-adaptivity
k-distinctness
Group Testing
Problem Formulation
Examples
Adversary Bound
Probabilistic Language
Hard Case
Constraint
Objective Value
Analysis
General Case
Analysis
Gapped version

Bernstein-Vazirani problem (1993)

Solve the case of $h =$ XOR in 1 quantum query exactly.

(Combinatorial) Group Testing problem, Dorfman (1943)

The case of $h =$ OR.



Examples

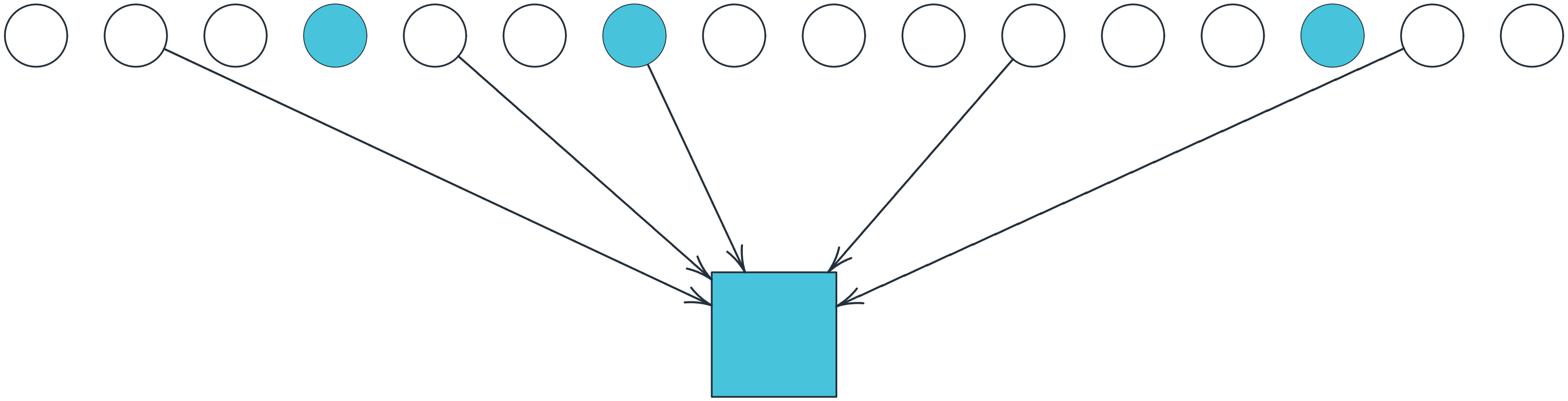
Beyond non-adaptivity
k-distinctness
Group Testing
Problem Formulation
Examples
Adversary Bound
Probabilistic Language
Hard Case
Constraint
Objective Value
Analysis
General Case
Analysis
Gapped version

Bernstein-Vazirani problem (1993)

Solve the case of $h = \text{XOR}$ in 1 quantum query exactly.

(Combinatorial) Group Testing problem, Dorfman (1943)

The case of $h = \text{OR}$.

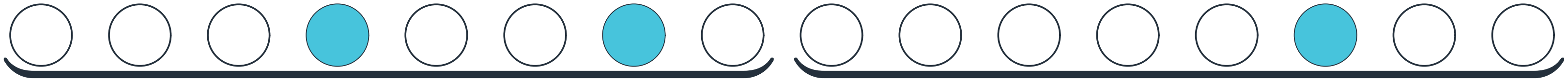


Examples

Beyond non-adaptivity
k-distinctness
Group Testing
Problem Formulation
Examples
Adversary Bound
Probabilistic Language
Hard Case
Constraint
Objective Value
Analysis
General Case
Analysis
Gapped version

Bernstein-Vazirani problem (1993)
Solve the case of $h =$ XOR in 1 quantum query exactly.

(Combinatorial) Group Testing problem, Dorfman (1943)
The case of $h =$ OR.



Examples

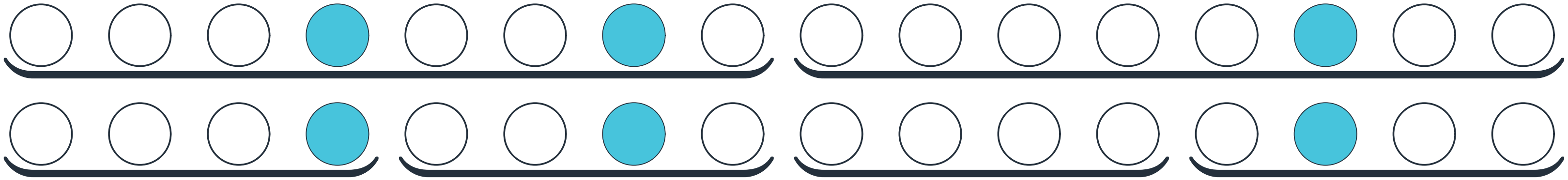
Beyond non-adaptivity
k-distinctness
Group Testing
  Problem Formulation
  Examples
  Adversary Bound
  Probabilistic Language
  Hard Case
    Constraint
    Objective Value
    Analysis
  General Case
    Analysis
  Gapped version

Bernstein-Vazirani problem (1993)

Solve the case of $h =$ XOR in 1 quantum query exactly.

(Combinatorial) Group Testing problem, Dorfman (1943)

The case of $h =$ OR.



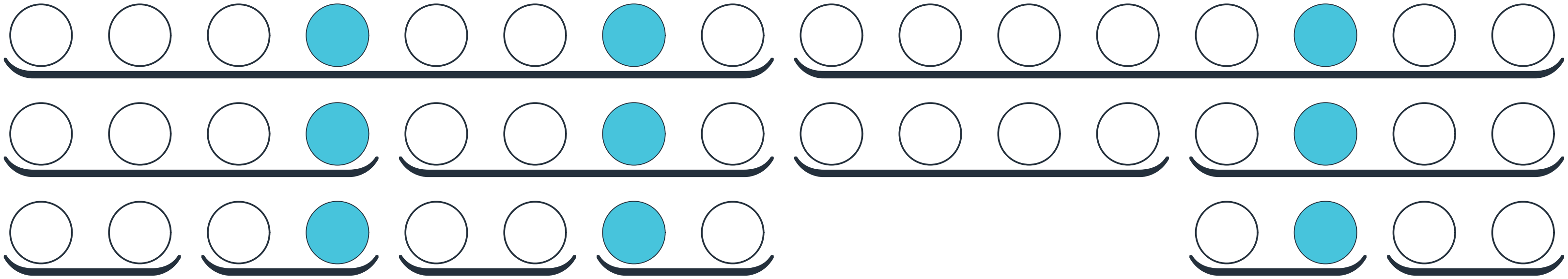
Examples

Bernstein-Vazirani problem (1993)

Solve the case of $h = $ XOR in 1 quantum query exactly.

(Combinatorial) Group Testing problem, Dorfman (1943)

The case of $h = $ OR.



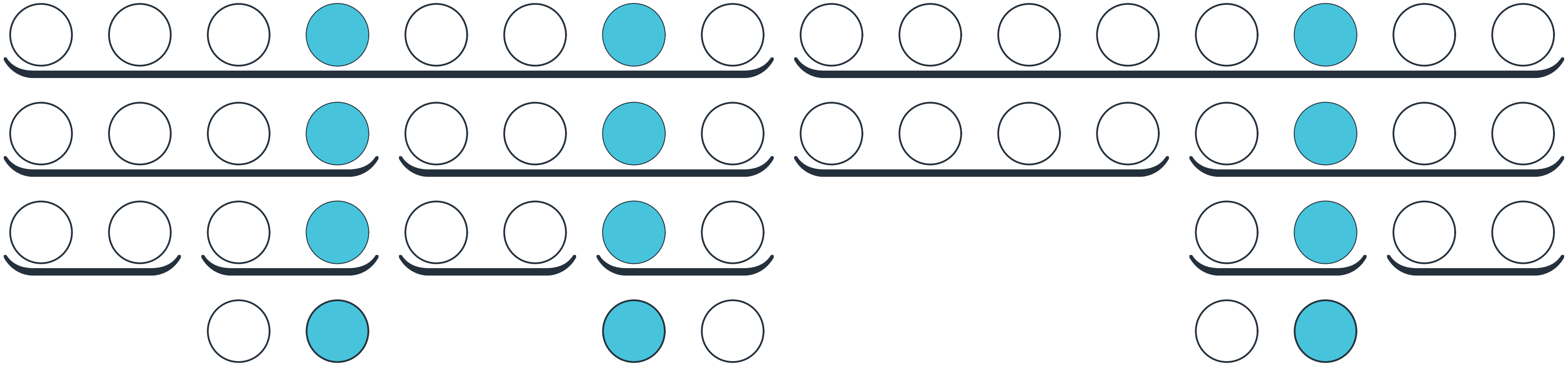
Examples

Bernstein-Vazirani problem (1993)

Solve the case of $h =$ XOR in 1 quantum query exactly.

(Combinatorial) Group Testing problem, Dorfman (1943)

The case of $h =$ OR.



Examples

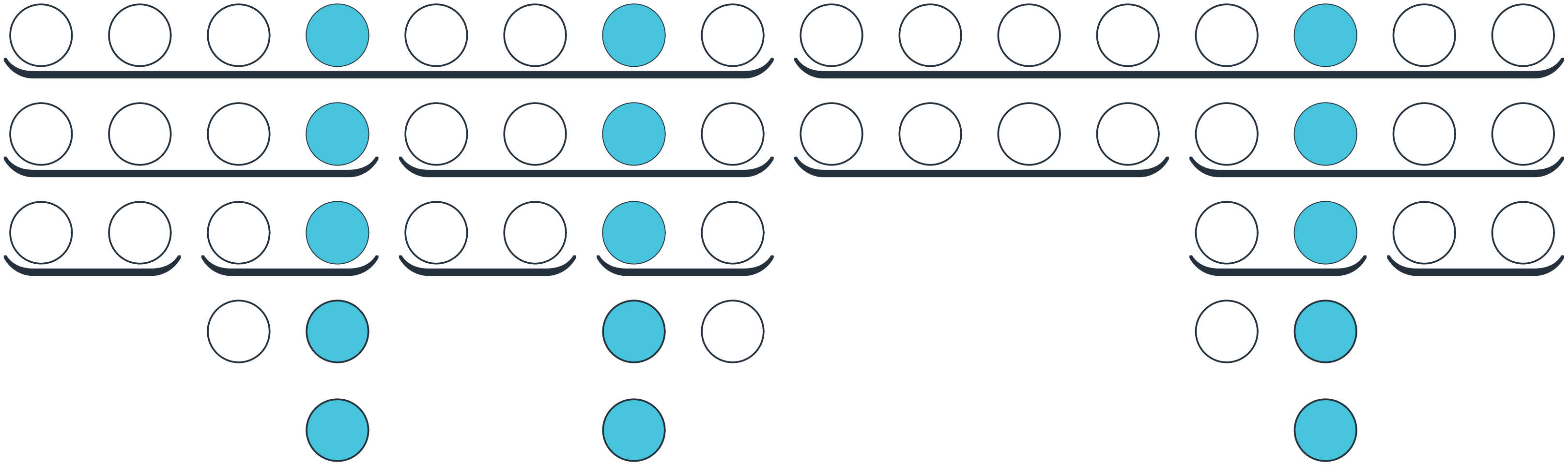
Beyond non-adaptivity
k-distinctness
Group Testing
Problem Formulation
Examples
Adversary Bound
Probabilistic Language
Hard Case
Constraint
Objective Value
Analysis
General Case
Analysis
Gapped version

Bernstein-Vazirani problem (1993)

Solve the case of $h = \text{XOR}$ in 1 quantum query exactly.

(Combinatorial) Group Testing problem, Dorfman (1943)

The case of $h = \text{OR}$.



Gives $O(k \log n)$ algorithm. Can be reduced to $O\left(k \log \frac{n}{k}\right)$.

Adversary Bound

Beyond non-adaptivity
k-distinctness
Group Testing
Problem Formulation
Examples
Adversary Bound
Probabilistic Language
Hard Case
Constraint
Objective Value
Analysis
General Case
Analysis
Gapped version

minimise $\max_{A\in\mathcal{C}} \sum_{S\subseteq[n]} X_S[\![A,A]\!]$

subject to $\sum_{S:f_A(S)\neq f_B(S)} X_S[\![A,B]\!] = 1$ for all $A \neq B$ in $\mathcal{C}$;

X_S is a p.s.d. $\mathcal{C} \times \mathcal{C}$ matrix for all $S \subseteq [n]$,

Probabilistic Language

Beyond non-adaptivity
k-distinctness
Group Testing
  Problem Formulation
  Examples
  Adversary Bound
  Probabilistic Language
  Hard Case
    Constraint
    Objective Value
    Analysis
  General Case
    Analysis
  Gapped version

$$X_S = \left(\boxed{\Pr[S]} \right)$$

$$\Pr[S] = p^{|S|}(1-p)^{n-|S|} \qquad \text{for some } 0 < p < 1$$

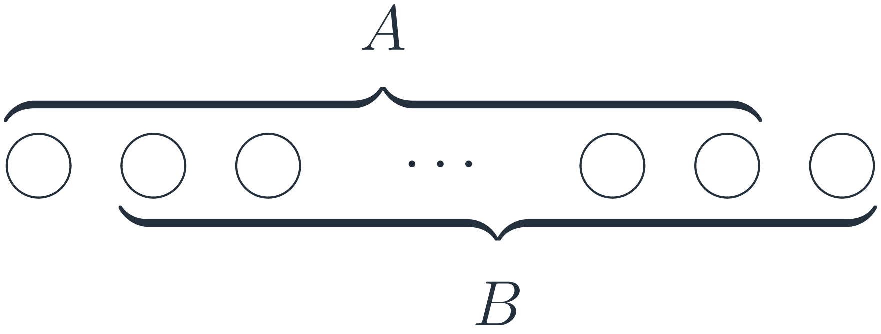
Which subsets A do we include?

Hard Case

We have constraint

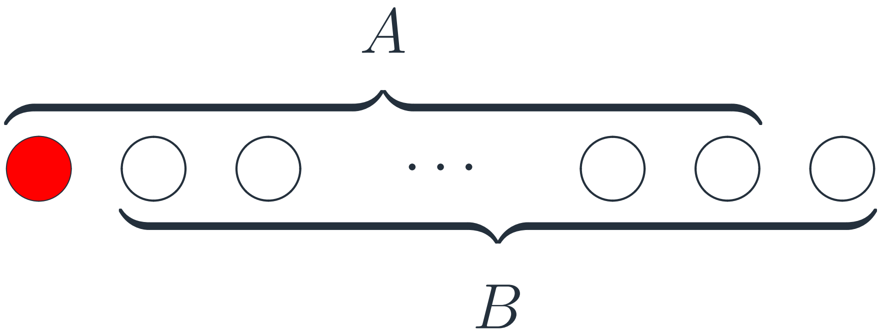
$$\sum_{S: f_A(S) \neq f_B(S)} X_S[\![A, B]\!] = 1.$$

“Hardest” when A and B differ in 1 element:



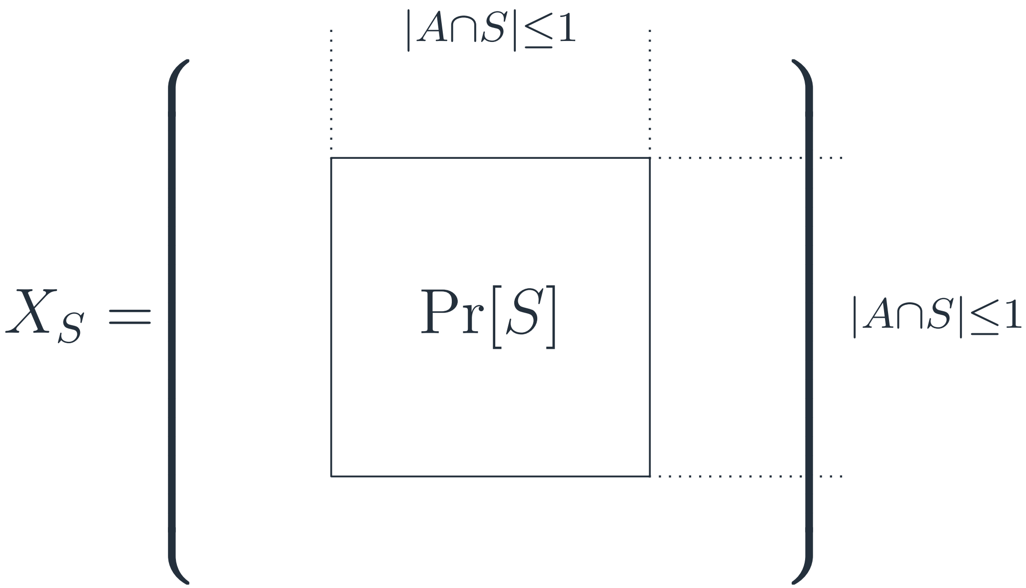
Hard Case: Constraint

$$\sum_{S: f_A(S)\neq f_B(S)} X_S[\![A, B]\!]$$ is the probability of $f_A(S) \neq f_B(S)$:



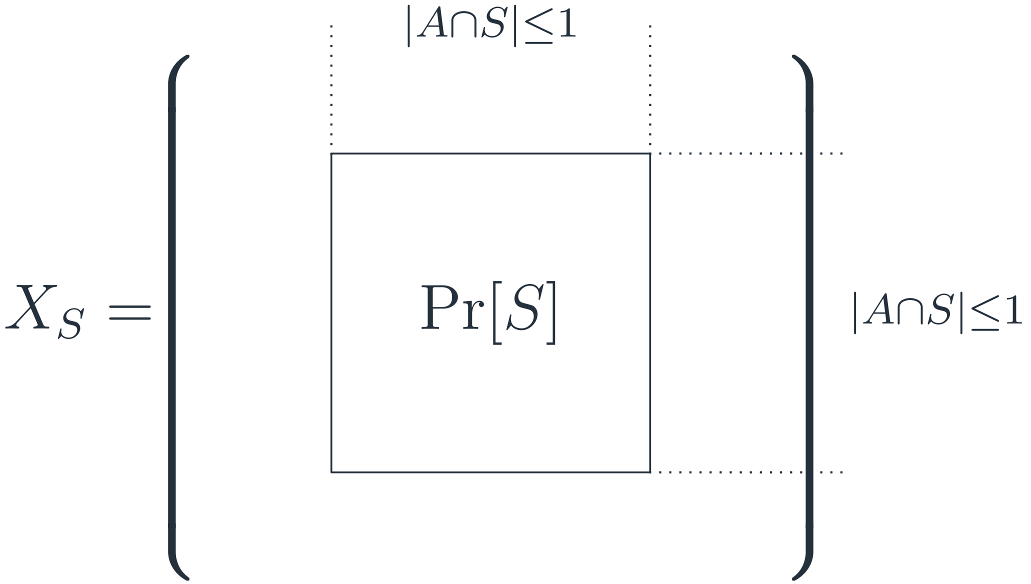
- It equals $2p(1-p)^k$.
- In X_S we include A satisfying $|A \cap S| \leq 1$.

Hard Case: Objective Value



$$
\begin{aligned}
\sum_{S \subseteq [n]} X_S[\![A, A]\!] &= \sum_{S : |S \cap A| = 0} X_S[\![A, A]\!] + \sum_{S : |S \cap A| = 1} X_S[\![A, A]\!] \\
&= \Pr_S[S \cap A = \emptyset] + \Pr_S[|S \cap A| = 1] \\
&= (1-p)^k + kp(1-p)^{k-1}.
\end{aligned}
$$

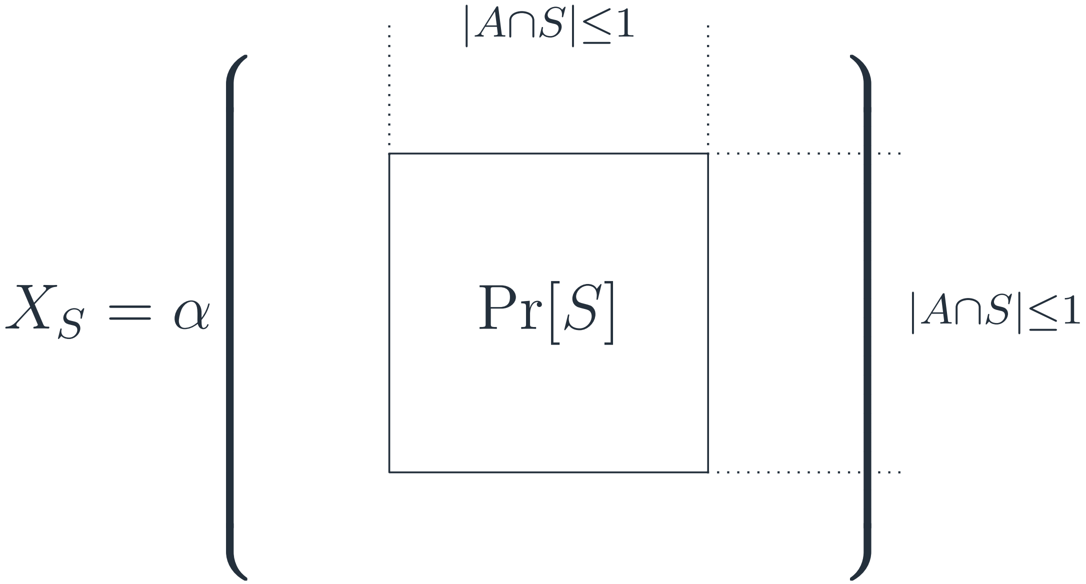
Hard Case: Analysis



Objective:
$$(1-p)^k$$
$$kp(1-p)^{k-1}$$

Constraint: $2p(1-p)^k$

Hard Case: Analysis



Objective: $\cancel{(1-p)^k}$ $1-p$

$\cancel{kp(1-p)^{k-1}}$ kp

Constraint: $\cancel{2p(1-p)^k}$ $2p(1-p)$

Hard Case: Analysis

$$X_S = \alpha \left(\begin{array}{cc} A\cap S=\emptyset & |A\cap S|=1 \\ \beta \Pr[S] & \Pr[S] \\ \Pr[S] & \frac{\Pr[S]}{\beta} \end{array} \right) \begin{array}{l} \\ A\cap S=\emptyset \\ |A\cap S|=1 \end{array}$$

Objective:

$$\begin{tikzcd} \cancel{(1-p)^k} & 1-p \arrow[dr] & \\ & & \sqrt{kp(1-p)} \\ \cancel{kp(1-p)^{k-1}} & kp \arrow[ur] & \end{tikzcd}$$

Constraint: $\cancel{2p(1-p)^k}$ $\qquad 2p(1-p)$

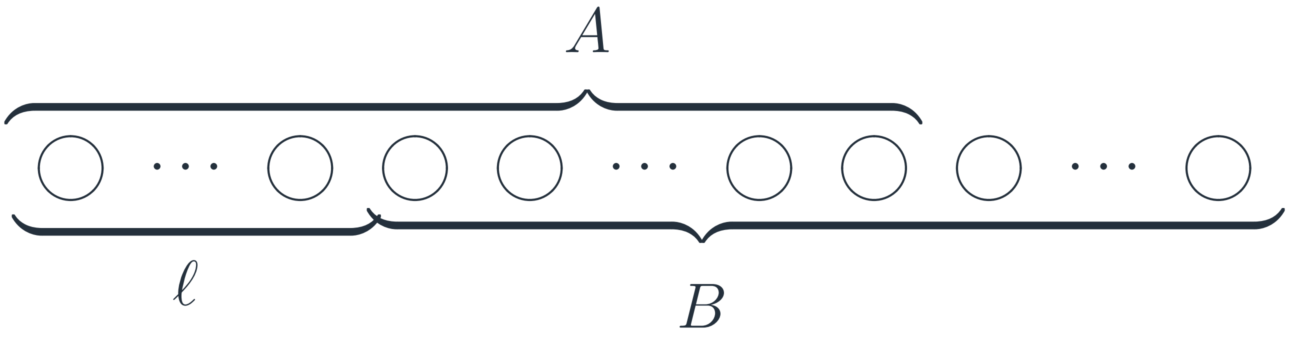
By plugging $p = 1/2$ and rescaling, we get complexity $O(\sqrt{k})$.

General Case

Beyond non-adaptivity
k-distinctness
Group Testing
  Problem Formulation
  Examples
  Adversary Bound
  Probabilistic Language
  Hard Case
    Constraint
    Objective Value
    Analysis
  General Case
    Analysis
  Gapped version

BUT!

What if A and B differ in $\ell > 1$ elements?

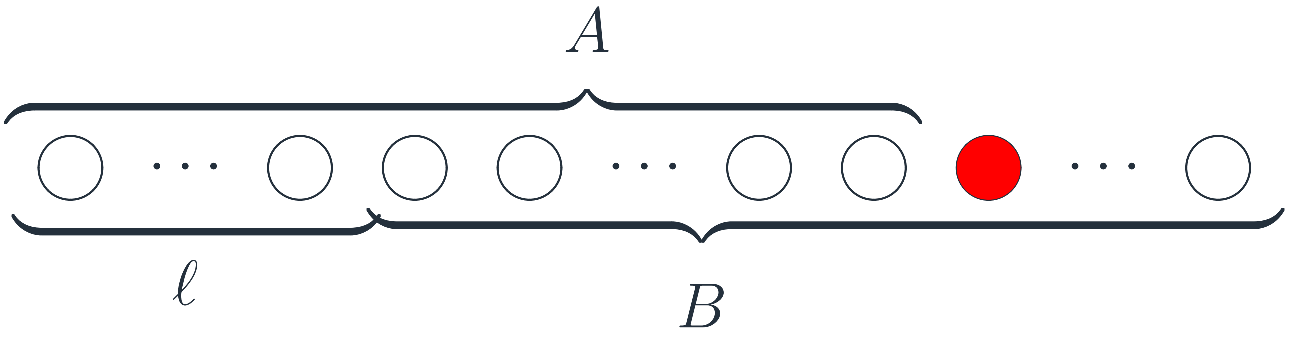


General Case

Beyond non-adaptivity
k-distinctness
Group Testing
  Problem Formulation
  Examples
  Adversary Bound
  Probabilistic Language
  Hard Case
    Constraint
    Objective Value
    Analysis
  General Case
    Analysis
  Gapped version

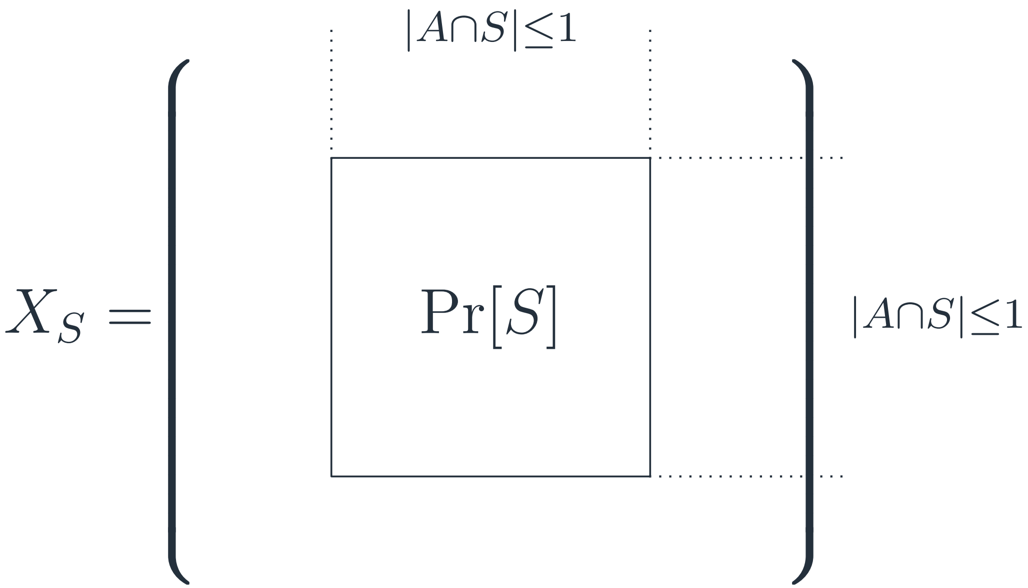
BUT!

What if A and B differ in $\ell > 1$ elements?



The probability is $2\ell p(1-p)^{k+\ell-1}$.

General Case: Analysis



Objective: $$(1-p)^k$$ $$kp(1-p)^{k-1}$$

Constraint: $2\ell p(1-p)^{k+\ell-1}$

General Case: Analysis

$$X_S = \alpha \left(\begin{array}{cc|c} A\cap S=\emptyset & |A\cap S|=1 & \\ \hline \beta\Pr[S] & \Pr[S] & A\cap S=\emptyset \\ \Pr[S] & \frac{\Pr[S]}{\beta} & |A\cap S|=1 \end{array} \right)$$

Objective:

$$\begin{array}{ll} \cancel{(1-p)^k} & 1/(2p) \\ \cancel{kp(1-p)^{k-1}} & k/(2(1-p)) \end{array} \longrightarrow \sqrt{\frac{k}{4p(1-p)}}$$

Constraint:

$$\cancel{2\ell p(1-p)^{k+\ell-1}} \quad \ell(1-p)^{\ell-1}$$

General Case: Analysis

Objective:

$$\begin{array}{ll} \cancel{(1-p)^k} & 1/(2p) \\ \cancel{kp(1-p)^{k-1}} & k/(2(1-p)) \end{array} \longrightarrow \sqrt{\frac{k}{4p(1-p)}}$$

Constraint:

$$\cancel{2\ell p(1-p)^{k+\ell-1}} \qquad \ell(1-p)^{\ell-1}$$

Now we integrate by p from 0 to 1:

$$X_S = \int_0^1 X_S(p)\,dp$$

$$\frac{\sqrt{k}}{2}\int_0^1 \frac{dp}{\sqrt{p(1-p)}} = \frac{\pi\sqrt{k}}{2}$$

$$\int_0^1 \ell(1-p)^{\ell-1}\,dp = 1.$$

Gapped version

Given: Oracle access to $f_A\colon \{0,1\}^n \to \{0,1\}$ with $n \gg k$ defined by

$$f_A(x) = OR(x_A)$$

for some subset A.

Task: Distinguish the cases

$$|A| \leq k \qquad \text{and} \qquad |A| \geq k + d$$

- Randomised query complexity is
  - ☐ $\Theta(k)$ for $d \leq \sqrt{k}$; and
  - ☐ $\Theta((k/d)^2)$ for $d \geq \sqrt{k}$.
- Quantum query complexity is $\Theta(\sqrt{k/d})$.
- We get a quartic improvement when $d = \sqrt{k}$.

Beyond non-adaptivity
k-distinctness
Group Testing
Problem Formulation
Examples
Adversary Bound
Probabilistic Language
Hard Case
Constraint
Objective Value
Analysis
General Case
Analysis
Gapped version

Thank you!