Noisy Computing of the Threshold Function

Nadim Ghaddar

University of Toronto

Joint work with Ziao Wang, Banghua Zhu, and Lele Wang

Remarkable 2025, Vector Institute
Toronto, Canada

March 4, 2025

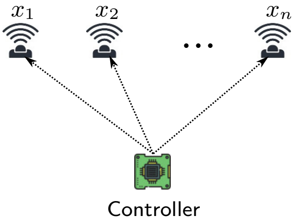
Motivation

- Computing in the presence of noise

Motivation

- Computing in the presence of noise

  E.g. n sensors make noisy measurements of signals $x_1, \ldots, x_n$



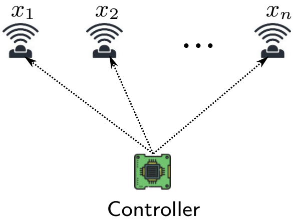
  Controller adaptively probes one of the sensors to make a measurement.

- Goal: Compute a function $f(x_1, \ldots, x_n)$ from noisy measurements

Motivation

- Computing in the presence of noise

  E.g. n sensors make noisy measurements of signals $x_1, \ldots, x_n$



  Controller adaptively probes one of the sensors to make a measurement.

- Goal: Compute a function $f(x_1, \ldots, x_n)$ from noisy measurements

- Applications
  - Fault tolerance
  - Active ranking
  - Recommendation systems
  - ...

Problem Statement (Threshold Function)

- Let $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$.
- Threshold function: For $\mathbf{x} \in \{0,1\}^n$,

$$\mathsf{TH}_k(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq k, \\ 0 & \text{otherwise.} \end{cases}$$

  Note: $\mathsf{TH}_1(\mathbf{x}) = \mathsf{OR}(\mathbf{x})$ and $\mathsf{TH}_{n/2}(\mathbf{x}) = \mathsf{MAJORITY}(\mathbf{x})$.

- **Goal:** Find an estimate of $\mathsf{TH}_k(\mathbf{x})$ using noisy bit readings.

Problem Statement (Threshold Function)

- Let $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$.
- Threshold function: For $\mathbf{x} \in \{0,1\}^n$,

$$\mathsf{TH}_k(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq k, \\ 0 & \text{otherwise.} \end{cases}$$

  Note: $\mathsf{TH}_1(\mathbf{x}) = \mathsf{OR}(\mathbf{x})$ and $\mathsf{TH}_{n/2}(\mathbf{x}) = \mathsf{MAJORITY}(\mathbf{x})$.

- **Goal:** Find an estimate of $\mathsf{TH}_k(\mathbf{x})$ using noisy bit readings.
  - At ith time step, submit query $U_i = j$ for some $j \in [n]$.
  - Receive noisy response

    $$Y_i = x_{U_i} \oplus Z_i,$$

    where $Z_i \sim \mathrm{Bern}(p)$, for some fixed and known $p < 1/2$.
  - After M queries, compute estimate $\widehat{\mathsf{TH}}_k$ of $\mathsf{TH}_k(\mathbf{x})$.

Problem Statement (Threshold Function)

- Let $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$.
- Threshold function: For $\mathbf{x} \in \{0,1\}^n$,

$$\mathsf{TH}_k(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq k, \\ 0 & \text{otherwise.} \end{cases}$$

  Note: $\mathsf{TH}_1(\mathbf{x}) = \mathsf{OR}(\mathbf{x})$ and $\mathsf{TH}_{n/2}(\mathbf{x}) = \mathsf{MAJORITY}(\mathbf{x})$.

- **Goal:** Find an estimate of $\mathsf{TH}_k(\mathbf{x})$ using noisy bit readings.
  - At ith time step, submit query $U_i = j$ for some $j \in [n]$.
  - Receive noisy response

    $$Y_i = x_{U_i} \oplus Z_i,$$

    where $Z_i \sim \mathrm{Bern}(p)$, for some fixed and known $p < 1/2$.
  - After M queries, compute estimate $\widehat{\mathsf{TH}}_k$ of $\mathsf{TH}_k(\mathbf{x})$.

- **Question:** How many queries are needed to find $\widehat{\mathsf{TH}}_k$ s.t.

$$\sup_{\mathbf{x}} \mathsf{P}(\widehat{\mathsf{TH}}_k \neq \mathsf{TH}_k(\mathbf{x})) \leq \delta?$$

Related Work

- Special case: OR function ($k = 1$)
  - $\Omega(n \log n)$ queries are necessary for non-adaptive query strategies[1][2][3]
  - $\Theta(n \log(1/\delta))$ queries are both necessary and sufficient for adaptive query strategies[4]

[1] R. L. Dobrushin and S. I. Ortyukov. "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements". In: *Problemy Peredachi Informatsii* 13.1 (1977), pp. 82–89.

[2] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis. "On a lower bound for the redundancy of reliable networks with noisy gates". In: *IEEE Trans. Inf. Theory* 37.3 (1991), pp. 639–643.

[3] P. Gács and A. Gál. "Lower bounds for the complexity of reliable Boolean circuits with noisy gates". In: *IEEE Trans. Inf. Theory* 40.2 (1994), pp. 579–583.

[4] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Related Work

- Special case: OR function ($k = 1$)
  - $\Omega(n \log n)$ queries are necessary for non-adaptive query strategies[1,2,3]
  - $\Theta(n \log(1/\delta))$ queries are both necessary and sufficient for adaptive query strategies[4]
- General k
  - Best known results: $\Theta(n \log(k/\delta))$ queries are both necessary and sufficient[4]

[1] R. L. Dobrushin and S. I. Ortyukov. "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements". In: *Problemy Peredachi Informatsii* 13.1 (1977), pp. 82–89.

[2] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis. "On a lower bound for the redundancy of reliable networks with noisy gates". In: *IEEE Trans. Inf. Theory* 37.3 (1991), pp. 639–643.

[3] P. Gács and A. Gál. "Lower bounds for the complexity of reliable Boolean circuits with noisy gates". In: *IEEE Trans. Inf. Theory* 40.2 (1994), pp. 579–583.

[4] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Related Work

- Special case: OR function ($k = 1$)
  - $\Omega(n \log n)$ queries are necessary for non-adaptive query strategies[1,2,3]
  - $\Theta(n \log(1/\delta))$ queries are both necessary and sufficient for adaptive query strategies[4]
- General k
  - Best known results: $\Theta(n \log(k/\delta))$ queries are both necessary and sufficient[4]

Exact dependence on p is not known in prior work.

[1] R. L. Dobrushin and S. I. Ortyukov. "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements". In: *Problemy Peredachi Informatsii* 13.1 (1977), pp. 82–89.

[2] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis. "On a lower bound for the redundancy of reliable networks with noisy gates". In: *IEEE Trans. Inf. Theory* 37.3 (1991), pp. 639–643.

[3] P. Gács and A. Gál. "Lower bounds for the complexity of reliable Boolean circuits with noisy gates". In: *IEEE Trans. Inf. Theory* 40.2 (1994), pp. 579–583.

[4] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Main Result

Theorem (Informal)

Suppose $k \le n/2$ *and* $\delta = o(1)$. *The optimal query complexity* M *satisfies*

$$(1 - o(1)) \frac{(n-k) \log \frac{k}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)} \le \mathsf{E}[M] \le (1 + o(1)) \frac{n \log \frac{k}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)},$$

where $D_{\mathrm{KL}}(p \| 1-p)$ *is the KL divergence between* $\mathrm{Bern}(p)$ *and* $\mathrm{Bern}(1-p)$.

[5] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Main Result

Theorem (Informal)

Suppose $k \le n/2$ and $\delta = o(1)$. The optimal query complexity M satisfies

$$(1 - o(1)) \frac{(n-k) \log \frac{k}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)} \le \mathsf{E}[M] \le (1 + o(1)) \frac{n \log \frac{k}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)},$$

where $D_{\mathrm{KL}}(p \| 1-p)$ is the KL divergence between $\mathrm{Bern}(p)$ and $\mathrm{Bern}(1-p)$.

- Bounds are tight when $k = o(n)$. E.g., when $k = 1$ (OR function),

$$\mathsf{E}[M] = (1 \pm o(1)) \frac{n \log \frac{1}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)}$$

- Multiplicative gap is ≤ 2 when $k = \Theta(n)$. E.g., when $k = n/2$ (MAJORITY function)

[5] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. “Computing with noisy information”. In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Main Result

Theorem (Informal)

Suppose $k \le n/2$ *and* $\delta = o(1)$. *The optimal query complexity* M *satisfies*

$$(1 - o(1))\frac{(n-k)\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p||1-p)} \le \mathsf{E}[M] \le (1 + o(1))\frac{n\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p||1-p)},$$

where $D_{\mathrm{KL}}(p||1-p)$ *is the KL divergence between* $\mathrm{Bern}(p)$ *and* $\mathrm{Bern}(1-p)$.

- Bounds are tight when $k = o(n)$. E.g., when $k = 1$ (OR function),

$$\mathsf{E}[M] = (1 \pm o(1))\frac{n\log\frac{1}{\delta}}{D_{\mathrm{KL}}(p||1-p)}$$

- Multiplicative gap is ≤ 2 when $k = \Theta(n)$. E.g., when $k = n/2$ (MAJORITY function)
- Comparison to existing bounds[5]: for $k = n^{1/3}$, $\delta = n^{-1/4}$, and $p = 1/3$,
  - Our result: $\mathsf{E}[M] \approx 2.5247 n\log n$
  - Existing bounds: $0.0506 n\log n \le \mathsf{E}[M] \le 433.7518 n\log n$

[5] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Lower Bound Proof Sketch

- Challenge: need to consider arbitrary adaptive querying strategies

Lower Bound Proof Sketch

- Challenge: need to consider arbitrary adaptive querying strategies
- Idea: For any querying strategy $\mathcal{A}$, consider an enhanced version $\mathcal{A}'$
  - $\mathcal{A}'$ accesses more information about each bit compared to $\mathcal{A}$
  - Analysis of $\mathcal{A}'$ is more tractable
- Lower bound on the number of queries for $\mathcal{A}'$ implies lower bound for $\mathcal{A}$

Enhanced Querying Strategy $\mathcal{A}'$

Key Fact

Fix $\epsilon = o(1)$. For any querying strategy $\mathcal{A}$ that uses $\frac{(n-k-\epsilon n)\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ queries, the number of bits that are queried for more than $\alpha := \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ times is at most $\beta := n - k - \epsilon n$.

[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Enhanced Querying Strategy $\mathcal{A}'$

Key Fact

Fix $\epsilon = o(1)$. For any querying strategy $\mathcal{A}$ that uses $\frac{(n-k-\epsilon n)\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ queries, the number of bits that are queried for more than $\alpha := \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ times is at most $\beta := n - k - \epsilon n$.

- Enhanced querying strategy $\mathcal{A}'$ consists of two phases[6]:
  1. Non-adaptive phase: Query each bit for α times
  2. Adaptive phase: Choose β bits adaptively, and noiselessly obtain their values

[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Enhanced Querying Strategy $\mathcal{A}'$

Key Fact

Fix $\epsilon = o(1)$. For any querying strategy $\mathcal{A}$ that uses $\frac{(n-k-\epsilon n)\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ queries, the number of bits that are queried for more than $\alpha := \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ times is at most $\beta := n - k - \epsilon n$.

- Enhanced querying strategy $\mathcal{A}'$ consists of two phases[6]:
  1. Non-adaptive phase: Query each bit for α times
  2. Adaptive phase: Choose β bits adaptively, and noiselessly obtain their values

- Any strategy $\mathcal{A}$ that uses $\frac{(n-k-\epsilon n)\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ queries can be simulated using $\mathcal{A}'$.

[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Enhanced Querying Strategy $\mathcal{A}'$

Key Fact

Fix $\epsilon = o(1)$. For any querying strategy $\mathcal{A}$ that uses $\frac{(n-k-\epsilon n)\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ queries, the number of bits that are queried for more than $\alpha := \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ times is at most $\beta := n - k - \epsilon n$.

- Enhanced querying strategy $\mathcal{A}'$ consists of two phases[6]:
  1. Non-adaptive phase: Query each bit for α times
  2. Adaptive phase: Choose β bits adaptively, and noiselessly obtain their values

- Any strategy $\mathcal{A}$ that uses $\frac{(n-k-\epsilon n)\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ queries can be simulated using $\mathcal{A}'$.

- It suffices to show that $P_e(\mathcal{A}') > \delta$.

[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Non-Adaptive Phase

- Non-adaptive phase: Query each bit for $\alpha = \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p||1-p)}$ times

Non-Adaptive Phase

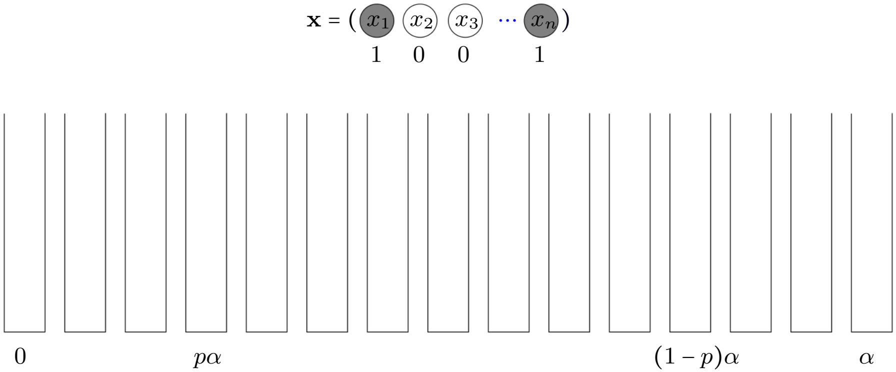
- Non-adaptive phase: Query each bit for $\alpha = \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p||1-p)}$ times
- View this phase through a balls and bins model

$$\mathbf{x} = (x_1 \; x_2 \; x_3 \; \cdots \; x_n)$$

$$1 \quad 0 \quad 0 \qquad 1$$

Non-Adaptive Phase

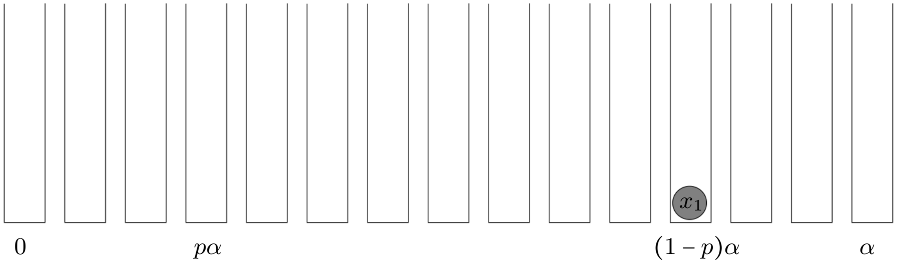
- Non-adaptive phase: Query each bit for $\alpha = \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p||1-p)}$ times
- View this phase through a balls and bins model



Non-Adaptive Phase

- Non-adaptive phase: Query each bit for $\alpha = \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ times
- View this phase through a balls and bins model

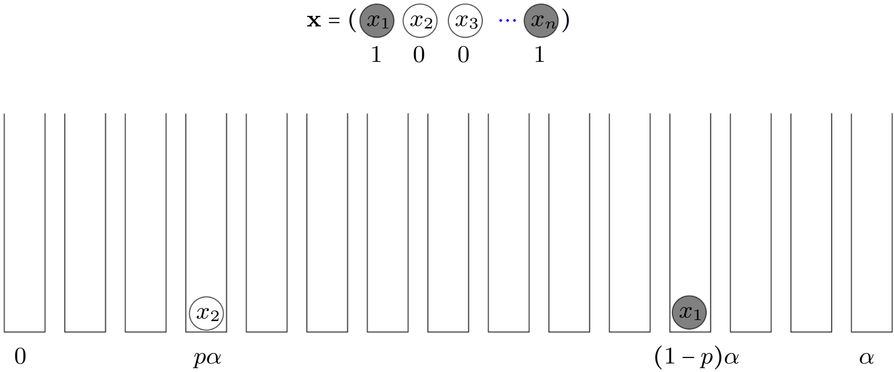
$$\mathbf{x} = (x_1\ x_2\ x_3\ \cdots\ x_n)$$
$$1\quad 0\quad 0\quad\quad 1$$



Non-Adaptive Phase

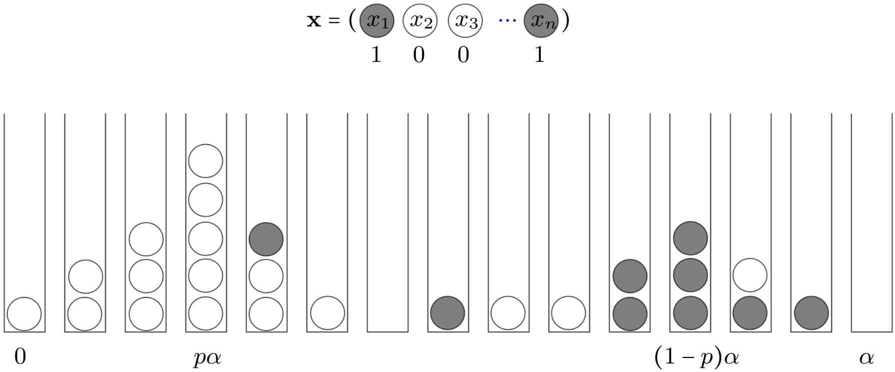
- Non-adaptive phase: Query each bit for $\alpha = \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ times
- View this phase through a balls and bins model

$$\mathbf{x} = (x_1 \; x_2 \; x_3 \; \cdots \; x_n)$$



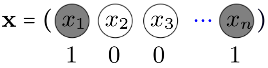
Non-Adaptive Phase

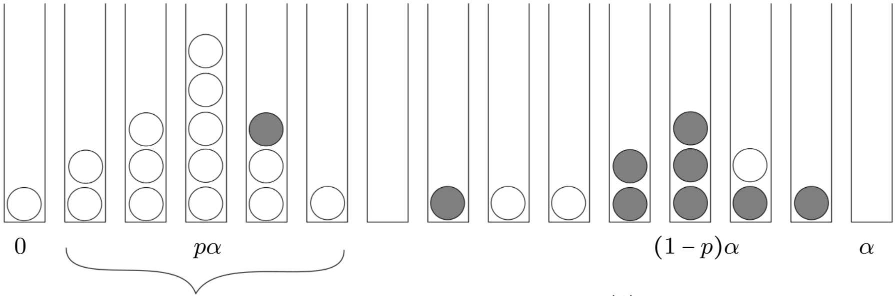
- Non-adaptive phase: Query each bit for $\alpha = \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p||1-p)}$ times
- View this phase through a balls and bins model



Non-Adaptive Phase

- Non-adaptive phase: Query each bit for $\alpha = \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p||1-p)}$ times
- View this phase through a balls and bins model

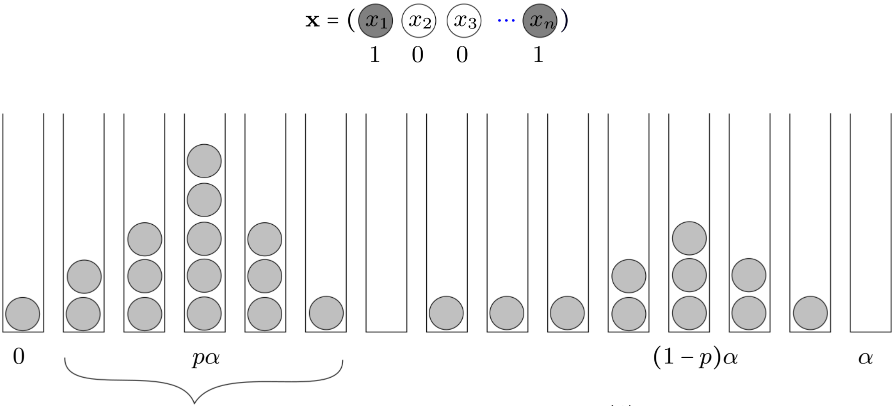




Each of these bins contains an outlier with probability $\omega(\delta)$

Non-Adaptive Phase

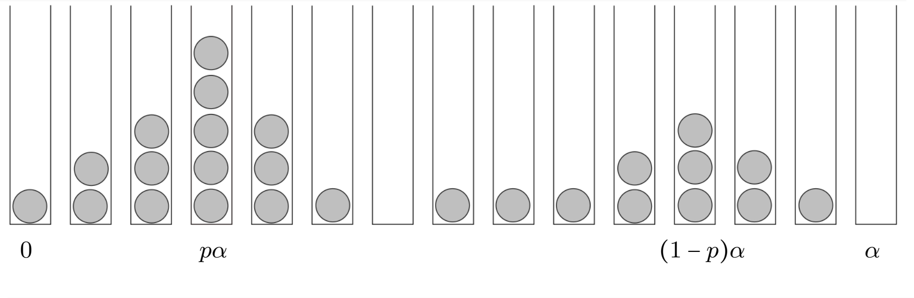
- Non-adaptive phase: Query each bit for $\alpha = \frac{\log(k/\delta)}{D_{\mathrm{KL}}(p||1-p)}$ times
- View this phase through a balls and bins model



Each of these bins contains an outlier with probability $\omega(\delta)$

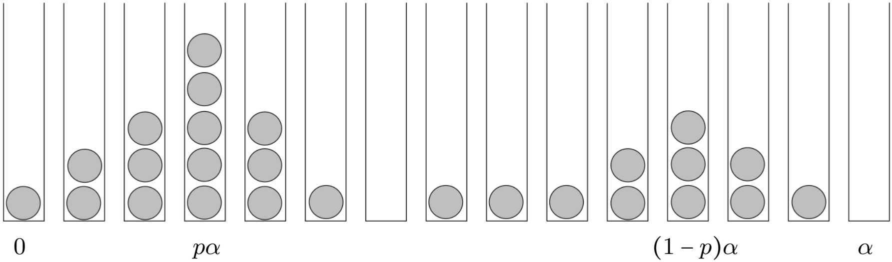
Adaptive Phase

- Adaptive phase: choose $\beta := n - k - \epsilon n$ bits adaptively, and reveal values noiselessly



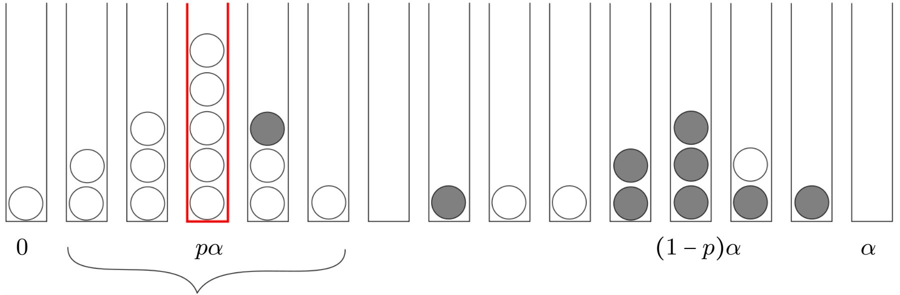
Adaptive Phase

- Adaptive phase: choose $\beta := n - k - \epsilon n$ bits adaptively, and reveal values noiselessly
- Genie provides free information through a random process
- Demonstration: when $\mathbf{x}$ has k ones



Adaptive Phase

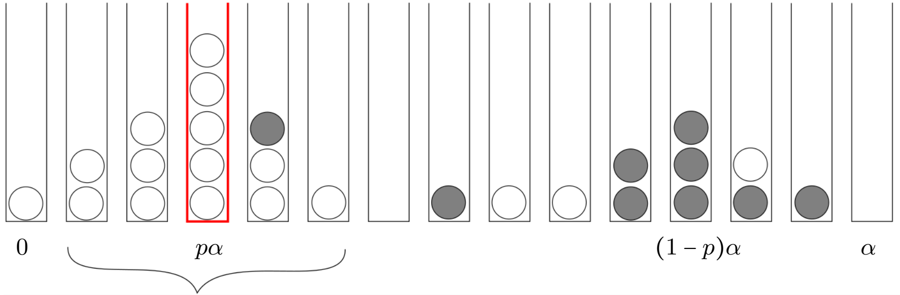
- Adaptive phase: choose $\beta := n - k - \epsilon n$ bits adaptively, and reveal values noiselessly
- Genie provides free information through a random process
- Demonstration: when $\mathbf{x}$ has k ones



Genie chooses one of these bins with probability proportional to the size of the bin

Adaptive Phase

- Adaptive phase: choose $\beta := n - k - \epsilon n$ bits adaptively, and reveal values noiselessly
- Genie provides free information through a random process
- Demonstration: when $\mathbf{x}$ has k ones

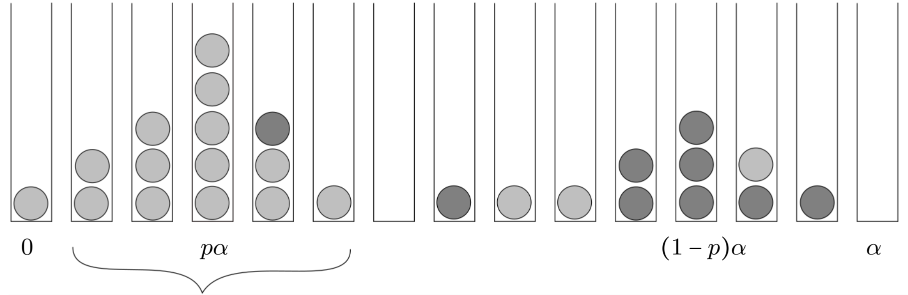


Genie chooses one of these bins with probability proportional to the size of the bin

- **If the chosen bin does not contain a heavy ball, genie reveals all k heavy balls**

Adaptive Phase

- Adaptive phase: choose $\beta := n - k - \epsilon n$ bits adaptively, and reveal values noiselessly
- Genie provides free information through a random process
- Demonstration: when $\mathbf{x}$ has k ones

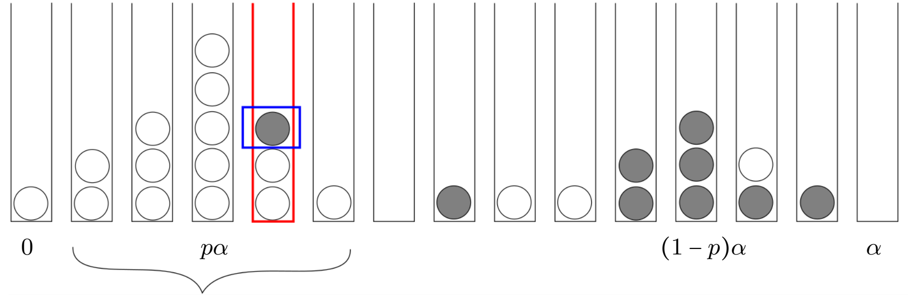


Genie chooses one of these bins with probability proportional to the size of the bin

- **If the chosen bin does not contain a heavy ball, genie reveals all k heavy balls**

Adaptive Phase

- Adaptive phase: choose $\beta := n - k - \epsilon n$ bits adaptively, and reveal values noiselessly
- Genie provides free information through a random process
- Demonstration: when $\mathbf{x}$ has k ones

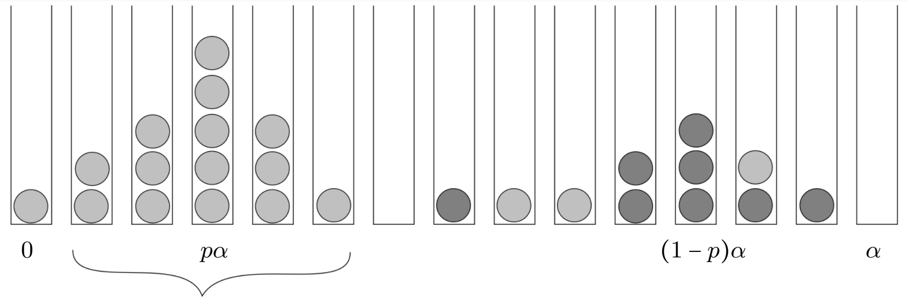


Genie chooses one of these bins with probability proportional to the size of the bin

- If the chosen bin does not contain a heavy ball, genie reveals all k heavy balls
- **If the chosen bin contains a heavy ball (w.p. $\omega(\delta)$), genie reveals $k-1$ heavy balls (all except one in the chosen bin)**

Adaptive Phase

- Adaptive phase: choose $\beta := n - k - \epsilon n$ bits adaptively, and reveal values noiselessly
- Genie provides free information through a random process
- Demonstration: when $\mathbf{x}$ has k ones

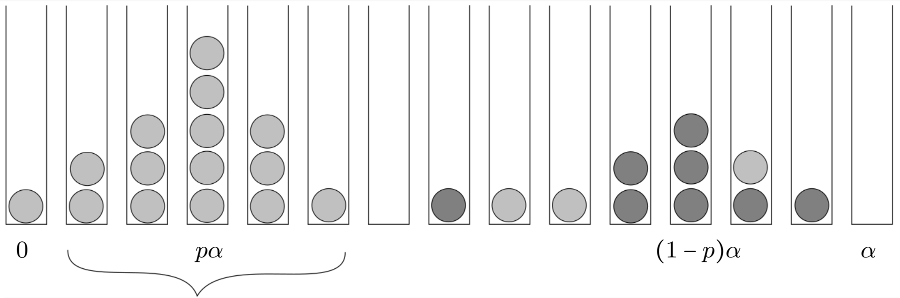


Genie chooses one of these bins with probability proportional to the size of the bin

- If the chosen bin does not contain a heavy ball, genie reveals all k heavy balls
- **If the chosen bin contains a heavy ball (w.p. $\omega(\delta)$), genie reveals $k-1$ heavy balls (all except one in the chosen bin)**

Adaptive Phase

- Adaptive phase: choose $\beta := n - k - \epsilon n$ bits adaptively, and reveal values noiselessly
- Genie provides free information through a random process
- Demonstration: when $\mathbf{x}$ has k ones



Genie chooses one of these bins with probability proportional to the size of the bin

- If the chosen bin does not contain a heavy ball, genie reveals all k heavy balls
- **If the chosen bin contains a heavy ball (w.p. $\omega(\delta)$), genie reveals $k-1$ heavy balls (all except one in the chosen bin)**
  - **In this case, probability of not finding the hidden ball is $\geq \epsilon$**
  - **This implies that $P_e(\mathcal{A}') = \omega(\epsilon\delta) > \delta$**

Upper Bound Proof Sketch

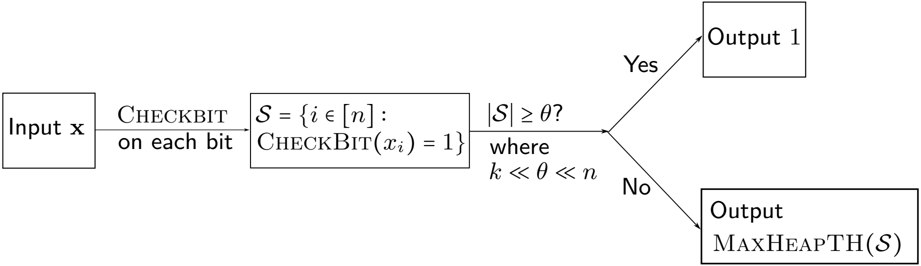
- Upper bound: $\mathsf{E}[M] \leq (1+o(1))\frac{n\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$
- Querying strategy that uses at most $(1+o(1))\frac{n\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ queries on average

[7] Y. Gu and Y. Xu. "Optimal Bounds for Noisy Sorting". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*. Orlando, FL, USA, 2023, 1502–1515.

[8] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Upper Bound Proof Sketch

- Upper bound: $\mathsf{E}[M] \leq (1+o(1))\frac{n\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$
- Querying strategy that uses at most $(1+o(1))\frac{n\log(k/\delta)}{D_{\mathrm{KL}}(p\|1-p)}$ queries on average
- When $k = o(n)$:
  - Subroutine CheckBit[7]: Repeatedly query a single bit until its value is known with confidence level γ
  - Subroutine MaxHeapTH[8]: Existing querying strategy for computing $\mathsf{TH}_k(\mathbf{x})$.



[7] Y. Gu and Y. Xu. "Optimal Bounds for Noisy Sorting". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*. Orlando, FL, USA, 2023, 1502–1515.

[8] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with noisy information". In: *SIAM Journal on Computing* 23.5 (1994), pp. 1001–1018.

Beyond the Threshold Function

- **Noisy Comparison Model:** When $\mathbf{x} \in \mathbb{R}^n$,
  - At kth time step, query $(U_k, V_k) \triangleq (x_i, x_j)$ for $i \neq j$.
  - Receive noisy response $Y_k = \mathbb{1}_{\{U_k < V_k\}} \oplus Z_k$, where $Z_k \sim \mathrm{Bern}(p)$.

[9] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. "Noisy Computing of the OR and MAX Functions". In: *IEEE Journal on Selected Areas in Information Theory* 5 (2024), pp. 302–313.

[10] M. V. Burnashev. "Data Transmission over a Discrete Channel with Feedback. Random Transmission Time". In: *Problemy Peredachi Informatsii* 12.4 (1976), pp. 10–30.

[11] Z. Wang, N. Ghaddar, B. Zhu, and L. Wang. "Variable-Length Insertion-Based Noisy Sorting". In: *Proc. IEEE Internat. Symp. Inf. Theory*. 2023, pp. 1782–1787.

[12] Y. Gu and Y. Xu. "Optimal Bounds for Noisy Sorting". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*. Orlando, FL, USA, 2023, 1502–1515.

Beyond the Threshold Function

- **Noisy Comparison Model:** When $\mathbf{x} \in \mathbb{R}^n$,
  - At kth time step, query $(U_k, V_k) \triangleq (x_i, x_j)$ for $i \neq j$.
  - Receive noisy response $Y_k = \mathbb{1}_{\{U_k < V_k\}} \oplus Z_k$, where $Z_k \sim \text{Bern}(p)$.

Function	Description	Optimal Query Complexity ($\delta = o(1)$)
MAX[9]	**Input:** $\mathbf{x} \in \mathbb{R}^n$ **Output:** max. of $\mathbf{x}$	$\dfrac{n \log \frac{1}{\delta}}{D_{\mathsf{KL}}(p \| 1-p)}$
SEARCH[10]	**Input:** sorted $\mathbf{x} \in \mathbb{R}^n$, $w \in \mathbb{R}$ **Output:** index i s.t. $x_i < w < x_{i+1}$	$\dfrac{\log n}{1 - H(p)} + \dfrac{\log \frac{1}{\delta}}{D_{\mathsf{KL}}(p \| 1-p)}$
SORT[11][12]	**Input:** $\mathbf{x} \in \mathbb{R}^n$ **Output:** sorted version of $\mathbf{x}$	$\left[\dfrac{1}{1 - H(p)} + \dfrac{1}{D_{\mathsf{KL}}(p \| 1-p)}\right] n \log n$

[9] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. "Noisy Computing of the OR and MAX Functions". In: *IEEE Journal on Selected Areas in Information Theory* 5 (2024), pp. 302–313.

[10] M. V. Burnashev. "Data Transmission over a Discrete Channel with Feedback. Random Transmission Time". In: *Problemy Peredachi Informatsii* 12.4 (1976), pp. 10–30.

[11] Z. Wang, N. Ghaddar, B. Zhu, and L. Wang. "Variable-Length Insertion-Based Noisy Sorting". In: *Proc. IEEE Internat. Symp. Inf. Theory*. 2023, pp. 1782–1787.

[12] Y. Gu and Y. Xu. "Optimal Bounds for Noisy Sorting". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*. Orlando, FL, USA, 2023, 1502–1515.

Final Remarks

- Optimal bounds for noisy computing: TH_k, MAX, SEARCH, SORT functions

[13] Y. Gu, X. Li, and Y. Xu. *Tight Bounds for Noisy Computation of High-Influence Functions, Connectivity, and Threshold*. 2025. arXiv: 2502.04632.

Final Remarks

- Optimal bounds for noisy computing: TH_k, MAX, SEARCH, SORT functions
- Bounds on optimal query complexity for computing $\mathsf{TH}_k(\mathbf{x})$:

$$(1-o(1))\frac{(n-k)\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)} \le \mathsf{E}[M] \le (1+o(1))\frac{n\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)}.$$

- Tight when $k = o(n)$
- Multiplicative gap is at most 2 when $k = \Theta(n)$

[13] Y. Gu, X. Li, and Y. Xu. *Tight Bounds for Noisy Computation of High-Influence Functions, Connectivity, and Threshold.* 2025. arXiv: 2502.04632.

Final Remarks

- Optimal bounds for noisy computing: TH_k, MAX, SEARCH, SORT functions
- Bounds on optimal query complexity for computing $\mathsf{TH}_k(\mathbf{x})$:

$$(1-o(1))\frac{(n-k)\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)} \leq \mathsf{E}[M] \leq (1+o(1))\frac{n\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)}.$$

- Tight when $k = o(n)$
- Multiplicative gap is at most 2 when $k = \Theta(n)$
- Recent work by Gu, Li and Xu[13]: Tight bounds for any k

$$(1-o(1))\frac{n\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)} \leq \mathsf{E}[M] \leq (1+o(1))\frac{n\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)}.$$

[13] Y. Gu, X. Li, and Y. Xu. *Tight Bounds for Noisy Computation of High-Influence Functions, Connectivity, and Threshold*. 2025. arXiv: 2502.04632.

Final Remarks

- Optimal bounds for noisy computing: TH_k, MAX, SEARCH, SORT functions
- Bounds on optimal query complexity for computing $\mathsf{TH}_k(\mathbf{x})$:

$$(1-o(1))\frac{(n-k)\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)} \leq \mathsf{E}[M] \leq (1+o(1))\frac{n\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)}.$$

- Tight when $k = o(n)$
- Multiplicative gap is at most 2 when $k = \Theta(n)$
- Recent work by Gu, Li and Xu[13]: Tight bounds for any k

$$(1-o(1))\frac{n\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)} \leq \mathsf{E}[M] \leq (1+o(1))\frac{n\log\frac{k}{\delta}}{D_{\mathrm{KL}}(p\|1-p)}.$$

- Future work
  - $p \to 0$ or $p \to 1/2$? $\delta = \Theta(1)$?
  - Other binary functions (e.g.: PARITY)
  - ...

[13] Y. Gu, X. Li, and Y. Xu. *Tight Bounds for Noisy Computation of High-Influence Functions, Connectivity, and Threshold*. 2025. arXiv: 2502.04632.

Any Questions?

arXiv link