

Asymptotics of Fingerprinting and Group Testing

Thijs Laarhoven

mail@thijs.com

<http://www.thijs.com/>

WIC 2014, Eindhoven, The Netherlands
(May 12, 2014)

Outline

Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$

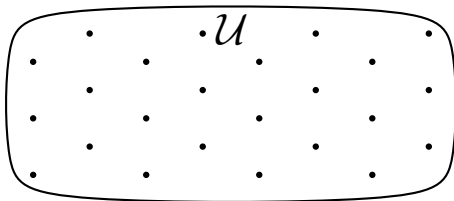
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



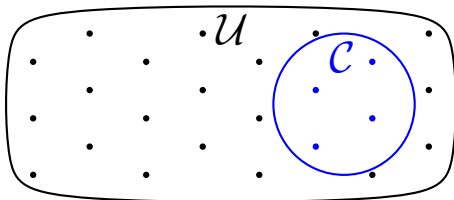
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



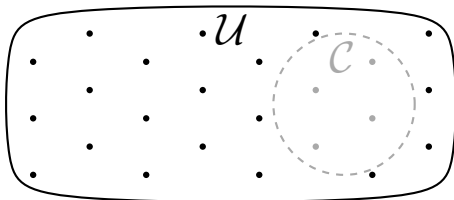
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



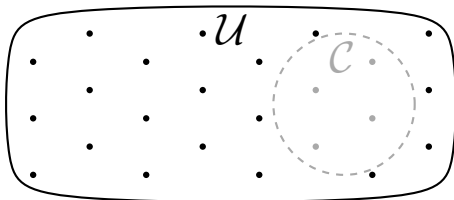
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.30 \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix}$$

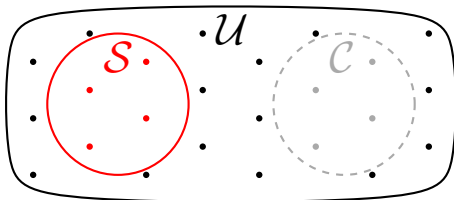
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.30 \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix}$$

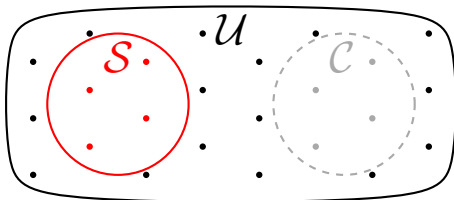
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0.01} \\ 0.30 \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix}$$

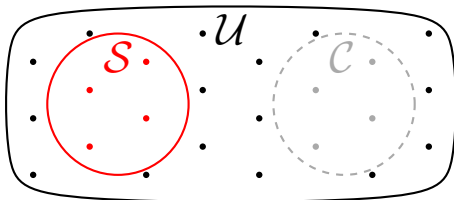
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0.01} \\ 0.30 \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix} \implies (y_{\mathcal{S}})_{k \in \mathbb{N}} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \dots)$$

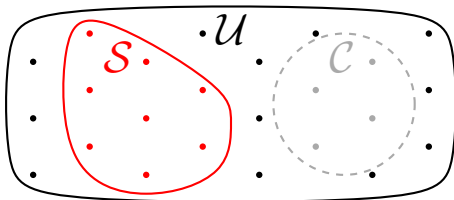
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0.01} \\ 0.30 \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix} \implies (y_{\mathcal{S}})_{k \in \mathbb{N}} = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0, \dots)$$

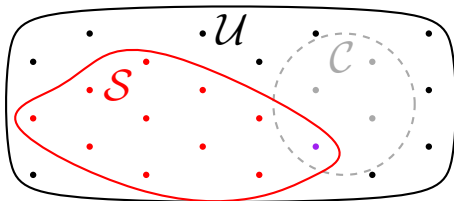
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.01 \\ \mathbf{0.30} \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix} \implies (y_{\mathcal{S}})_{k \in \mathbb{N}} = (0, 0, 1, 0, 1, 0, 0, 0, 0, 1, \dots)$$

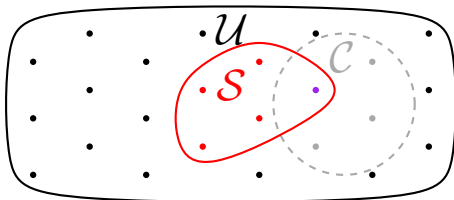
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.01 \\ \mathbf{0.30} \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix} \implies (y_{\mathcal{S}})_{k \in \mathbb{N}} = (1, 0, 0, 1, 0, 0, 0, 0, 1, 0, \dots)$$

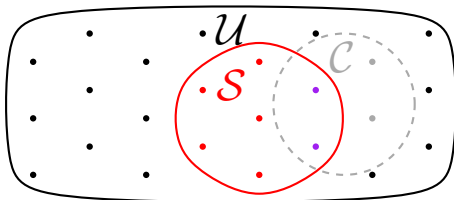
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.30 \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix} \implies (y_{\mathcal{S}})_{k \in \mathbb{N}} = (1, 0, 0, 1, 1, 0, 1, 1, 0, 1, \dots)$$

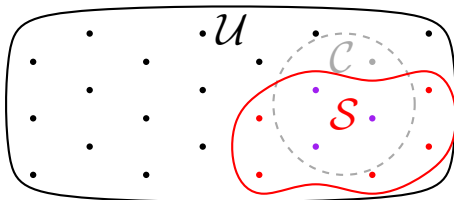
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.30 \\ 0.50 \\ \mathbf{0.95} \\ 0.99 \end{bmatrix} \implies (y_{\mathcal{S}})_{k \in \mathbb{N}} = (1, 1, 1, 0, 1, 1, 1, 1, 1, 1, \dots)$$

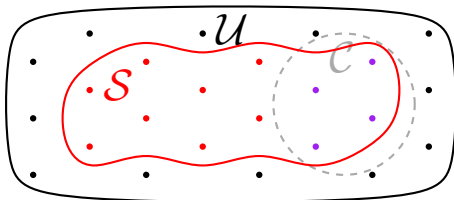
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.30 \\ 0.50 \\ 0.95 \\ \mathbf{0.99} \end{bmatrix} \implies (y_{\mathcal{S}})_{k \in \mathbb{N}} = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \dots)$$

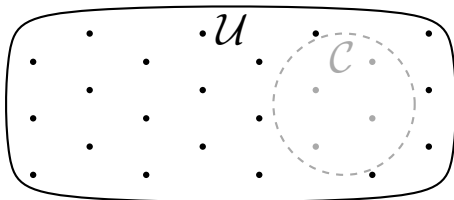
Search problems

Introduction

Problem: Given a universe \mathcal{U} of n elements and a model $\vec{\theta}$, find a hidden subset $\mathcal{C} \subset \mathcal{U}$ of size $c \ll n$ using *subset queries*.

Subset query: Given a subset $\mathcal{S} \subseteq \mathcal{U}$, an oracle returns a bit $y_{\mathcal{S}}$:

$$\mathbb{P}(Y_{\mathcal{S}} = 1) = \theta_z. \quad (z = |\mathcal{S} \cap \mathcal{C}| \in \{0, \dots, c\})$$



$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.30 \\ 0.50 \\ 0.95 \\ 0.99 \end{bmatrix}$$

Search problems

Fingerprinting and group testing models

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_{c-1} \\ \theta_c \end{bmatrix} =$$

Search problems

Fingerprinting and group testing models

Fingerprinting

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_{c-1} \\ \theta_c \end{bmatrix} = \begin{bmatrix} 0 \\ * \\ * \\ * \\ \dots \\ * \\ 1 \end{bmatrix}$$

Search problems

Fingerprinting and group testing models

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_{c-1} \\ \theta_c \end{bmatrix} = \begin{bmatrix} 0 \\ * \\ * \\ * \\ \dots \\ * \\ 1 \end{bmatrix} \quad \begin{array}{l} \text{Fingerprinting} \\ \dots \text{coinflip atk.} \\ \dots \text{majority atk.} \\ \dots \text{linear atk.} \end{array}$$

$$\begin{bmatrix} 0 \\ 1/2 \\ 1/2 \\ 1/2 \\ \vdots \\ 1/2 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1/c \\ 2/c \\ 3/c \\ \vdots \\ (c-1)/c \\ 1 \end{bmatrix}$$

Search problems

Fingerprinting and group testing models

$$\begin{array}{c}
 \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_{c-1} \\ \theta_c \end{bmatrix} = \begin{array}{c} \text{Fingerprinting} \\ \begin{bmatrix} 0 \\ * \\ * \\ * \\ \vdots \\ * \\ 1 \end{bmatrix} \\ \dots \text{coinflip atk.} \\ \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \\ 1/2 \\ \vdots \\ 1/2 \\ 1 \end{bmatrix} \\ \dots \text{majority atk.} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \\ \dots \text{linear atk.} \\ \begin{bmatrix} 0 \\ 1/c \\ 2/c \\ 3/c \\ \vdots \\ (c-1)/c \\ 1 \end{bmatrix}
 \end{array}
 \end{array}$$

- **Fingerprinting:** Model corresponds to [adversary](#); unknown

Search problems

Fingerprinting and group testing models

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_{c-1} \\ \theta_c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ * \\ * \\ * \\ \vdots \\ * \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 1/2 \\ 1/2 \\ 1/2 \\ \vdots \\ 1/2 \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 1/c \\ 2/c \\ 3/c \\ \vdots \\ (c-1)/c \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

Fingerprinting ...coinflip atk. ...majority atk. ...linear atk. **Group testing**

- **Fingerprinting:** Model corresponds to adversary; unknown

Search problems

Fingerprinting and group testing models

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_{c-1} \\ \theta_c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ * \\ * \\ * \\ \vdots \\ * \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 1/2 \\ 1/2 \\ 1/2 \\ \vdots \\ 1/2 \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 1/c \\ 2/c \\ 3/c \\ \vdots \\ (c-1)/c \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} r \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} \\ 1/3 \\ 2/3 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

Fingerprinting ...coinflip atk. ...majority atk. ...linear atk. **Group testing** ...with noise ...with gaps

- **Fingerprinting:** Model corresponds to adversary; unknown

Search problems

Fingerprinting and group testing models

	Fingerprinting	...coinflip atk.	...majority atk.	...linear atk.	Group testing	...with noise	...with gaps
$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_{c-1} \\ \theta_c \end{bmatrix} =$	$\begin{bmatrix} 0 \\ * \\ * \\ * \\ \vdots \\ * \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1/2 \\ 1/2 \\ 1/2 \\ \vdots \\ 1/2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1/c \\ 2/c \\ 3/c \\ \vdots \\ (c-1)/c \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} r \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1/3 \\ 2/3 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$

- **Fingerprinting:** Model corresponds to adversary; unknown
- **Group testing:** Model generally known but possibly noisy

Overview

Lower bounds on fingerprinting

How many queries ℓ are necessary for non-adaptive fingerprinting?

- 1998: $\ell = \Omega(c \log n)^{[1]}$
- 2003: $\ell = \Omega(c^2 \log \frac{n}{c})^{[2]}$
- 2003: $\ell = \Omega(c^2 \log n)^{[3]}$
- 2009: $\ell \stackrel{?}{\sim} 2c^2 \ln n^{[4]}$
- 2012: $\ell \sim 2c^2 \ln n^{[5]}$
 - ▶ asymptotic optimal attack is the linear attack ($\theta_z = z/c$)

[1] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[2] C. Peikert et al., "Lower bounds for collusion-secure fingerprinting," in *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003, pp. 472–479.

[3] G. Tardos, "Optimal probabilistic fingerprint codes," in *ACM Symposium on Theory of Computing (STOC)*, 2003, pp. 116–125.

[4] E. Amiri and G. Tardos, "High rate fingerprinting codes and the fingerprinting capacity," in *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009, pp. 336–345.

[5] Y.-W. Huang and P. Moulin, "On the saddle-point solution and the large-coalition asymptotics of fingerprinting games," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 160–175, 2012.

Overview

Upper bounds on fingerprinting

How many queries ℓ are sufficient for non-adaptive fingerprinting?

- 1995: $\ell = O(c^4 \log n)^{[1]}$
- 2003: $\ell = 100c^2 \ln n^{[2]}$ (“the Tardos scheme”)
- 2006: $\ell \sim 4\pi^2 c^2 \ln n^{[6]}$
- 2008: $\ell \sim \pi^2 c^2 \ln n^{[7]}$
- 2011: $\ell \sim \frac{1}{2}\pi^2 c^2 \ln n^{[8]}$
- 2013: $\ell \sim 2c^2 \ln n^{[9]}$
 - ▶ decoder designed against the linear attack is ‘optimal’

[6] B. Skoric et al., “Tardos fingerprinting is better than we thought,” *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3663–3676, 2008.

[7] B. Skoric et al., “Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes,” *Designs, Codes and Cryptography*, vol. 46, no. 2, pp. 137–166, 2008.

[8] T. Laarhoven and B. de Weger, “Optimal symmetric Tardos traitor tracing schemes,” *Designs, Codes and Cryptography*, vol. 71, no. 1, pp. 83–103, 2014.

[9] J.-J. Oosterwijk et al., “A capacity-achieving simple decoder for bias-based traitor tracing schemes,” *Cryptology ePrint Archive*, 2013.

Overview

Lower bounds on group testing

How many queries ℓ are necessary for non-adaptive group testing?

- 1985: $\ell \sim c \log_2 n$ ^[10]
- 1989: $\ell = \Omega(c^2 \log n)$ ^[11] (deterministic tracing)
- 2009: $\ell = \Omega\left(\frac{c \log n}{(1-r)^2}\right)$ ^[12] (noise)
- ...

^[10]A. Sebő, "On two random search problems," *Journal of Statistical Planning and Inference*, vol. 11, no. 1, pp. 23–31, 1985.

^[11]A. G. Dyachkov et al., "Superimposed distance codes," *Problems of Control and Information Theory*, vol. 18, no. 4, pp. 237–250, 1989.

^[12]G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1880–1901, 2012.

Overview

Upper bounds on group testing

How many queries ℓ are sufficient for non-adaptive group testing?

- 2005: $\ell \sim 2c^2 \log_2 n$ ^[13] (linear gap)
- 2009: $\ell = O\left(\frac{c \log n}{(1-r)^3}\right)$ ^[14] (noise)
- 2011: $\ell \sim ec \ln n$ ^[15]
- 2013: $\ell \sim O(\sqrt{g}c \log n)$ ^[16] (coinflip gap)
- 2013: $\ell \sim \pi c \ln n$ ^[17] (majority)
- ...

^[13]A. D. Lungo et al., “The guessing secrets problem: a probabilistic approach,” *Journal of Algorithms*, vol. 55, pp. 142–176, 2005.

^[14]M. Cheraghchi et al., “Group testing with probabilistic tests: theory, design and application,” *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 7057–7067, 2011.

^[15]C. L. Chan et al., “Non-adaptive probabilistic group testing with noisy measurements,” in *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2011, pp. 1832–1839.

^[16]C. L. Chan et al., “Stochastic threshold group testing,” in *IEEE Information Theory Workshop (ITW)*, 2013, pp. 1–5.

^[17]T. Laarhoven, “Efficient probabilistic group testing based on traitor tracing,” in *Annual Allerton Conference on Communication, Control and Computing (Allerton)*, 2013, pp. 1358–1365.

Explicit asymptotics of the capacities of various models^[18]

- Information-theoretic approach: Mutual information game
- Both simple (efficient) and joint (optimal) decoding
- Can be applied to arbitrary models $\vec{\theta}$

[18] T. Laarhoven, "Asymptotics of fingerprinting and group testing: tight bounds from channel capacities," submitted to *IEEE Transactions on Information Theory*, pp. 1–14, 2014.

[19] T. Laarhoven, "Asymptotics of fingerprinting and group testing: capacity-achieving log-likelihood decoders," submitted to *IEEE Transactions on Information Theory*, pp. 1–13, 2014.

Explicit asymptotics of the capacities of various models^[18]

- Information-theoretic approach: Mutual information game
- Both simple (efficient) and joint (optimal) decoding
- Can be applied to arbitrary models $\vec{\theta}$

Capacity-achieving decoders for arbitrary models^[19]

- Statistical approach: Neyman-Pearson hypothesis testing
- Both simple and joint decoding
- Asymptotically optimal regardless of the model

[18] T. Laarhoven, "Asymptotics of fingerprinting and group testing: tight bounds from channel capacities," submitted to *IEEE Transactions on Information Theory*, pp. 1–14, 2014.

[19] T. Laarhoven, "Asymptotics of fingerprinting and group testing: capacity-achieving log-likelihood decoders," submitted to *IEEE Transactions on Information Theory*, pp. 1–13, 2014.

Explicit asymptotics of the capacities of various models^[18]

- Information-theoretic approach: Mutual information game
- Both simple (efficient) and joint (optimal) decoding
- Can be applied to arbitrary models $\vec{\theta}$
- **Focus of this talk**

Capacity-achieving decoders for arbitrary models^[19]

- Statistical approach: Neyman-Pearson hypothesis testing
- Both simple and joint decoding
- Asymptotically optimal regardless of the model

[18] T. Laarhoven, "Asymptotics of fingerprinting and group testing: tight bounds from channel capacities," submitted to *IEEE Transactions on Information Theory*, pp. 1–14, 2014.

[19] T. Laarhoven, "Asymptotics of fingerprinting and group testing: capacity-achieving log-likelihood decoders," submitted to *IEEE Transactions on Information Theory*, pp. 1–13, 2014.

Capacities

Randomized construction

Choosing the subsets $\mathcal{S} \subseteq \mathcal{U}$ to query

- Choose a parameter $p \in (0, 1)$
- For every query and element j : $\mathbb{P}(j \in \mathcal{S}) = p$
 - ▶ Different queries and elements are independent

Capacities

Randomized construction

Choosing the subsets $\mathcal{S} \subseteq \mathcal{U}$ to query

- Choose a parameter $p \in (0, 1)$
- For every query and element j : $\mathbb{P}(j \in \mathcal{S}) = p$
 - ▶ Different queries and elements are independent

Finding the hidden subset $\mathcal{C} \subseteq \mathcal{U}$

- Simple decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ X : The information whether $j \in \mathcal{S}$ or not
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p
- Joint decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ X' : The information whether $j' \in \mathcal{S}$ or not, for all $j' \in \mathcal{U}$
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p

Capacities

Simple decoding

- Simple decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ X : The information whether $j \in \mathcal{S}$ or not
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p

Capacities

Simple decoding

- Simple decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ X : The information whether $j \in \mathcal{S}$ or not
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p

For fixed $\vec{\theta}$, the simple capacity is given by^[5]

$$C^{\text{simple}}(\vec{\theta}) = \max_{p \in (0,1)} I(X; Y | P = p).$$

Capacities

Simple decoding

- Simple decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ X : The information whether $j \in \mathcal{S}$ or not
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p

For fixed $\vec{\theta}$, the simple capacity is given by^[5]

$$C^{\text{simple}}(\vec{\theta}) = \max_{p \in (0,1)} I(X; Y | P = p).$$

$I(X; Y | P = p)$ is an explicit function of $\vec{\theta}$ and p .

Capacities

Joint decoding

- Joint decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ X' : The information whether $j' \in \mathcal{S}$ or not, for all $j' \in \mathcal{U}$
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p

Capacities

Joint decoding

- Joint decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ X' : The information whether $j' \in \mathcal{S}$ or not, for all $j' \in \mathcal{U}$
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p

For fixed $\vec{\theta}$, the simple capacity is given by^[5]

$$C^{\text{joint}}(\vec{\theta}) = \max_{p \in (0,1)} I(X'; Y | P = p).$$

Capacities

Joint decoding

- Joint decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ Z : The size of $\mathcal{S} \cap \mathcal{C}$
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p

For fixed $\vec{\theta}$, the simple capacity is given by^[5]

$$C^{\text{joint}}(\vec{\theta}) = \max_{p \in (0,1)} I(Z; Y | P = p).$$

Capacities

Joint decoding

- Joint decoding: Decide whether $j \in \mathcal{C}$ based on...
 - ▶ Z : The size of $\mathcal{S} \cap \mathcal{C}$
 - ▶ Y : The output bits $y_{\mathcal{S}}$
 - ▶ P : The randomly drawn parameters p

For fixed $\vec{\theta}$, the simple capacity is given by^[5]

$$C^{\text{joint}}(\vec{\theta}) = \max_{p \in (0,1)} I(Z; Y | P = p).$$

$I(Z; Y | P = p)$ is an explicit function of $\vec{\theta}$ and p .

Capacities

Results

Model $\vec{\theta}$	$C^{\text{simple}}(\vec{\theta})$	$C^{\text{joint}}(\vec{\theta})$
$(0, *, *, *, \dots, *, *, *, 1)$	$1/(2c^2 \ln 2)^{[5]}$	$1/(2c^2 \ln 2)^{[5]}$
$(0, 1/c, 2/c, \dots, (c-1)/c, 1)$	$1/(2c^2 \ln 2)^{[5]}$	$1/(2c^2 \ln 2)^{[5]}$
$(0, 1/2, 1/2, \dots, 1/2, 1/2, 1)$	$\ln 2/(4c)$	$\log_2(5/4)/c$
$(0, 0, 0, 0, \dots, 1, 1, 1, 1)$	$1/(\pi c \ln 2)$	$1/c$
$(0, 1, 1, 1, \dots, 1, 1, 1, 1)$	$\ln 2/c$	$1/c^{[14]}$
$(r, 1, 1, 1, \dots, 1, 1, 1, 1)$	$[\ln 2 - r + O(r^2)]/c$	$[1 - \frac{1}{2}h(r) + O(r^2)]/c$
...

Capacities

Results

Model $\vec{\theta}$	$C^{\text{simple}}(\vec{\theta})$	$C^{\text{joint}}(\vec{\theta})$
$(0, *, *, *, \dots, *, *, *, 1)$	$1/(2c^2 \ln 2)^{[5]}$	$1/(2c^2 \ln 2)^{[5]}$
$(0, 1/c, 2/c, \dots, (c-1)/c, 1)$	$1/(2c^2 \ln 2)^{[5]}$	$1/(2c^2 \ln 2)^{[5]}$
$(0, 1/2, 1/2, \dots, 1/2, 1/2, 1)$	$\ln 2/(4c)$	$\log_2(5/4)/c$
$(0, 0, 0, 0, \dots, 1, 1, 1, 1)$	$1/(\pi c \ln 2)$	$1/c$
$(0, 1, 1, 1, \dots, 1, 1, 1, 1)$	$\ln 2/c$	$1/c^{[14]}$
$(r, 1, 1, 1, \dots, 1, 1, 1, 1)$	$[\ln 2 - r + O(r^2)]/c$	$[1 - \frac{1}{2}h(r) + O(r^2)]/c$
...

$$\max_{p \in (0,1)} I(\dots) = \dots = \max_{t \in (0,1)} [h(\frac{t}{2}) - t] = h(\frac{1}{5}) - \frac{2}{5} = \log_2(\frac{5}{4}).$$

Capacities

Results

Model $\vec{\theta}$	$C^{\text{simple}}(\vec{\theta})$	$C^{\text{joint}}(\vec{\theta})$
$(0, *, *, *, \dots, *, *, *, 1)$	$1/(2c^2 \ln 2)^{[5]}$	$1/(2c^2 \ln 2)^{[5]}$
$(0, 1/c, 2/c, \dots, (c-1)/c, 1)$	$1/(2c^2 \ln 2)^{[5]}$	$1/(2c^2 \ln 2)^{[5]}$
$(0, 1/2, 1/2, \dots, 1/2, 1/2, 1)$	$\ln 2/(4c)$	$\log_2(5/4)/c$
$(0, 0, 0, 0, \dots, 1, 1, 1, 1)$	$1/(\pi c \ln 2)$	$1/c$
$(0, 1, 1, 1, \dots, 1, 1, 1, 1)$	$\ln 2/c$	$1/c^{[10]}$
$(r, 1, 1, 1, \dots, 1, 1, 1, 1)$	$[\ln 2 - r + O(r^2)]/c$	$[1 - \frac{1}{2}h(r) + O(r^2)]/c$
...

Results

Lower bounds on fingerprinting

How many queries ℓ are necessary for non-adaptive fingerprinting?

- 1998: $\ell = \Omega(c \log n)^{[1]}$
- 2003: $\ell = \Omega(c^2 \log \frac{n}{c})^{[2]}$
- 2003: $\ell = \Omega(c^2 \log n)^{[3]}$
- 2009: $\ell \stackrel{?}{\sim} 2c^2 \ln n^{[4]}$
- 2012: $\ell \sim 2c^2 \ln n^{[5]}$

Results

Upper bounds on fingerprinting

How many queries ℓ are sufficient for non-adaptive fingerprinting?

- 1995: $\ell = O(c^4 \log n)^{[1]}$
- 2003: $\ell = 100c^2 \ln n^{[2]}$ (“the Tardos scheme”)
- 2006: $\ell \sim 4\pi^2 c^2 \ln n^{[6]}$
- 2008: $\ell \sim \pi^2 c^2 \ln n^{[7]}$
- 2011: $\ell \sim \frac{1}{2}\pi^2 c^2 \ln n^{[8]}$
- 2013: $\ell \sim 2c^2 \ln n^{[9][23]}$

Results

Lower bounds on group testing

How many queries ℓ are necessary for non-adaptive group testing?

- 1985: $\ell \sim c \log_2 n$ ^[10]
- 1989: $\ell = \Omega(c^2 \log n)$ ^[11] (deterministic tracing)
- 2009: $\ell = \Omega\left(\frac{c \log n}{(1-r)^2}\right)$ ^[12] (noise)
- 2014: $\ell \sim \frac{c \log_2 n}{\ln 2}$ ^[18] (simple decoding)
- 2014: $\ell \sim \frac{c \log_2 n}{\ln 2 - r + O(r^2)}$ ^[18] (simple decoding, noise)
- 2014: $\ell \sim \frac{c \log_2 n}{1 - \frac{1}{2}h(r) + O(r^2)}$ ^[18] (joint decoding, noise)
- ...

Results

Upper bounds on group testing

How many queries ℓ are sufficient for non-adaptive group testing?

- 2005: $\ell \sim 2c^2 \log_2 n^{[13]}$ (linear gap)
- 2009: $\ell = O\left(\frac{c \log n}{(1-r)^3}\right)^{[14]}$ (noise)
- 2011: $\ell \sim ec \ln n^{[15]}$ (simple decoding)
- 2013: $\ell \sim O(\sqrt{g}c \log n)^{[16]}$ (coinflip gap)
- 2013: $\ell \sim \pi c \ln n^{[17]}$ (majority)
- 2014: $\ell \sim \frac{c \log_2 n}{\ln 2}^{[19]}$ (simple decoding)
- 2014: $\ell \sim \frac{4c \log_2 n}{\ln 2}^{[19]}$ (simple decoding, coinflip)
- 2014: $\ell \sim c \log_{5/4} n^{[19]}$ (joint decoding, coinflip)
- ...

Conclusion

Explicit asymptotics of the capacities of various models [18]

- Information-theoretic approach: Mutual information game
- Both simple (efficient) and joint (optimal) decoding
- Can be applied to arbitrary models $\vec{\theta}$

Capacity-achieving decoders for arbitrary models [19]

- Statistical approach: Neyman-Pearson hypothesis testing
- Both simple and joint decoding
- Asymptotically optimal regardless of the model

Questions?