Cookbook: Lower Bounds for Statistical Inference in Distributed and Constrained Settings

Jayadev Acharya, **Clément Canonne**, Himanshu Tyagi

FOCS 2020

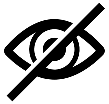
Part IV: Upper bounds and discussion

Those were lower bounds.

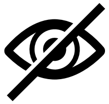
Those were lower bounds.

Are they tight?

Upper bounds for learning

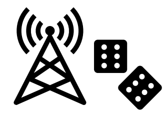
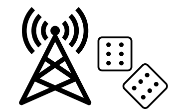
Estimation		
Δ_k, ℓ_1	$\frac{k}{\varepsilon^2} \cdot \frac{k}{\min\{2^\ell, k\}}$	$\frac{k}{\varepsilon^2} \cdot \frac{k}{\varrho^2}$
$\mathcal{B}_d, \ell_2$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\varrho^2}$
$\mathcal{G}_d, \ell_2$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\varrho^2}$

Upper bounds for learning

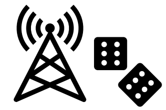
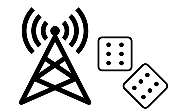
Estimation		
Δ_k, ℓ_1	$\frac{k}{\varepsilon^2} \cdot \frac{k}{\min\{2^\ell, k\}}$	$\frac{k}{\varepsilon^2} \cdot \frac{k}{\varrho^2}$
$\mathcal{B}_d, \ell_2$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\varrho^2}$
$\mathcal{G}_d, \ell_2$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\varrho^2}$

Focus on communication for this part

Upper bounds for testing

Testing		
Δ_k, ℓ_1	$\frac{\sqrt{k}}{\varepsilon^2} \cdot \frac{k}{\min\{2^\ell, k\}}$	$\frac{\sqrt{k}}{\varepsilon^2} \cdot \sqrt{\frac{k}{\min\{2^\ell, k\}}}$
$\mathcal{B}_d, \ell_2$	$\frac{\sqrt{d}}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \sqrt{\frac{d}{\min\{\ell, d\}}}$
"Hide-and-Seek"	$\frac{\log d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	

Upper bounds for testing

Testing		
Δ_k, ℓ_1	$\frac{\sqrt{k}}{\varepsilon^2} \cdot \frac{k}{\min\{2^\ell, k\}}$	$\frac{\sqrt{k}}{\varepsilon^2} \cdot \sqrt{\frac{k}{\min\{2^\ell, k\}}}$
$\mathcal{B}_d, \ell_2$	$\frac{\sqrt{d}}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \sqrt{\frac{d}{\min\{\ell, d\}}}$
"Hide-and-Seek"	$\frac{\log d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	

(analogous for privacy)

That’s seven upper bounds to prove.

(in ≈30 minutes)

That's seven upper bounds to prove.
(in ≈30 minutes)

Discrete distributions under ℓ_1 loss: 3

Bernoulli product under ℓ_2 loss: 3

Bernoulli product hide-and-seek: 1

Let’s do several at once: simulate-and-infer

Idea: if, under constraints, given messages from s users the server can simulate one sample from the unknown $\boldsymbol{p}$, then

$$n = s \cdot n_{centralized}$$

users suffice.

Let’s do several at once: simulate-and-infer

Theorem (easy). For $\mathcal{B}_d$, noninteractive private-coin simulate-and-infer is possible with $s = \frac{d}{\ell}$.

Let's do several at once: simulate-and-infer

Theorem (easy). For $\mathcal{B}_d$, noninteractive private-coin simulate-and-infer is possible with $s = \frac{d}{\ell}$.

Proof. First user sends the first ℓ bits of X_1, …, s-th user sends last ℓ bits of X_s. Server creates

$$\tilde{X} := (X_{11}, \dots, X_{1\ell}, X_{21}, \dots, X_{2\ell}, \dots, X_{s1}, \dots, X_{s\ell}) \in \{\pm 1\}^d$$

Since $\boldsymbol{p}$ is a product distribution, $\tilde{X} \sim \boldsymbol{p}$. □

Let's do several at once: simulate-and-infer

Corollary. For $\mathcal{B}_d$, noninteractive private-coin mean estimation under ℓ_2 loss is possible with $n = O\left(\frac{d}{\varepsilon^2} \cdot \frac{d}{\ell}\right)$.

Proof. Recall that the centralized sample complexity is $O\left(\frac{d}{\varepsilon^2}\right)$, by taking the empirical mean. □

Let's do several at once: simulate-and-infer

Corollary. For $\mathcal{B}_d$, noninteractive private-coin mean testing under ℓ_2 loss is possible with $n = O\left(\frac{\sqrt{d}}{\varepsilon^2} \cdot \frac{d}{\ell}\right)$.

Proof. Recall that the centralized sample complexity is $O\left(\frac{\sqrt{d}}{\varepsilon^2}\right)$, by taking the squared ℓ_2 norm empirical mean (and computing its expectation and variance). □

Let's do several at once: simulate-and-infer

- **Corollary.** For $\mathcal{B}_d$, noninteractive private-coin hide-and-seek can be performed with $n = O\left(\frac{\log d}{\varepsilon^2} \cdot \frac{d}{\ell}\right)$.

Proof. Recall that the centralized sample complexity is $O\left(\frac{\log d}{\varepsilon^2}\right)$, by computing the empirical mean of each coordinate to $\pm\frac{\varepsilon}{2}$ (and taking a union bound). □

Let’s do several at once: simulate-and-infer

That’s three upper bounds via simulate-and-infer. Let’s do two more.

Let’s do several at once: simulate-and-infer

That’s three upper bounds via simulate-and-infer. Let’s do two more.

Theorem ([ACT20d]). For Δ_k, noninteractive private-coin simulate-and-infer is possible with $s = \frac{k}{2^\ell}$.

Let's do several at once: simulate-and-infer

- **Theorem.** For Δ_k, noninteractive private-coin simulate-and-infer is possible with $s \asymp \frac{k}{2^\ell}$ (in expectation).

Let's do several at once: simulate-and-infer

Theorem. For Δ_k, noninteractive private-coin simulate-and-infer is possible with $s \asymp \frac{k}{2^\ell}$ (in expectation).

Proof. First, $\ell = 1$. Take $s = 2k$ users, pair them: users $2i - 1$ and $2i$ send $Y_{2i-1} = \mathbb{I}_{X_{2i-1}=i}$ and $Y_{2i} = \mathbb{I}_{X_{2i}=i}$, resp.
If
- there is a unique $i \in [k]$ s.t. $Y_{2i-1} = 1$, and
- for that i we also have $Y_{2i} = 0$

then the server outputs that i. Otherwise, it outputs $\bot$.

Let's do several at once: simulate-and-infer

Theorem. For Δ_k, noninteractive private-coin simulate-and-infer is possible with $s \asymp \frac{k}{2^\ell}$ (in expectation).

Proof. First, $\ell = 1$. Take $s = 2k$ users, pair them: users $2i - 1$ and $2i$ send $Y_{2i-1} = \mathbb{I}_{X_{2i-1}=i}$ and $Y_{2i} = \mathbb{I}_{X_{2i}=i}$, resp.

If

- there is a unique $i \in [k]$ s.t. $Y_{2i-1} = 1$, and
- for that i we also have $Y_{2i} = 0$

then the server outputs $\tilde{X} = i$. Otherwise, $\tilde{X} = \bot$.

$$\Pr[\tilde{X} = i \mid \tilde{X} \neq \bot] = \boldsymbol{p}_i \prod_{j \neq i} (1 - \boldsymbol{p}_j) \cdot (1 - \boldsymbol{p}_i) = \boldsymbol{p}_i \cdot \prod_j (1 - \boldsymbol{p}_j)$$

Let’s do several at once: simulate-and-infer

Theorem. For Δ_k, noninteractive private-coin simulate-and-infer is possible with $s \asymp \frac{k}{2^\ell}$ (in expectation).

Proof (cont’d). So

$$\Pr[\tilde{X} = i \mid \tilde{X} \neq \perp] \propto \boldsymbol{p}_i$$

which is good. Moreover,

$$\Pr[\tilde{X} \neq \perp] = \prod_j \left(1 - \boldsymbol{p}_j\right) \geq \prod_j 4^{-\boldsymbol{p}_j} = \frac{1}{4}$$

using that $1 - x \geq 4^{-x}$ for $0 \leq x \leq \frac{1}{2}$. So we are good as long as $\|\boldsymbol{p}\|_\infty \leq \frac{1}{2}$ … which we can assume via a simple trick using and losing a factor 2: $\boldsymbol{p}'$ on $[2k]$ with $\boldsymbol{p}'_i = \boldsymbol{p}'_{i+k} = \frac{\boldsymbol{p}_i}{2}$).

Let's do several at once: simulate-and-infer

Theorem. For Δ_k, noninteractive private-coin simulate-and-infer is possible with $s \asymp \frac{k}{2^\ell}$ (in expectation).

Proof (cont'd). We just proved that $\mathbb{E}[s] \leq 4k$, for $\ell = 1$. For $\ell \geq 1$, partition $[k]$ in sets $S_1, \dots, S_{\frac{k}{2^\ell - 1}}$ of size $2^\ell - 1$. Users $2i - 1$ and $2i$ send 0 if their sample is outside S_i, or the index of their sample inside S_i otherwise. Same analysis as for the case $\ell = 1$. □

Let's do several at once: simulate-and-infer

Corollary. For Δ_k, noninteractive private-coin estimation under ℓ_1 loss is possible with $n = O\left(\frac{k}{\varepsilon^2} \cdot \frac{k}{2^\ell}\right)$.

Proof. Recall that the centralized sample complexity is $O\left(\frac{k}{\varepsilon^2}\right)$, by taking the empirical distribution. □

Let's do several at once: simulate-and-infer

Corollary. For Δ_k, noninteractive private-coin identity testing under ℓ_1 distance is possible with $n = O\left(\frac{\sqrt{k}}{\varepsilon^2} \cdot \frac{k}{2^\ell}\right)$.

Proof. Recall that the centralized sample complexity is $O\left(\frac{\sqrt{k}}{\varepsilon^2}\right)$, e.g., via a χ^2-type test (and computing its expectation and variance). □

Two more to go, and public coins to use

We just proved 5 out of 7 upper bounds, via distribution simulation: all were **private-coin**, noninteractive.

The last two are public-coin upper bounds, and both will rely on some type of dimensionality reduction: use public randomness to project $\boldsymbol{p}$ to a lower-dimensional random subspace $\rightsquigarrow$ "domain compression"

Domain compression for $\mathcal{B}_d$

Theorem. For $\mathcal{B}_d$, noninteractive public-coin mean testing under ℓ_2 loss is possible with $n = O\left(\frac{\sqrt{d}}{\varepsilon^2} \cdot \sqrt{\frac{d}{\ell}}\right)$.

Domain compression for $\mathcal{B}_d$

Theorem. For $\mathcal{B}_d$, noninteractive public-coin mean testing under ℓ_2 loss is possible with $n = O\left(\frac{\sqrt{d}}{\varepsilon^2} \cdot \sqrt{\frac{d}{\ell}}\right)$.

Proof. Pick a common u.a.r. random vector $Z \in \{\pm 1\}^d$: all users replace their X_i by $X_i' \coloneqq Z \cdot X_i \in \{\pm 1\}^d$.
Conditioned on Z, new mean s.t. $\|Z \cdot \mu\|_2^2 = \|\mu\|_2^2$.

Partition the d coordinates in ℓ groups $S_1, \ldots, S_\ell$ of same size. User i computes $\mathbb{I}[\sum_{j \in S_t} X_{ij}' > 0]$ for all $1 \leq t \leq \ell$ and send those ℓ bits.

So the server gets n i.i.d. samples from some $\boldsymbol{p}_Z$ on $\{\pm 1\}^\ell$.

Domain compression for $\mathcal{B}_d$

Theorem. For $\mathcal{B}_d$, noninteractive public-coin mean testing under ℓ_2 loss is possible with $n = O\left(\frac{\sqrt{d}}{\varepsilon^2} \cdot \sqrt{\frac{d}{\ell}}\right)$.

Proof (cont'd). Why is this good?

- This $\boldsymbol{p}_Z$ is a product distribution on $\{\pm 1\}^\ell$
- If $\boldsymbol{p}$ has mean $\mu = \mathbf{0}$, then $\boldsymbol{p}_Z$ has mean $\mu_Z = \mathbf{0}$
- If $\boldsymbol{p}$ has mean $\|\mu\|_2 > \varepsilon$, "then"

$$\Pr_Z\left[\|\mu_Z\|_2 > \varepsilon \cdot \sqrt{\ell/d}\ \right] = \Omega(1)$$

Domain compression for $\mathcal{B}_d$

Theorem. For $\mathcal{B}_d$, noninteractive public-coin mean testing under ℓ_2 loss is possible with $n = O\left(\frac{\sqrt{d}}{\varepsilon^2} \cdot \sqrt{\frac{d}{\ell}}\right)$.

Proof (cont'd). This last part is not quite obvious. Helps to think of each $\frac{1}{\sqrt{|S_t|}} \sum_{j \in S_t} X'_{ij} = \sqrt{\frac{\ell}{d}} \sum_{j \in S_t} X_{ij} Z_j$ as roughly normal:

$$\mathrm{N_t} \approx \mathcal{N}\left(\sqrt{\frac{\ell}{d}} \sum_{j \in S_t} Z_j \mu_j \, , 1\right)$$

So t-th bit has parameter $\Pr[\, N_t \geq 0] = \Omega\left(\sqrt{\frac{\ell}{d}} \sum_{j \in S_t} Z_j \mu_j\right)$ …

Domain compression for $\mathcal{B}_d$

Theorem. For $\mathcal{B}_d$, noninteractive public-coin mean testing under ℓ_2 loss is possible with $n = O\left(\frac{\sqrt{d}}{\varepsilon^2} \cdot \sqrt{\frac{d}{\ell}}\right)$.

Proof (cont'd). The mean vector then satisfies

$$\mathbb{E}_Z[\|\mu_Z\|_2^2] \geq \frac{\ell}{d} \sum_{t=1}^{\ell} \left(\sum_{j \in S_t} Z_j \mu_j \right)^2 = \frac{\ell}{d} \|\mu\|_2^2$$

and (handwaving) we can show that

$$\Pr_Z\left[\|\mu_Z\|_2 > \varepsilon \cdot \sqrt{\ell/d}\,\right] = \Omega(1).$$

We are done: the server can do mean testing over $\{\pm 1\}^\ell$ with $\varepsilon' := \varepsilon\sqrt{\ell/d}$, for which $n = O\left(\frac{\sqrt{\ell}}{\varepsilon'^2}\right) = O\left(\frac{d}{\varepsilon^2\sqrt{\ell}}\right)$ is enough. □

Domain compression for Δ_k

Theorem ([ACT20d,ACHST20]). For Δ_k, noninteractive public-coin identity testing under ℓ_1 distance is possible with $n = O\left(\frac{\sqrt{k}}{\varepsilon^2} \cdot \sqrt{\frac{k}{2^\ell}}\right)$.

Domain compression for Δ_k

Theorem. For Δ_k, noninteractive public-coin identity testing under ℓ_1 distance is possible with $n = O\left(\frac{\sqrt{k}}{\varepsilon^2} \cdot \sqrt{\frac{k}{2^\ell}}\right)$.

Proof. Pick a common u.a.r. hash function $h: [k] \to [2^\ell]$: all users replace their X_i by $X_i' := h(X_i)$, which they can send.

So server gets n i.i.d. samples from some $\boldsymbol{p}_h$ on $[2^\ell]$. It also knows h, so can compute $\boldsymbol{q}_h$ (where $\boldsymbol{q}$ is the reference distribution).

All that remains is to do identity testing of $\boldsymbol{p}_h$ to $\boldsymbol{q}_h$…

Domain compression for Δ_k

Theorem. For Δ_k, noninteractive public-coin identity testing under ℓ_1 distance is possible with $n = O\left(\frac{\sqrt{k}}{\varepsilon^2} \cdot \sqrt{\frac{k}{2^\ell}}\right)$.

Proof (cont'd). Why is this good?

- Server has n i.i.d. samples from this $\boldsymbol{p}_h$ on $[2^\ell]$
- If $\boldsymbol{p} = \boldsymbol{q}$ then $\boldsymbol{p}_h = \boldsymbol{q}_h$
- If $\|\boldsymbol{p} - \boldsymbol{q}\|_1 > \varepsilon$, "then"

$$\Pr_h\left[\|\boldsymbol{p}_h - \boldsymbol{q}_h\|_1 > \varepsilon \cdot \sqrt{2^\ell / k}\right] = \Omega(1)$$

Domain compression for Δ_k

Theorem. For Δ_k, noninteractive public-coin identity testing under ℓ_1 distance is possible with $n = O\left(\frac{\sqrt{k}}{\varepsilon^2} \cdot \sqrt{\frac{k}{2^\ell}}\right)$.

Proof (cont'd). This last part is not obvious: going to handwave the argument. Proving the analogous statement for ℓ_2 is a bit simpler:

1. Check that $\mathbb{E}_h[\|\boldsymbol{p}_h - \boldsymbol{q}_h\|_2^2] \asymp \|\boldsymbol{p} - \boldsymbol{q}\|_2^2$
2. Bound the variance of $\|\boldsymbol{p}_h - \boldsymbol{q}_h\|_2^2$
3. Apply Paley-Zygmund's inequality.

(For the ℓ_1 statement, a few more ingredients are needed.)

Domain compression for Δ_k

Theorem. For Δ_k, noninteractive public-coin identity testing under ℓ_1 distance is possible with $n = O\left(\frac{\sqrt{k}}{\varepsilon^2} \cdot \sqrt{\frac{k}{2^\ell}}\right)$.

Proof (cont'd). Once we have this, we are done: the server can do identity testing to q_h over $[2^\ell]$with $\varepsilon' := \varepsilon\sqrt{2^\ell/k}$, for which

$$n = O\left(\frac{\sqrt{2^\ell}}{\varepsilon'^2}\right) = O\left(\frac{k}{\varepsilon^2\sqrt{2^\ell}}\right)$$

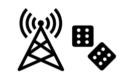
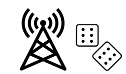
is enough. □

That’s seven upper bounds we proved.

(in ≈30 minutes)

That’s seven upper bounds we proved.
(in ≈30 minutes)

Estimation		
Δ_k, ℓ_1	$\frac{k}{\varepsilon^2} \cdot \frac{k}{\min\{2^\ell, k\}}$	$\frac{k}{\varepsilon^2} \cdot \frac{k}{\varrho^2}$
$\mathcal{B}_d, \ell_2$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\varrho^2}$
$\mathcal{G}_d, \ell_2$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \frac{d}{\varrho^2}$

Testing		
Δ_k, ℓ_1	$\frac{\sqrt{k}}{\varepsilon^2} \cdot \frac{k}{\min\{2^\ell, k\}}$	$\frac{\sqrt{k}}{\varepsilon^2} \cdot \sqrt{\frac{k}{\min\{2^\ell, k\}}}$
$\mathcal{B}_d, \ell_2$	$\frac{\sqrt{d}}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	$\frac{d}{\varepsilon^2} \cdot \sqrt{\frac{d}{\min\{\ell, d\}}}$
“Hide-and-Seek”	$\frac{\log d}{\varepsilon^2} \cdot \frac{d}{\min\{\ell, d\}}$	

Summary

This tutorial: techniques for proving lower bounds, in both **interactive** and **noninteractive** settings, for statistical estimation and testing under "local constraints."

Summary

This tutorial: techniques for proving lower bounds, in both **interactive** and **noninteractive** settings, for statistical estimation and testing under "local constraints."

I. Introduction — Clément

II. Lower Bounds for Estimation — Jayadev

III. Lower Bounds for Testing — Himanshu

IV. Some upper bounds, and discussion — Clément

Some open problems

First, happy to discuss those (and more) in detail during the conference, interactively! Please feel free reach out.

Some directions

First, happy to discuss those (and more) in detail during the conference, interactively! Please feel free reach out.

Open Problem #1: What if all users had different constraints? E.g., different bandwidth constraints, or different privacy requirements…

Open Problem #2: Other types of constraints! Linear measurements, threshold measurements (univariate case), or malicious noise à la Massart…

References and previous work

For a detailed bibliography:
www.cs.columbia.edu/~ccanonne/tutorial-focs2020/bibliography.html