Amplification of One-Way Information Complexity via Codes and Noise Sensitivity

Presenter: Omri Weinstein (NYU)

Marco Molinaro
TU Delft

David Woodruff
IBM Almaden

Grigory Yaroslavtsev
U. Penn

COMMUNICATION MODEL

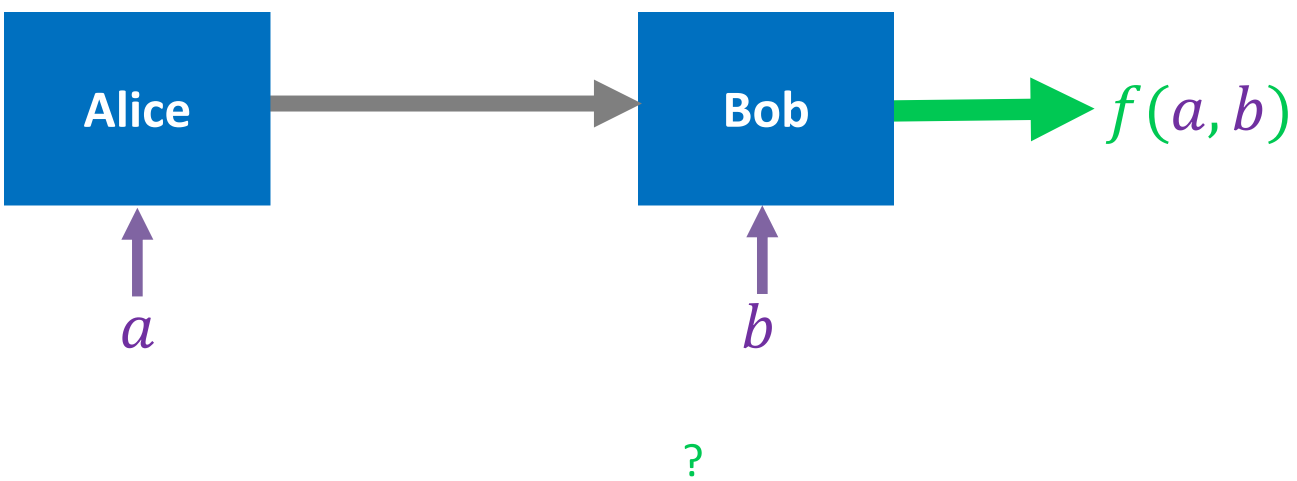
We consider one-way communication:

- 2 deterministic players: Alice has input a and Bob input b
- Joint function f

COMMUNICATION MODEL

We consider one-way communication:

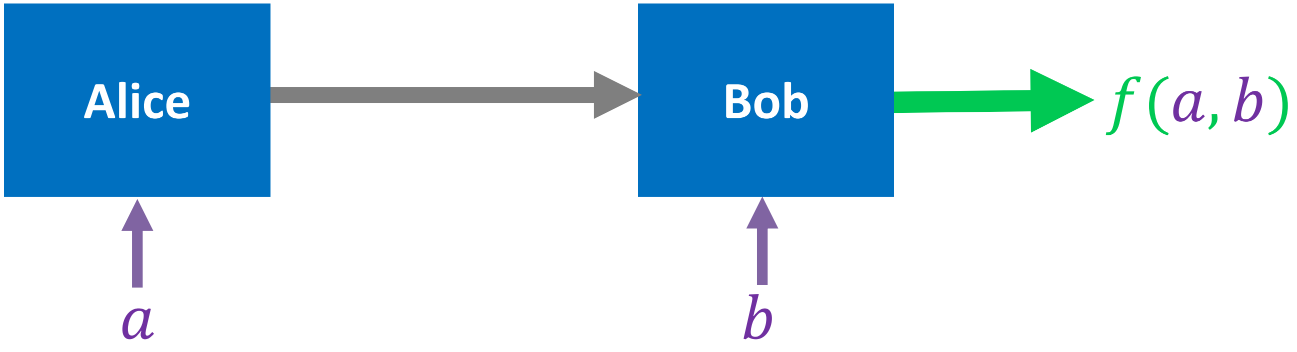
- 2 deterministic players: Alice has input a and Bob input b
- Joint function f
- Alice sends a message to Bob, Bob tries to output $f(a,b)$



COMMUNICATION MODEL

We consider one-way communication:

- 2 deterministic players: Alice has input a and Bob input b
- Joint function f
- Alice sends a message to Bob, Bob tries to output $f(a, b)$

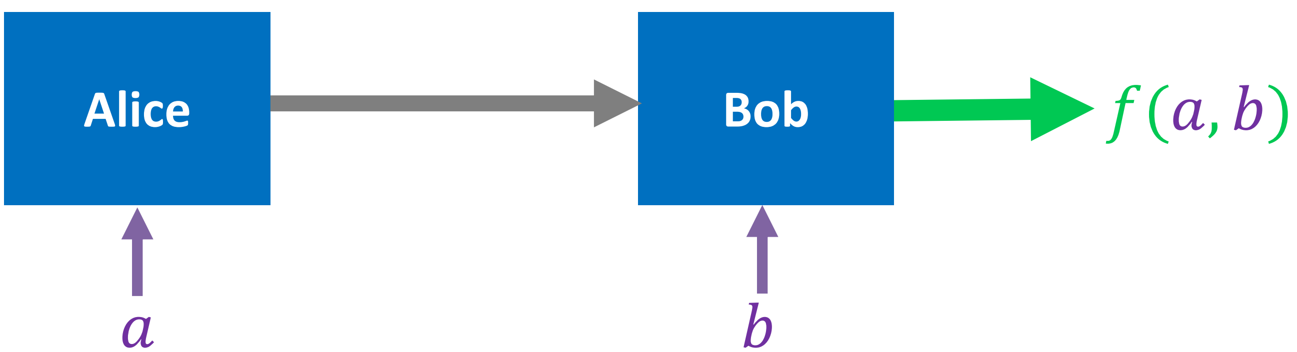


Ex: $x, y \in \{0,1\}^n$, want to output $x \stackrel{?}{=} y$

COMMUNICATION MODEL

We consider one-way communication:

- 2 deterministic players: Alice has input a and Bob input b
- Joint function f
- Alice sends a message to Bob, Bob tries to output $f(a,b)$



Ex: $x, y \in \{0,1\}^n$, want to output $x \stackrel{?}{=} y$

Want small communication from Alice to Bob

COMMUNICATION MODEL

We consider product distribution $\mu \times \nu$ over Alice $\times$ Bob's inputs

COMMUNICATION MODEL

We consider product distribution $\mu \times \nu$ over Alice $\times$ Bob's inputs

Goal: Compute $f(a, b)$ for all but δ $(\mu \times \nu)$-fraction of inputs while minimizing size of Alice's longest message

COMMUNICATION MODEL

We consider product distribution $\mu \times \nu$ over Alice $\times$ Bob's inputs

Goal: Compute $f(a,b)$ for all but δ $(\mu \times \nu)$-fraction of inputs while minimizing size of Alice's longest message

matrix view:

function f

	b1	b2	b3	b4	b5
a1	0	1	1	0	0
a2	0	0	1	0	1
a3	1	0	0	0	0
a4	0	0	0	1	0

$\xrightarrow{\text{protocol}}$

	b1	b2	b3	b4	b5
a1	0	0	1	0	0
a2	0	0	1	0	1
a3	1	0	1	0	0
a4	1	0	0	1	0

COMMUNICATION MODEL

We consider product distribution $\mu \times \nu$ over Alice $\times$ Bob's inputs

Goal: Compute $f(a,b)$ for all but δ $(\mu \times \nu)$-fraction of inputs while minimizing size of Alice's longest message

matrix view:

function f

a1	0	1	1	0	0
a2	0	0	1	0	1
a3	1	0	0	0	0
a4	0	0	0	1	0
	b1	b2	b3	b4	b5

protocol

a1	0	0	1	0	0
a2	0	0	1	0	1
a3	1	0	1	0	0
a4	1	0	0	1	0
	b1	b2	b3	b4	b5

$D_{\mu\times\nu,\delta}(f) =$ minimum communication over all δ-error protocols

COMMUNICATION MODEL

We consider product distribution $\mu \times \nu$ over Alice $\times$ Bob's inputs

Goal: Compute $f(a, b)$ for all but δ $(\mu \times \nu)$-fraction of inputs while minimizing size of Alice's longest message

matrix view:

function f

	b1	b2	b3	b4	b5
a1	0	1	1	0	0
a2	0	0	1	0	1
a3	1	0	0	0	0
a4	0	0	0	1	0

protocol

	b1	b2	b3	b4	b5
a1	0	0	1	0	0
a2	0	0	1	0	1
a3	1	0	1	0	0
a4	1	0	0	1	0

$D_{\mu\times\nu,\delta}(f)$ = minimum communication over all δ-error protocols

One-way product complexity

$D_{\times,\delta}(f) = \max D_{\mu\times\nu,\delta}(f)$ over product distributions

COMMUNICATION MODEL

We consider product distribution $\mu \times \nu$ over Alice $\times$ Bob's inputs

Goal: Compute $f(a,b)$ for all but δ $(\mu \times \nu)$-fraction of inputs while minimizing size of Alice's longest message

Our goal: Lower bound one-way product complexity

a4	0	0	0	1	0
	b1	b2	b3	b4	b5

a4	1	0	0	1	0
	b1	b2	b3	b4	b5

$D_{\mu\times\nu,\delta}(f)$ = minimum communication over all δ-error protocols

One-way product complexity

$$D_{\times,\delta}(f) = \max D_{\mu\times\nu,\delta}(f) \text{ over product distributions}$$

ROWS AS METRIC SPACE (1/2)

We can see the rows of matrix f as a metric space:

- Each row is 0-1 vector
- Use weighted hamming distance, weighted by distribution ν

$$f$$

$$\begin{array}{c|ccccc|}
a1 & 0 & 1 & 1 & 0 & 0 \\
a2 & 0 & 0 & 1 & 0 & 1 \\
a3 & 1 & 0 & 0 & 0 & 0 \\
a4 & 0 & 0 & 0 & 1 & 0 \\
\hline
 & b1 & b2 & b3 & b4 & b5
\end{array}$$

ROWS AS METRIC SPACE (1/2)

We can see the rows of matrix f as a metric space:

- Each row is 0-1 vector
- Use weighted hamming distance, weighted by distribution ν

f

a1	0	1	1	0	0
a2	0	0	1	0	1
a3	1	0	0	0	0
a4	0	0	0	1	0
	b1	b2	b3	b4	b5

Ex: $dist\big(row(a1), row(a2)\big) = \nu(b2) + \nu(b5)$

ROWS AS METRIC SPACE (1/2)

We can see the rows of matrix f as a metric space:

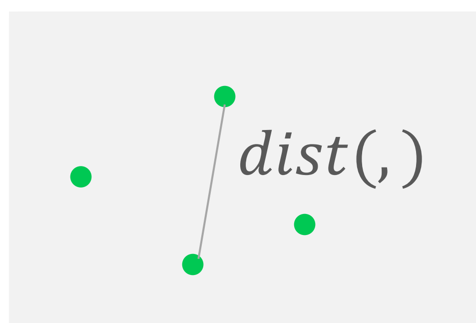
- Each row is 0-1 vector
- Use weighted hamming distance, weighted by distribution ν

f

	b1	b2	b3	b4	b5
a1	0	1	1	0	0
a2	0	0	1	0	1
a3	1	0	0	0	0
a4	0	0	0	1	0

Ex: $dist\big(row(a1), row(a2)\big) = \nu(b2) + \nu(b5)$

metric space view of f:



ROWS AS METRIC SPACE (2/2)

Can think protocol provides approximation to the rows of f

f

	b1	b2	b3	b4	b5
a1	0	1	1	0	0
a2	0	0	1	0	1
a3	1	0	0	0	0
a4	0	0	0	1	0

$\xrightarrow{\text{protocol}}$

	b1	b2	b3	b4	b5
a1	0	0	1	0	0
a2	0	0	1	0	1
a3	1	0	1	0	0
a4	1	0	0	1	0

ROWS AS METRIC SPACE (2/2)

Can think protocol provides approximation to the rows of f

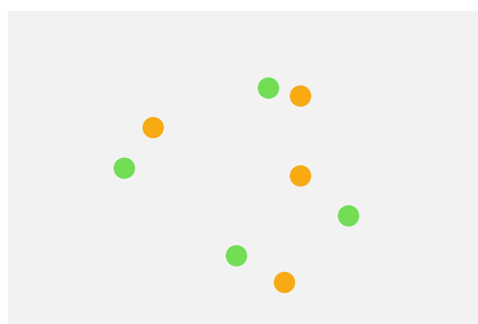
f

	b1	b2	b3	b4	b5
a1	0	1	1	0	0
a2	0	0	1	0	1
a3	1	0	0	0	0
a4	0	0	0	1	0

protocol $\longrightarrow$

	b1	b2	b3	b4	b5
a1	0	0	1	0	0
a2	0	0	1	0	1
a3	1	0	1	0	0
a4	1	0	0	1	0

metric space view:

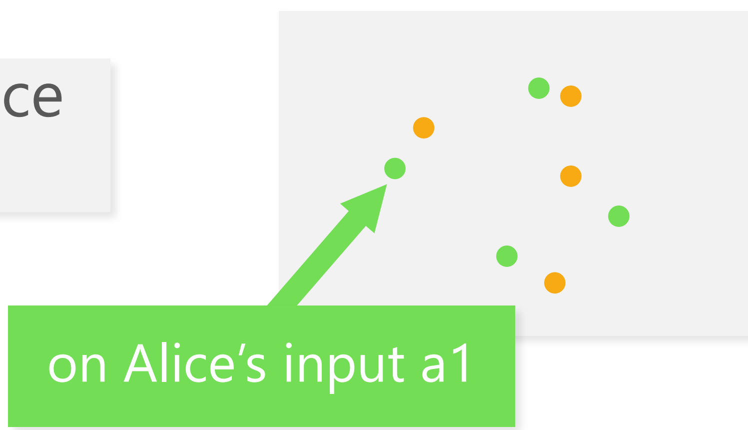


ROWS AS METRIC SPACE (2/2)

Can think protocol provides approximation to the rows of f

$$
\begin{array}{c|ccccc|}
 & & & f & & \\
a1 & 0 & 1 & 1 & 0 & 0 \\
a2 & 0 & 0 & 1 & 0 & 1 \\
a3 & 1 & 0 & 0 & 0 & 0 \\
a4 & 0 & 0 & 0 & 1 & 0 \\
 & b1 & b2 & b3 & b4 & b5
\end{array}
\xrightarrow{\text{protocol}}
\begin{array}{c|ccccc|}
a1 & 0 & 0 & 1 & 0 & 0 \\
a2 & 0 & 0 & 1 & 0 & 1 \\
a3 & 1 & 0 & 1 & 0 & 0 \\
a4 & 1 & 0 & 0 & 1 & 0 \\
 & b1 & b2 & b3 & b4 & b5
\end{array}
$$

metric space view:



ROWS AS METRIC SPACE (2/2)

Can think protocol provides approximation to the rows of f

f

	b1	b2	b3	b4	b5
a1	0	1	1	0	0
a2	0	0	1	0	1
a3	1	0	0	0	0
a4	0	0	0	1	0

$\xrightarrow{\text{protocol}}$

	b1	b2	b3	b4	b5
a1	0	0	1	0	0
a2	0	0	1	0	1
a3	1	0	1	0	0
a4	1	0	0	1	0

metric space view:

ROWS AS METRIC SPACE (2/2)

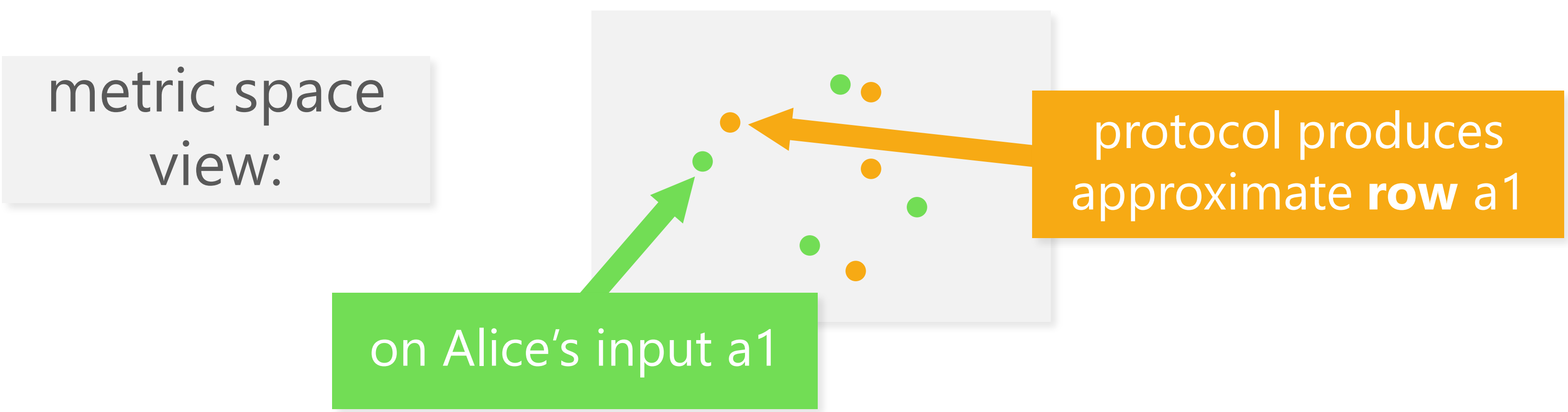
Idea for lower bound:

If protocol has low error (orange points close to green) and rows of f are far apart…

⇒ can use protocol's output to recover Alice's input

⇒ protocol reveals a lot of information

⇒ protocol has large communication!



LOWER BOUND VIA CODES

First result: Suppose the rows of f form a (δ, β)-code (far apart). Then

$$D_{\mu \times \nu, \delta}(f) \geq \log \frac{1}{\beta}$$

LOWER BOUND VIA CODES

First result: Suppose the rows of f form a (δ, β)-code (far apart). Then

$$D_{\mu\times\nu,\delta}(f) \geq \log\frac{1}{\beta}$$

Def: Rows form (δ, β)-code if the probability (wrt μ) that rows are within distance δ is at most β

LOWER BOUND VIA CODES

First result: Suppose the rows of f form a (δ, β)-code (far apart). Then

$$D_{\mu\times\nu,\delta}(f) \geq \log\frac{1}{\beta}$$

Def: Rows form (δ, β)-code if the probability (wrt μ) that rows are within distance δ is at most β

Obs: Readily recovers lower bound of Dasgupta-Kumar-Sivakumar '12 on Sparse Set Disjointness function

CONSEQUENCE 1: VC DIMENSION

CONSEQUENCE 1: VC DIMENSION

Obs: Connection (α, β)-codes ≈ packing numbers ≈ VC-dim

CONSEQUENCE 1: VC DIMENSION

Obs: Connection (α, β)-codes ≈ packing numbers ≈ VC-dim

Kremer-Nisan-Ron '99 characterizes one-way product complexity via VC dimension of rows, for constant error δ

CONSEQUENCE 1: VC DIMENSION

Obs: Connection (α, β)-codes ≈ packing numbers ≈ VC-dim

Kremer-Nisan-Ron '99 characterizes one-way product complexity via VC dimension of rows, for constant error δ

Thm: [KNR99] Let VC be the VC-dimension of rows of f. Then

$$VC * \left(1 - H(\delta)\right) \leq D_{\times,\delta}(f) \leq VC * O\left(\frac{1}{\delta}\log\frac{1}{\delta}\right)$$

CONSEQUENCE 1: VC DIMENSION

Obs: Connection (α, β)-codes ≈ packing numbers ≈ VC-dim

Kremer-Nisan-Ron '99 characterizes one-way product complexity via VC dimension of rows, for constant error δ

Thm: [KNR99] Let VC be the VC-dimension of rows of f. Then

$$VC * \left(1 - H(\delta)\right) \leq D_{\times,\delta}(f) \leq VC * O\left(\frac{1}{\delta}\log\frac{1}{\delta}\right)$$

We get exponential improvement wrt error δ

CONSEQUENCE 1: VC DIMENSION

Obs: Connection (α,β)-codes ≈ packing numbers ≈ VC-dim

Kremer-Nisan-Ron '99 characterizes one-way product complexity via VC dimension of rows, for constant error δ

Thm: [KNR99] Let VC be the VC-dimension of rows of f. Then

$$VC * (1 - H(\delta)) \le D_{\times,\delta}(f) \le VC * O\left(\frac{1}{\delta}\log\frac{1}{\delta}\right)$$

We get exponential improvement wrt error δ

Thm: $VC * (1 - H(\delta)) \le D_{\times,\delta}(f) \le VC * O\left(\log\frac{1}{\delta}\right)$

CONSEQUENCE 2: COMPOSITION

CONSEQUENCE 2: COMPOSITION

Now Alice and Bob want to compute a composed function:

$$g\big(f(a_1, b_1), f(a_2, b_2), \dots, f(a_n, b_n)\big)$$

CONSEQUENCE 2: COMPOSITION

Now Alice and Bob want to compute a composed function:

$$g\big(f(a_1, b_1), f(a_2, b_2), \dots, f(a_n, b_n)\big)$$

Ex: g = XOR, g = AND, g = OR

CONSEQUENCE 2: COMPOSITION

Now Alice and Bob want to compute a composed function:

$$g\big(f(a_1, b_1), f(a_2, b_2), \dots, f(a_n, b_n)\big)$$

Ex: g = XOR, g = AND, g = OR

(α, β)-codes give generic way for lower bounding the communication for composed function $g(f)$ based on f and g

CONSEQUENCE 2: COMPOSITION

Now Alice and Bob want to compute a composed function:

$$g\big(f(a_1, b_1), f(a_2, b_2), \dots, f(a_n, b_n)\big)$$

Ex: g = XOR, g = AND, g = OR

(α, β)-codes give generic way for lower bounding the communication for composed function $g(f)$ based on f and g

Thm (oversimplified): Suppose the rows of f form a (δ, β)-code. Then computing $g(f)$ with error $NS_\delta(g)$ needs comm.

$$D_{\times, NS_\delta(g)}(g(f)) \geq n * \Omega\left(\log \frac{1}{\beta}\right)$$

CONSEQUENCE 2: COMPOSITION

Now Alice and Bob want to compute a composed function:

$$g\big(f(a_1, b_1), f(a_2, b_2), \dots, f(a_n, b_n)\big)$$

Ex: g = XOR, g

(α, β)-codes g
communicatio

Similar to noise sensitivity of g:

≈ probability that g's output changes if we flip inputs with probability δ

Thm (oversimplified): Suppose the rows of f form a (δ, β)-code. Then computing $g(f)$ with error $NS_\delta(g)$ needs comm.

$$D_{\times, NS_\delta(g)}(g(f)) \geq n * \Omega\left(\log \frac{1}{\beta}\right)$$

CONSEQUENCE 2: COMPOSITION

Thm (oversimplified): If the rows of f form a (δ, β)-code then computing $g(f)$ with error $NS_\delta(g)$ requires communication:

$$D_{\times, NS_\delta(g)}(g(f)) \geq n * \Omega\left(\log \frac{1}{\beta}\right)$$

CONSEQUENCE 2: COMPOSITION

Thm (oversimplified): If the rows of f form a (δ, β)-code then computing $g(f)$ with error $NS_\delta(g)$ requires communication:

$$D_{\times, NS_\delta(g)}(g(f)) \geq n * \Omega\left(\log \frac{1}{\beta}\right)$$

Ex: If g = XOR on n bits, $NS_{\frac{1}{n}}(g) = \frac{1}{4}$

CONSEQUENCE 2: COMPOSITION

Thm (oversimplified): If the rows of f form a (δ, β)-code then computing $g(f)$ with error $NS_\delta(g)$ requires communication:

$$D_{\times, NS_\delta(g)}(g(f)) \geq n * \Omega\left(\log \frac{1}{\beta}\right)$$

Ex: If g = XOR on n bits, $NS_{\frac{1}{n}}(g) = \frac{1}{4}$

The theorem then gives a stronger direct sum result for XOR (need to solve each "copy" of f with much higher probability)

CONSEQUENCE 2: COMPOSITION

Thm (oversimplified): If the rows of f form a (δ, β)-code then computing $g(f)$ with error $NS_\delta(g)$ requires communication:

$$D_{\times, NS_\delta(g)}(g(f)) \geq n * \Omega\left(\log \frac{1}{\beta}\right)$$

Ex: If g = XOR on n bits, $NS_{\frac{1}{n}}(g) = \frac{1}{4}$

The theorem then gives a stronger direct sum result for XOR (need to solve each "copy" of f with much higher probability)

Corollary: For any function f

$$D_{\times, 1/4}(XOR(f)) \geq n * D_{\times, \frac{1}{n}}(f)$$

CONSEQUENCE 3: STREAMING

CONSEQUENCE 3: STREAMING

Approximate closest pair $\ell_p(\boldsymbol{n}, \boldsymbol{d}, \boldsymbol{M}, \boldsymbol{\epsilon}, \boldsymbol{\theta})$:

CONSEQUENCE 3: STREAMING

Approximate closest pair $\ell_p(\boldsymbol{n}, \boldsymbol{d}, \boldsymbol{M}, \boldsymbol{\epsilon}, \boldsymbol{\theta})$**:**

Given one pass over a stream $v_1 \ldots v_{\boldsymbol{n}}$ of vectors in $[\pm \boldsymbol{M}]^{\boldsymbol{d}}$ decide whether:

1. For all $i \neq j$ it holds that $\left|\left|v^i - v^j\right|\right|_p^p \geq (1 + \boldsymbol{\epsilon})\theta$
2. There exist $i \neq j$ such that $\left|\left|v^i - v^j\right|\right|_p^p \leq (1 - \boldsymbol{\epsilon})\theta$

CONSEQUENCE 3: STREAMING

Approximate closest pair $\ell_p(\boldsymbol{n}, \boldsymbol{d}, \boldsymbol{M}, \boldsymbol{\epsilon}, \boldsymbol{\theta})$:
Given one pass over a stream $v_1 \dots v_{\boldsymbol{n}}$ of vectors in $[\pm \boldsymbol{M}]^{\boldsymbol{d}}$ decide whether:

1. For all $i \neq j$ it holds that $\left|\left|v^i - v^j\right|\right|_p^p \geq (1 + \boldsymbol{\epsilon})\theta$
2. There exist $i \neq j$ such that $\left|\left|v^i - v^j\right|\right|_p^p \leq (1 - \boldsymbol{\epsilon})\theta$

Theorem (simplified):
Any streaming algorithms for approximate closest pair problem $\ell_p(\boldsymbol{n}, \boldsymbol{d}, \boldsymbol{M}, \boldsymbol{\epsilon}, \boldsymbol{\theta})$ with error δ takes space:

$$\Omega\left(\frac{\boldsymbol{n}}{\boldsymbol{\epsilon}^2} \log \frac{\boldsymbol{n}}{\delta} \left(\log \boldsymbol{d} + \log \boldsymbol{M}\right)\right)$$

CONSEQUENCE 4: SKETCHING

CONSEQUENCE 4: SKETCHING

Approximate largest entry in matrix product:

CONSEQUENCE 4: SKETCHING

Approximate largest entry in matrix product:
Given one pass over a stream representing entries of a matrix $\boldsymbol{A} \in [\pm\boldsymbol{M}]^{\boldsymbol{n}\times\boldsymbol{n}}$ construct an $\boldsymbol{n} \times \boldsymbol{d}$ sketch matrix $\boldsymbol{S}$ such that for any $\boldsymbol{B} \in [\pm\boldsymbol{M}]^{\boldsymbol{n}\times\boldsymbol{n}}$ from $\boldsymbol{AS}$ and **B** only it is possible to compute whether:

1. $(\boldsymbol{AB})_{ij} \geq (1+\boldsymbol{\epsilon})\theta$ for some $i, j \in [n]$
2. $(\boldsymbol{AB})_{ij} \leq \theta$ for all $i, j \in [n]$

CONSEQUENCE 4: SKETCHING

Approximate largest entry in matrix product:
Given one pass over a stream representing entries of a matrix $\boldsymbol{A} \in [\pm \boldsymbol{M}]^{\boldsymbol{n} \times \boldsymbol{n}}$ construct an $\boldsymbol{n} \times \boldsymbol{d}$ sketch matrix $\boldsymbol{S}$ such that for any $\boldsymbol{B} \in [\pm \boldsymbol{M}]^{\boldsymbol{n} \times \boldsymbol{n}}$ from $\boldsymbol{AS}$ and **B** only it is possible to compute whether:

1. $(\boldsymbol{AB})_{ij} \geq (1 + \boldsymbol{\epsilon})\theta$ for some $i, j \in [n]$
2. $(\boldsymbol{AB})_{ij} \leq \theta$ for all $i, j \in [n]$

Theorem (simplified):
Number of bits to specify linear sketch AS:

$$\Omega\left(\frac{\boldsymbol{n}}{\boldsymbol{\epsilon}^2} \log \frac{\boldsymbol{n}}{\delta} \left(\log \boldsymbol{d} + \log \boldsymbol{M}\right)\right)$$

(matching upper bounds for this and streaming via JL).

THANK YOU!