Communication Complexity

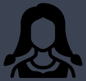
by Omid Yaghoubi

on February 8, 2023

Introduction

Two-Party Communication Complexity

- X, Y, Z are finite
- $f : X \times Y \to Z$
- (alice) $x \in X$ and (bob) $y \in Y$
- $f(x, y) = ?$



Alice $x \in X$

$$\downarrow 0 \quad \uparrow 1 \quad \downarrow 1 \quad \uparrow 0 \quad \downarrow 1$$

Bob $y \in Y$

The model

» Protocol

* Specifies whether the execution terminated
* Specifies what is the output
* Specifies the sender (Alice or Bob)
* Specifies what message the sender should send next [1]
* Transcript: *The sequence of bits sent back and forth*
* Let $s_\pi(x,y)$ denote the transcript $\pi(x,y)$

$$A: \quad X \times \underbrace{\{0,1\}^*}_{\text{Transcript}} \to \{0,1\}$$

$$B: \quad Y \times \underbrace{\{0,1\}^*}_{\text{Transcript}} \to \{0,1\}$$

$$N: \quad \underbrace{\{0,1\}^*}_{\text{Transcript}} \to \{A, B, STOP\}$$

» Example: Naïve or

OR function

$$OR(x,y) = 1 \Leftrightarrow (x_1 \vee x_2 \vee \ldots x_n) \vee (y_1 \vee y_2 \vee \ldots y_n) = True$$

- Alice sends x to Bob
- Bob computes $z = f(x,y)$ and send z to Alice

$$
\begin{array}{llcl}
(1) & N(\epsilon) = A & \Rightarrow & A(00,\epsilon) = 0 \\
(2) & N(0) = A & \Rightarrow & A(00,0) = 0 \\
(3) & N(00) = B & \Rightarrow & B(01,00) = 1 \\
(4) & N(001) = STOP & \Rightarrow & A(00,001) = B(01,001) = 1 = \pi(00,01)
\end{array}
$$

» Cost and Communication Complexity

* π computes f iff $\forall(x,y) : f(x,y) = \pi(x,y)$
* Cost: Worst case (over all $(x,y) \in X \times Y$) of $|s_\pi(x,y)|$ (example?)
* Communication complexity of f: The cost of best π which computes f

$$cost(\pi) \quad := \quad \max_{\{(x,y)\in X\times Y : |x|=|y|=n\}} |s_\pi(x,y)|$$

$$D(f) \quad := \quad \min_{\{\pi : \pi \text{ computes } f\}} cost(\pi)$$

» Naïve solution

For every $f : X \times Y \to Z$:

$$D(f) \leq \lceil \log X \rceil + \lceil \log Z \rceil$$

$$D(f) \leq \lceil \log Y \rceil + \lceil \log Z \rceil$$

Upper bound

* Parity
* Majority
* Median
* P^{cc}

Upper bound

- Parity
- Majority
- Median
- P^{cc}

» PARITY

$$PARITY(x,y) = \left(\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i\right) \mod 2$$

$D(PARITY) \leq 2$

1. $\text{Alice} \xrightarrow{\left(\sum_{i=1}^{n} x_i\right) \mod 2} \text{Bob}$
2. $\text{Bob} \xrightarrow{\left(\sum_{i=1}^{n} y_i\right) \mod 2} \text{Alice}$

» PARITY

$D(PARITY) \geq 2$

* Suppose $D(PARITY) < 2$
* (wlog) $N(\epsilon) = A$
* $A(x, b) = A(x, A(x, \epsilon)) = PARITY(x, y)$
* $PARITY(x, y)$ not depends on y!
* Flip one bit in y to change $PARITY(x, y)$ (Contradiction!)

Corollary: $D(PARITY) = 2$

Upper bound

- Parity
- Majority
- Median
- P^{cc}

» MAJORITY

$$MAJ(x,y) = 1 \Leftrightarrow \#_1(x.y) \geq \#_0(x.y)$$

$$D(MAJ) \leq O(\log n)$$

1. Alice $\xrightarrow{\#_1 x}$ Bob
2. Bob $\xrightarrow{\#_1 y}$ Alice

Upper bound

* Parity
* Majority
* Median
* P^{cc}

» Median problem

* Characteristic vector: $s_i = 1 \Leftrightarrow i \in [s]$
* Input: $[x] \subseteq \{2, 4, \ldots, 2n\}$
* Input: $[y] \subseteq \{1, 3, \ldots, 2n-1\}$
* Goal: $\mathit{Median}\,\{[x] \cup [y]\}$
* Note: $x, y \in \{0,1\}^n$ (Real inputs!)
* Note: $[x] \cup [y] \subseteq \{1, 2, \ldots, 2n\}$
* Naïve : $D(\mathit{MED}) \leq n + \lceil \log 2n \rceil$

» Median protocol

Claim: $D(MED) \leq O(\log^2 n)$!

Idea: Binary search !

Protocol:

1. Suppose $MED(x,y) \in [i,j]$
2. $mid = \lfloor \frac{i+j}{2} \rfloor$
3. Alice: $R_x = |[mid+1,j] \cap [x]|$ and $L_x = |[i,mid] \cap [x]|$
4. Bob : $R_y = |[mid+1,j] \cap [y]|$ and $L_y = |[i,mid] \cap [y]|$
5. Alice $\xrightarrow{L_x, R_x}$ Bob
6. Bob $\xrightarrow{L_y, R_y}$ Alice
7. Update $[i,j]$ to $[i,mid]$ or $[mid+1,j]$

[2]

» Cost of protocol

* $|L_x| + |R_x| + |L_y| + |R_y| \leq 4.\lceil \log 2n \rceil$
* Number of iterations $\leq O(\log 2n)$
* Hence $D(MED) \leq O(\log^2 n)$

Upper bound

* Parity
* Majority
* Median
* P^{cc}

» P^{cc}

$$P^{cc} := \{f : D(f) = O(poly(\log n))\}$$

$$MED \in P^{cc}$$

Lower bound

* EQUALITY
* Fooling set method

Lower bound

* EQUALITY
* Fooling set method

» EQUALITY

* $EQ(x, y) = 1 \Leftrightarrow x = y$
* $D(EQ) \leq n + 1$ (Naïve)
* Claim: $D(EQ) \geq n$

» Mix and match lemma

If

$$s_\pi(x,y) = s_\pi(x',y')$$

then

$$s_\pi(x,y) = s_\pi(x',y') = s_\pi(x,y') = s_\pi(x',y)$$

Proof: By induction on length of the transcript.

Intuition: :)

Induction Step

* By assumption:
$s_\pi(x,y) = s_\pi(x',y') = b_1, b_2, \dots, b_i, b_{i+1} \dots, b_k$
* By induction hypothesis:
$s_\pi(x,y)[1,i] = s_\pi(x',y')[1,i]$
$= s_\pi(x,y')[1,i] = s_\pi(x',y)[1,i] = b_1, b_2, \dots, b_i$
* (wlog) $N(b_1, b_2, \dots, b_i) = A$
* $s_\pi(x',y)[i+1] = A(x', b_1, b_2, \dots, b_i)$
* $s_\pi(x,y')[i+1] = A(x, b_1, b_2, \dots, b_i)$
*

$$\begin{aligned} s_\pi(x,y)[i+1] &= s_\pi(x',y')[i+1] \\ &= A(x, b_1, b_2, \dots, b_i) \\ &= A(x', b_1, b_2, \dots, b_i) \\ &= s_\pi(x,y')[i+1] \\ &= s_\pi(x',y)[i+1] \\ &= b_{i+1} \end{aligned}$$

» Corollary

Corollary: If the previous lemma holds and π computes f then:

$$f(x,y) = f(x',y') = f(x',y) = f(x,y')$$

Proof:

$$\begin{array}{llllllll}
(1) & f(x,y) & = & \pi(x,y) & = & A(x,s_\pi(x,y)) & = & B(y,s_\pi(x,y)) \\
(2) & f(x',y') & = & \pi(x',y') & = & A(x',s_\pi(x,y)) & = & B(y',s_\pi(x,y)) \\
(3) & f(x',y) & = & \pi(x',y) & = & A(x',s_\pi(x,y)) & = & B(y,s_\pi(x,y)) \\
(4) & f(x,y') & = & \pi(x,y') & = & A(x,s_\pi(x,y)) & = & B(y',s_\pi(x,y))
\end{array}$$

» Lower bound for *EQ*

Claim: $D(EQ) \geq n$

* Assume $D(EQ) < n$
* $|\{s : |s| < n\}| = 2^n - 1$
* $2^n - 1$ distinct transcripts
* $FS := \{(x, x) : x \in \{0, 1\}^n\}$
* $|FS| = 2^n$
* By Pigeonhole Principle :
  $\exists\{(x, x), (y, y)\} \subseteq FS$ s.t $x \neq y$ and $s_\pi(x, x) = s_\pi(y, y)$
* By previous lemma
  $s_\pi(x, x) = s_\pi(y, y) = s_\pi(x, y) = s_\pi(y, x)$
* Hence $EQ(x, x) = EQ(x, y)$ which is a contradiction

$\square$

Lower bound

* EQUALITY
* Fooling set method

» Fooling set

* $FS \subseteq X \times Y$
* For all $\{(x,y),(x',y')\} \subseteq FS \Rightarrow f(x,y) = f(x',y')$
* $f(x,y) \neq f(x',y) \vee f(x,y) \neq f(x,y')$
* $cost(\pi) \geq \lceil \log |FS| \rceil$

Applications

* Turing machines

Applications

* Turing machines

» *PAL* lowerbound

$$PAL := \{w \in \{0,1\}^* : w = w^R\}$$

Claim: $PAL \in \Omega(n^2)$ (one-tape TM)

Idea: $D(EQ) \geq n \Rightarrow PAL \in \Omega(n^2)$ [1]

» $PAL \in \Omega(n^2)$

x_1 ... $x_{\frac{n}{3}}$	$x_{\frac{n}{3}+1}$ ... $x_{\frac{2n}{3}}$	$x_{\frac{2n}{3}+1}$ ... x_n	$\square$ ...
First	Middle	Last	
	$\frac{n}{3} \leq i \leq \frac{2n}{3}$		

- $EQ(x, y) = 1 \Leftrightarrow x0^m y^R \in PAL$
- Suppose $cross(i) = k$
- Claim: $\exists \pi$ s.t. computes EQ and $cost(\pi) \leq k.\lceil \log |Q| \rceil + 1$
- $D(EQUALITY) \geq m$
- Hence $k.\lceil \log |Q| \rceil + 1 \geq m$
- $m = \frac{n}{3} \Rightarrow k \in \Omega(n)$
- $\frac{n}{3}$ different choices for i
- Hence $PAL \in \Omega(n^2)$

Matrix form

» Matrix form

$$M_{f_{2^{n=3}\times 2^{n=3}}} := \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

» M_{EQ}

$$M_{OR_{2^3\times 2^3}} := \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

» M_{EQ}

$$M_{EQ_{2^3\times 2^3}} := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Thank you!

E. Kushilevitz, "Communication complexity," in *Advances in Computers*, vol. 44, pp. 331–360, Elsevier, 1997.

A. Rao and A. Yehudayoff, *Communication Complexity: and Applications*.
Cambridge University Press, 2020.