Communication Complexity

Jake Kinsella and Max von Hippel

Northeastern University

April 9, 2021

If Alice knows x, and Bob knows y, how many bits of information must they communicate, in order for both Alice and Bob to know $f(x, y)$?

(Max)

Consider a two-party communication problem, in which the participants

(a) Alice

and

(b) Bob

participate to compute a function:

$$f : \underbrace{\mathbb{B}^n}_{\text{Alice's input}} \times \underbrace{\mathbb{B}^n}_{\text{Bob's input}} \to \underbrace{\mathbb{B}}_{\text{global output}}$$

(Max)

The players can come up with a *protocol* $\Pi = (p_1, \ldots, p_t)$, namely, for some natural $t \in \mathbb{N}$, a sequence of t-many functions $p_i : \mathbb{B}^* \to \mathbb{B}^*$ such that the communication between the players looks like this ...

(Max)

Alice is given input x.

(Max)

Alice is given input x.

Hello Bob. I can't reveal x, but $p_1(x)$ is $\texttt{p1}$.

(Max)

Alice is given input x.

> Hello Bob. I can't reveal x, but $p_1(x)$ is $\texttt{p1}$.

Bob is given input y.

(Max)

Alice is given input x.

Hello Bob. I can't reveal x, but $p_1(x)$ is $\texttt{p1}$.

Bob is given input y.

Thanks Alice. I can't reveal y, but $p_2(y, \texttt{p1})$ is $\texttt{p2}$.

(Max)

Alice is given input x.

Hello Bob. I can't reveal x, but $p_1(x)$ is $\mathtt{p1}$.

Bob is given input y.

Thanks Alice. I can't reveal y, but $p_2(y, \mathtt{p1})$ is $\mathtt{p2}$.

... yada yada yada ...

(Max)

Alice is given input x.

Hello Bob. I can't reveal x, but $p_1(x)$ is $\texttt{p1}$.

Bob is given input y.

Thanks Alice. I can't reveal y, but $p_2(y, \texttt{p1})$ is $\texttt{p2}$.

... yada yada yada ...

Pleasure doing business with you Bob. My final clue for you is that $p_{n-1}(x, \texttt{p1}, \ldots, \texttt{pn-2})$ is $\texttt{pn-1}$.

(Max)

Alice is given input x.

Hello Bob. I can't reveal x, but $p_1(x)$ is $\texttt{p1}$.

Bob is given input y.

Thanks Alice. I can't reveal y, but $p_2(y, \texttt{p1})$ is $\texttt{p2}$.

... yada yada yada ...

Pleasure doing business with you Bob. My final clue for you is that $p_{n-1}(x, \texttt{p1}, \ldots, \texttt{pn-2})$ is $\texttt{pn-1}$.

Rad. Then $p_n(y, \texttt{p1}, \ldots, \texttt{pn-1})$ is $\texttt{pn}$. TTFN!

(Max)

- The functions p_i can be *anything* so long as they are well-defined. E.g., could solve the Halting Problem.
- After the final message, *both parties* must know $f(x, y)$.

(Max)

- The functions p_i can be *anything* so long as they are well-defined. E.g., could solve the Halting Problem.
- After the final message, *both parties* must know $f(x, y)$.

Definition (Communication Complexity)

Suppose Π is a protocol for f in which at most t bits are communicated between Alice and Bob. Then the *communication complexity* of Π, denoted $C(\Pi)$, is t.

(Max)

- The functions p_i can be *anything* so long as they are well-defined. E.g., could solve the Halting Problem.
- After the final message, *both parties* must know $f(x, y)$.

Definition (Communication Complexity)

Suppose Π is a protocol for f in which at most t bits are communicated between Alice and Bob. Then the *communication complexity* of Π, denoted $C(\Pi)$, is t.

Definition ($C(f)$)

The communication complexity of f, denoted $C(f)$, is the minimum communication complexity achieved by any protocol for f.

Parity (Jake)

Example (Are the number of 1s in xy even (0), or odd (1)?)

$f : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is precisely $(x, y) \mapsto \bigoplus xy$.

Parity (Jake)

Example (Are the number of 1s in xy even (0), or odd (1)?)

$f : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is precisely $(x, y) \mapsto \bigoplus xy$.

Example protocol Π:

$\mathtt{P1} = \text{parity}(x)$.

Parity (Jake)

Example (Are the number of 1s in xy even (0), or odd (1)?)

$f : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is precisely $(x, y) \mapsto \bigoplus xy$.

Example protocol Π:

$\text{P1} = \text{parity}(x)$.

$\text{P2} = \text{parity}(y) \bigoplus \text{P1}$

Parity (Jake)

Example (Are the number of 1s in xy even (0), or odd (1)?)

$f : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is precisely $(x, y) \mapsto \bigoplus xy$.

Example protocol Π:

$\mathtt{P1} = \text{parity}(x)$.

$\mathtt{P2} = \text{parity}(y) \oplus \mathtt{P1}$

Now both Alice and Bob know $f(x, y) = \mathtt{P2}$. $C(f) \leq 2$ because $C(\Pi) = 2$ and Π implements f. But $C(f) \geq 2$ because f depends on x and y. Hence $C(f) = 2$.

Halting (Jake)

Example (A_{TM})

$H : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is precisely $\langle M, x \rangle \mapsto 1$ if M halts on x else 0.

Halting (Jake)

Example (A_{TM})

$H : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is precisely $\langle M, x \rangle \mapsto 1$ if M halts on x else 0.

Example protocol Π:

P1 = y.

Halting (Jake)

Example (A_{TM})

$H : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is precisely $\langle M, x \rangle \mapsto 1$ if M halts on x else 0.

Example protocol Π:

P1 = y.

P2= (M does/doesn't accept y).

Halting (Jake)

Example (A_{TM})

$H : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is precisely $\langle M, x \rangle \mapsto 1$ if M halts on x else 0.

Example protocol Π:

$\mathtt{P1} = y.$

$\mathtt{P2}=$ (M does/doesn't accept y).

Both players have unlimited computation power. We are only interest in communication complexity.

Methods (Max)

If we find a protocol Π, then we know $C(f)$ is at most $C(\Pi)$.

Methods (Max)

If we find a protocol Π, then we know $C(f)$ is at most $C(\Pi)$. What if we don't know any protocol Π?

Methods (Max)

If we find a protocol Π, then we know $C(f)$ is at most $C(\Pi)$.
What if we don't know any protocol Π?

- Could we upper-bound $C(f)$ without knowing Π?

Methods (Max)

If we find a protocol Π, then we know $C(f)$ is at most $C(\Pi)$.
What if we don't know any protocol Π?

- Could we upper-bound $C(f)$ without knowing Π?

What if the only protocols we find seem really lousy?

Methods (Max)

If we find a protocol Π, then we know $C(f)$ is at most $C(\Pi)$. What if we don't know any protocol Π?

- Could we upper-bound $C(f)$ without knowing Π?

What if the only protocols we find seem really lousy?

- Could we lower-bound $C(f)$ without finding a better protocol?

Methods (Max)

If we find a protocol Π, then we know $C(f)$ is at most $C(\Pi)$.
What if we don't know any protocol Π?

- Could we upper-bound $C(f)$ without knowing Π?

What if the only protocols we find seem really lousy?

- Could we lower-bound $C(f)$ without finding a better protocol?

TL;DR: yup.

Fooling Set Method (Jake)

We begin with a motivating observation.

Fooling Set Method (Jake)

We begin with a motivating observation.

Lemma (Communication Equality is Image Equality)

If Alice and Bob exchange the same sequence of messages when Alice gets x *and Bob gets* y *as they do when Alice gets* x' *and Bob gets* y'*, then* $f(x, y) = f(x', y')$*.*

Fooling Set Method (Jake)

We begin with a motivating observation.

Lemma (Communication Equality is Image Equality)

If Alice and Bob exchange the same sequence of messages when Alice gets x *and Bob gets* y *as they do when Alice gets* x' *and Bob gets* y'*, then* $f(\mathrm{x}, \mathrm{y}) = f(\mathrm{x}', \mathrm{y}')$.

Proof.

Π is deterministic and f is a function. $\square$

Fooling Set Method (Jake)

We begin with a motivating observation.

Lemma (Communication Equality is Image Equality)

If Alice and Bob exchange the same sequence of messages when Alice gets x *and Bob gets* y *as they do when Alice gets* x' *and Bob gets* y'*, then* $f(\mathrm{x},\mathrm{y}) = f(\mathrm{x}',\mathrm{y}')$*.*

Proof.

Π is deterministic and f is a function. $\square$

Idea: an efficient protocol will efficiently group together inputs that go to the same output.

Fooling Set Method (Jake)

Idea: an efficient protocol will efficiently **group together inputs that go to the same output.**

Definition (Fooling Set)

If $f : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is a function, a *fooling set* for f is a set $S \subseteq \mathbb{B}^n \times \mathbb{B}^n$ such that for some choice $b \in \mathbb{B}$ $f(S) = \{b\}$ but, for all distinct $(x, y), (x', y') \in S$, $(\neg b) \in f(\{x, x'\} \times \{y, y'\})$.

Basically, a fooling set is a group of inputs that go to the same output, but which is *brittle* to argument-swapping. In some sense these *brittle* sets lower-bound the difficulty in grouping like inputs.

Fooling Set Method (Jake)

Idea: an efficient protocol will efficiently **group together inputs that go to the same output.**

Definition (Fooling Set)

If $f : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is a function, a *fooling set* for f is a set $S \subseteq \mathbb{B}^n \times \mathbb{B}^n$ such that for some choice $b \in \mathbb{B}$ $f(S) = \{b\}$ but, for all distinct $(x, y), (x', y') \in S$, $(\neg b) \in f(\{x, x'\} \times \{y, y'\})$.

Basically, a fooling set is a group of inputs that go to the same output, but which is *brittle* to argument-swapping. In some sense these *brittle* sets lower-bound the difficulty in grouping like inputs.

Lemma (Fooling Set Method)

If f has a size-M fooling set, then $C(f) \geq log_2(M)$.

Fooling Set Method (Jake)

Example (Set-Disjointness)

$\textsc{Disj} : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is the function that maps (A, B) to 1 if $A \cap B = \varnothing$ else 0.

How many fooling sets does $\textsc{Disj}$ have?

Fooling Set Method (Jake)

Example (Set-Disjointness)

$\textsc{Disj} : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is the function that maps (A, B) to 1 if $A \cap B = \varnothing$ else 0.

How many fooling sets does $\textsc{Disj}$ have? Notice A, B are disjoint iff $A \oplus B$ is $\mathbf{1}$.

Fooling Set Method (Jake)

Example (Set-Disjointness)

$\textsc{Disj} : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is the function that maps (A, B) to 1 if $A \cap B = \varnothing$ else 0.

How many fooling sets does $\textsc{Disj}$ have? Notice A, B are disjoint iff $A \oplus B$ is $\mathbf{1}$. There are 2^n possible values A.

Fooling Set Method (Jake)

Example (Set-Disjointness)

$\textsc{Disj} : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is the function that maps (A, B) to 1 if $A \cap B = \varnothing$ else 0.

How many fooling sets does $\textsc{Disj}$ have? Notice A, B are disjoint iff $A \oplus B$ is $\mathbf{1}$. There are 2^n possible values A. Hence 2^n values (A, B) s.t. $A \oplus B = \mathbf{1}$. None of these distinct $(A, B), (A', B')$ satisfy $A \oplus B = A \oplus B'$ or $A \oplus B = A' \oplus B$ else they wouldn't be distinct.

Fooling Set Method (Jake)

Example (Set-Disjointness)

$\textsc{Disj} : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is the function that maps (A, B) to 1 if $A \cap B = \varnothing$ else 0.

How many fooling sets does $\textsc{Disj}$ have? Notice A, B are disjoint iff $A \oplus B$ is $\mathbf{1}$. There are 2^n possible values A. Hence 2^n values (A, B) s.t. $A \oplus B = \mathbf{1}$. None of these distinct $(A, B), (A', B')$ satisfy $A \oplus B = A \oplus B'$ or $A \oplus B = A' \oplus B$ else they wouldn't be distinct. So we get a 2^n-size fooling set.

Fooling Set Method (Jake)

Example (Set-Disjointness)

$\textsc{Disj} : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ is the function that maps (A, B) to 1 if $A \cap B = \varnothing$ else 0.

How many fooling sets does $\textsc{Disj}$ have? Notice A, B are disjoint iff $A \oplus B$ is $\mathbf{1}$. There are 2^n possible values A. Hence 2^n values (A, B) s.t. $A \oplus B = \mathbf{1}$. None of these distinct $(A, B), (A', B')$ satisfy $A \oplus B = A \oplus B'$ or $A \oplus B = A' \oplus B$ else they wouldn't be distinct. So we get a 2^n-size fooling set.

$$\therefore C(\textsc{Disj}) \geq \log_2(2^n) = n$$

Fooling Set Method (Jake)

NTS: If f has a size-M fooling set then $C(f) \geq \log_2(M)$.

Proof.

For a contradiction suppose a protocol Π exists for f s.t. $C(\Pi) < \log_2(M)$.

Fooling Set Method (Jake)

NTS: If f has a size-M fooling set then $C(f) \geq \log_2(M)$.

Proof.

For a contradiction suppose a protocol Π exists for f s.t. $C(\Pi) < \log_2(M)$. Then Π yields at most $2^{C(\Pi)} < 2^{\log_2(M)} = M$ distinct communication patterns.

Fooling Set Method (Jake)

NTS: If f has a size-M fooling set then $C(f) \geq \log_2(M)$.

Proof.

For a contradiction suppose a protocol Π exists for f s.t. $C(\Pi) < \log_2(M)$. Then Π yields at most $2^{C(\Pi)} < 2^{\log_2(M)} = M$ distinct communication patterns. But there are $M * (M-1)$ disjoint choices of $(x, y), (x', y') \in S$.

Fooling Set Method (Jake)

NTS: If f has a size-M fooling set then $C(f) \geq \log_2(M)$.

Proof.

For a contradiction suppose a protocol Π exists for f s.t. $C(\Pi) < \log_2(M)$. Then Π yields at most $2^{C(\Pi)} < 2^{\log_2(M)} = M$ distinct communication patterns. But there are $M * (M-1)$ disjoint choices of $(x, y), (x', y') \in S$. Since $M * (M-1) > M$ there must be some $(x, y), (x', y')$ on which Π yields the same communication pattern.

Fooling Set Method (Jake)

NTS: If f has a size-M fooling set then $C(f) \geq \log_2(M)$.

Proof.

For a contradiction suppose a protocol Π exists for f s.t. $C(\Pi) < \log_2(M)$. Then Π yields at most $2^{C(\Pi)} < 2^{\log_2(M)} = M$ distinct communication patterns. But there are $M * (M-1)$ disjoint choices of $(x,y),(x',y') \in S$. Since $M*(M-1) > M$ there must be some $(x,y),(x',y')$ on which Π yields the same communication pattern.

Then (x,y') must yield the same communication pattern as (x,y) as Bob cannot possibly tell the difference. The argument is symmetric for (x',y) and Alice. One of the two must yield a contradiction and we are done. $\square$

Tiling Method (Max)

TODO

2-Party Discrepency Method (Max)

TODO

Multi-Party Problem (Jake)

TODO

Multi-Party Discrepency Method (Max)

TODO

Non-Deterministic (Jake)

TODO

Randomized (Max)

TODO