Data Streams & Communication Complexity

Lecture 3: Communication Complexity and Lower Bounds

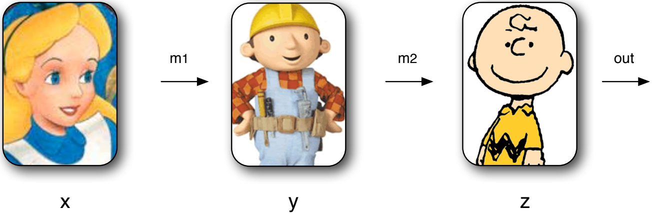
Andrew McGregor, UMass Amherst

Basic Communication Complexity

- Three friends Alice, Bob, and Charlie each have some information x, y, z and Charlie wants to compute some function $P(x, y, z)$.

Basic Communication Complexity

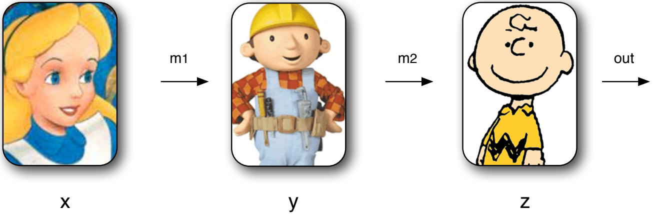
- Three friends Alice, Bob, and Charlie each have some information x, y, z and Charlie wants to compute some function $P(x, y, z)$.



- To help Charlie, Alice sends a message m_1 to Bob, and then Bob sends a message m_2 to Charlie.

Basic Communication Complexity

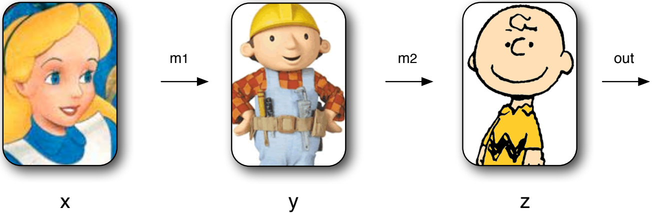
- Three friends Alice, Bob, and Charlie each have some information x, y, z and Charlie wants to compute some function $P(x, y, z)$.



- To help Charlie, Alice sends a message m_1 to Bob, and then Bob sends a message m_2 to Charlie.
- *Question:* How large must the total length of the messages be for Charlie to evaluate $P(x, y, z)$ correctly?

Basic Communication Complexity

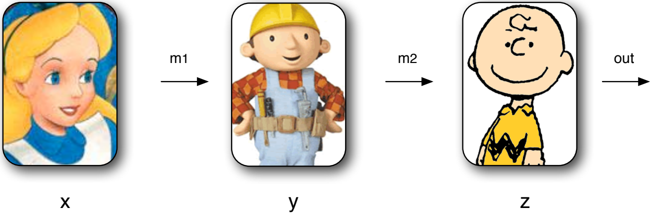
- Three friends Alice, Bob, and Charlie each have some information x, y, z and Charlie wants to compute some function $P(x, y, z)$.



- To help Charlie, Alice sends a message m_1 to Bob, and then Bob sends a message m_2 to Charlie.
- *Question:* How large must the total length of the messages be for Charlie to evaluate $P(x, y, z)$ correctly?
  - *Deterministic:* $m_1(x)$, $m_2(m_1, y)$, $\text{out}(m_2, z) = P(x, y, z)$

Basic Communication Complexity

- Three friends Alice, Bob, and Charlie each have some information x, y, z and Charlie wants to compute some function $P(x, y, z)$.

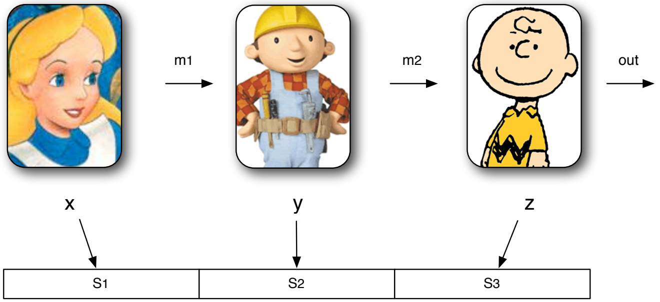


- To help Charlie, Alice sends a message m_1 to Bob, and then Bob sends a message m_2 to Charlie.
- *Question:* How large must the total length of the messages be for Charlie to evaluate $P(x, y, z)$ correctly?
  - *Deterministic:* $m_1(x)$, $m_2(m_1, y)$, $\text{out}(m_2, z) = P(x, y, z)$
  - *Random:* $m_1(x, r)$, $m_2(m_1, y, r)$, $\text{out}(m_2, z, r)$ where r is public random string. Require $\mathbb{P}_r[\text{out}(m_2, z, r) = P(x, y, z)] \geq 9/10$.

Stream Algorithms Yield Communication Protocols

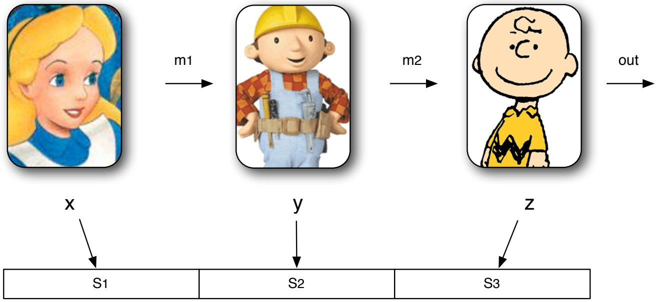
Stream Algorithms Yield Communication Protocols

- Let Q be some stream problem. Suppose there's a reduction $x \to S_1$, $y \to S_2$, $z \to S_3$ such that knowing $Q(S_1 \circ S_2 \circ S_3)$ solves $P(x, y, z)$.



Stream Algorithms Yield Communication Protocols

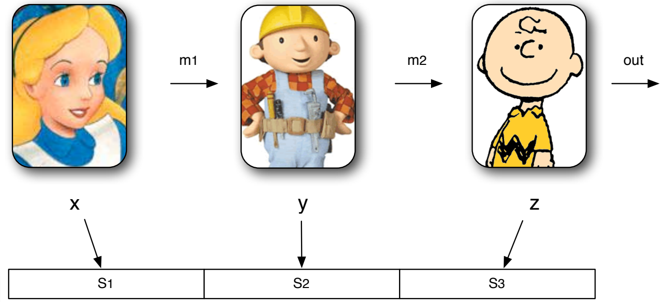
- Let Q be some stream problem. Suppose there's a reduction $x \to S_1$, $y \to S_2$, $z \to S_3$ such that knowing $Q(S_1 \circ S_2 \circ S_3)$ solves $P(x, y, z)$.



- An s-bit stream algorithm $\mathcal{A}$ for Q yields $2s$-bit protocol for P:

Stream Algorithms Yield Communication Protocols

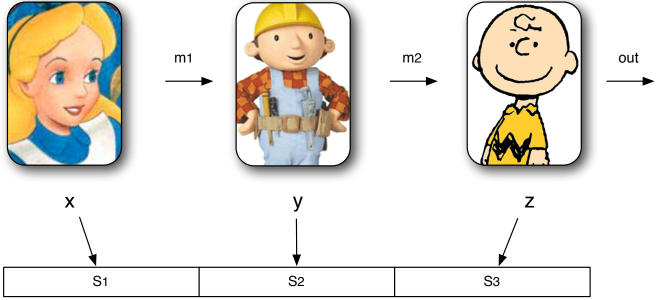
- Let Q be some stream problem. Suppose there's a reduction $x \to S_1$, $y \to S_2$, $z \to S_3$ such that knowing $Q(S_1 \circ S_2 \circ S_3)$ solves $P(x, y, z)$.



- An s-bit stream algorithm $\mathcal{A}$ for Q yields $2s$-bit protocol for P: Alice runs $\mathcal{A}$ of S_1;

Stream Algorithms Yield Communication Protocols

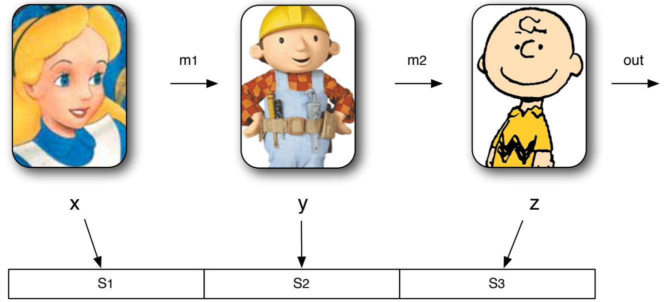
- Let Q be some stream problem. Suppose there's a reduction $x \to S_1$, $y \to S_2$, $z \to S_3$ such that knowing $Q(S_1 \circ S_2 \circ S_3)$ solves $P(x, y, z)$.



- An s-bit stream algorithm $\mathcal{A}$ for Q yields $2s$-bit protocol for P: Alice runs $\mathcal{A}$ of S_1; sends memory state to Bob;

Stream Algorithms Yield Communication Protocols

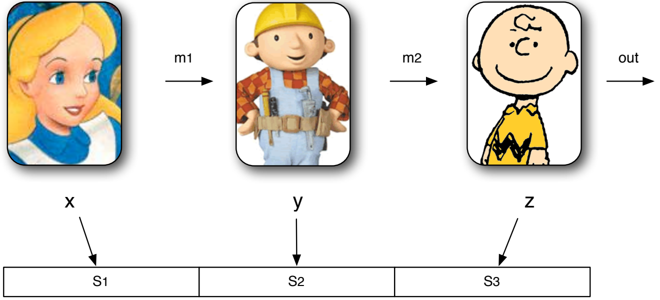
- Let Q be some stream problem. Suppose there's a reduction $x \to S_1$, $y \to S_2$, $z \to S_3$ such that knowing $Q(S_1 \circ S_2 \circ S_3)$ solves $P(x, y, z)$.



- An s-bit stream algorithm $\mathcal{A}$ for Q yields $2s$-bit protocol for P: Alice runs $\mathcal{A}$ of S_1; sends memory state to Bob; Bob instantiates $\mathcal{A}$ with state and runs it on S_2;

Stream Algorithms Yield Communication Protocols

- Let Q be some stream problem. Suppose there's a reduction $x \to S_1$, $y \to S_2$, $z \to S_3$ such that knowing $Q(S_1 \circ S_2 \circ S_3)$ solves $P(x, y, z)$.



- An s-bit stream algorithm $\mathcal{A}$ for Q yields $2s$-bit protocol for P: Alice runs $\mathcal{A}$ of S_1; sends memory state to Bob; Bob instantiates $\mathcal{A}$ with state and runs it on S_2; sends state to Charlie who finishes running $\mathcal{A}$ on S_3 and infers $P(x, y, z)$ from $Q(S_1 \circ S_2 \circ S_3)$.

Communication Lower Bounds imply Stream Lower Bounds

- Had there been t players, the s-bit stream algorithm for Q would have lead to a $(t-1)s$ bit protocol P.

Communication Lower Bounds imply Stream Lower Bounds

- Had there been t players, the s-bit stream algorithm for Q would have lead to a $(t-1)s$ bit protocol P.
- Hence, a lower bound of L on the communication required for P implies $s \geq L/(t-1)$ bits of space are required to solve Q.

Outline of Lecture

Classic Problems and Reductions

Information Statistics Approach

Hamming Approximation

Outline

Classic Problems and Reductions

Information Statistics Approach

Hamming Approximation

Indexing

- Consider a binary string $x \in \{0,1\}^n$ and $j \in [n]$, e.g.,

$$x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad j = 3$$

and define $\textsc{Index}(x, j) = x_j$

Indexing

- Consider a binary string $x \in \{0,1\}^n$ and $j \in [n]$, e.g.,

$$x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad j = 3$$

  and define $\textsc{Index}(x, j) = x_j$
- Suppose Alice knows x and Bob knows j.

Indexing

- Consider a binary string $x \in \{0,1\}^n$ and $j \in [n]$, e.g.,

$$x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad j = 3$$

  and define $\textsc{Index}(x, j) = x_j$
- Suppose Alice knows x and Bob knows j.
- How many bits need to be sent by Alice for Bob to determine $\textsc{Index}(x, j)$ with probability $9/10$?

Indexing

- Consider a binary string $x \in \{0,1\}^n$ and $j \in [n]$, e.g.,

$$x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad j = 3$$

  and define $\textsc{Index}(x, j) = x_j$
- Suppose Alice knows x and Bob knows j.
- How many bits need to be sent by Alice for Bob to determine $\textsc{Index}(x, j)$ with probability $9/10$? $\Omega(n)$

Application: Median Finding

- *Thm:* Any algorithm that returns the exact median of length $2n-1$ stream requires $\Omega(n)$ memory.

Application: Median Finding

- *Thm:* Any algorithm that returns the exact median of length $2n-1$ stream requires $\Omega(n)$ memory.
- *Reduction from Index:* On input $x \in \{0,1\}^n$, Alice generates $S_1 = \{2i + x_i : i \in [n]\}$. On input $j \in [n]$, Bob generates $S_2 = \{n-j \text{ copies of } 0 \text{ and } j-1 \text{ copies of } 2n+2\}$. E.g.,

$$\begin{array}{rcl} x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} & \to & \{2,5,6,9,11,12\} \\ j = 3 & \to & \{0,0,0,14,14\} \end{array}$$

Application: Median Finding

- *Thm:* Any algorithm that returns the exact median of length $2n-1$ stream requires $\Omega(n)$ memory.
- *Reduction from Index:* On input $x \in \{0,1\}^n$, Alice generates $S_1 = \{2i + x_i : i \in [n]\}$. On input $j \in [n]$, Bob generates $S_2 = \{n-j \text{ copies of } 0 \text{ and } j-1 \text{ copies of } 2n+2\}$. E.g.,

$$\begin{array}{rcl} x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} & \rightarrow & \{2,5,6,9,11,12\} \\ j = 3 & \rightarrow & \{0,0,0,14,14\} \end{array}$$

- Then $\text{median}(S_1 \cup S_2) = 2j + x_j$ and this determines $\textsc{Index}(x,j)$.

Application: Median Finding

- *Thm:* Any algorithm that returns the exact median of length $2n-1$ stream requires $\Omega(n)$ memory.
- *Reduction from Index:* On input $x \in \{0,1\}^n$, Alice generates $S_1 = \{2i + x_i : i \in [n]\}$. On input $j \in [n]$, Bob generates $S_2 = \{n-j$ copies of 0 and $j-1$ copies of $2n+2\}$. E.g.,

$$\begin{array}{rcl} x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} & \rightarrow & \{2,5,6,9,11,12\} \\ j = 3 & \rightarrow & \{0,0,0,14,14\} \end{array}$$

- Then $\text{median}(S_1 \cup S_2) = 2j + x_j$ and this determines $\textsc{Index}(x,j)$.
- An s-space algorithm implies an s-bit protocol so

$$s = \Omega(n)$$

by the communication complexity of indexing.

Multi-Party Set-Disjointness

- Consider a $t \times n$ matrix where column has weight $0, 1$, or t, e.g.,

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

  and let $\textsc{Disj}_t(C) = 1$ if there is an all 1's column and 0 otherwise.

Multi-Party Set-Disjointness

- Consider a $t \times n$ matrix where column has weight $0, 1$, or t, e.g.,

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

  and let $\textsc{Disj}_t(C) = 1$ if there is an all 1's column and 0 otherwise.
- Consider t players where P_i knows i-th row of C.

Multi-Party Set-Disjointness

- Consider a $t \times n$ matrix where column has weight $0, 1$, or t, e.g.,

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

  and let $\textsc{Disj}_t(C) = 1$ if there is an all 1's column and 0 otherwise.
- Consider t players where P_i knows i-th row of C.
- How many bits need to be communicated between the players to determine $\textsc{Disj}_t(C)$?

Multi-Party Set-Disjointness

- Consider a $t \times n$ matrix where column has weight $0, 1$, or t, e.g.,

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

  and let $\textsc{Disj}_t(C) = 1$ if there is an all 1's column and 0 otherwise.
- Consider t players where P_i knows i-th row of C.
- How many bits need to be communicated between the players to determine $\textsc{Disj}_t(C)$? $\Omega(n/t)$

Application: Frequency Moments

- *Thm:* A 2-approximation algorithm for F_k needs $\Omega(n^{1-2/k})$ space.

Application: Frequency Moments

- *Thm:* A 2-approximation algorithm for F_k needs $\Omega(n^{1-2/k})$ space.
- *Reduction from Set Disjointness:*

Application: Frequency Moments

- *Thm:* A 2-approximation algorithm for F_k needs $\Omega(n^{1-2/k})$ space.
- *Reduction from Set Disjointness:* The i-th player generates set $S_i = \{j : C_{ij} = 1\}$, e.g.,

$$\left(\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right) \longrightarrow \{4, 1, 4, 5, 2, 4, 4\}$$

Application: Frequency Moments

- *Thm:* A 2-approximation algorithm for F_k needs $\Omega(n^{1-2/k})$ space.
- *Reduction from Set Disjointness:* The i-th player generates set $S_i = \{j : C_{ij} = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \longrightarrow \{4, 1, 4, 5, 2, 4, 4\}$$

- If all columns have weight 0 or 1: $F_k(S) \leq n$

Application: Frequency Moments

- *Thm:* A 2-approximation algorithm for F_k needs $\Omega(n^{1-2/k})$ space.
- *Reduction from Set Disjointness:* The i-th player generates set $S_i = \{j : C_{ij} = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \longrightarrow \{4, 1, 4, 5, 2, 4, 4\}$$

- If all columns have weight 0 or 1: $F_k(S) \leq n$
- If there's column of weight t: $F_k(S) \geq t^k$

Application: Frequency Moments

- *Thm:* A 2-approximation algorithm for F_k needs $\Omega(n^{1-2/k})$ space.
- *Reduction from Set Disjointness:* The i-th player generates set $S_i = \{j : C_{ij} = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \longrightarrow \{4, 1, 4, 5, 2, 4, 4\}$$

- If all columns have weight 0 or 1: $F_k(S) \leq n$
- If there's column of weight t: $F_k(S) \geq t^k$
- If $t > 2^{1/k} n^{1/k}$ then a 2 approximation of $F_k(S)$ distinguishes cases.

Application: Frequency Moments

- *Thm:* A 2-approximation algorithm for F_k needs $\Omega(n^{1-2/k})$ space.
- *Reduction from Set Disjointness:* The i-th player generates set $S_i = \{j : C_{ij} = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \longrightarrow \{4, 1, 4, 5, 2, 4, 4\}$$

- If all columns have weight 0 or 1: $F_k(S) \leq n$
- If there's column of weight t: $F_k(S) \geq t^k$
- If $t > 2^{1/k} n^{1/k}$ then a 2 approximation of $F_k(S)$ distinguishes cases.
- An s-space 2-approximation implies an $s(t-1)$ bit protocol so

$$s = \Omega(n/t^2) = \Omega(n^{1-2/k})$$

by the communication complexity of set-disjointness.

Hamming Approximation

- Consider 2 binary vectors $x, y \in \{0,1\}^n$, e.g.,

$$x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$y = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

  and define the Hamming distance $\Delta(x,y) = |\{i : x_i \neq y_i\}|$.

Hamming Approximation

- Consider 2 binary vectors $x, y \in \{0,1\}^n$, e.g.,

$$x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$y = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

  and define the Hamming distance $\Delta(x, y) = |\{i : x_i \neq y_i\}|$.
- Suppose Alice knows x and Bob knows y.

Hamming Approximation

- Consider 2 binary vectors $x, y \in \{0,1\}^n$, e.g.,

$$x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$y = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

  and define the Hamming distance $\Delta(x,y) = |\{i : x_i \neq y_i\}|$.
- Suppose Alice knows x and Bob knows y.
- How many bits need to be communicated to estimate $\Delta(x,y)$ up to an additive $\sqrt{n}$ error?

Hamming Approximation

- Consider 2 binary vectors $x, y \in \{0,1\}^n$, e.g.,

$$x = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$y = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

  and define the Hamming distance $\Delta(x,y) = |\{i : x_i \neq y_i\}|$.
- Suppose Alice knows x and Bob knows y.
- How many bits need to be communicated to estimate $\Delta(x,y)$ up to an additive $\sqrt{n}$ error? $\Omega(n)$ bits.

Application: Distinct Elements

- *Thm:* A $(1+\epsilon)$-approximation algorithm for F_0 needs $\Omega(\epsilon^{-2})$ space.

Application: Distinct Elements

- *Thm:* A $(1+\epsilon)$-approximation algorithm for F_0 needs $\Omega(\epsilon^{-2})$ space.
- *Reduction from Hamming Approximation:* On input $x, y \in \{0,1\}^n$, players form $S_1 = \{j : x_j = 1\}$ and $S_2 = \{j : y_j = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \longrightarrow \{2,4,5,1,2,5,6\}$$

Application: Distinct Elements

- *Thm:* A $(1+\epsilon)$-approximation algorithm for F_0 needs $\Omega(\epsilon^{-2})$ space.
- *Reduction from Hamming Approximation:* On input $x, y \in \{0,1\}^n$, players form $S_1 = \{j : x_j = 1\}$ and $S_2 = \{j : y_j = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \longrightarrow \{2,4,5,1,2,5,6\}$$

- Note that $2F_0(S) = |x| + |y| + \Delta(x,y)$.

Application: Distinct Elements

- *Thm:* A $(1+\epsilon)$-approximation algorithm for F_0 needs $\Omega(\epsilon^{-2})$ space.
- *Reduction from Hamming Approximation:* On input $x, y \in \{0,1\}^n$, players form $S_1 = \{j : x_j = 1\}$ and $S_2 = \{j : y_j = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \longrightarrow \{2,4,5,1,2,5,6\}$$

- Note that $2F_0(S) = |x| + |y| + \Delta(x,y)$.
- We may assume $|x|$ and $|y|$ are known Bob. Hence, a $(1+\epsilon)$ approximation of F_0 yields an additive approximation to $\Delta(x,y)$ of

$$\epsilon(|x| + |y| + \Delta(x,y))/2 \leq n\epsilon$$

Application: Distinct Elements

- *Thm:* A $(1+\epsilon)$-approximation algorithm for F_0 needs $\Omega(\epsilon^{-2})$ space.
- *Reduction from Hamming Approximation:* On input $x, y \in \{0,1\}^n$, players form $S_1 = \{j : x_j = 1\}$ and $S_2 = \{j : y_j = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \longrightarrow \{2,4,5,1,2,5,6\}$$

- Note that $2F_0(S) = |x| + |y| + \Delta(x,y)$.
- We may assume $|x|$ and $|y|$ are known Bob. Hence, a $(1+\epsilon)$ approximation of F_0 yields an additive approximation to $\Delta(x,y)$ of

$$\epsilon(|x| + |y| + \Delta(x,y))/2 \leq n\epsilon$$

- This is less than $\sqrt{n}$ if $\epsilon < 1/\sqrt{n}$

Application: Distinct Elements

- *Thm:* A $(1+\epsilon)$-approximation algorithm for F_0 needs $\Omega(\epsilon^{-2})$ space.
- *Reduction from Hamming Approximation:* On input $x, y \in \{0,1\}^n$, players form $S_1 = \{j : x_j = 1\}$ and $S_2 = \{j : y_j = 1\}$, e.g.,

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \longrightarrow \{2,4,5,1,2,5,6\}$$

- Note that $2F_0(S) = |x| + |y| + \Delta(x,y)$.
- We may assume $|x|$ and $|y|$ are known Bob. Hence, a $(1+\epsilon)$ approximation of F_0 yields an additive approximation to $\Delta(x,y)$ of

$$\epsilon(|x| + |y| + \Delta(x,y))/2 \leq n\epsilon$$

- This is less than $\sqrt{n}$ if $\epsilon < 1/\sqrt{n}$
- An s-space $(1+\epsilon)$-approximation implies an s bit protocol so

$$s = \Omega(n) = \Omega(1/\epsilon^2)$$

by communication complexity of approximating Hamming distance.

Outline

Classic Problems and Reductions

Information Statistics Approach

Hamming Approximation

Information Statistics Approach

- Information statistics approach is based on analyzing the "information revealed" about the input from the messages.

Information Statistics Approach

- Information statistics approach is based on analyzing the "information revealed" about the input from the messages.
- Useful for proving bounds on complicated functions in terms of simpler problems, e.g., proving a bound on

$$\textsc{Disj}_t(M) = \bigvee_{j \in [n]} \textsc{And}_t(M_{1,j}, \ldots, M_{t,j})$$

by first establishing a bound on $\textsc{And}_t$.

Information Statistics Approach

- Information statistics approach is based on analyzing the "information revealed" about the input from the messages.
- Useful for proving bounds on complicated functions in terms of simpler problems, e.g., proving a bound on

$$\textsc{Disj}_t(M) = \bigvee_{j \in [n]} \textsc{And}_t(M_{1,j}, \ldots, M_{t,j})$$

  by first establishing a bound on $\textsc{And}_t$.
- We'll first give some definitions and then run through an example.

Information Theory Definitions

- Let X and Y be random variables.

Information Theory Definitions

- Let X and Y be random variables.
- *Entropy:* $H(X) := \sum_i -\mathbb{P}[X = i] \lg \mathbb{P}[X = i]$

Information Theory Definitions

- Let X and Y be random variables.
- *Entropy:* $H(X) := \sum_i -\mathbb{P}[X = i] \lg \mathbb{P}[X = i]$
- *Conditional Entropy:* $H(X|Y) := \mathbb{E}_{y \sim Y}[H(X|Y = y)]$

Information Theory Definitions

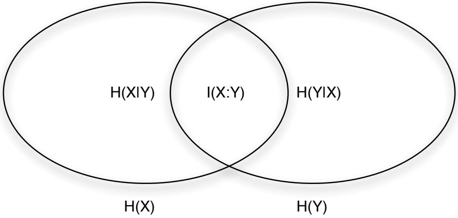
- Let X and Y be random variables.
- *Entropy:* $H(X) := \sum_i -\mathbb{P}[X = i] \lg \mathbb{P}[X = i]$
- *Conditional Entropy:* $H(X|Y) := \mathbb{E}_{y \sim Y}[H(X|Y = y)] \leq H(X)$

Information Theory Definitions

- Let X and Y be random variables.
- *Entropy:* $H(X) := \sum_i -\mathbb{P}[X = i] \lg \mathbb{P}[X = i]$
- *Conditional Entropy:* $H(X|Y) := \mathbb{E}_{y \sim Y}[H(X|Y = y)] \leq H(X)$
- *Mutual Information:* $I(X : Y) = H(X) - H(X|Y)$

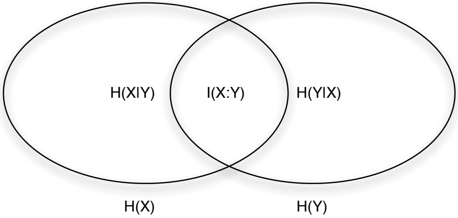
Information Theory Definitions

- Let X and Y be random variables.
- *Entropy:* $H(X) := \sum_i -\mathbb{P}[X=i] \lg \mathbb{P}[X=i]$
- *Conditional Entropy:* $H(X|Y) := \mathbb{E}_{y \sim Y}[H(X|Y=y)] \leq H(X)$
- *Mutual Information:* $I(X:Y) = H(X) - H(X|Y)$



Information Theory Definitions

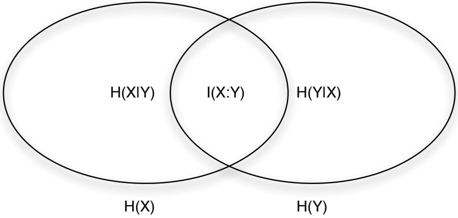
- Let X and Y be random variables.
- *Entropy:* $H(X) := \sum_i -\mathbb{P}[X = i] \lg \mathbb{P}[X = i]$
- *Conditional Entropy:* $H(X|Y) := \mathbb{E}_{y \sim Y}[H(X|Y = y)] \leq H(X)$
- *Mutual Information:* $I(X : Y) = H(X) - H(X|Y)$



- Useful Facts:

Information Theory Definitions

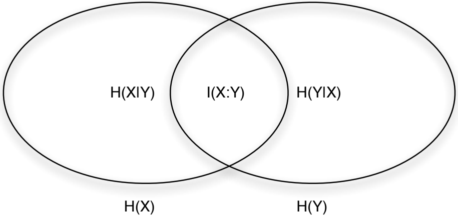
- Let X and Y be random variables.
- *Entropy:* $H(X) := \sum_i -\mathbb{P}[X = i] \lg \mathbb{P}[X = i]$
- *Conditional Entropy:* $H(X|Y) := \mathbb{E}_{y \sim Y}[H(X|Y = y)] \leq H(X)$
- *Mutual Information:* $I(X : Y) = H(X) - H(X|Y)$



- Useful Facts:
  - If X takes at most 2^ℓ values, then $H(X) \leq \ell$.

Information Theory Definitions

- Let X and Y be random variables.
- *Entropy:* $H(X) := \sum_i -\mathbb{P}[X = i] \lg \mathbb{P}[X = i]$
- *Conditional Entropy:* $H(X|Y) := \mathbb{E}_{y \sim Y}[H(X|Y = y)] \leq H(X)$
- *Mutual Information:* $I(X : Y) = H(X) - H(X|Y)$



- Useful Facts:
  - If X takes at most 2^ℓ values, then $H(X) \leq \ell$.
  - If X and Y are independent, then $I(XY : Z) \geq I(X : Z) + I(Y : Z)$.

Information Cost

- Suppose you have a protocol Π for a two-party communication problem P in which Alice and Bob have random inputs X and Y.

Information Cost

- Suppose you have a protocol Π for a two-party communication problem P in which Alice and Bob have random inputs X and Y.
- Let M be the (random) message sent by Alice and define:

$$\mathrm{cost}(\Pi) = \max |M|$$

and

$$\mathrm{icost}(\Pi) = I(M : X)$$

Information Cost

- Suppose you have a protocol Π for a two-party communication problem P in which Alice and Bob have random inputs X and Y.
- Let M be the (random) message sent by Alice and define:

$$\text{cost}(\Pi) = \max |M|$$

and

$$\text{icost}(\Pi) = I(M : X)$$

- Note that

$$\text{icost}(\Pi) = I(M : X) \leq H(M) \leq \text{cost}(\Pi) \ .$$

Example: Indexing

- We'll prove a lower bound on the information cost of $\textsc{Index}$ where $X \in_R \{0,1\}^n$ in terms a simpler problem "$\textsc{Echo}$."

Example: Indexing

- We'll prove a lower bound on the information cost of INDEX where $X \in_R \{0,1\}^n$ in terms a simpler problem "ECHO."
- ECHO: Alice has a single bit $B \in_R \{0,1\}$ and Bob wants to output B with probability at least $1 - \delta$.

Example: Indexing

- We'll prove a lower bound on the information cost of $\textsc{Index}$ where $X \in_R \{0,1\}^n$ in terms a simpler problem "$\textsc{Echo}$."
- $\textsc{Echo}$: Alice has a single bit $B \in_R \{0,1\}$ and Bob wants to output B with probability at least $1-\delta$.
- A protocol $\Pi_{\textsc{Index}}$ for $\textsc{Index}$ yields a protocol $\Pi_{\textsc{Echo},i}$ for $\textsc{Echo}$ where i is hard-coded into the protocol:

Example: Indexing

- We'll prove a lower bound on the information cost of $\textsc{Index}$ where $X \in_R \{0,1\}^n$ in terms a simpler problem "$\textsc{Echo}$."
- $\textsc{Echo}$: Alice has a single bit $B \in_R \{0,1\}$ and Bob wants to output B with probability at least $1-\delta$.
- A protocol $\Pi_{\textsc{Index}}$ for $\textsc{Index}$ yields a protocol $\Pi_{\textsc{Echo},i}$ for $\textsc{Echo}$ where i is hard-coded into the protocol:
  1. Given B, Alice picks $X_j \in_R \{0,1\}$ for $j \neq i$ and generates:

$$X = (X_1, X_2, \ldots, X_{i-1}, B, X_{i+1}, \ldots, X_n)$$

Example: Indexing

- We'll prove a lower bound on the information cost of $\textsc{Index}$ where $X \in_R \{0,1\}^n$ in terms a simpler problem "$\textsc{Echo}$."
- $\textsc{Echo}$: Alice has a single bit $B \in_R \{0,1\}$ and Bob wants to output B with probability at least $1-\delta$.
- A protocol $\Pi_{\textsc{Index}}$ for $\textsc{Index}$ yields a protocol $\Pi_{\textsc{Echo},i}$ for $\textsc{Echo}$ where i is hard-coded into the protocol:
  1. Given B, Alice picks $X_j \in_R \{0,1\}$ for $j \neq i$ and generates:
  $$X = (X_1, X_2, \ldots, X_{i-1}, B, X_{i+1}, \ldots, X_n)$$
  2. She sends the message M she'd have sent in $\Pi_{\textsc{Index}}$ if she'd had X.

Example: Indexing

- We'll prove a lower bound on the information cost of $\textsc{Index}$ where $X \in_R \{0,1\}^n$ in terms a simpler problem "$\textsc{Echo}$."
- $\textsc{Echo}$: Alice has a single bit $B \in_R \{0,1\}$ and Bob wants to output B with probability at least $1-\delta$.
- A protocol $\Pi_{\textsc{Index}}$ for $\textsc{Index}$ yields a protocol $\Pi_{\textsc{Echo},i}$ for $\textsc{Echo}$ where i is hard-coded into the protocol:
  1. Given B, Alice picks $X_j \in_R \{0,1\}$ for $j \neq i$ and generates:
  $$X = (X_1, X_2, \ldots, X_{i-1}, B, X_{i+1}, \ldots, X_n)$$
  2. She sends the message M she'd have sent in $\Pi_{\textsc{Index}}$ if she'd had X.
  3. Bob receives M and outputs the value he'd have returned in $\Pi_{\textsc{Index}}$ had his input been i.

Example: Indexing

- We'll prove a lower bound on the information cost of $\textsc{Index}$ where $X \in_R \{0,1\}^n$ in terms a simpler problem "$\textsc{Echo}$."
- $\textsc{Echo}$: Alice has a single bit $B \in_R \{0,1\}$ and Bob wants to output B with probability at least $1-\delta$.
- A protocol $\Pi_{\textsc{Index}}$ for $\textsc{Index}$ yields a protocol $\Pi_{\textsc{Echo},i}$ for $\textsc{Echo}$ where i is hard-coded into the protocol:
  1. Given B, Alice picks $X_j \in_R \{0,1\}$ for $j \neq i$ and generates:
  $$X = (X_1, X_2, \ldots, X_{i-1}, B, X_{i+1}, \ldots, X_n)$$
  2. She sends the message M she'd have sent in $\Pi_{\textsc{Index}}$ if she'd had X.
  3. Bob receives M and outputs the value he'd have returned in $\Pi_{\textsc{Index}}$ had his input been i.
- *Note 1:* If $\Pi_{\textsc{Index}}$ is correct with probability $1-\delta$ then $\Pi_{\textsc{Echo},i}$ is also correct with probability $1-\delta$.

Example: Indexing

- We'll prove a lower bound on the information cost of $\textsc{Index}$ where $X \in_R \{0,1\}^n$ in terms a simpler problem "$\textsc{Echo}$."
- $\textsc{Echo}$: Alice has a single bit $B \in_R \{0,1\}$ and Bob wants to output B with probability at least $1-\delta$.
- A protocol $\Pi_{\textsc{Index}}$ for $\textsc{Index}$ yields a protocol $\Pi_{\textsc{Echo},i}$ for $\textsc{Echo}$ where i is hard-coded into the protocol:
  1. Given B, Alice picks $X_j \in_R \{0,1\}$ for $j \neq i$ and generates:
  $$X = (X_1, X_2, \ldots, X_{i-1}, B, X_{i+1}, \ldots, X_n)$$
  2. She sends the message M she'd have sent in $\Pi_{\textsc{Index}}$ if she'd had X.
  3. Bob receives M and outputs the value he'd have returned in $\Pi_{\textsc{Index}}$ had his input been i.
- *Note 1:* If $\Pi_{\textsc{Index}}$ is correct with probability $1-\delta$ then $\Pi_{\textsc{Echo},i}$ is also correct with probability $1-\delta$.
- *Note 2:* The message in $\Pi_{\textsc{Index}}$ on input $X \in_R \{0,1\}^n$ is distributed identically to the message in $\Pi_{\textsc{Echo},i}$ on input $B \in_R \{0,1\}$.

Relating Information Cost of INDEX and ECHO

- Since $X_1, X_2, \ldots, X_n$ are independent:

$$\mathrm{cost}(\Pi_{\mathrm{INDEX}}) \quad \geq \quad \mathrm{icost}(\Pi_{\mathrm{INDEX}})$$

Relating Information Cost of $\textsc{Index}$ and $\textsc{Echo}$

- Since $X_1, X_2, \ldots, X_n$ are independent:

$$\begin{aligned}\mathrm{cost}(\Pi_{\textsc{Index}}) &\geq \mathrm{icost}(\Pi_{\textsc{Index}}) \\ &= I(X_1 X_2 \ldots X_n : M)\end{aligned}$$

Relating Information Cost of Index and Echo

- Since $X_1, X_2, \ldots, X_n$ are independent:

$$
\begin{aligned}
\mathrm{cost}(\Pi_{\textsc{Index}}) &\geq \mathrm{icost}(\Pi_{\textsc{Index}}) \\
&= I(X_1 X_2 \ldots X_n : M) \\
&\geq I(X_1 : M) + I(X_2 : M) + \ldots + I(X_n : M)
\end{aligned}
$$

Relating Information Cost of Index and Echo

- Since $X_1, X_2, \ldots, X_n$ are independent:

$$\begin{aligned}
\text{cost}(\Pi_{\text{INDEX}}) &\geq \text{icost}(\Pi_{\text{INDEX}}) \\
&= I(X_1 X_2 \ldots X_n : M) \\
&\geq I(X_1 : M) + I(X_2 : M) + \ldots + I(X_n : M) \\
&= \text{icost}(\Pi_{\text{ECHO},1}) + \text{icost}(\Pi_{\text{ECHO},2}) + \ldots + \text{icost}(\Pi_{\text{ECHO},n})
\end{aligned}$$

Relating Information Cost of $\textsc{Index}$ and $\textsc{Echo}$

- Since $X_1, X_2, \ldots, X_n$ are independent:

$$\begin{aligned}
\mathrm{cost}(\Pi_{\textsc{Index}}) &\geq \mathrm{icost}(\Pi_{\textsc{Index}}) \\
&= I(X_1 X_2 \ldots X_n : M) \\
&\geq I(X_1 : M) + I(X_2 : M) + \ldots + I(X_n : M) \\
&= \mathrm{icost}(\Pi_{\textsc{Echo},1}) + \mathrm{icost}(\Pi_{\textsc{Echo},2}) + \ldots + \mathrm{icost}(\Pi_{\textsc{Echo},n})
\end{aligned}$$

- By Fano's inequality, solving $\textsc{Echo}$ with probability $> 1 - \delta$ requires

$$\mathrm{icost}(\Pi_{\textsc{Echo},i}) = H(B) - H(B|M) \geq 1 - H_2(\delta)$$

where $H_2(p) = -p \lg p - (1-p)\lg(1-p)$.

Relating Information Cost of Index and Echo

- Since $X_1, X_2, \ldots, X_n$ are independent:

$$\begin{aligned}
\mathrm{cost}(\Pi_{\mathrm{INDEX}}) &\geq \mathrm{icost}(\Pi_{\mathrm{INDEX}}) \\
&= I(X_1 X_2 \ldots X_n : M) \\
&\geq I(X_1 : M) + I(X_2 : M) + \ldots + I(X_n : M) \\
&= \mathrm{icost}(\Pi_{\mathrm{ECHO},1}) + \mathrm{icost}(\Pi_{\mathrm{ECHO},2}) + \ldots + \mathrm{icost}(\Pi_{\mathrm{ECHO},n})
\end{aligned}$$

- By Fano's inequality, solving Echo with probability $> 1 - \delta$ requires

$$\mathrm{icost}(\Pi_{\mathrm{ECHO},i}) = H(B) - H(B|M) \geq 1 - H_2(\delta)$$

where $H_2(p) = -p \lg p - (1-p)\lg(1-p)$.

- Hence, $\mathrm{cost}(\Pi_{\mathrm{INDEX}}) \geq (1 - H_2(\delta))n$.

Outline for $\textsc{Disj}_t$ Lower Bound

- Express $\textsc{Disj}_t$ in terms of $\textsc{And}_t$ where $\textsc{And}_t(x_1, \ldots, x_t) = \prod_i x_i$:

$$\textsc{Disj}_t(C) = \bigvee_{j \in [n]} \textsc{And}_t(C_{1,j}, \ldots, C_{t,j})$$

Outline for $\textsc{Disj}_t$ Lower Bound

- Express $\textsc{Disj}_t$ in terms of $\textsc{And}_t$ where $\textsc{And}_t(x_1, \ldots, x_t) = \prod_i x_i$:

$$\textsc{Disj}_t(C) = \bigvee_{j \in [n]} \textsc{And}_t(C_{1,j}, \ldots, C_{t,j})$$

- Define input C by $C_{D_j j} \in_R \{0, 1\}$ for $D_j \in_R [t]$. All other entries 0.

Outline for $\textsc{Disj}_t$ Lower Bound

- Express $\textsc{Disj}_t$ in terms of $\textsc{And}_t$ where $\textsc{And}_t(x_1, \ldots, x_t) = \prod_i x_i$:

$$\textsc{Disj}_t(C) = \bigvee_{j \in [n]} \textsc{And}_t(C_{1,j}, \ldots, C_{t,j})$$

- Define input C by $C_{D_j j} \in_R \{0,1\}$ for $D_j \in_R [t]$. All other entries 0.
- Let $M = (M_1, \ldots, M_{t-1})$ be the messages sent in a t-party protocol and define the information cost of a protocol as:

$$\text{icost}(\Pi|D) = I(C : M|D) \quad \text{where} \quad D = (D_1, \ldots, D_n) \ .$$

Outline for $\textsc{Disj}_t$ Lower Bound

- Express $\textsc{Disj}_t$ in terms of $\textsc{And}_t$ where $\textsc{And}_t(x_1, \ldots, x_t) = \prod_i x_i$:

$$\textsc{Disj}_t(C) = \bigvee_{j \in [n]} \textsc{And}_t(C_{1,j}, \ldots, C_{t,j})$$

- Define input C by $C_{D_j j} \in_R \{0,1\}$ for $D_j \in_R [t]$. All other entries 0.
- Let $M = (M_1, \ldots, M_{t-1})$ be the messages sent in a t-party protocol and define the information cost of a protocol as:

$$\text{icost}(\Pi|D) = I(C : M|D) \quad \text{where} \quad D = (D_1, \ldots, D_n) \ .$$

- A protocol for $\textsc{Disj}_t$ yields n different protocols $\Pi_{\textsc{And}_t, i}$ for $\textsc{And}_t$:

$$\text{icost}(\Pi_{\textsc{Disj}_t}|D) \geq \sum_{i \in [n]} \text{icost}(\Pi_{\textsc{And}_t, i}|D) \ .$$

Outline for $\textsc{Disj}_t$ Lower Bound

- Express $\textsc{Disj}_t$ in terms of $\textsc{And}_t$ where $\textsc{And}_t(x_1, \ldots, x_t) = \prod_i x_i$:

$$\textsc{Disj}_t(C) = \bigvee_{j \in [n]} \textsc{And}_t(C_{1,j}, \ldots, C_{t,j})$$

- Define input C by $C_{D_j j} \in_R \{0, 1\}$ for $D_j \in_R [t]$. All other entries 0.
- Let $M = (M_1, \ldots, M_{t-1})$ be the messages sent in a t-party protocol and define the information cost of a protocol as:

$$\text{icost}(\Pi | D) = I(C : M | D) \quad \text{where} \quad D = (D_1, \ldots, D_n) \ .$$

- A protocol for $\textsc{Disj}_t$ yields n different protocols $\Pi_{\textsc{And}_t, i}$ for $\textsc{And}_t$:

$$\text{icost}(\Pi_{\textsc{Disj}_t} | D) \geq \sum_{i \in [n]} \text{icost}(\Pi_{\textsc{And}_t, i} | D) \ .$$

- Result follows by showing $\text{icost}(\Pi_{\textsc{And}_t, i} | D) = \Omega(1/t)$.

Outline

Classic Problems and Reductions

Information Statistics Approach

Hamming Approximation

Hamming Approximation Lower Bound

Some communication results can be proved via a reduction from other communication results.

Theorem

If Alice and Bob have $x, y \in \{0,1\}^n$ and Bob wants to determine $\Delta(x, y)$ up to $\pm\sqrt{n}$ with probability $9/10$, then Alice must send $\Omega(n)$ bits.

Hamming Approximation Lower Bound

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.

Hamming Approximation Lower Bound

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^t$ using public random bits.

Hamming Approximation Lower Bound

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^t$ using public random bits.
- Alice computes $\text{sign}(r.z)$ and Bob computes $\text{sign}(r_j)$

Hamming Approximation Lower Bound

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^t$ using public random bits.
- Alice computes $\text{sign}(r.z)$ and Bob computes $\text{sign}(r_j)$
- *Lemma:* For some constant $c > 0$,

$$\mathbb{P}\left[\text{sign}(r.z) = \text{sign}(r_j)\right] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

Hamming Approximation Lower Bound

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^t$ using public random bits.
- Alice computes $\text{sign}(r.z)$ and Bob computes $\text{sign}(r_j)$
- *Lemma:* For some constant $c > 0$,

$$\mathbb{P}\left[\text{sign}(r.z) = \text{sign}(r_j)\right] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- Repeat $n = 25t/c^2$ times to construct

$$x_i = I[\text{sign}(r.z) = +] \quad \text{and} \quad y_i = I[\text{sign}(r_j) = +]$$

Hamming Approximation Lower Bound

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^t$ using public random bits.
- Alice computes $\text{sign}(r.z)$ and Bob computes $\text{sign}(r_j)$
- *Lemma:* For some constant $c > 0$,

$$\mathbb{P}[\text{sign}(r.z) = \text{sign}(r_j)] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- Repeat $n = 25t/c^2$ times to construct

$$x_i = I[\text{sign}(r.z) = +] \quad \text{and} \quad y_i = I[\text{sign}(r_j) = +]$$

- Note that

$$z_j = 0 \Rightarrow \mathbb{E}[\Delta(x,y)] = n/2$$
$$z_j = 1 \Rightarrow \mathbb{E}[\Delta(x,y)] = n/2 - 5\sqrt{n}$$

and by Chernoff bounds $\mathbb{P}\left[|\Delta(x,y) - \mathbb{E}[\Delta(x,y)]| \geq 2\sqrt{n}\right] < 1/10$.

Hamming Approximation Lower Bound

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^t$ using public random bits.
- Alice computes $\text{sign}(r.z)$ and Bob computes $\text{sign}(r_j)$
- *Lemma:* For some constant $c > 0$,

$$\mathbb{P}\left[\text{sign}(r.z) = \text{sign}(r_j)\right] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- Repeat $n = 25t/c^2$ times to construct

$$x_i = I[\text{sign}(r.z) = +] \quad \text{and} \quad y_i = I[\text{sign}(r_j) = +]$$

- Note that

$$z_j = 0 \Rightarrow \mathbb{E}\left[\Delta(x,y)\right] = n/2$$
$$z_j = 1 \Rightarrow \mathbb{E}\left[\Delta(x,y)\right] = n/2 - 5\sqrt{n}$$

  and by Chernoff bounds $\mathbb{P}\left[|\Delta(x,y) - \mathbb{E}\left[\Delta(x,y)\right]| \geq 2\sqrt{n}\right] < 1/10$.
- Hence, a $\pm\sqrt{n}$ approx. of $\Delta(x,y)$ determines z_j with prob. $> 9/10$.

Proof of Lemma

Claim

Let A be the event $A = \{\text{sign}(r.z) = r_j\}$. For some constant $c > 0$,

$$\mathbb{P}[A] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

Proof of Lemma

Claim

Let A be the event $A = \{\text{sign}(r.z) = r_j\}$. For some constant $c > 0$,

$$\mathbb{P}[A] = \begin{cases} 1/2 & \textit{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \textit{if } z_j = 1 \end{cases}$$

- If $z_j = 0$: sign$(r.z)$ and r_j are independent so $\mathbb{P}[A] = 1/2$.

Proof of Lemma

Claim

Let A be the event $A = \{\text{sign}(r.z) = r_j\}$. For some constant $c > 0$,

$$\mathbb{P}[A] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- If $z_j = 0$: $\text{sign}(r.z)$ and r_j are independent so $\mathbb{P}[A] = 1/2$.
- If $z_j = 1$: Let $s = r.z - r_j$, the sum of an even number ($\ell = t/2 - 1$) of independent ± 1 values.

Proof of Lemma

Claim

Let A be the event $A = \{\text{sign}(r.z) = r_j\}$. For some constant $c > 0$,

$$\mathbb{P}[A] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- If $z_j = 0$: $\text{sign}(r.z)$ and r_j are independent so $\mathbb{P}[A] = 1/2$.
- If $z_j = 1$: Let $s = r.z - r_j$, the sum of an even number ($\ell = t/2 - 1$) of independent ± 1 values. Then,

$$\mathbb{P}[A] = \mathbb{P}[A|s = 0]\,\mathbb{P}[s = 0] + \mathbb{P}[A|s \neq 0]\,\mathbb{P}[s \neq 0]$$

Proof of Lemma

Claim

Let A be the event $A = \{\text{sign}(r.z) = r_j\}$. For some constant $c > 0$,

$$\mathbb{P}[A] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- If $z_j = 0$: $\text{sign}(r.z)$ and r_j are independent so $\mathbb{P}[A] = 1/2$.
- If $z_j = 1$: Let $s = r.z - r_j$, the sum of an even number ($\ell = t/2 - 1$) of independent ± 1 values. Then,

$$\mathbb{P}[A] = \mathbb{P}[A|s = 0]\,\mathbb{P}[s = 0] + \mathbb{P}[A|s \neq 0]\,\mathbb{P}[s \neq 0]$$

  - $\mathbb{P}[s = 0] = \binom{\ell}{\ell/2}/2^\ell = 2c/\sqrt{t}$ for some constant $c > 0$.

Proof of Lemma

Claim

Let A be the event $A = \{\text{sign}(r.z) = r_j\}$. For some constant $c > 0$,

$$\mathbb{P}[A] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- If $z_j = 0$: $\text{sign}(r.z)$ and r_j are independent so $\mathbb{P}[A] = 1/2$.
- If $z_j = 1$: Let $s = r.z - r_j$, the sum of an even number ($\ell = t/2 - 1$) of independent ± 1 values. Then,

$$\mathbb{P}[A] = \mathbb{P}[A|s=0]\,\mathbb{P}[s=0] + \mathbb{P}[A|s \neq 0]\,\mathbb{P}[s \neq 0]$$

  - $\mathbb{P}[s=0] = \binom{\ell}{\ell/2}/2^{\ell} = 2c/\sqrt{t}$ for some constant $c > 0$.
  - $\mathbb{P}[A|s=0] = 1$ since $s = 0 \Rightarrow r.z = r_j \Rightarrow A$.

Proof of Lemma

Claim

Let A be the event $A = \{\text{sign}(r.z) = r_j\}$. For some constant $c > 0$,

$$\mathbb{P}[A] = \begin{cases} 1/2 & \textit{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \textit{if } z_j = 1 \end{cases}$$

- If $z_j = 0$: $\text{sign}(r.z)$ and r_j are independent so $\mathbb{P}[A] = 1/2$.
- If $z_j = 1$: Let $s = r.z - r_j$, the sum of an even number ($\ell = t/2 - 1$) of independent ± 1 values. Then,

$$\mathbb{P}[A] = \mathbb{P}[A|s = 0]\,\mathbb{P}[s = 0] + \mathbb{P}[A|s \neq 0]\,\mathbb{P}[s \neq 0]$$

  - $\mathbb{P}[s = 0] = \binom{\ell}{\ell/2}/2^{\ell} = 2c/\sqrt{t}$ for some constant $c > 0$.
  - $\mathbb{P}[A|s = 0] = 1$ since $s = 0 \Rightarrow r.z = r_j \Rightarrow A$.
  - $\mathbb{P}[A|s \neq 0] = 1/2$ since $s \neq 0 \Rightarrow s = \{\ldots, -4, -2, 2, 4, \ldots\}$. Hence, $\text{sign}(r.z) = \text{sign}(s)$ which is independent of r_j.

Proof of Lemma

Claim

Let A be the event $A = \{\text{sign}(r.z) = r_j\}$. For some constant $c > 0$,

$$\mathbb{P}[A] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- If $z_j = 0$: $\text{sign}(r.z)$ and r_j are independent so $\mathbb{P}[A] = 1/2$.
- If $z_j = 1$: Let $s = r.z - r_j$, the sum of an even number ($\ell = t/2 - 1$) of independent ± 1 values. Then,

$$\mathbb{P}[A] = \mathbb{P}[A|s = 0]\,\mathbb{P}[s = 0] + \mathbb{P}[A|s \neq 0]\,\mathbb{P}[s \neq 0]$$

  - $\mathbb{P}[s = 0] = \binom{\ell}{\ell/2}/2^\ell = 2c/\sqrt{t}$ for some constant $c > 0$.
  - $\mathbb{P}[A|s = 0] = 1$ since $s = 0 \Rightarrow r.z = r_j \Rightarrow A$.
  - $\mathbb{P}[A|s \neq 0] = 1/2$ since $s \neq 0 \Rightarrow s = \{\ldots, -4, -2, 2, 4, \ldots\}$. Hence, $\text{sign}(r.z) = \text{sign}(s)$ which is independent of r_j.
- So $\mathbb{P}[A] = \mathbb{P}[s = 0] + \frac{\mathbb{P}[s \neq 0]}{2} = \frac{1}{2} + \frac{\mathbb{P}[s = 0]}{2} = \frac{1}{2} + \frac{c}{\sqrt{t}}$.