1 Bookkeeping

1.1 Outline

Today: Communication Complexity

1. Upper Bounds
2. Lower Bounds for IP (Inner Product)
   - Distributional Complexity
   - Discrepancy, Spectrum

1.2 Administrative Details

- Problem set 3 is out, due Friday March 15
- Professor Sudan has extra office hours this week on Friday from 1-3pm
- List of topics for the project will come out shortly. Find partners! Use Piazza!

2 Communication Complexity Review

Recall that our model of communication is for Alice and Bob, given \( x \in \{0,1\}^n \) and \( y \in \{0,1\}^n \) respectively, to send binary strings to each other in rounds in order for Bob to compute \( f: \{0,1\}^{2n} \to S \) on \( (x,y) \), where \( S \) is a finite set often chosen to be \( \{0,1\} \). We can also add randomness to this model in two ways: either by public randomness, where a random string is available to both Alice and Bob simultaneously, or by private randomness, where a random string is available to Alice and not Bob and similarly a random string is available to Bob but not Alice.

As before, we have:

**Definition 1** (Communication Complexity). We define the communication complexity of \( f: \{0,1\}^{2n} \to S \) to be

\[
CC(f) \triangleq \min_{\pi} \{ \text{# bits exchanged by } \pi \},
\]

where the min is taken over all protocols \( \pi \) computing \( f \). Similarly, we define the private randomness communication complexity of \( f \) to be

\[
CC_{\text{Priv}}(f) \triangleq \min_{\pi} \{ \text{# bits exchanged by } \pi \text{ with private randomness} \},
\]

and the public randomness communication complexity of \( f \) to be

\[
CC_{\text{Pub}}(f) \triangleq \min_{\pi} \{ \text{# bits exchanged by } \pi \text{ with public randomness} \}.
\]
Note that it’s clear from these definitions that
\[ \text{CC}^\text{Pub}(f) \leq \text{CC}^\text{Priv}(f) \leq \text{CC}(f), \]
for any \( f \). We also have the following inequalities in the other direction:

**Proposition 2.** For all \( f : \{0,1\}^{2n} \rightarrow S \), we have \( \text{CC}^\text{Priv}(f) \leq \text{CC}^\text{Pub}(f) + O(\log(n)) \).

**Proposition 3.** For all \( f : \{0,1\}^{2n} \rightarrow S \), we have \( \text{CC}(f) \leq 2^{O(\text{CC}^\text{Priv}(f))} \).

Note that these two inequalities are tight for Equality(\( x,y \)).

### 3 Upper Bound Examples

#### 3.1 Hamming Distance

Consider the function
\[ \text{HammingDist}_k(x,y) = \begin{cases} 1 & \text{if } \Delta(x,y) \leq k, \\ 0 & \text{if } \Delta(x,y) > k, \end{cases} \]
for some parameter \( k \), where \( \Delta(x,y) = \#\{i : x_i \neq y_i\} \) is the Hamming distance between \( x,y \in \{0,1\}^n \). It turns out that there is a \( \Theta(k \log k + 1) \) bit protocol with shared randomness (does not depend on \( n \)). Today, we will see a \( \Theta(k^2 + 1) \) bit protocol with shared randomness. Note that if \( k = 0 \), this is the Equality function, which we know has \( \Theta(1) \) public randomness communication complexity, so this protocol is reasonably tight for small \( k \).

#### 3.2 Small Set Disjointness

Consider the Small Set Disjointness problem, where Alice gets \( S \subseteq [n] \) and Bob gets \( T \subseteq [n] \) (both represented as a characteristic vector) with the condition that \( |S|, |T| \leq k \) for some parameter \( k \). The goal is to output whether \( S \cap T = \emptyset \). Hastad and Wigderson give a \( \Theta(k) \) bit protocol, but we will see a \( \Theta(k \log k) \) bit protocol today.

#### 3.3 Protocols using hash functions

Both of these problems can be solved by protocols that publicly pick a completely random hash function \( h : [n] \rightarrow [m] \), which can be shown to have the property that for all \( W \subseteq [n] \) with \( |W| \leq k \), we have
\[ \Pr_h[\exists i \neq j \in W \text{ s.t. } h(i) = h(j)] \leq \frac{1}{100}, \]
for some \( m = O(k^2) \).

**Exercise 4.** Prove that a uniformly random function \( h : [n] \rightarrow [m] \) satisfies the above property for some \( m = O(k^2) \).

For Small Set Disjointness, we can apply this to \( W = S \cup T \), and Alice can send \( \{h(i)\}_{i \in S} \) to Bob, which takes \( |S| \log m \leq O(k \log k) \) bits. Since the probability of any collision is small, we know that Bob can recover \( S \) with high probability and thus compute whether \( S \cap T = \emptyset \).

For HammingDist\(_k\), for all \( j \in [m] \), Alice can compute
\[ u_j = \bigoplus_{i \in h^{-1}(j)} x_i \]
and send the message \( \{u_j\}_{j \in [m]} \). Then, Bob can similarly compute

\[
v_j = \bigoplus_{i \in k^{-1}(j)} y_i,
\]

and check whether \( \Delta(u,v) \leq k \). If \( \Delta(x,y) \leq k \), then \( x \) and \( y \) differ in at most \( k \) indices \( \subseteq \{i_1, \ldots, i_k\} \), which implies that \( u \) and \( v \) differ only on a subset of the indices \( \{h(i_1), \ldots, h(i_k)\} \), which implies \( \Delta(u,v) \leq k \). If \( \Delta(x,y) > k \), then one can show that \( \Delta(u,v) > k \) with probability \( \geq 2/3 \), which completes the analysis of this \( \Theta(k^2 + 1) \) bit protocol for HammingDist_k.

### 3.4 “Distance” problems in \( \mathbb{R}^n \)

Here, Alice and Bob are given \( x, y \in \mathbb{R}^n \) respectively with \( \|x\|_2 = \|y\|_2 = 1 \). First, consider the function

\[
f(x,y) = \sum_{i=1}^{n} x_i - y_i
\]

where we allow an additive error of up to \( \varepsilon \).

**Remark** The requirement that \( \|x\|_2 = \|y\|_2 = 1 \) is only so that the error term \( \varepsilon \) makes sense, as otherwise, we could scale \( x \) and \( y \) up without any change in \( \varepsilon \), which would be too good to be true.

For this function, the protocol is easy: Alice sends \( \sum x_i \pm \varepsilon \) in \( O(\log 1/\varepsilon) \) bits, and Bob can compute the rest.

What about the function

\[
f(x,y) = \sum_{i=1}^{n} (x_i - y_i)^2
\]

with an additive error of up to \( \varepsilon \)? Here, the cross-terms \( x_iy_i \) cause us difficulty. However, with randomness, Alice and Bob can overcome this obstacle. Specifically, Alice can send \( \sum x_i^2 \pm \varepsilon \), similar to before, and now she can also send \( \sum R_ix_i \), where \( R_1, \ldots, R_n \) are “bits” identically and independently distributed uniformly over \( \{-1,1\} \). For Bob to decode this, note that

\[
E[R] \left( \sum_i R_ix_i \right) \left( \sum_j R_jy_j \right)
= E[R] \left[ \sum_i R_i^2x_iy_i \right] + E[R] \left[ \sum_{i \neq j} R_iR_jx_iy_j \right]
= \sum_i x_iy_i + 0,
\]

where the last equality comes from the fact that \( R_i^2 = 1 \) and \( E[R_iR_j] = 0 \) for all \( i \neq j \). Therefore, Bob can take \( \sum_i R_ix_i \) from Alice and \( \sum_j R_jy_j \) directly from its input and multiply them to get an estimate for \( \sum_i x_iy_i \). Given that Alice sends \( \sum x_i^2 \) and Bob can deduce \( \sum y_i^2 \), Bob can estimate \( \sum_i (x_i - y_i)^2 \). For this to work with high probability, we need to squash the variance of the random variable \( \sum_{i \neq j} R_iR_jx_iy_j \). We can squash this variance successfully with \( O(1/\varepsilon^2) \) bits of communication. In fact, this is the best we can hope for:

**Exercise 5.** Prove that \( 1/\varepsilon^2 \) bits are required for any protocol to compute \( f(x,y) = \sum_{i=1}^{n} (x_i - y_i)^2 \) up to an additive error of \( \varepsilon \).

In summary, for the function \( f(x,y) = (\sum_{i=1}^{n} x_i - y_i)^2 \pm \varepsilon \), there is a protocol that uses \( O(\log 1/\varepsilon) \) bits, but for the function \( f(x,y) = (\sum_{i=1}^{n} (x_i - y_i)^2)^2 \pm \varepsilon \), the best protocol uses \( \Theta(1/\varepsilon^2) \) bits.
4 Lower Bounds for Inner Product

Recall that \( IP(x, y) \triangleq \sum_{i=1}^{n} x_i y_i \mod 2 \), for \( x, y \in \{0, 1\}^n \). How can we prove a \( \Omega(n) \) lower bound on communication complexity of \( IP \) with shared randomness? One avenue to pursue would be to look at the rank of the matrix \( M_{IP} \), but we saw for Equality that rank was not helpful in proving lower bounds for protocols with randomness. So, we need something new.

4.1 Distributional Complexity

**Idea:** Put a distribution \( \mu \) on \( \{0, 1\}^n \times \{0, 1\}^n \). We can define 
\[
\delta_\mu(f, g) \triangleq \Pr_{(x, y) \sim \mu}[f(x, y) \neq g(x, y)]
\]
and
\[
D_{\varepsilon, \mu}(f) \triangleq \min_g \text{s.t. } \delta_\mu(f, g) \leq \varepsilon \text{ CC}(g).
\]

4.2 Randomized Protocol \( \implies \) Distributional Deterministic Protocol

With this setup, we can prove distributional lower bounds by putting some distribution \( \mu \) on \( \{0, 1\}^n \times \{0, 1\}^n \), and prove that no deterministic protocol \( \pi \) using \( k \) bits achieves small error on \( (x, y) \sim \mu \).

Why is this helpful?

**Proposition 6.** For all functions \( f : \{0, 1\}^{2n} \to S \) and distributions \( \mu \) over \( \{0, 1\}^{2n} \), we have
\[
\text{CC}^{\text{Pub}}(f) \geq \frac{D_{\varepsilon, \mu}(f)}{O(\log 1/\varepsilon)}.
\]

Thus, if we have a lower bound on \( k \) for any deterministic protocol computing \( f \) achieving small error for some distribution \( \mu \), then we must have a lower bound for any random protocol with public randomness computing \( f \).

**Proof of Proposition 6.** Suppose we have some \( k \)-bit protocol \( \pi \) that gets error less than \( 1/3 \) probability for every \( (x, y) \in \{0, 1\}^{2n} \). By repeating this protocol \( O(\log 1/\varepsilon) \) times and taking the majority of the outputs, we have a protocol \( \tilde{\pi} \) using \( O(k \log 1/\varepsilon) \) bits that errors with probability \( \leq \varepsilon \). That is, for all \( (x, y) \), we have
\[
\mathbb{E}_R[\mathbb{1}_{f(x,y)\neq \hat{\pi}(x,y, R)}] \leq \varepsilon,
\]
where the randomness \( R \) denotes the randomness of the protocol. Now, we can take the expectation over \( \mu \) and switch the order to get
\[
\varepsilon \geq \mathbb{E}_{(x,y) \in \mu} \mathbb{E}_R[\mathbb{1}_{f(x,y)\neq \hat{\pi}(x,y, R)}]
= \mathbb{E}_{R} \mathbb{E}_{(x,y) \in \mu}[\mathbb{1}_{f(x,y)\neq \hat{\pi}(x,y, R)}].
\]
This means that there exists some \( R \) such that \( \mathbb{E}_{(x,y) \in \mu}[\mathbb{1}_{f(x,y)\neq \hat{\pi}(x,y, R)}] \leq \varepsilon \), i.e. \( \Pr_{(x,y) \in \mu}[f(x,y) \neq \hat{\pi}(x,y, R)] \leq \varepsilon \). Now, we can hardcode \( R \) into \( \hat{\pi} \) to get a deterministic protocol \( \pi' \) using \( O(k \log 1/\varepsilon) \) bits, where we have \( \Pr_{(x,y) \in \mu}[f(x,y) \neq \pi'(x,y)] \leq \varepsilon \), i.e. \( \delta_{\mu}(f, \pi') \leq \varepsilon \), as desired. \( \Box \)

The idea here is that we can view randomized protocols as distributions over deterministic protocols.
4.3 Discrepancy

Now, we would like to show $D_{\mu, \varepsilon}(\text{IP}_n) \geq \Omega(n)$ for some distribution $\mu$, as from the proposition above, this would give a $\Omega(n)/\log(1/\varepsilon)$ lower bound on the number of bits of any protocol computing $\text{IP}_n$ with public randomness. In this case, thankfully choosing $\mu$ to be uniform will suffice, i.e. $\mu(x, y) = 4^{-n}$ for all $x, y \in \{0, 1\}^n$.

Suppose $\pi$ is a protocol for $f$ using $k$ bits, with error probability $\leq \varepsilon$ over $\mu$ (or equivalently, $D_{\mu, \varepsilon}(f) \leq k$). Without loss of generality, we can assume that the final bit communicated by $\pi$ is the function value (as this adds at most 1 round and 1 bit). Considering the usual matrix $M_{\text{IP}}$, we know that the $k$ bit protocol splits the matrix into $K = 2^k$ rectangles $R_1, \ldots, R_K$, where by a rectangle, we mean a Cartesian product of some $S \subseteq [n]$ and $T \subseteq [n]$. Let $p_i$ denote the probability that $\pi$ is correct and ends up in rectangle $R_i$, and let $\varepsilon_i$ denote the probability that $\pi$ is wrong and ends up in rectangle $R_i$. Then, we have

\[
\sum_{i=1}^{K} p_i \geq 1 - \varepsilon,
\]

\[
\sum_{i=1}^{K} \varepsilon_i \leq \varepsilon.
\]

Subtracting the two equations, we have $\sum_{i=1}^{K} p_i - \varepsilon_i \geq 1 - 2\varepsilon$, which implies that for some $i \in [K]$, we have

\[
p_i - \varepsilon_i \geq \frac{1 - 2\varepsilon}{K} = \frac{1 - 2\varepsilon}{2^k}.
\]

(1)

Now, we are ready for another definition. In addition to the matrix $M_f(x, y) = f(x, y) \in \{0, 1\}$ as we saw in the last lecture, we can now define $M_{f, \mu}(x, y) \triangleq \mu(x, y)(-1)^{f(x, y)}$.

Translating equation (1) into this new notation, for rectangle $R_i$, which we can say is given by rectangle $S \times T$, we have

\[
\left| \sum_{x, y \in \{0, 1\}^n} 1_S(x)1_T(y)M_{f, \mu}(x, y) \right| = |p_i - \varepsilon_i| \geq \frac{1 - 2\varepsilon}{2^k}.
\]

This motivates the following definition:

**Definition 7** (Discrepancy). We can define the discrepancy of $f$ with respect to $\mu$ to be

\[
\text{Disc}_{\mu}(f) \triangleq \max_{S, T \subseteq [n]} \left| \sum_{x, y \in \{0, 1\}^n} 1_S(x)1_T(y)M_{f, \mu}(x, y) \right|.
\]

We have just shown:

**Proposition 8.** If $D_{\mu, \varepsilon}(f) \leq k$, then we have

\[
\text{Disc}_{\mu}(f) \geq \frac{1 - 2\varepsilon}{2^k}.
\]

Our goal now is to show that $\text{Disc}_{\mu}(\text{IP}_n)$ is small, as this would imply $D_{\mu, \varepsilon}(f)$ is big by (the contrapositive of) proposition (8) which would imply that $\text{CC}^{\text{Pub}}(f)$ is big by proposition (6).
4.4 Spectrum bounds Discrepancy

We can bound $\text{Disc}_\mu(\text{IP}_n)$ directly, where we represent $S, T$ by characteristic column vectors $U, V \in \{0, 1\}^{2^n}$. Recall that $\mu$ is uniform over $\{0, 1\}^{2^n}$. Then, we have

$$\text{Disc}_\mu(\text{IP}_n) = \max_{S, T \subseteq [n]} \left| \sum_{x,y} \mathbb{1}_S(x) \mathbb{1}_T(y) M_{\text{IP}_n, \mu}(x, y) \right| \quad (2)$$

$$= \max_{U, V \in \{0, 1\}^{2^n}} |U^T M_{\text{IP}_n, \mu} V| \quad (3)$$

$$\leq \max_{U, V \in \mathbb{R}^{2^n}} |U^T M_{\text{IP}_n, \mu} V| \quad (4)$$

$$= 2^n \max_{U, V \in \mathbb{R}^{2^n}} |U^T M_{\text{IP}_n, \mu} V| \quad (5)$$

$$= 2^n \lambda_{\text{max}}(M_{\text{IP}_n, \mu}). \quad (6)$$

Thankfully, $M_{\text{IP}_n, \mu}$ has enough structure to make computing its maximum eigenvalue tractable. In fact,

Exercise 9. $M_{\text{IP}_n, \mu} = (M_{\text{IP}_1, \mu_1})^\otimes n$, where $\mu_i$ is uniform over $\{0, 1\}^i \times \{0, 1\}^i$.

Corollary 10. $\lambda_{\text{max}}(M_{\text{IP}_n, \mu}) = (\lambda_{\text{max}}(M_{\text{IP}_1, \mu_1}))^n$.

We can explicitly write $M_{\text{IP}_1, \mu_1}$ as

$$M_{\text{IP}_1, \mu_1} = \begin{bmatrix} 1/4 & 1/4 \\ 1/4 & -1/4 \end{bmatrix}$$

as $\mu_1 = 1/4$ for all inputs, and $(-1)^{xy}$ is $-1$ if $x = y = 1$ and 1 otherwise. A computation shows that $\lambda_{\text{max}}(M_{\text{IP}_1, \mu_1}) = 1/\sqrt{8}$, so $\lambda_{\text{max}}(M_{\text{IP}_n, \mu}) = (1/\sqrt{8})^n$. Thus, plugging back into (6), we get

$$\text{Disc}_\mu(\text{IP}_n) \leq 2^n \lambda_{\text{max}}(M_{\text{IP}_n, \mu}) = 2^{-n/2}.$$ 

Thus, for $k = n/2 - 1$ and $\varepsilon < 1/4$, we can apply the contrapositive of Proposition 8 to get that $D_{\mu, \varepsilon} \geq n/2 - 1$. For constant $\varepsilon < 1/4$ and applying Proposition 6, we have $\text{CC}^{\text{Pab}}(\text{IP}_n) \geq \Omega(n)$. 

CS 229r Information Theory in Computer Science-6