1 Recap

- Last class
  - \( R(\text{DISJ}) = \Omega(\sqrt{n}) \), where \( \text{DISJ}(x, y) = \land_i \text{NAND}(x_i, y_i) \). Achieved this bound by using product distribution.
  - Hellinger Distance: \( \Delta_{\text{Hel}}^2(p, q) = 1 - \sum_x \sqrt{p(x)q(x)} \).
  - \( \Delta_{\text{Hel}}^2(p, q) \leq \Delta_{\text{TV}}(p, q) \leq \sqrt{2} \Delta_{\text{Hel}}(p, q) \)

- Today
  - \( R(\text{DISJ}) = \Omega(n) \)

2 \( \Omega(n) \) DISJ bound

The high level idea is to find a distribution on the inputs, which gives a distribution on the transcript, and finding a way to get individual NANDs from the transcript.

2.1 Input distribution

The strings \((x_1, y_1) \ldots (x_n, y_n)\) will be independent across the \(n\) coordinates, but each \((x_i, y_i)\) are correlated.

Let \( \sigma \in \{A, B\}^n \).

\((x_i, y_i)\) is sampled independently from \( \eta_A \) if \( \sigma_i = A \) and from \( \eta_B \) if \( \sigma_i = B \), where:

\[
\begin{align*}
\eta_A(1, 0) &= \eta_A(0, 0) = 1/2, \eta_A(x, 1) = 0 \\
\eta_B(0, 1) &= \eta_B(0, 0) = 1/2, \eta_B(1, x) = 0
\end{align*}
\]

(In a sense, \( \sigma_i \) defines “who is active” for the \(i\)th bit.)

2.2 Bounding protocol information

Now suppose a protocol \( \Pi \) communicates with less than \( \delta n \) bits for some constant \( \delta \) and errs with probability at most \( 1/2 - \varepsilon \). We can bound

\[
I(X, Y; \Pi) \leq H(\Pi) \leq \delta n,
\]

where we also use \( \Pi \) to refer to the transcript of this protocol. Also, \( I(X, Y; \Pi) \geq \sum_{k=1}^{n} I(X_k, Y_k; \Pi) \). Putting these together gives

\[
E_{\text{uniform}}[I(X, Y; \Pi)] \leq \delta
\]
So far nothing we have done depends on $\sigma$. Since the above is true for fixed $\sigma$, it is true for distributional $\sigma$. Thus we have

$$\implies E_{\sigma \text{unif}} E_k I(X_k, Y_k; \Pi) \leq \delta$$
$$\implies E_k E_{\sigma} I(X_k, Y_k; \Pi) \leq \delta$$

Thus there is a fixed $k$ such that $E_{\sigma} I(X_k, Y_k; \Pi) \leq \delta$. We can decompose $\sigma$ into coordinates; define $\sigma_{-k} := (\sigma_1, \ldots, \sigma_{k-1}, \sigma_{k+1}, \ldots, \sigma_n)$. Continuing,

$$\implies E_{\sigma_{-k}} E_{\sigma_k} I(X_k, Y_k; \Pi) \leq \delta$$
$$\implies \text{fixed } \sigma_{-k} \text{ such that } E_{\sigma_k} I(X_k, Y_k; \Pi) \leq \delta$$
$$\implies I(X_k, Y_k; \Pi | \sigma_k = A) + I(X_k, Y_k; \Pi | \sigma_k = B) \leq 2\delta$$

Intuitively, the protocol does not carry much information about $x_k, y_k$, which will give a contradiction if we try to compute NAND as the protocol should.

### 2.3 Computing NAND(x,y)

Alice and Bob receive 1 bit $x, y \in \{0, 1\}$ and want to compute NAND($x, y$) using $\Pi$. Set $X_k = x, Y_k = y$, sample $X_{-k}, Y_{-k}$ randomly from $\sigma_{-k}, \eta_A, \eta_B$.

Run $\Pi$ on $(X, Y)$ and note that $DISJ(X, Y) = NAND(x, y)$. By definition of protocol $\Pi$, Alice and Bob compute $NAND(x, y)$ with error at most $1/2 - \varepsilon$. Call this whole NAND protocol $\pi$.

By what we showed before,

$$I((X_k, Y_k); \pi(X_k, Y_k) | (X_k, Y_k) \sim \eta_A) + I((X_k, Y_k); \pi(X_k, Y_k) | (X_k, Y_k) \sim \eta_B) \leq 2\delta$$
$$\implies I(Z; \pi(Z, 0)) + I(Z; \pi(0, Z)) \leq 2\delta$$

where $Z$ is uniform at random in $\{0, 1\}$.

Recall from Problem Set 1, Problem 6 that

$$I(Z, \pi(Z, 0)) \geq \frac{1}{2} \left[ \Delta_{TV}^2(\pi(Z, 0),\pi(0, 0)) + \Delta_{TV}^2(\pi(Z, 0),\pi(1, 0)) \right]$$

where $\Delta_{TV}(p, q) = \frac{1}{2} \sum_x |p(x) - q(x)| = \max_{S \subseteq \text{supp}(p)} |p(S) - q(S)|$. Coming this with Cauchy-Schwartz and the Triangle Inequality gives

$$I(Z; \pi(Z, 0)) + I(Z; \pi(0, Z)) \geq \frac{1}{2} \left[ \Delta_{TV}^2(\pi(Z, 0),\pi(0, 0)) + \Delta_{TV}^2(\pi(Z, 0),\pi(1, 0)) \right]$$
$$+ \Delta_{TV}^2(\pi(0, Z),\pi(0, 0)) + \Delta_{TV}^2(\pi(0, Z),\pi(0, 1))$$
$$\geq \frac{1}{8} \left( \Delta_{TV}(\pi(Z, 0),\pi(0, 0)) + \Delta_{TV}(\pi(Z, 0),\pi(1, 0)) \right)$$
$$+ \Delta_{TV}(\pi(0, Z),\pi(0, 0)) + \Delta_{TV}(\pi(0, Z),\pi(0, 1))$$
$$\geq \frac{1}{8} \left[ \Delta_{TV}(\pi(0, 0),\pi(1, 0)) + \Delta_{TV}(\pi(0, 0),\pi(0, 1)) \right]^2$$
$$\geq \frac{1}{8} \Delta_{TV}^2(\pi(1, 0),\pi(0, 1))$$

Actually, we could have worked directly with the Hellinger distance using:

**Exercise:** $I(Z, f(Z)) \geq \Delta_{Hell}^2(f(0), f(1))$ where $f$ is any randomized function.
This exercise gives the bound
\[ 2\delta \geq I(Z, \pi(Z, 0)) + I(Z, \pi(0, Z)) \]
\[ \geq \frac{1}{2} \Delta^2_{Hel} (\pi(1, 0), \pi(0, 1)) \]
\[ = \frac{1}{2} \Delta^2_{Hel} (\pi(0, 0), \pi(1, 1)) \]
\[ = \frac{1}{4} \Delta^2_{TV} (\pi(0, 0), \pi(1, 1)) \]
where the last equality is the lemma we showed last class. Now this is interesting because NAND is different on (0, 0) and (1, 1). In particular,
\[ \Delta_{TV}(\pi(0, 0), \pi(1, 1)) \geq |\Pr(\pi(0, 0) = 0) - \Pr(\pi(1, 1) = 0)| \geq 2\varepsilon \]
Putting the last few inequalities together gives
\[ 2\delta \geq \varepsilon^2 \implies \delta \geq \frac{\varepsilon^2}{2} \]
This implies \( R_{1/2-\varepsilon}(DISJ) \geq \frac{\varepsilon^2}{4} n \), completing the proof.

In fact, it was recently showed that \( R_{1/2-\varepsilon}(DISJ) = \Omega(\varepsilon n) \) (Braverman, Moitra ’12)

### 3 Application: Moments in the streaming model

**Setting:** We have a sequence \( a_1, a_2, \ldots, a_m \). \( a_i \in [n] \) arrives as a stream. For all \( i \), \( f_i := |\{j \in [m], a_j = i\}| \) (frequency).

**Goal:** Compute \( \max_i f_i \). Not very hard (might want to compute other moments but turns out infinite moment is hardest).

**Challenge:** Use as little memory as possible. Obviously we can do it in linear memory, can we do better?

**Theorem 1.** Any streaming algorithm needs \( \Omega(n) \) memory.

**Proof:** We will reduce from DISJ. Given \((x, y)\) to DISJ and streaming algorithm \( A \), we can construct a protocol for computing DISJ:

Alice maps \( x \) to the stream \( a_x = \{i \mid x_i = 1\} \). She runs \( A \) on \( a_x \), and sends the state of \( A \) to Bob. Bob continues the execution of \( A \) with sequence \( b_y = \{i \mid y_i = 1\} \). Then the output \( \max_i f_i \) is 1 if \( DISJ(x, y) = 1 \), and 2 if \( DISJ(x, y) = 2 \).

The communication cost of this protocol is the memory footprint of \( A \), which must be \( \Omega(n) \) by the bound on DISJ. Note that this shows \( A \) can’t even estimate the answer probabilistically.

### 4 Information Cost

**Def:** \( IC_{ext}(\Pi, \mu) = I_{(X,Y) \sim \mu} (X,Y; \Pi) \), referring to the information cost for an external observer.

We can also define a similar idea about what Alice and Bob learn about each other’s input from \( \Pi \):

**Def:** \( IC(\Pi, \mu) = I(\Pi; Y|X) + I(\Pi; X|Y) \), where \((X,Y) \sim \mu\).