Last time: \( R(\text{DISJ}) = \Omega(\sqrt{n}) \) using product distributions

\[ \Delta_{\text{Hel}}^2(p, q) = 1 - \sum_x \sqrt{p(x) \cdot q(x)} \]

\[ \Delta_{\text{Hel}}^2(p, q) \leq \Delta_{\text{TV}}(p, q) \leq \sqrt{2} \Delta_{\text{Hel}}^2(p, q) \]

Today: \( R(\text{DISJ}) = \Omega(n) \) \[\text{[KS'87, Razborov '90, BJKS '04]}\]

\( \text{DISJ}(x_i, y_i) = \bigwedge_i (\overline{x_i} \lor y_i) = \bigwedge_i \text{NAND}(x_i, y_i) \)

High-level idea: protocol computing \( \text{DISJ} \) must carry enough information for estimating the individual \( \text{NANDs} \).

Let \( \sigma \in \{A, B\}^n \). Consider the input distribution \( (x_1, y_1), \ldots, (x_n, y_n) \), where \( (x_i, y_i) \) is sampled independently:

- sampled from \( \eta_A \) if \( \sigma_i = A \) and from \( \eta_B \) if \( \sigma_i = B \).
\[
\begin{align*}
\eta_A(1,0) &= \eta_A(0,0) = \frac{1}{2}, & \eta_A(0,1) = \eta_A(1,1) &= 0 \\
\eta_B(0,1) &= \eta_B(0,0) = \frac{1}{2}, & \eta_B(1,0) = \eta_B(1,1) &= 0.
\end{align*}
\]

"Alice active in \( \eta_A \) and Bob active in \( \eta_B \)."

\[ \text{NB. } \text{DISJ}(X,Y) = 1 \text{ w.p. } 1 \]

[Similar to the "fooling set argument".]

\[ \forall I \]

Suppose the protocol communicates \( \leq \delta n \) bits \( (|\Pi| \leq \delta n) \) and \( \delta \) errs with

\[ \text{prob } \leq \frac{1}{2} - \varepsilon. \]

\[ \forall I(X,Y;i\Pi) \leq H(\Pi) \leq |\Pi| \leq \delta n. \]

Since \( (X_k,Y_k) \) is independent for different \( k \),

\[ I(X,Y;\Pi) = \sum_{k=1}^{n} I(X_k,Y_k;\Pi) \]

\[ \Rightarrow \mathbb{E}_{k \text{ uniform in } [n]} I(X_k,Y_k;\Pi) \leq \delta \]

\[ \Rightarrow \mathbb{E}_{k \text{ uniform random}} \leq \delta \Rightarrow \mathbb{E}_{k \text{ uniform random}} I(X_k,Y_k;\Pi) \leq \delta \]
\[ \exists \text{ fixed } k \text{ s.t. } \mathbb{E}_{\sigma} I(X_k, Y_k; \Pi) \leq \delta. \]

\[ \forall k: \quad \sigma_k = (\sigma_1, -\sigma_{k-1}, \sigma_{k+1}, \ldots, \sigma_n). \]

\[ \mathbb{E}_{\sigma_k} \mathbb{E}_{\sigma_k} I(X_k, Y_k; \Pi) \leq \delta. \]

\[ \exists \text{ fixed } \sigma_k \text{ s.t. } \mathbb{E}_{\sigma_k} I(X_k, Y_k; \Pi) \leq \delta. \]

\[ I(X_k, Y_k; \Pi | \sigma_k = A) + I(X_k, Y_k; \Pi | \sigma_k = B) \leq 2\delta. \]

Now, from \( \Pi \) we construct a protocol for computing NAND of two bits \( x, y \in \{0, 1\} \):

* Alice and Bob set \( X_k = x, Y_k = y \),

and sample \((X_i, Y_i), i \neq k\), according to \( \sigma_k \) and \( \eta_A, \eta_B \). Then they run \( \Pi \).

Note: \( \text{dist}(X, Y) = \text{NAND}(x, y) \).

\[ \Rightarrow \text{ Alice and Bob compute } \text{NAND}(x, y) \]

with error \( \leq \frac{1}{2} - \varepsilon \).
Call the NAND protocol $\pi(x, y)$.

\[ I(x, y; \pi(x, y) \mid (x, y) \sim \eta_A) + I(x, y; \pi(x, y) \mid (x, y) \sim \eta_B) \leq 2\delta \]

Define $\eta_A, \eta_B \Rightarrow I(Z; \pi(Z, 0)) + I(Z; \pi(0, Z)) \leq 2\delta,

$Z \sim$ uniform random on \{0, 1\}.

Recall again from P.S.1 Problem 6:

\[ I(Z; \pi(Z, 0)) \geq \frac{1}{2} \left( \Delta^2_{TV}(\pi(Z, 0), \pi(0, 0)) \right) \]

(similarly for $\pi(0, Z)$)

\[ \geq \frac{1}{4} \left( \Delta^2_{TV}(\pi(Z, 0), \pi(0, 0)) + \Delta^2_{TV}(\pi(Z, 0), \pi(1, 0)) \right) \]

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

Cauchy-Schwarz

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

\[ \geq \frac{1}{8} \left( \Delta^2_{TV}(\Pi(1,0), \Pi(0,1)) \right) \]

In fact, we could have worked with Hellinger distance, and it's true that for \( Z \sim \{0,1\} \),

\[ I(Z; f(Z)) \geq \Delta^2_{\text{Hell}} \left( f(0), f(1) \middle| f_0(x) \rightarrow \text{other distribution} \right) \]

where \( f(.) \) is a randomized function. (Exercise)

\[ \Rightarrow 2 \Delta \geq I(Z; \Pi(z,0)) + I(Z; \Pi(0,z)) \]

\[ \geq \frac{1}{2} \Delta^2_{\text{Hell}} (\Pi(1,0), \Pi(0,1)) \]

* Recall: \[ \Delta_{\text{Hell}} (\Pi(1,0), \Pi(0,1)) = \Delta_{\text{Hell}} (\Pi(0,0), \Pi(1,1)) \]

\[ \Rightarrow 2 \Delta \geq \frac{1}{2} \Delta^2_{\text{Hell}} (\Pi(0,0), \Pi(1,1)) \]

(\text{Hell} \rightarrow \text{TV})

\[ \geq \frac{1}{2} \Delta^2_{TV} (\Pi(0,0), \Pi(1,1)) \]
\[ \Delta_{TV}(p, q) = \max_{S \subseteq \text{supp}(p)} |p(S) - q(S)|. \]

\[ \Rightarrow \forall S \subseteq \text{supp}(p), \Delta_{TV}(p, q) \geq |p(S) - q(S)|. \]

\[ \Delta_{TV}(\pi(0,0), \pi(1,1)) \]

\[ \Rightarrow \frac{\delta}{\delta} \geq \frac{1}{\delta} |\Pr(\text{output of transcript } \pi(0,0) = 0) - \Pr(\pi(1,1) = 0)|. \]

\[ \geq 2\varepsilon. \]

\[ \Rightarrow 2\delta \geq \frac{1}{4} (2\varepsilon)^2 = \varepsilon^2. \]

\[ \Rightarrow R_{\frac{1}{2} - \varepsilon} (\text{DISJ}) \geq \frac{\varepsilon^2}{2}. \]

\[ \square \]

In fact, \[ R_{\frac{1}{2} - \varepsilon} (\text{DISJ}) = \Omega(\varepsilon \cdot n) \quad \text{(optimal)} \]

(Braverman, Mahtra '12)
Simple application: Moments in the streaming model.

Setting: A sequence \( a_1, a_2, \ldots, a_m, a_i \in [n] \)

arrives as a stream.

\[ \forall i \in [n], \ f_i \triangleq \left| \{ j \in [m], a_j = i \} \right| \]

Goal: Compute \( \max f_i \).

Challenge: Use as little memory as possible.

Theorem: Any streaming algorithm requires \( \Omega(n) \) space.

Proof: Reduction from disjointness.

Given \( (x, y) \) to DISJ, streaming algorithm A.

Alice: \( x \mapsto \text{sequence } a_x = \{ i | x_i = 1 \} \).

Alice runs A on \( a_x \) and sends the state of A (\( C \) bits), \( C \leq \text{memory usage of A} \) to Bob.

Bob: \( y \mapsto \text{sequence } b_y = \{ i | y_i = 1 \} \).
Bob continues execution of A with by.

\[
\operatorname{max} f_i = \begin{cases} 
0 & \text{if } \operatorname{DISJ}(x, y) = 0 \\
1 & \text{else }
\end{cases}
\]

\[
(\text{Communication cost } = C + 1)
\]

\[
\implies C = \Omega(n). \quad (\text{by disjointness lower bound})
\]

Information Cost:

Recall that for the lower bounds we looked at:

\[
\text{Def: } IC_{\text{ext}}(\Pi, \mu) = I(X, Y; \Pi | (X, Y) \sim \mu)
\]

That is, what an "external observer" learns from \((X, Y)\) by observing the transcript \(\Pi\).

* We can also define a similar notion on what Alice and Bob learn about each other's input from \(\Pi\):

\[
\text{Def: } IC(\Pi, \mu) = I(\Pi; X | X) + I(\Pi; X | Y)
\]

\[
((X, Y) \sim \mu)
\]