DSC180A FA25: Notes

Ciro, Darren, Ivy, Ryan

October 20, 2025

Contents

1 Meeting 1

Introductions;

2 Meeting 2

Notes on Ch1 of the Rao-Yehudayoff book from the second meeting, Monday Oct 6 2025. Speakers: Ciro and Ivy. This chapter covers deterministic protocols; rectangles; richness; fooling sets. Basic lower bound methods.

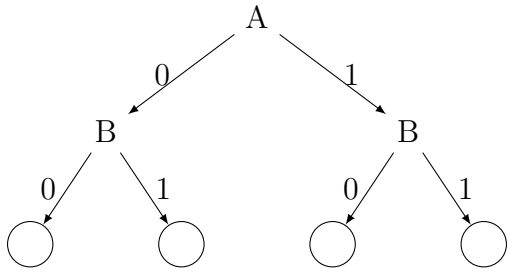
2.1 Rectangles

A **deterministic protocol** is a pre-determined procedure between how patries commnicate for whatever problem we give it. The goal of this communication is to compute something deterministic based on the input.
One way to model this is as a tree. Consider two parties, Alice and Bob. Alice has input x, Bob has y, and they wish to compute something based on x, y.

Example 2.1 (**Eq** example) Suppose Alice and Bob work to find if $x = y$, equivalently compute

$$\mathrm{Eq}(x, y) = \begin{cases} 1 & x = y, \\ 0 & x \neq y. \end{cases}$$

Suppose further x, y are two bits. We can represent this communication as a **tree**:



Remark 2.2 Notice that communication complexity presents a combinatorial setting to solve problems, which have much more mathematical tools compared to general complexity. Therefore,

Lemma 2.3 A protocol that sends c bits has a protocol tree of depth c. This implies you can partition your matrix (whose rows are the possibilities of x and columns y) into 2^c monochromatic rectangles. (Recall that a partition is a disjoint collection whose union is the whole thing.)

Definition 2.4 A **monochromatic rectangle** is a subset of the matrix $R \subseteq X \times Y$ such that if $(x, y), (x', y') \in R$, then $(x, y'), (x', y) \in R$.

Example 2.5 Consider the following matrix:

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

This cannot be one monochromatic rectangle. However,

$$\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$$

can be expressed as one monochromatic rectangle.

Example 2.6 Notice that the matrix for Eq from Example 1.1 is just the identity matrix. By 1.4 there must be at least n monochromatic rectangles, one for each 1 on the diagonal. Therefore, no protocol exists that does better than n bits communicated for Eq.

Remark 2.7 Notice that since we care about the worst case, that is all x, y, we care more about the number of rectangles, not the size of the rectangles; it is possible that our matrix is covered mostly by 1 rectangle, but many (hard) inputs can only be covered by very small rectangle.

2.2 Fooling Sets

Definition 2.8 TODO; A **fooling set** is a subset $F \subseteq X \times Y$ such that if $(x, y), (x', y') \in F$ and $f : X \times Y \to 0, 1$ is the evaluation function, then either $f(x, y') \neq f(x, y)$ or $f(x', y) \neq f(x, y)$ such that it cannot be a single monochromatic rectangle.

2.3 Richness

Definition 2.9 Define **richness** as a tuple (u, v) describing the matrix of a protocol, particularly when $|X| < |Y|$ or vice versa. u refers to the number of rows that have v 1s. Together with the size of X, Y, and the number of 1s in the matrix, you can give an upper bound on the maximum size of a 1-monochromatic-rectangle and therefore a lower bound on the number of monochromatic rectangles (and thus communication complexity in general).

Example 2.10 Consider the following matrix:

$$\begin{matrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

The above has richness $(3, 1), (2, 2), (1, 3)$. This tells us a maximum size of a 1-monochormatic rectangle (in particular, 2 by 2), which just happens to be tight here.

Remark 2.11 Note that rectangles and richness easily extend to relations, where there may be more than one possible input (for example, consider indexed sets X, Y and finding i where $X_i = Y_i$).

2.4 Computing Functions and Relations

Definition 2.12 Define the **computational complexity** of $g : X \times Y \to \{0, 1\}$ denoted $\mathrm{CC}(g)$ is the minimum depth of the protocol tree of g. Define $P(g)$ as the minimum number of leaves, equivalently, the number of monochromatic triangles of the protocol tree.

LEMMA 2.13 $P(g) \leq 2^{\mathrm{CC}(g)}$. Also, $\sqrt{\mathrm{CC}(g)} \leq \log_2 P(g) \leq \mathrm{CC}(g)$.

EXAMPLE 2.14 Suppose $g : X \times Y \to \{0,1\}$ and g requires c bits of communication. Suppose we wish to compute g^m. If g is a function, $\mathrm{CC}(g^m) = \mathrm{CC}(g)^m$ is an open problem. If g is a relation, you cannot do this. TODO; See book why.

THEOREM 2.15 $P(g^m) = P(g)^m$.

REMARK 2.16 We can study the number of leaves but not the depth...

3

Stuff

4 Randomized Protocols

4.1 Defining Randomness

DEFINITION 4.1 To talk about randomness in communication protocols, we begin by giving a careful definition of it. A protocol uses **public randomness** or **public coins** when all parties have access to a common shared random string. We denote this string by R.

A protocol uses **private randomness** or **private coins** if each party privately samples an independent random string. In the case of Alice and Bob, we call these strings R_a and R_b respectively.

We require R, R_a, R_b, and the input (X, Y) as independent uniform and the only source of randomness in the protocol. Thus, a random protocol is simply a distribution over deterministic protocols.

LEMMA 4.2 Every private coin protocol can be simulated by a public coin protocol.

Proof. Simply use the public coin as the private coin. □

LEMMA 4.3 Every public coin protocol can be simulated by a private coin protocol, albeit with a small increase in the length of the protocol.

Proof. Deferred. □

DEFINITION 4.4 By introducting randomness, we also allow **error** into our functions. There are two ways to measure the probability that a random protocol makes an error:

A random protocol has error ϵ in the **worst-case error** if for every input, the probability that the protocol makes an error is at most ϵ. That is, $\forall x, y$,

$$\Pr_{R,R_a,R_b}[\pi_{R,R_a,R_b}(x,y) \text{ is wrong}] \leq \epsilon,$$

where π_{R,R_a,R_b} is the deterministic protocol obtained by fixing said variables.

Given a distribution on inputs μ, we say that a protocol has error ϵ with respect to μ if the probability that the protocol makes an error is at most ϵ when the inputs are sampled from μ. That is,

$$\Pr_{R,R_a,R_b,(X,Y)}[\pi_{R,R_a,R_b}(X,Y) \text{ is wrong}] \leq \epsilon.$$

The **length** of a protocol is defined to be the maximum depth of all the deterministic protocol tress the random protocol may generate.

LEMMA 4.5 If a randomized protocol never errors, then we can fix the randomness to obtain a deterministic protocol that is always correct.

COROLLARY If a randomized protocol has less length than its deterministic counterparts, then it must introduce some form of error to the output of the protocol.

4.2 Equality

In equality, Alice and Bob are given n bit strings x, y and wish to know if these strings are the same or not. We know its determinstic communciation complexity is $n + 1$.

EXAMPLE 4.6 **Simple Random Equality Protocol:** Let Alice and Bob have a shared random coin of length $k2^n$. This allows Alice and Bob to recover a random function

$$h : \{0,1\}^n \to \{0,1\}^k$$

which we can treat as a hashmap. Then Alice can send $h(x)$ to Bob and Bob responds with a bit indicating if $h(x) = h(y)$.

Here $k+1$ bits are communicated and the probability of making an error is at most 2^{-k}; the probability $h(x) = h(y)$ yet $x \neq y$ is 2^{-k}, the probability k bits of x, y are the same (when hashed).

REMARK 4.7 Though the protocol is efficient it requires a large number of shared random bits. We can try to reduce the number of bits, introducing a new dimension to optimize in our protocols.

EXAMPLE 4.8 **Error-correcting Equality Protocol** An (example of an) **error correcting code** is a function

$$C : \{0,1\}^n \to [2k]^m$$

such that if $x \neq y$, then $C(x)$ and $C(y)$ differ in all but $m/2^{-\Omega(k)}$ coordinates. (When $m = 10n$, for any k most functions C will be error-correcting codes.)

Given a code, Alice can pick a random coordinate of $C(x)$ and send it to Bob, and Bob can check whether $C(x) = C(y)$. Alice takes $\log_2 m$ bits to send the index and $\log_2 2^k = k$ bits to send the value. Thus the protocol takes $O(\log n + k)$ bits of communication.

The probability of making an error is at most $2^{-\Omega(k)}$, by the definition of the error-correcting code.

EXAMPLE 4.9 **Greater-than Random Protocol** Deterministic protocols take $\log n$ bits. A randomized protocol taking $O(\log \log n)$ bits exists.

A simpler random protocol in $O(\log \log n \cdot \log \log \log n)$ can be done by using the above equality protocol to check if the first $n/2$ most significant bits of x, y are the same. This allows us to perform a binary search on which bit differs first.

4.3 Disjointness

EXAMPLE 4.10 ***k*-Disjoint Random Protocol** Suppose Alice and Bob are given sets $X, Y \subseteq [n]$, each of size at most k. They wish to check for intersection. By the rank method, at least $\log \binom{n}{k}$ bits are required.

When $k \ll n$, the deterministic protocol of length $O(k)$ is more efficient than any deterministic protocol. Here is how it goes:

Shared randomness is used to generate $R_1, R_2, \ldots \subseteq [n]$. Alice announces the smallest index i with $X \subseteq R_i$ and Bob does the symmetry with j. This takes $2(\log i + \log j + 2)$ bits. Alice replaces her set with $X \cap R_j$ and Bob with $Y \cap R_i$. Notice that if $X \cap Y = \varnothing$ this remains true, and same if $X \cap Y \neq \varnothing$.

Repeat the above until $O(k)$ bits are communicated. If either set becomes empty, X, Y are disjoint. If both sets are non-empty, they conclude X, Y intersect.

THEOREM 4.11 In the above, Alice and Bob arrive at the correct conclusion with probability at least $2/3$.

LEMMA 4.12 The expected length of the first step is at most $2(|X| + |Y| + 2)$.

Proof. The probability R_1 contains $|X|$ is $2^{-|X|}$. If it does not, then

$$\mathbb{E}[i] = 2^{-|X|} \cdot 1 + (1 - 2^{-|X|})(\mathbb{E}[i] + 1) \implies \mathbb{E}[i] = 2^{|X|}.$$

Same for Y. Then

$$\mathbb{E}[2 \log i + 2 \log j + 4] \leq 2(\log \mathbb{E}[i] + \log \mathbb{E}[j] + 2) = 2(|X| + |Y| + 2). \qquad \square$$

LEMMA 4.13 In the second round, the expectation is that the sets are half as large. Formally, for $i \in [n]$ let $X_{i,s}$ be the indicator variable for if $i \in X$ before step s on the above process. Define $Y_{i,s}$ similarly. If X, Y are disjoint,

$$\mathbb{E}\left[\sum_{i \in [n]} \sum_{s=1}^{\infty} X_{i,s} + Y_{i,s}\right] \leq 4k.$$

Proof. Notice

$$\mathbb{E}\left[\sum_{i\in[n]}\sum_{s=1}^{\infty} X_{i,s}+Y_{i,s}\right]=\sum_{i\in[n]}\sum_{s=1}^{\infty}\mathbb{E}[X_{i,s}]+\mathbb{E}[Y_{i,s}].$$

If $X_{i,1}=0$, then $X_{i,s}=0$ for all s. If $X_{i,1}=1$, the probability $X_{i,s}=1$ is 2^{-s+1} so

$$\sum_{i\in X}\sum_{s=1}^{\infty}\mathbb{E}[X_{i,s}]=\sum_{i\in X}\sum_{s=1}^{\infty}2^{-s+1}\leq 2k.$$

Applying the same to $Y_{i,s}$ gives $\leq 4k$. □

THEOREM 4.14 (MARKOV'S INEQUALITY) For any random variable $X\geq 0$,

$$\mathbb{P}(X\geq a)\leq\frac{\mathbb{E}[X]}{a}.$$

LEMMA 4.15 The previous lemma implies that the parties communicate at most $16k$ bits. By the Markov inequality, the probability the protocol communicates more than $3\cdot 16k=48k$ bits is at most $1/3$.

4.4 Hamming Distance

DEFINITION 4.16 If Alice and Bob are given two strings $x,y\in\{\pm 1\}^n$, define the **Hamming distance** as

$$H(x,y)=\Delta(x,y)=|i\in[n]\mid x_i\neq y_i|=\frac{n-\langle x,y\rangle}{2}.$$

DEFINITION 4.17 Say that a protocol π approximates Δ up to m if

$$|\pi(x,y)-\Delta(x,y)|\leq m$$

for all inputs x,y. For $alpha<1/2$, approximating Δ up to αn requires deterministic CC $\Omega(n)$.

THEOREM 4.18 **Hamming Distance Random Protocol** Suppose Alice and Bob wish to estimate Δ up to m. Both use shared randomness to sample $i_1,\ldots,i_k\in[n]$ independently and uniformly. They communicate $2k$ bits the **empirical distance**

$$\gamma=(1/k)\cdot|j\in[k]\mid x_{i_j}\neq y_{i_j}|$$

and output γn.

LEMMA 4.19 The probability of making an error is at most $1/e$.

Proof. Deferred. □

Term Index

Example Index