THE SINGLE BIGGEST PROBLEM IN COMMUNICATION IS THE ILLUSION THAT IT HAS TAKEN PLACE.

GEORGE BERNARD SHAW

WORDS EMPTY AS THE WIND ARE BEST LEFT UNSAID.

HOMER

EVERYTHING BECOMES A LITTLE DIFFERENT AS SOON AS IT IS SPOKEN OUT LOUD.

HERMANN HESSE

LANGUAGE IS A VIRUS FROM OUTER SPACE.

WILLIAM S. BUROUGHS

ANUP RAO, AMIR YEHUDAYOFF

COMMUNICATION COMPLEXITY (EARLY DRAFT)

Contents

Introduction

This is a book about *interactive communication*, a concept first made mathematically rigorous by Yao[1]. In the decades since Yao's work, the study of communication protocols has been extremely fruitful. This is largely due to two important features:

[1] Yao, 1979

- the concept is general enough that it captures something important about many other models of computation. Efficient streaming algorithms, data structures, linear programs or circuits for various problems all give rise to efficient communication protocols for solving related tasks.
- the concept is simple and natural enough that methods from combinatorics, analysis and information theory can be leveraged to understand the complexity of communication problems.

In this book, we explain some of the central results in the area of communication complexity, and show how they can be used to prove surprising results about several other models. This book is a living document: comments about the content are always appreciated, and the authors intend to keep the book up to date for the foreseeable future.

Acknowledgements

Thanks to Abe Friesen, Jeff Heer, Morgan Dixon, Guy Kindler, Vincent Liew, Venkatesh Medabalimi, Rotem Oshman, Kayur Patel, Sivaramakrishnan Natarajan Ramamoorthy, Cyrus Rashtchian, Thomas Rothvoß, and Makrand Sinha for comments that helped to improve this book.

Conventions and Preliminaries

Sets, Numbers and Functions

For a positive integer h, we use $[h]$ to denote the set $\{1, 2, \ldots, h\}$. $2^{[h]}$ denotes the power set, namely the family of all subsets of $[h]$. All logarithms are computed base 2 unless otherwise specified. A boolean function is a function whose values are in the set $\{0, 1\}$.

Random variables are denoted by capital letters (e.g. A) and values they attain are denoted by lower-case letters (e.g. a). Events in a probability space will be denoted by calligraphic letters (e.g. $\mathcal{E}$). Given $a = a_1, a_2, \ldots, a_n$, we write $a_{\leq i}$ to denote $a_1, \ldots, a_i$. We define $a_{<i}$ similarly. We write a_S to denote the projection of a to the coordinates specified in the set $S \subseteq [n]$. $[k]$ to denotes the set $\{1, 2, \ldots, k\}$, and $[k]^{<n}$ denotes the set of all strings of length less than n over the alphabet $[k]$, including the empty string. $|z|$ denotes the length of the string z.

Graphs

A graph on the set $[n]$ (called the vertices) is a collection of sets of size 2 (called edges). A clique $C \subseteq [n]$ in the graph is a subset where every edge is present in the graph. An independent set $I \subseteq [n]$ in the graph is a set that does not contain any edges.

Probability

We use the notation $p(a)$ to denote both the distribution on the variable a, and the number $\Pr_p[A = a]$. The meaning will be clear from context. We write $p(a|b)$ to denote either the distribution of A conditioned on the event $B = b$, or the number $\Pr[A = a|B = b]$. Given a distribution $p(a, b, c, d)$, we write $p(a, b, c)$ to denote the marginal distribution on the variables a, b, c (or the corresponding probability). We often write $p(ab)$ instead of $p(a, b)$ for conciseness of notation. If $\mathcal{E}$ is an event, we write $p(\mathcal{E})$ to denote its probability according to p. We denote by $\mathbb{E}_{p(a)}[g(a)]$ the expected value of $g(a)$

in p. We write $A - M - B$ to assert that $p(amb) = p(m) \cdot p(a|m) \cdot p(b|m)$.

The statistical distance between $p(x)$ and $q(x)$ is defined to be:

$$|p - q| = (1/2) \sum_x |p(x) - q(x)| = \max_T p(T) - q(T),$$

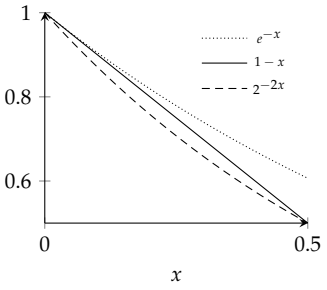
where the maximum is taken over all subsets T of the universe. We sometimes write $p(x) \overset{\epsilon}{\approx} q(x)$, to indicate that $|p(x) - q(x)| \leq \epsilon$. Suppose a, b are two variables in a probability space p. For ease of notation, we write $p(a|b) \overset{\epsilon}{\approx} p(a)$ for average b, to mean that

$$\mathop{\mathbb{E}}_{p(b)} [|p(a|b) - p(a)|] \leq \epsilon.$$

Some Useful Inequalities

Approximating linear functions with exponentials

We will often need to approximate linear functions with exponentials. For this it is often useful to note that $e^{-x} \geq 1 - x$ when $x \geq 0$, and $1 - x \geq 2^{-2x}$ when $0 \leq x \leq 1/2$.



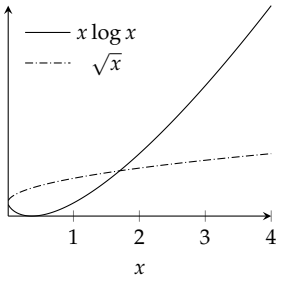
Cauchy-Schwartz Inequality

The Cauchy-Schwartz inequality says that for two vectors $x, y \in \mathbb{R}^n$, $\langle x, y \rangle \leq \|x\| \cdot \|y\|$.

Convexity

A function $f : \mathbb{R} \to \mathbb{R}$ is said to be *convex* if $\frac{f(x)+f(y)}{2} \geq f\left(\frac{x+y}{2}\right)$, for all x, y in the domain. It is said to be concave if $\frac{f(x)+f(y)}{2} \leq f\left(\frac{x+y}{2}\right)$. Some convex functions: $x^2, e^x, x \log x$. Some concave functions: $\log x, \sqrt{x}$.

Jensen's inequality says if a function f is convex, then $\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$, for any real-valued random variable X. Similarly, if f is concave, then $\mathbb{E}[f(X)] \leq f(\mathbb{E}[X])$.



Part I

Fundamentals

1
Deterministic Protocols

A PROTOCOL SPECIFIES A WAY for k parties who each have access to different inputs to communicate about their inputs, in order to learn about some property of all the inputs. Each of the k parties may have access to different bits of information. We begin by giving some interesting examples of communication problems. Many of these examples will be discussed in much more detail in future chapters of the book.

Equality Alice and Bob are given two n-bit strings $x, y \in \{0,1\}^n$ and want to know if $x = y$. There is a trivial solution: Alice can send her input to Bob, and Bob can let her know if $x = y$. This is a *deterministic*[1] protocol that takes $n+1$ bits of communication, and no deterministic protocol can do better. On the other hand, there is a *randomized*[1] protocol that uses only $O(1)$ bits of communication: the parties can hash their inputs and check that the hashes are the same. There is a *non-deterministic*[1] protocol that uses $O(\log n)$ bits of communication: If Alice guessed an index i where $x_i \neq y_i$, she could send it to Bob and they could confirm that their inputs are not the same.

[1] These terms will be made clear in due course.

Cliques and Independent Sets Alice and Bob are given two sets $A, B \subseteq [n]$ and both know a graph G on the vertex set $[n]$, with the promise that A is a clique and B is an independent set. They want to know whether A intersects B or not. There is no one-way protocol that solves this problem efficiently using less than n bits of communication. However, there is an interactive protocol that uses $O(\log^2 n)$ bits of communication. If A contains a vertex v of degree less than $n/2$, Alice announces the name of the vertex. Either $v \in B$ or Alice and Bob can safely discard all the non-neighbors of v, since these cannot be a part of A. This reduces the size of the graph by a factor of 2. Similarly, if B contains a vertex v of degree at least $n/2$, Bob announces the name of v. Again, either $v \in A$,

or Alice and Bob can safely discard all the neighbors of v which reduces the size of the graph by a factor of 2. After at most $\log n$ such steps, Alice and Bob will have determined the answer.

k-Disjointness Alice and Bob are given two sets $A, B \subseteq [n]$, each of size k, and want to know if the sets share a common element. Alice can send her set over, which takes $k \log n$ bits of communication. There is a randomized protocol that uses only $O(k)$ bits of communication. Alice and Bob sample a random sequence of sets in the universe, Alice announces the name of the first set that contains A. If A and B are disjoint, this eliminates half of B. Repeating this procedure gives a protocol with $O(k)$ bits of communication. There is a non-deterministic protocol that uses $O(\log n)$ bits of communication.

k-party Disjointness The input is k sets $A_1, \ldots, A_k \subseteq [n]$, and there are k parties. The i'th party knows all the sets except for the i'th one. The parties want to know if there is a common element in all sets. There is a deterministic protocol with $O(n/2^k)$ bits of communication, and this is known to be essentially the best protocol. We know that no randomized protocol can have communication less than $\sqrt{n}/2^k$, but it is not known whether this bound is tight.

3-Sum The input is three numbers $x, y, z \in [n]$. Alice knows (x, y), Bob knows (y, z) and Charlie knows (x, z). The parties want to know whether or not $x + y + z = n$. Alice can tell Bob x, which would allow Bob to announce the answer. This takes $O(\log n)$ bits of communication. There is a deterministic protocol that communicates $o(\log n)$ bits, but one can show that any deterministic protocol must communicate $\omega(1)$ bits. There is a randomized protocol that communicates $O(1)$ bits.

Pointer Chasing The input consists of two functions $f, g : [n] \to [n]$, where Alice knows f and Bob knows g. Let $a_0, a_1, \ldots, a_k \in [n]$ be defined by setting $a_0 = 1$, and $a_i = f(g(a_{i-1}))$. The goal is to compute a_k. There is a simple k round protocol with communication $O(k \log n)$ that solves this problem, but any protocol with fewer than k rounds requires $\Omega(n)$ bits of communication.

Graph Connectivity The input is an undirected graph on the vertices $[n]$. There are k parties, and the j'th party knows all of the edges except those that touch the vertices of $[(j-1)n/k, jn/k]$. The parties want to know whether 1 is connected to n in the graph. The trivial deterministic protocol takes $O(n^2/k)$ bits of communication. One can show that there is no randomized protocol with less than $n/2^k$ bits of communication.

Defining 2 party protocols

LET US DEFINE exactly what we mean by a 2 party deterministic protocol. Suppose the inputs come from two sets $\mathcal{X}, \mathcal{Y}$. A protocol π is specified by a rooted tree, where every internal vertex v has 2 children. Every such vertex v is associated with either the first or second party, and a function $f_v : \mathcal{X} \to \{0,1\}$ (or $f_v : \mathcal{Y} \to \{0,1\}$) mapping an input of that party to a child of the vertex.

Given inputs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the outcome of the protocol $\pi(x, y)$ is a leaf in the protocol tree, computed as follows. The parties begin by setting the current vertex v to be the root of the tree. If the first party (resp. second party) is associated with the vertex v, she announces the value of $f_v(x)$ (resp. $f_v(y)$). Both parties set the current vertex to be the child of v indicated by the value of f_v . This process is repeated until the current vertex is a leaf, and this leaf is the outcome of the protocol.

Given a boolean function $g : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ we say that π computes g if $\pi(x, y)$ determines $g(x, y)$ for every input in $\mathcal{X} \times \mathcal{Y}$. It is sometimes convenient to imagine that the leaves of the protocol tree are labeled by the value of the function that was computed.

The *communication complexity* of the protocol π, denoted $\|\pi\|$, is the depth of the protocol tree[2]. The communication complexity of a function is c if there is protocol that computes the function with c bits of communication, but no protocol can compute the function with less than c bits of communication. The *number of rounds* of the protocol is the maximum number of alternations that occur between messages of the first party and messages of the second party on any root to leaf path in the tree. An efficient protocol is one of minimal communication complexity and minimal number of rounds.

Some basic observations:

Fact 1.1. *For any protocol π, the number of rounds in $\|\pi\|$ is always at most $\|\pi\| - 1$.*

Lemma 1.2. *The number of leaves in the protocol tree for $\|\pi\|$ is at most $2^{\|\pi\|}$.*

Proof. We prove this by induction on the communication of the protocol. When the communication is 0, the number of leaves is exactly $2^0 = 1$. In general, if the communication is $\|\pi\|$, then the number of leaves in the left subtree and the right subtree of the root is at most $2^{\|\pi\|-1}$ by induction, so the total number of leaves is at most $2 \cdot 2^{\|\pi\|-1} = 2^{\|\pi\|}$. □

The setup is analogous for k party protocols. Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ be k sets. A k-party communication protocol defines a way for k parties to communicate information about their inputs, where the i'th party gets an input from the set $\mathcal{X}_i$. Every vertex v is associated with a party i and a function $f_v : \mathcal{X}_i \to \{0,1\}$

In the special case that $\mathcal{X} = \mathcal{Y} = \{0,1\}^n$, and each of the functions f_v is restricted to being equal to a bit of the input, the resulting protocol is called a *decision tree*, a model worthy of study in its own right.

One can easily generalize the definitions to handle functions that are not boolean, but we restrict our attention to booleann functions for simplicity.

For k-party protocols we may also consider functions $g : \mathcal{D} \to \mathcal{R}$ mapping some domain $\mathcal{D} \subseteq \mathcal{X}_1, \ldots, \mathcal{X}_k$ to points in $\mathcal{R}$. This generalization will become important when we study the Number-on-Forehead model.

[2] The length of the longest path from root to leaf.

Example: Alice sends 2 bits, Bob sends 3 bits, Alice sends 1 bit. Number of rounds is 2.

Balancing Protocols

LEMMA 1.2 IS TIGHT EXACTLY WHEN the protocol tree is a balanced binary tree. Does it make sense to ever have a protocol tree that is not balanced? It turns out that one can always balance an unbalanced tree.

Theorem 1.3. *If π is a protocol with ℓ leaves, then there is a protocol that computes the outcome $\pi(x, y)$ with communication at most $2\log_{3/2}\ell$.*

To prove the theorem, we need a simple lemma about trees.

Lemma 1.4. *In every protocol tree that has ℓ leaves, there is a vertex v such that the subtree rooted at v contains r leaves, and $\ell/3 \leq r \leq 2\ell/3$.*

Proof. Let r be the root of the protocol tree. Consider the sequence of vertices $r = v_1, v_2, \ldots$ defined as follows. v_1 is the root of the tree, and for each i, v_{i+1} is the child of v_i that has the most leaves under it. Let ℓ_i denote the number of leaves in the subtree rooted at v_i. By the definition of v_i, we have that $\ell_{i+1} \geq \ell_i/2$, and $\ell_{i+1} < \ell_i$. Since $\ell_1 = \ell$, and the sequence is decreasing until it hits 1, there must be some i for which $\ell/3 \leq \ell_i \leq 2\ell/3$. □

In each step of the balanced protocol, the parties pick a vertex v as promised by Lemma 1.4 and each verify whether their input is consistent with the entire path leading up to v. If this is the case, the parties repeat the procedure at the subtree rooted at v. If not, the parties delete the vertex v from the protocol tree and continue the protocol. In each step, the number of leaves of the protocol tree is reduced by a factor of at least $\frac{2}{3}$, so there can be at most $\log_{3/2}\ell$ such steps.

Input: Alice knows $x \in \mathcal{X}$, Bob knows $y \in \mathcal{Y}$, both know a protocol π that has ℓ leaves.
Output: The outcome of the protocol π.

while π *has more than* 1 *leaf* **do**
  Find a vertex v as promised by Lemma 1.4;
  Alice and Bob exchange two bits to confirm that their inputs are consistent with the path in the protocol tree to v;
  if *both inputs are consistent with* v **then**
    Replace π with the subtree rooted at v;
  else
    Remove v from the protocol tree, and replace v's parent with v's sibling;
  end
end
Output the unique leaf in π;

Figure 1.1: Rebalancing Protocol

Rectangles

A VERY USEFUL CONCEPT to understand communication protocols is the concept of a *rectangle* in the inputs. A rectangle is a subset $R = A \times B \subseteq \mathcal{X} \times \mathcal{Y}$, where $A \subseteq \mathcal{X}$ and $B \subseteq \mathcal{Y}$.

Lemma 1.5. *R is a rectangle if and only if whenever $(x, y), (x', y') \in R$, then $(x', y), (x, y') \in R$.*

Proof. If $R = A \times B$ is a rectangle, then $(x, y), (x', y') \in R$ means that $x, x' \in A$ and $y, y' \in B$. Thus $(x, y'), (x', y) \in A \times B$. On the other hand, if R is an arbitrary set with the given property, if R is empty, it is a rectangle. If R is not empty, let $(x, y) \in R$ be an element.

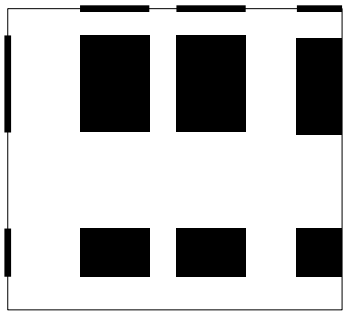


Figure 1.2: A rectangle.

For k party protocols, a rectangle is the cartesian product of k sets.

Define $A = \{x' : (x', y) \in R\}$ and $B = \{y' : (x, y') \in R\}$. Then by the property of R, $A \times B \subseteq R$, and for every element $(x', y') \in R$, $x' \in A, y' \in B$, so $R \subseteq A \times B$. Thus $R = A \times B$. □

It turns out that every vertex of the protocol tree corresponds to a rectangle of the inputs. For every vertex v, let $R_v \subseteq \mathcal{X} \times \mathcal{Y}$ denote the set of inputs (x, y) that would lead the protocol to pass through the vertex v during the execution. For the root vertex r, we see that $R_r = \mathcal{X} \times \mathcal{Y}$, so R_r is a rectangle. Now consider an arbitrary vertex v such that $R_v = A \times B$ is a rectangle. Let u, w be the children of v in the protocol tree. Suppose the first party is associated with v, and u is the vertex that the players move to when $f_v(x) = 0$. Then set

$$A_0 = \{x \in A : f_v(x) = 0\}$$
$$A_1 = \{y \in A : f_v(x) = 1\}$$

A_0, A_1 are a partition of A, and moreover $R_u = A_0 \times B$, $R_w = A_1 \times B$. Thus R_u, R_v are rectangles that partition v. Continuing in this way, we get that R_v is a rectangle for every vertex in the protocol tree. We have shown:

Lemma 1.6. *For every vertex v in the protocol tree, R_v is a rectangle. Moreover, the rectangles given by all the leaves of the protocol tree form a partition of the inputs.*

A RECTANGLE R IS SAID TO BE *monochromatic* UNDER g if g is constant on R. In other words, for every two points $(x, y), (x', y') \in R$, $g(x, y) = g(x', y')$. We say that the rectangle is 1-*monochromatic* if the function takes on the value 1 on the rectangle. We shall use Lemmas 1.2 and 1.6 to show that every function with small communication complexity induces a small partition of the inputs into monochromatic rectangles.

Suppose a protocol π computes a function $g : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, and let v be a leaf in π. If there are two inputs $(x, y), (x', y') \in R_v$ such that $g(x, y) \neq g(x', y')$, then π cannot compute g, since the outcomes of $\pi(x, y)$ and $\pi(x', y')$ are the same, and so must be incorrect in at least one case. So every leaf v of the protocol corresponds to a monochromatic rectangle R_v under g. Combining this fact with Lemmas 1.2 and 1.6 gives:

Theorem 1.7. *If the communication complexity of $g : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ is c, then $\mathcal{X} \times \mathcal{Y}$ can be partitioned into at most 2^c monochromatic rectangles.*

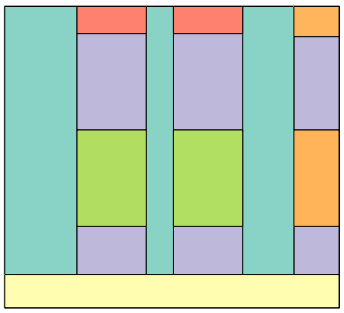


Figure 1.3: A partition of the space into rectangles.

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

Figure 1.4: A 0-monochromatic rectangle.

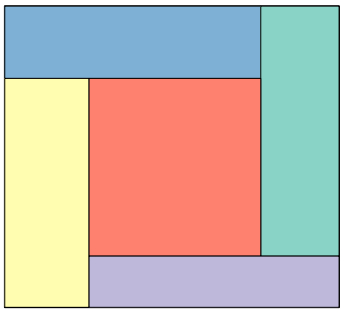


Figure 1.5: A partition into rectangles that does not correspond to a protocol.

From Rectangles to Protocols

GIVEN THEOREM 1.7, ONE MIGHT WONDER whether every partition of the inputs can be realized by a protocol. While this is not true (see Figure 1.5), we can show that a small partition of the inputs into monochromatic rectangles under g can be used to give an efficient protocol for computing g. A partial answer to this question is given by the following theorem[3]:

[3] Yannakakis, 1991; and Aho et al., 1983

Theorem 1.8. *If g admits 2^c monochromatic rectangles whose union is $\mathcal{X} \times \mathcal{Y}$, then there is a protocol that computes g with $O(c^2)$ bits of communication.*

An efficient partition of the 1's of the input space into rectangles also leads to an efficient protocol (Exercise 1.2).

A key concept we will need to understand Yannakakis's protocol is the notion of two rectangles intersecting *horizontally* and *vertically*. We say that two rectangles $R = A \times B$ and $R' = A' \times B'$ intersect horizontally if A intersects A', and intersect vertically if B intersects B'. If $x \in A \cap A'$ and $y \in B \cap B'$, then $(x, y) \in A \times B$ and $(x, y) \in A' \times B'$, proving:

Fact 1.9. *If R, R' are disjoint rectangles, they cannot intersect horizontally and intersect vertically.*

The parties are given inputs (x, y) and know a collection of monochromatic rectangles $\mathcal{R}$ that cover all inputs. The aim of the protocol is to find a rectangle $R_{x,y}$ such that $(x, y) \in R \in \mathcal{R}$. In each step, one of the parties will announce the name of a rectangle $R = A \times B$ that is consistent with their input. If Alice announces such a rectangle, then it must be that $x \in A$, so both parties can safely discard all rectangles in $\mathcal{R}$ that do not vertically intersect R. Any rectangle that contains (x, y) will not be discarded. Similarly, if Bob announces R, then both parties can safely discard all rectangles that do not horizontally intersect R. We shall show that there will always be an R that one of the parties can announce that will allow for many other rectangles to be discarded.

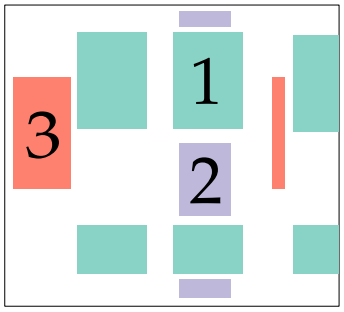


Figure 1.6: Rectangles 1 and 2 intersect vertically, while rectangles 1 and 3 intersect horizontally.

Let $\mathcal{R}_0 = \{R \in \mathcal{R} : g(R) = 0\}$, be the set of rectangles that have value 0, and $\mathcal{R}_1 = \{R \in \mathcal{R} : g(R) = 1\}$ be the set of rectangles with value 1.

Definition 1.10. *Say that a rectangle $R = (A \times B) \in \mathcal{R}_0$ is*

- *horizontally good if $x \in A$, and R horizontally intersects at most half of the rectangles in $\mathcal{R}_1$, and*
- *vertically good if $y \in B$, and R vertically intersects at most half of the rectangles in $\mathcal{R}_1$.*

Observe that Alice can compute which rectangles are horizontally good, and Bob can find all rectangles that are vertically good without any communication.

Suppose $g(x,y) = 0$. Then there must be a rectangle $R_{x,y} \in \mathcal{R}_0$ that contains (x,y). Since the rectangles of $\mathcal{R}_1$ are disjoint from $R_{x,y}$, Fact 1.9 implies that every rectangle in $\mathcal{R}_1$ can intersect $R_{x,y}$ horizontally, or vertically, but not both horizontally and vertically. Thus either at most half of the rectangles in $\mathcal{R}_1$ intersect $R_{x,y}$ horizontally, or at most half of them intersect $R_{x,y}$ vertically. Moreover, any such rectangle is consistent with both Alice and Bob's input. So we have shown:

Claim 1.11. *Any rectangle of $\mathcal{R}_0$ that contains (x,y) is either horizontally good, or vertically good.*

In each step of the protocol, one of the parties announce the name of a rectangle that is either horizontally good or vertically good, if such a rectangle exists. This leads to half of the rectangles in $\mathcal{R}_1$ being discarded. If no such rectangle exists, then it must mean that no rectangle of $\mathcal{R}_0$ covers (x,y), and so $g(x,y) = 1$. Since $\mathcal{R}_1$ can survive at most $c+1$ such discards, and a rectangle in the family can be described with c bits of communication, the communication complexity of the protocol is at most $O(c^2)$.

Open Problem 1.12. *Recent work[4] has shown that there is function g under which the inputs can be partitioned into 2^c monochromatic rectangles, yet no protocol can compute g using $o(c^{3/2})$ bits of communication. What are the best parameters with which one can prove Theorem 1.8?*

Input: Alice knows $x \in \mathcal{X}$, Bob knows $y \in \mathcal{Y}$, both know a set of monochromatic rectangles $\mathcal{R}$ whose union contains (x,y).
Output: $g(x,y)$.

while $\mathcal{R}_1$ *is not empty* **do**
  if $\exists R \in \mathcal{R}_0$ *that is horizontally good* **then**
    Alice sends Bob the name of R;
    Both parties discard all rectangles from $\mathcal{R}_1$ that do not horizontally intersect R;
  else if $\exists R \in \mathcal{R}_0$ *that is vertically good* **then**
    Bob sends Alice the name of R;
    Both parties discard all rectangles from $\mathcal{R}_1$ that do not vertically intersect R;
  else
    The parties output 1;
  end
end
The parties output 0;

Figure 1.7: A Protocol from Monochromatic Rectangle Covers

[4] Göös et al., 2015

Some lower bounds

We turn to proving that some problems do not have efficient protocols. The easiest way to prove a lower bound is to use the characterization provided by Theorem 1.7. If we can show that the inputs cannot be partitioned into 2^c monochromatic rectangles, or do not have large monochromatic rectangles, then that proves that there is no protocol computing the function with c bits of communication.

Using bounds on the size of Monochromatic Rectangles

Equality Consider the equality function $\mathsf{EQ} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ defined as:

$$\mathsf{EQ}(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

Alice can send Bob her input, and Bob can respond with the value of a function, giving a protocol with communication $n+1$. Is there a protocol with communication n? Since any such protocol induces a partition into 2^n monochromatic rectangles, a first attempt at proving a lower bound might to try and show that there is no large monochromatic rectangle. If we could prove that, then we could argue that many monochromatic rectangles are needed to cover the whole input. Unfortunately, equality *does* have large monochromatic rectangles, for example, the rectangle $R = \{(x,y) : x_1 = 0, y_1 = 1\}$. This is a rectangle that has density $\frac{1}{4}$, and it is monochromatic, since $\mathsf{EQ}(x,y) = 0$ for every $(x,y) \in R$. Instead, we will try to show that equality does not have any large 1-monochromatic rectangle.

Observe that if $x \neq x'$, then the points (x,x) and (x,x') cannot be in the same monochromatic rectangle. Otherwise, by Lemma 1.5, (x,x') would also have to be included in this rectangle. Since the rectangle is monochromatic, we would have $\mathsf{EQ}(x,x') = \mathsf{EQ}(x,x)$, which is a contradiction. We have shown:

Claim 1.13. *Every* 1*-monochromatic rectangle of* EQ *has size at most* 1.

Since there are 2^n inputs x where $\mathsf{EQ}(x,x) = 1$, this means that you need 2^n rectangles just to cover the 1's. Thus we have shown:

Theorem 1.14. *The deterministic communication complexity of* EQ *is at least* $n+1$.

Disjointness Next, consider the disjointness function, $\mathsf{Disj} : 2^{[n]} \times 2^{[n]} \to 1$ defined by:

$$\mathsf{Disj}(X,Y) = \begin{cases} 1 & \text{if } X \cap Y = \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

Alice can send her whole set X to Bob, which gives a protocol with communication $n+1$. Can we prove that this is optimal? Once again, this function does have large monochromatic rectangles, for example the rectangle $R = \{(X,Y) : 1 \in X, 1 \in Y\}$, but we shall show that there are no large monochromatic 1-rectangles. Indeed, suppose $R = A \times B$ is a 1- monochromatic rectangle. Let $X' = \cup_{X \in A} X$ and $Y' = \cup_{Y \in B} Y$. Then X' and Y' must be disjoint, so $|X'| + |Y'| \leq n$. On the other hand, $|A| \leq 2^{|X'|}$, $|B| \leq 2^{|Y'|}$, so $|R| = |A||B| \leq 2^n$. We have shown:

Claim 1.15. *Every* 1*-monochromatic rectangle of* Disj *has size at most* 2^n.

On the other hand, the number of disjoint pairs (X,Y) is exactly 3^n. That's because for every element of the universe, there are

3 possibilities: to be in X, be in Y or be in neither. Thus, at least $3^n/2^n = 2^{(\log 3-1)n}$ monochromatic rectangles are needed to cover the 1's of Disj, an so :

We shall soon prove a stronger lower bound for disjointness.

Theorem 1.16. *The deterministic communication complexity of* Disj *is at least* $(\log 3 - 1)n$.

Richness

Sometimes we need to understand *asymmetric* communication protocols, where we need separate bounds on the communication complexity of Alice and Bob. The concept of *richness*[5] is useful here:

[5] Miltersen et al., 1998

Definition 1.17. *A function* $g : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ *is said to be* (u,v) *rich if there is a set* $V \subseteq Y$, $|V| = v$, *such that for all* $y \in V$, *there is a set* $U_y \subseteq \mathcal{X}$, *with* $g(U_y, y) = 1$.

A rich function has large 1-monochromatic rectangles:

Lemma 1.18. *If* $g : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ *is* (u,v) *rich, and if there is a protocol for computing* g *where Alice sends at most* a *bits and Bob sends at most* b *bits, then* g *admits a* $\frac{u}{2^a} \times \frac{v}{2^{a+b}}$ 1*-monochromatic rectangle.*

Proof. The statement is proved inductively. For the base case, if the protocol does not communicate at all, then clearly $g(x,y) = 1$ for all $x \in \mathcal{X}, y \in \mathcal{Y}$, and the statement holds.

If Bob sends the first bit of the protocol, then Bob partitions $\mathcal{Y} = \mathcal{Y}_0 \cup \mathcal{Y}_1$. One of these two sets must have $v/2$ off the inputs y that make g (u,v) rich. By induction, this set contains a $\frac{u}{2^a} \times \frac{v/2}{2^{a+b-1}}$ 1-monochromatic rectangle, as required. On the other hand, if Alice sends the first bit, then this bit partitions $\mathcal{X}$ into two sets $\mathcal{X}_0, \mathcal{X}_1$. Every input y that has u 1's must have $u/2$ 1's in either $\mathcal{X}_0$ or $\mathcal{X}_1$. Thus there must be $v/2$ choices of inputs $y \in \mathcal{Y}$ that have $u/2$ 1's for g restricted to $\mathcal{X}_0 \times \mathcal{Y}$ or for g restricted to $\mathcal{X}_1 \times \mathcal{Y}$. By induction, we get that there is a 1-monochromatic rectangle with dimensions $\frac{u/2}{2^{a-1}} \times \frac{v/2}{2^{a-1+b}}$, as required. □

Now let us see some examples where richness can be used to prove lower bounds.

Lopsided Disjointness Suppose Alice is given a set $X \subseteq [n]$ of size $k < n$, and Bob is given a set $Y \subseteq [n]$, and they want to compute whether the sets are disjoint or not. Now the obvious protocol is for Alice to send her input to Bob, which takes $\log \binom{n}{k}$ bits[6]. However, what can we say about the communication of this problem if Alice is forced to send much less than $\log \binom{n}{k}$ bits?

[6] In Chapter 2, we show that the communication complexity of this problem is at least $\log \binom{n}{k}$.

To prove a lower bound, we need to analyze rectangles of a certain *shape*. We restrict our attention to special family of sets for

Alice and Bob. Let $n = 2kt$, and suppose Y contains exactly one element of $2i-1, 2i$, for each i, and that X contains exactly one element from $2t(i-1)+1, \ldots, 2ti$ for each i.

Figure 1.8: An input with $n = 12, k = 3, t = 2$.

Claim 1.19. *If $A \times B$ is a 1-monochromatic rectangle, then $|B| \leq 2^{kt-k|A|^{1/k}}$.*

Proof. We claim that $|\bigcup_{X \in A} X| \geq k|A|^{1/k}$. Indeed, if the union $\bigcup_{X \in A} X$ has a_i elements in $\{2t(i-1)+1, \ldots, 2ti\}$, then

$$\left|\bigcup_{X \in A} X\right| = \sum_{i=1}^{k} a_i \geq k \left(\prod_{i=1}^{k} a_i\right)^{1/k} \geq k|A|^{1/k}.$$

By the arithmetic-mean, geometric mean inequality.

$\bigcup_{X \in A} X$ cannot contain both $2i, 2i+1$ for any i, since one of these elements intersects every set of B. Thus, $\bigcup_{X \in A} X$ determines at least $k|A|^{1/k}$ elements of every set of B, and the number of possible choices for sets in B is at most $2^{kt-k|A|^{1/k}}$. □

The disjointness matrix here is at least $(t^k, 2^{kt})$-rich, since every choice Y allows for t^k possible choices for X that are disjoint. By Lemma 1.18, any protocol where Alice sends a bits and Bob sends b bits induces a 1-monochromatic rectangle with dimensions $t^k/2^a \times 2^{kt-a-b}$, so Claim 1.19 gives:

$$\begin{aligned} 2^{kt-a-b} &\leq 2^{kt-kt/2^{a/k}} \\ \Rightarrow a + b &\geq \frac{n}{2^{a/k+1}}. \end{aligned}$$

We conclude:

Theorem 1.20. *If $X, Y \subseteq [n], |X| = k$ and Alice sends at most a bits and Bob sends at most b bits in a protocol computing $\mathsf{Disj}(X, Y)$, then $a + b \geq \frac{n}{2^{a/k+1}}$.*

Span Suppose Alice is given a vector $x \in \{0,1\}^n$, and Bob is given a $n/2$ dimensional subspace $V \subseteq \{0,1\}^n$. Their goal is figure out whether or not $x \in V$. As in the case of disjointness, we start by claiming that the inputs do not have 1-monochromatic rectangles of a certain shape:

Claim 1.21. *If $A \times B$ is a 1-monochromatic rectangle, then $|B| \leq 2^{n^2/2 - n\log|A|}$.*

Proof. The set of x's in the rectangle must span a subspace of dimension at least $\log|A|$. The number of $n/2$ dimensional subspaces that contain the span of x is thus at most $\binom{2^n}{n/2 - \log|A|} \leq 2^{n^2/2 - n\log|A|}$. □

The problem we are working with is at least $(2^{n/2}, 2^{n^2/4}/n!)$-rich, since there are at least $2^{n^2/4}/n!$ subspaces, and each contains $2^{n/2}$ vectors. Applying Lemma 1.18 and Claim 1.21, we get that if there is a protocol where Alice sends a bits and Bob sends b bits,

$$\begin{aligned} & 2^{n^2/4-a-b}/n! \leq 2^{n^2/2-n\log 2^{n/2-a}} \\ \Rightarrow\ & n^2/4 - a - b - n\log n \leq na \\ \Rightarrow\ & n^2/4 - a(n+1) - n\log n \leq b. \end{aligned}$$

Theorem 1.22. *If Alice sends a bits and Bob sends b bits to solve the span problem, then $b \geq n^2/4 - a(n+1) - n\log n$.*

Using Fooling Sets

Greater-than Our next example is the greater-than function, $\mathsf{GT} : [n] \times [n] \to \{0,1\}$ defined as:

$$\mathsf{GT}(x,y) = \begin{cases} 1 & \text{if } x > y, \\ 0 & \text{otherwise.} \end{cases} \tag{1.3}$$

The trivial protocol has communication complexity $\lceil \log n \rceil$ bits, and we shall show that this is tight. The methods we used for the last two examples will surely not work here, because GT has large 0-monochromatic rectangles (like $R = \{(x,y) : x < n/2, y > n/2\}$) and large 1 monochromatic rectangles (like $R = \{(x,y) : x > n/2, y < n/2\}$). Instead we shall use a *fooling set* to to prove the bound. Consider the set of n points $S = \{(x,x)\}$. We claim:

Claim 1.23. *Two points of S cannot lie in the same monochromatic rectangle.*

Indeed, if R is monochromatic, and $x < x'$, but $(x,x), (x',x') \in R$, then since R is a rectangle, $(x',x) \in R$. This is contradicts the fact that R is monochromatic, since $\mathsf{GT}(x',x) \neq \mathsf{GT}(x',x')$. So once again, we have shown that the number of monochromatic rectangles must be at least n, proving:

Theorem 1.24. *The deterministic communication complexity of* GT *is at least* $\log n$.

Disjointness Fooling sets also allow us to prove tighter lower bounds on the communication complexity of disjointness. Consider the set $S = \{(X, \overline{X})\}$, namely X paired with its complement, for every set X. Once again, we see that no monochromatic rectangle can contain two such pairs, because if such a rectangle contained $(X,\overline{X}), (Y,\overline{Y})$ for $X \neq Y$, then it would also contain $(X,\overline{Y})$, but this

last pair of sets must intersect, while the other two pairs are disjoint. Since S has 2^n pairs, and at least one more monochromatic rectangle is required, this proves:

Theorem 1.25. *The deterministic communication complexity of disjointness is at least $n+1$.*

Krapchenko's Method

We end this chapter with a clever idea of Krapchenko. Let $\mathcal{X} = \{x \in \{0,1\}^n : \sum_{i=1}^n x_i = 0 \mod 2\}$ and $\mathcal{Y} = \{y \in \{0,1\}^n : \sum_{i=1}^n y_i = 1 \mod 2\}$. Since $\mathcal{X}$ and $\mathcal{Y}$ are disjoint, for every $x \in \mathcal{X}, y \in \mathcal{Y}$, there is an index i such that $x_i \neq y_i$. Suppose Alice is given x and Bob is given y, and they want to find an index i. How much communication is required?

Perhaps the most trivial protocol is for Alice to send Bob her entire string, but we can use binary search to do better. Since

$$\sum_{i \leq n/2} x_i + \sum_{i > n/2} x_i \neq \sum_{i \leq n/2} y_i + \sum_{i > n/2} y_i \mod 2,$$

Alice and Bob can exchange $\sum_{i \leq n/2} x_i \mod 2$ and $\sum_{i \leq n/2} y_i \mod 2$. If these values are not the same, they can safely restrict their attention to the strings $x_{\leq n/2}, y_{\leq n/2}$ and continue. On the other hand, if the values are the same, they can continue the protocol on the strings $x_{>n/2}, y_{>n/2}$. In this way, in every step they communicate 2 bits and eliminate half of their input string, giving a protocol of communication complexity $2 \log n$.

It is easy to see that $\log n$ bits of communication are necessary, because that's how many bits it takes to write down the answer. Now we shall prove that $2 \log n$ bits are necessary, using a variant of fooling sets. Consider the set of inputs

We need at least n monochromatic rectangles to cover pairs of the type $(0, e_i)$, where e_i is the i'th unit vector.

$$S = \{(x,y) \in \mathcal{X} \times \mathcal{Y} : x, y \text{ differ in only 1 coordinate}\}.$$

S contains $n \cdot 2^{n-1}$ inputs, since one can pick an input of S by picking $x \in \mathcal{X}$ and flipping any of the n coordinates. We will not be able to argue that every monochromatic rectangle must contain only one element of S or bound the number of elements in any way. Instead, we will prove that if such a rectangle does contain many elements of S, then it is big:

Claim 1.26. *Suppose R is a monochromatic rectangle that contains r elements of S. Then $|R| \geq r^2$.*

The key observation here is that two elements $(x,y), (x,y') \in S$ cannot be in the same monochromatic rectangle. For if the rectangle was labeled i, $(x,y), (x,y')$ must disagree in the i'th coordinate, but

since they both belong to S, that is the only coordinate on which they disagree. Thus $y = y'$. Similarly we cannot have two distinct elements $(x, y), (x', y) \in S$ that belong to the same monochromatic rectangle. Thus, if $R = A \times B$ has r elements of S, we must have $|A| \geq r, |B| \geq r$, proving that $|R| \geq r^2$.

Now suppose there are t monochromatic rectangles that cover the set S, and the i'th rectangle covers r_i elements of S. Then $|S| = \sum_{i=1}^{t} r_i$, but since the rectangles are disjoint, $2^{2n-2} \geq \sum_{i=1}^{t} r_i^2$. Using these facts and the Cauchy-Schwartz inequality:

$$2^{2n-2} \geq \sum_{i=1}^{t} r_i^2 \geq \left(\sum_{i=1}^{t} r_i / \sqrt{t} \right)^2 = n^2 2^{2n-2} / t,$$

proving that $t \geq n^2$. This shows that the binary search protocol is the best one can do.

Rectangle Covers

GIVEN THAT RECTANGLES PLAY such a crucial role in the communication complexity of protocols, it is worth studying alternative ways to measure the complexity of functions. Here we investigate what one can say if we count the number of monochromatic rectangles needed to cover all of the inputs.

Definition 1.27. *We say that a boolean function has a* 1*-cover of size* C *if there are* C *monochromatic rectangles whose union is all of the inputs that evaluate to* 1*. We say that the function has a* 0*-cover of size* C *if there are* C *monochromatic rectangles whose union is all of the inputs that evaluate to* 0*.*

By Theorem 1.7, every function that admits a protocol with communication c also admits a 1-cover of size at most 2^c and a 0-cover of size at most 2^c. Conversely, Theorem 1.8 shows that small covers can be used to give small communication.

Can the logarithm of the cover number be significantly different from the communication complexity? Consider the disjointness function, defined in (1.2). For $i = 1, 2, \ldots, n$, define the rectangle $R_i = \{X, Y : i \in X, i \in Y\}$. Then we see that $R_1, R_2, \ldots, R_n$ form a 0-cover for disjointness. So there is a 0 cover of size n, yet the the communication complexity of disjointness is linear is $n + 1$. However, we shall see in a later chapter that any 1-cover of disjointness must have $2^{\Omega(n)}$ rectangles.

Another interesting example is the k-disjointness function. Here Alice and Bob are given sets $X, Y \subseteq [n]$ of size k. We shall see in Chapter 2 that the communication complexity of k-disjointness is

Rectangle covers have an interesting interpretation in terms of *non-deterministic* communication complexity. If a function has a 1-cover of size C, then given any input that evaluates to 1, Alice and Bob can non-deterministically guess the name of a rectangle that covers their input, and then check that their inputs are consistent with the guessed rectangles. On the other hand, if their inputs correspond to a 0, no guess will convince them that their input is a 1. One can show that any non-deterministic protocol for a function corresponds to a 1-rectangle cover!

at least $\log \binom{n}{k} \approx k \log(n/k)$. As above, there is a 0-cover of k-disjointness using n rectangles.

Claim 1.28. *k-disjointness has a 1-cover of size $2^{2k} \ln(\binom{n}{k}^2)$.*

We prove Claim 1.28 using the probabilistic method. Sample a random 0-rectangle by picking a random set $S \subseteq [n]$ and using the rectangle $R = \{(X,Y) : X \subseteq S, Y \subseteq [n] \setminus S\}$. Namely, the set of all inputs X, Y where X is contained in S, and Y is contained in the complement of S. Now sample $t = 2^{2k} \ln\left(\binom{n}{k}^2\right)$ such rectangles independently. The probability that a particular disjoint pair X, Y is included in any single rectangle is 2^{-2k}. So the probability that the pair is excluded from all the rectangles is

$$(1 - 2^{-2k})^t \leq e^{-2^{-2k}t} < \binom{n}{k}^{-2},$$

Fact: $1 - x \leq e^{-x}$ for $x \geq 0$.

by the choice of t. Since the number of disjoint pairs X, Y is at most $\binom{n}{k}^2$, this means that the probability that *any* disjoint pair is excluded by the t rectangles is less than 1. So there must be t rectangles that cover all the 1 inputs.

Setting $k = \log n$, we have found 1-cover with $t = 2^{2\log n} \ln\binom{n}{\log n} = O(n^2 \log^2 n)$ rectangles. This example shows that Theorem 1.8 is tight, at least when it comes to rectangle covers.

Direct-sums in Communication Complexity

If a function requires c bits of communication, how much communication is required to compute k copies of the function? Given a function $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, we define $g^k : (\{0,1\}^n)^k \times (\{0,1\}^n)^k \to \{0,1\}^k$ by

$$g((x_1, \ldots, x_k), (y_1, \ldots, y_k)) = g(x_1, y_1), g(x_2, y_2), \ldots, g(x_k, y_k).$$

[7] Feder et al., 1995

We shall use many of the ideas we have developed so far to prove that[7]:

Theorem 1.29. *If g requires c bits of communication, then g^k requires at least $k(\sqrt{c} - \log n - 1)$ bits of communication.*

[8] See Exercise 1.11

In fact, one can show that even computing the two bits $\wedge_{i=1}^k g(x_i, y_i)$, and $\vee_{i=1}^k g(x_i, y_i)$ requires $k(\sqrt{c} - \log n - 1)$ bits of communication[8]. The main technical lemma we show is:

Lemma 1.30. *If g^k can be computed with ℓ bits of communication, then the inputs to g can be covered by $2n \cdot 2^{\ell/k}$ monochromatic rectangles.*

Theorem 1.8 and Lemma 1.30 imply that g has a protocol with communication $(\ell/k + \log n + 1)^2$. Thus,

$$\begin{aligned} c &\leq (\ell/k + \log n + 1)^2 \\ \Rightarrow \ell &\geq k(\sqrt{c} - \log n - 1), \end{aligned}$$

as required.

Now we turn to proving Lemma 1.30. We find the rectangles that cover the inputs to g iteratively. Let $S \subseteq \{0,1\}^n \times \{0,1\}^n$ denote the set of inputs to g that have not yet been covered by one of the monochromatic rectangles we have found. Initially, S is the set of all inputs. We claim:

Claim 1.31. *There is a rectangle that is monochromatic under g and covers at least $2^{-\ell/k}|S|$ of the inputs from S.*

Proof. Since g^k can be computed with ℓ bits of communication, by Theorem 1.7, the set S^k can be covered by 2^ℓ monochromatic rectangles, and so there must be some rectangle R that covers at least $2^{-\ell}|S|^k$ of these inputs. For each i, define

$$R_i = \{(x,y) \in \{0,1\}^n \times \{0,1\}^n : \exists (a,b) \in R, a_i = x, b_i = y\},$$

which is a rectangle, since R is a rectangle. Moreover, since this rectangle is monochromatic under g^k, it must be monochromatic under g. Now $|R| \leq \prod_{i=1}^k |R_i|$, so there must be some i for which $|R_i| \geq 2^{-\ell/k}|S|$. □

We repeatedly pick rectangles using Claim 1.31 until all of the inputs to g are covered. After $2n2^{\ell/k}$ steps, the number of uncovered inputs is at most

$$2^{2n} \cdot (1 - 2^{-\ell/k})^{2n2^{\ell/k}} \leq 2^{2n} e^{-2^{-\ell/k} \cdot 2n2^{\ell/k}} = 2^{2n} \cdot e^{-2n} < 1,$$

Fact: $1 - x \leq e^{-x}$ for $x \geq 0$.

proving that this process will stop after at most $2n \cdot 2^{\ell/k}$ steps.

Exercise 1.1

Define the inner product of two n-bit strings x, y to be $\sum_{i=1}^n x_i y_i \mod 2$. Use linear algebra to show that inner-product has no 0-rectangle of size bigger than 2^n. Conclude that the communication complexity of inner-product is $\Omega(n)$.

Exercise 1.2

Show that if $g : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is such that $g^{-1}(1)$ can be partitioned into 2^c rectangles, then g has communication complexity at most $O(c^2)$.

Exercise 1.3

Suppose Alice and Bob each get a subset of size k of $[n]$, and want to know whether these sets intersect or not. Show that at least $\log(\lfloor n/k \rfloor)$ bits are required.

Exercise 1.4

Suppose Alice gets a string $x \in \{0,1\}^n$ which has more 0's than 1's, and Bob gets a string $y \in \{0,1\}^n$ that has more 1's than 0's. They wish to communicate to find a coordinate i where $x_i \neq y_i$. Show that at least $2 \log n$ bits of communication are required.

Exercise 1.5

Show that almost all functions $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ require communication $\Omega(n)$, where Alice gets $x \in \{0,1\}^n$, Bob gets $y \in \{0,1\}^n$, and they must evaluate $f(x,y)$.

Exercise 1.6

Let X and Y be families of subsets of $[n]$. Assume for all $x \in X$ and $y \in Y$ the intersection of x and y contains at most 1 element, that is, $|x \cap y| \leq 1$. Define the communication problem as follows. Alice receives $x \in X$, Bob receives $y \in Y$, and they wish to evaluate the function $f : X \times Y \to \{0,1\}$ defined as $f(x,y) = |x \cap y|$. Show the deterministic complexity of f is $O(\log^2(n))$.

Exercise 1.7

Alice and Bob receive subsets of numbers $X, Y \subseteq [n]$. They wish to output the median of $X \cup Y$. Exhibit a deterministic protocol with $O(\log n)$ bits of communication. Show no protocol can do asymptotically better.

Exercise 1.8

Consider the partial function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, where the inputs to the parties are interpreted as 2 $n/2$ bit strings and

$$f(x,x',y,y') = \begin{cases} 1 & \text{if } x = y \text{ and } x' \neq y', \\ 0 & \text{if } x \neq y \text{ and } x' = y'. \end{cases}$$

Show that there are $2n$ monochromatic rectangles under f. Use fooling sets to show that the communication complexity of f is at least $\Omega(n)$. This proves that an analogue of Theorem 1.8 does not hold for partial functions.

Exercise 1.9

For a boolean function g, define $g^{\wedge k}$ by $g(x_1, x_2, \ldots, x_k, y_1, \ldots, y_k) = \wedge_{i=1}^{k} g(x_i, y_i)$. Show that if $g^{\wedge k}$ has a 1-cover of size 2^ℓ, then g has a 1-cover of size $2^{\ell/k}$.

Exercise 1.10

In this exercise, we will show[9] that an optimal direct sum theorem does not hold for the deterministic communication complexity of relations. Consider the problem where Alice is given a subset $X \subseteq [n]$ of size t, and Bob is given no input. The players want to output an element of X.

[9] Alon and Orlitsky, 1995

1. Show that $\log(n - t + 1)$ bits of communication are required for any deterministic protocol (and also sufficient).
2. Show that if Alice is given k sets $X_1, \ldots, X_k$, each of size t, and the parties want to compute an element from each of the sets, then there is a deterministic protocol that communicates only

$$O(k \log(n/t) + \log(kn))$$

bits, which is significantly less than $k \log(n - t + 1)$, when $t = n/2$.

Hint: Pick a random subset of $[n]^k$ of size $(n/t)^k \ln \binom{n}{t}^k$, and argue that it intersects $X_1 \times \cdots \times X_k$ with positive probability.

Exercise 1.11

Show that if $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ requires c bits of communication, then computing $\wedge_{i=1}^{k} g(x_i, y_i)$, and $\vee_{i=1}^{k} g(x_i, y_i)$ requires $k(\sqrt{c/2} - \log n - 1)$ bits of communication. HINT: Find a small 1-cover using the protocol for computing $\vee_{i=1}^{k} g(x_i, y_i)$, and 0-cover using the protocol for computing $\wedge_{i=1}^{k} g(x_i, y_i)$.

2
Rank

MATRICES GIVE A POWERFUL WAY to represent functions that depend on two inputs. We can represent $g : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ by an $m \times n$ matrix M, where $m = |\mathcal{X}|$ is the number of rows and $n = |\mathcal{Y}|$ is the number of columns, and the (i,j)'th entry is $M_{ij} = g(i,j)$. Given this interpretation of g, one can think of the inputs to the parties as unit column vectors e_i, e_j. The parties are trying to compute $e_i^T M e_j$. This view allows us to bring in the many tools of linear algebra to bear on understanding communication complexity.

Sometimes it is convenient to use $M_{ij} = (-1)^{g(i,j)}$ instead.

If there the function depends on the inputs of k parties, the natural representation is by a k-tensor.

Abusing notation, we shall sometimes refer to the communication complexity of M when we really mean to refer to the communication complexity of the associated boolean function.

Basic Properties of Rank

THE MOST BASIC QUANTITY associated with a matrix is its *rank*. The rank of a matrix is the maximum size of a set of linearly independent rows in the matrix. Its versatility stems from the fact that it has many interpretations:

Fact 2.1. *For an $m \times n$ matrix M, $\mathsf{rank}(M) = r$ if and only if:*

- *r is the smallest number such that M can be expressed as $M = AB$, where A is an $m \times r$ matrix, and B is an $r \times n$ matrix.*
- *r is the smallest number such that M can be expressed as the sum of r matrices of rank 1.*
- *r is the largest number such that M has r linearly independent columns (or rows).*

A useful property of rank that follows immediately from the definitions:

Fact 2.2. *If M' is a submatrix of M, then $\mathsf{rank}(M') \leq \mathsf{rank}(M)$.*

Another nice feature of the rank of matrices is that it behaves nicely under basic matrix operations. Since the rank of M is the minimum number of rank 1 matrices that add up to M, we get:

Fact 2.3. $|\mathsf{rank}(A) - \mathsf{rank}(B)| \leq \mathsf{rank}(A+B) \leq \mathsf{rank}(A) + \mathsf{rank}(B)$.

One consequence of Fact 2.3 is that many different representations of a matrix are more or less equivalent, when it comes to their rank. For example, if M is a boolean matrix, one can define a matrix M' of the same dimensions, with $M'_{i,j} = (-1)^{M_{i,j}}$, replacing 1's with -1 and 0's with 1. Then we see that $M' = J - 2M$, where J is the all 1's matrix, and so

Fact 2.4. $|\mathsf{rank}(M') - \mathsf{rank}(M)| \leq \mathsf{rank}(J) = 1$.

Since taking linear combinations of the rows or columns cannot increase the dimension of their span, we get:

Fact 2.5. $\mathsf{rank}(AB) \leq \min\{\mathsf{rank}(A), \mathsf{rank}(B)\}$.

The *tensor product* of an $m \times n$ matrix M and an $m' \times n'$ matrix M' is the $mm' \times nn'$ matrix $T = M \otimes M'$ whose entries are indexed by tuples $(i,i'),(j,j')$, with $T_{(i,i'),(j,j')} = M_{i,j} \cdot M_{i',j'}$. The tensor product multiplies the rank, a fact that is very useful for proving lower bounds:

Fact 2.6. $\mathsf{rank}(M \otimes M') = \mathsf{rank}(M) \cdot \mathsf{rank}(M')$.

The matrices we are working with are boolean, so one can view the entries of the matrix as real numbers, or rationals, or coming from the field of integers modulo 2: $\mathbb{F}_2$. This potentially leads to 3 different notions of rank, but we have:

Lemma 2.7. *The real rank of a boolean matrix is the same as its rational rank. The real rank is always at least as large as the rank over* $\mathbb{F}_2$.

The proof of the first fact follows from Gaussian elimination. If the rank over the rationals is r, we can always apply a linear transformation to the rows using rational coefficients to bring the matrix into this form:

$$\begin{bmatrix}
1 & 0 & 0 & \dots & 0 & M_{1,r+1} & \dots & M_{1,n} \\
0 & 1 & 0 & \dots & 0 & M_{2,r+1} & \dots & M_{2,n} \\
0 & 0 & 1 & \dots & 0 & M_{3,r+1} & \dots & M_{2,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots & & & \\
0 & 0 & 0 & \dots & 1 & M_{r,r+1} & \dots & M_{r,n} \\
0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}$$

This transformation does not affect the rank over the reals, and now it is clear that the rank is exactly r. Now if any set of rows is linearly

dependent over the rationals, then we can find an integer linear dependence between them, and so get a linear dependence over $\mathbb{F}_2$. This proves that the rank over $\mathbb{F}_2$ is at most the rank over the reals.

Throughout the rest of the book, unless we explicitly state otherwise, we shall always consider the rank over the reals. One consequence of Lemma 2.7 is:

Lemma 2.8. *A boolean matrix of rank r has at most 2^r distinct rows, and at most 2^r distinct columns.*

Proof. Since the rank over $\mathbb{F}_2$ is also at most r, every row must be expressible as the linear combination of some r rows over $\mathbb{F}_2$. There are only 2^r such linear combinations possible, so there can be at most 2^r distinct rows. □

Lower bounds using Rank

Lemma 2.8 immediately gives some bound on the communication in terms of the rank of the matrix. If the matrix has rank r, it has at most 2^r distinct rows. Alice only needs to communicate which one of these rows her row corresponds to. This takes r bits of communication. Bob can then respond with the value of the function. We have shown:

Theorem 2.9. *If a matrix has rank r then its communication complexity is at most $r + 1$.*

Theorem 2.9 is far from the last word on the subject. By the end of this chapter, we will prove that the communication is bounded by a quantity closer to $\sqrt{r}$.

The main reason that rank is useful in this context is that it can be used to prove lower bounds on communication, via the following theorem:

Lemma 2.10. *If a boolean matrix can be partitioned into 2^c monochromatic rectangles, then its rank is at most 2^c.*

Lemma 2.10 applies even if the matrix has $+1, -1$ entries.

Lemma 2.10 follows easily from Fact 2.1. For every rectangle $R = A \times B$, define the matrix where $R_{i,j} = 1$ if $(i, j) \in R$, and $R_{i,j} = 0$ otherwise. Then we see that R a matrix that has rank 1. Moreover, M can be expressed as the sum of at most 2^c such matrices, those that correspond to 1-rectangles.

Since every function with low communication gives rise to a partition into monochromatic rectangles (Theorem 1.7), we immediately get:

Theorem 2.11. *If a matrix has rank r, then its communication complexity is at least $\log r$.*

Theorem 2.11 allows us to prove lower bounds on many of the examples we have already considered. So let us revisit some of them.

Equality We start with the equality function, defined in (1.1). The matrix of the equality function is just the identity matrix. Since the rows of this matrix are all linearly independent, the rank of the matrix is 2^n, proving that the communication complexity of equality is at least n bits.

Greater-than Consider the greater than function, defined in (1.3). The matrix of this function is the upper-triangular matrix which is 1 above the diagonal and 0 on all other points. Once again we see that rows are linearly independent, and so the matrix has full rank. This proves that the communication complexity is at least $\log n$.

Disjointness Consider the disjointness function, defined in (1.2). Let D_n be boolean matrix that represents disjointness. Let us order the rows of the matrix in lexicographic order, so that the sets that contain the element n correspond to the last row and last column. If we partition the rows into two parts based on whether the row corresponds to a set that contains n or not, and do the same for the columns, we get that if the rows and columns come from the part where n is included in both, then the matrix is 0. However, if n is included in only the rows, or only the columns, we get a copy of the matrix D_{n-1}. So D_n can be expressed as:

$$D_n = \begin{bmatrix} D_{n-1} & D_{n-1} \\ D_{n-1} & 0 \end{bmatrix}$$

In other words $D_n = D_1 \otimes D_{n-1}$, and so $\mathsf{rank}(D_n) = 2 \cdot \mathsf{rank}(D_{n-1})$ by Fact 2.6. We conclude that $\mathsf{rank}(D_n) = 2^n$, proving that the communication complexity of disjointness is at least n.

k-disjointness Consider the disjointness function restricted to sets of size at most k. In this case, the matrix is an $\sum_{i=0}^{k} \binom{n}{i} \times \sum_{i=0}^{k} \binom{n}{i}$ matrix. Let us write $D_{n,k}$ to represent the matrix for this problem. For two sets $X, Y \subseteq [n]$, define the monomial $x = \prod_{i \in X} y_i$, and the string $y \in \{0,1\}^n$ such that $y_i = 0$ if and only if $i \in Y$. Then we see that $\mathsf{Disj}(X, Y) = x(y)$. Now any non-zero linear combination of the rows corresponds to a linear combination of the monomials we have defined, and so gives a non-zero polynomial f. To show that the matrix has full rank, we need to prove that there is a set Y that gives rise to an input y with $f(y) \neq 0$.

Alexander Razborov, 1987.

To show this, let X be a set that corresponds to a monomial of maximum degree in f. Let us restrict the values of all variables outside X to be equal to 1. After doing this, f becomes a non-zero polynomial that depends only on the variables of X. Since such polynomials are in one to one correspondence with the boolean functions on these variables, we get that there must be some

setting of the variables of X giving an assignment y with $f(y) = 1$. Moreover, this gives an assignment to y with at most k entries that are 0.

Inner-product Our final example the inner product function $\mathsf{IP} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, defined by

$$\mathsf{IP}(x, y) = \langle x, y \rangle \mod 2. \tag{2.1}$$

The trivial protocol takes n bits, and one case use bounds on the size of the largest rectangle to show that the communication is at least $\Omega(n)$. Here it will be helpful to use Fact 2.4. If P_n represents the matrix whose entries are , sorting the rows and columns lexicographically, we see that

See Exercise 1.1

$$P_n = \begin{bmatrix} P_{n-1} & P_{n-1} \\ P_{n-1} & -P_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes P_{n-1},$$

and so by Fact 2.6, $\mathsf{rank}(P_n) = 2\mathsf{rank}(P_{n-1})$. This proves that $\mathsf{rank}(P_n) = 2^n$, and so the communication complexity of IP is at least n.

Towards the Log-Rank Conjecture

Lovasz and Saks conjectured[1] that Theorem 2.11 is closer to the truth than Theorem 2.9:

[1] Lovász and Saks, 1988

Conjecture 2.12. *There is a constant α such that the communication complexity of a matrix M is at most $\log^{\alpha} \mathsf{rank}(M)$.*

Kushilevitz showed[2] that α must be at least $\log_3 6$ for the conjecture to hold, so we cannot expect the communication complexity of a matrix to be exactly equal to its rank. Our main goal in this section is to prove the following theorem[3]:

[2] Nisan and Wigderson, 1995

[3] Lovett, 2014

Theorem 2.13. *If the rank of a matrix is r, its communication complexity is at most $O(\sqrt{r} \log^2 r)$.*

Lovett actually proves that the communication is bounded by $O(\sqrt{r} \log r)$, but we prove the weaker bound here for ease of presentation.

The proof of Theorem 2.13 relies[4] on a powerful theorem from convex geometry called John's theorem[5]. We use it to show:

[4] Rothvoß, 2014

[5] John, 1948

Lemma 2.14. *Any $m \times n$ boolean matrix of rank $r > 1$ must have a monochromatic rectangle of size at least $mn \cdot 2^{-20\sqrt{r} \log r}$.*

Let us see how to use Lemma 2.14 to get a protocol. Let R be the rectangle promised by the lemma. Then, rearranging the rows and columns, we can write the matrix as: $\begin{bmatrix} R & A \\ B & C \end{bmatrix}$. Now we claim[6] that

$$\mathsf{rank}\left(\begin{bmatrix} R \\ B \end{bmatrix}\right) + \mathsf{rank}\left(\begin{bmatrix} R & A \end{bmatrix}\right) \leq \mathsf{rank}\left(\begin{bmatrix} R & A \\ B & C \end{bmatrix}\right) + 3.$$

Indeed, one can write

$$\begin{bmatrix} R & A \\ B & C \end{bmatrix} = \begin{bmatrix} 0 & A \\ B & C \end{bmatrix} + \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} R & A \end{bmatrix} = \begin{bmatrix} 0 & A \end{bmatrix} + \begin{bmatrix} R & 0 \end{bmatrix}$$
$$\begin{bmatrix} R \\ B \end{bmatrix} = \begin{bmatrix} 0 \\ B \end{bmatrix} + \begin{bmatrix} R \\ 0 \end{bmatrix},$$

So by Fact 2.3,

$$\begin{aligned} \mathsf{rank}\left(\begin{bmatrix} R \\ B \end{bmatrix}\right) + \mathsf{rank}\left(\begin{bmatrix} R & A \end{bmatrix}\right) &\leq \mathsf{rank}(A) + \mathsf{rank}(B) + 2 \\ &\leq \mathsf{rank}\left(\begin{bmatrix} 0 & A \\ B & C \end{bmatrix}\right) + 2 \\ &\leq \mathsf{rank}\left(\begin{bmatrix} R & A \\ B & C \end{bmatrix}\right) + 3. \end{aligned} \tag{2.2}$$

Now suppose $\begin{bmatrix} R \\ B \end{bmatrix}$ has the smaller rank. Then Bob sends the bit 0 if his input is consistent with R and 1 otherwise. If it is consistent, then if $\mathsf{rank}(M) > 9$, players have reduced[7] the rank of the matrix by a factor of at least $\frac{2}{3}$. If it is not consistent, the players have reduced the size of the matrix by a factor of $1 - 2^{-20\sqrt{r}\log r}$.

By Lemma 2.8, we can assume that any matrix of rank r has at most 2^r rows and columns. The number of 0 transmissions in this protocol is at most $2r \ln 2 \cdot 2^{20\sqrt{r}\log r}$, since after that many transmissions, the number of entries in the matrix have been reduced to[8]

$$\begin{aligned} 2^{2r}(1 - 2^{-20\sqrt{r}\log r})^{2r \cdot 2^{20\sqrt{r}\log r}} &< 2^{2r} e^{-2^{-20\sqrt{r}\log r} 2r \ln 2 \cdot 2^{20\sqrt{r}\log r}} \\ &= 2^{2r} e^{-2r\ln 2} = 1. \end{aligned}$$

The number of 1 transmissions is at most $O(\log_{3/2} r)$, since after that many transmissions, the rank of the matrix is reduced to less than 6. Thus, the number of leaves in this protocol is at most $\binom{2r \ln 2 \cdot 2^{20\sqrt{r}\log r}}{\log_{3/2} r} \leq 2^{O(\sqrt{r}\log^2 r)}$. By Theorem 1.3, we can balance the protocol tree to obtain a protocol with communication $O(\sqrt{r}\log^2 r)$ that computes the same function.

It only remains to prove Lemma 2.14. To prove it, we need to understand John's theorem. A set $K \subseteq \mathbb{R}^r$ is called *convex* if whenever $x, y \in K$, then all the points on the line from x to y are also in K. The

[7] $(t+3)/2 \leq 2t/3$, when $t \geq 9$.

[8] Fact: $1 - x \leq e^{-x}$, for $x \geq 0$.

Input: Alice knows i, Bob knows j.
Output: $M_{i,j}$.

while $\mathsf{rank}(M) > 9$ **do**
  Find a monochromatic rectangle R as promised by Lemma 2.14;
  Write $M = \begin{bmatrix} R & A \\ B & C \end{bmatrix}$;
  if $\mathsf{rank}\left(\begin{bmatrix} R \\ B \end{bmatrix}\right) > \mathsf{rank}\left(\begin{bmatrix} R & A \end{bmatrix}\right)$ **then**
    if *i is consistent with R* **then**
      Both parties replace M with $\begin{bmatrix} R & A \end{bmatrix}$;
    else
      Both parties replace M with $\begin{bmatrix} B & C \end{bmatrix}$;
    end
  else
    if *j is consistent with R* **then**
      Both parties replace M with $\begin{bmatrix} R \\ B \end{bmatrix}$;
    else
      Both parties replace M with $\begin{bmatrix} A \\ C \end{bmatrix}$;
    end
  end
end
The parties exchange at most 9 bits to compute $M_{i,j}$, using Theorem 2.9;

Figure 2.1: Protocol for Low Rank Matrices with $2^{O(\sqrt{r}\log^2 r)}$ leaves.

set is called *symmetric* if whenever $x \in K$, then $-x \in K$. An ellipsoid centered at 0 is a set of the form:

$$E = \left\{ x \in \mathbb{R}^r : \sum_{i=1}^{r} \langle x, u_i \rangle^2 / \alpha_i^2 \leq 1 \right\},$$

where $u_1, \ldots, u_r$ are a basis for $\mathbb{R}^r$. John's theorem shows[9]:

[9] John, 1948

Theorem 2.15 (John's Theorem)**.** *Let $K \subseteq \mathbb{R}^r$ be a symmetric convex body such that the unit ball is the most voluminous of all ellipsoids contained in K. Then every element of K is of length at most $\sqrt{r}$.*

The most voluminous ellipsoid contained in K behaves nicely when K is changed. Suppose the ellipsoid E above is the largest ellipsoid in K. Suppose that for some i, we multiply every element of K by a number ϵ in the direction of u_i: namely we consider the convex body:

$$K' = \left\{ x' : \exists x \in K, \langle x', u_j \rangle = \begin{cases} \epsilon \cdot \langle x, u_i \rangle & \text{if } j = i, \\ \langle x, u_j \rangle & \text{otherwise.} \end{cases} \right\}$$

Since scaling the space by β in any direction changes the volume of all objects by exactly β, the largest ellipsoid in K' is the scaled version of the largest ellipsoid in K:

Fact 2.16. *The largest ellipsoid in K' is*

$$E' = \left\{ x \in \mathbb{R}^r : \sum_{j=1}^{r} \langle x, u_j \rangle^2 / \beta_j^2 \leq 1 \right\},$$

where $\beta_j = \alpha_j$ if $j \neq i$, and $\beta_i = \epsilon \alpha_i$.

Lemma 2.14 is proved in two steps. In the first step, we use John's theorem to show that the matrix must contain a large *nearly* monochromatic rectangle. In the second step, we will show how any such rectangle of low rank must contain a large monochromatic rectangle itself.

Since the matrix has rank r, we know that M can be expressed as $M = AB$, where A is an $m \times r$ matrix, and B is an $r \times n$ matrix. We start by showing:

Lemma 2.17. *Any boolean matrix M of rank r can be expressed as $M = AB$, where A is an $m \times r$ matrix whose rows are vectors of length at most $\sqrt{r}$, and B is an $r \times n$ matrix whose columns are vectors of length at most 1.*

The "$\sqrt{r}$" and "1" in the statement of Lemma 2.17 can be replaced by any numbers whose product is $\sqrt{r}$.

Proof. Start with $M = AB$ for A, B not necessarily satisfying the length constraints. Let $v_1, \ldots, v_m$ be the rows of A, and let $w_1, \ldots, w_n$ be the columns of B. Let K be the convex hull of $\{\pm v_1, \ldots, \pm v_m\}$.

An ellipsoid centered at the origin in the space is specified by a basis $u_1, \ldots, u_r$ for the space, and numbers $\alpha_1, \ldots, \alpha_r$. The ellipsoid determined by these parameters is the set

$$E = \left\{ x \in \mathbb{R}^r : \sum_{i=1}^{r} \langle x, u_i \rangle^2 / \alpha_i^2 \leq 1 \right\}.$$

Our first goal is to ensure that the ellipsoid of maximum volume in K is the unit ball. This is the same as ensuring that $\alpha_1 = \alpha_2 = \ldots = \alpha_r = 1$. Suppose α_i is not 1 for some i. Then we can scale every vector v_j by a factor of α_i in the direction[10] u_i, and scale every vector w_j by a factor of $1/\alpha_i$ in the direction of u_i. This preserves the inner products of all pairs of vectors. By Fact 2.16, repeating this for each coordinate i ensures that the ellipsoid of maximum volume in K is the unit ball. Now, by John's theorem, every vector v_i must have length at most $\sqrt{r}$, since every vector in K has length at most $\sqrt{r}$.

[10] Formally, we write $v_j = \sum_{i'=1}^{r} \gamma_{i'} u_{i'}$, and $w_k = \sum_{i'=1}^{r} \beta_{i'} u_{i'}$ and replace γ_i with $\alpha_i \gamma_i$, and β_i with β_i / α_i. This preserves the inner product $\langle v_j, w_k \rangle$.

It only remains to argue that vectors $w_1, \ldots, w_n$ are of length at most 1. This is where we use the fact that the matrix is boolean. Consider any w_i, and the unit vector in the same direction: $e_i = w_i / \|w_i\|$. The length of w_i can be expressed as $\langle w_i, e_i \rangle$, but since e_i is in the unit ball, and so is contained in K, $e_i = \sum_j \mu_j v_j + \sum_j \kappa_j (-v_j)$ is a convex combination of the v_j's. Thus

$$\langle w_i, e_i \rangle = \sum_j \mu_j \langle w_i, v_j \rangle + \sum_j \kappa_j \langle w_i, -v_j \rangle \leq \sum_j \mu_j + \sum_j \kappa_j = 1,$$

where the inequality follows from the fact that M is boolean. □

For the rest of the proof, we assume that M has as least $mn/2$ 0's. We can do this, because if M has more 1's than 0's, we can replace M with $J - M$, where J is the all 1's matrix. This can increase the rank by at most 1, but now the role of 0's and 1's has been reversed.

Lemma 2.17 says something about the angles between the vectors we have found. Define $\theta_{i,j} = \arccos\left(\frac{\langle v_i, w_j \rangle}{\|v_i\| \|w_j\|} \right)$. Then observe that when v_i, w_j are orthogonal, the angle is $\pi/2$. But when the inner product is 1, the angle is at most $\arccos\left(\frac{1}{\sqrt{r}}\right) \leq \frac{\pi}{2} - \frac{2\pi}{7\sqrt{r}}$.

So we get:

$$\theta_{i,j} \begin{cases} = \frac{\pi}{2} & \text{if } M_{i,j} = 0, \\ \leq \frac{\pi}{2} - \frac{2\pi}{7\sqrt{r}} & \text{if } M_{i,j} = 1. \end{cases}$$

Consider the following random experiment. Sample t vectors of length 1 uniformly at random $z_1, \ldots, z_t$, and define the rectangle R by:

$$R = \{(i,j) : \forall k, \langle v_i, z_k \rangle > 0, \langle w_j, z_k \rangle < 0\}.$$

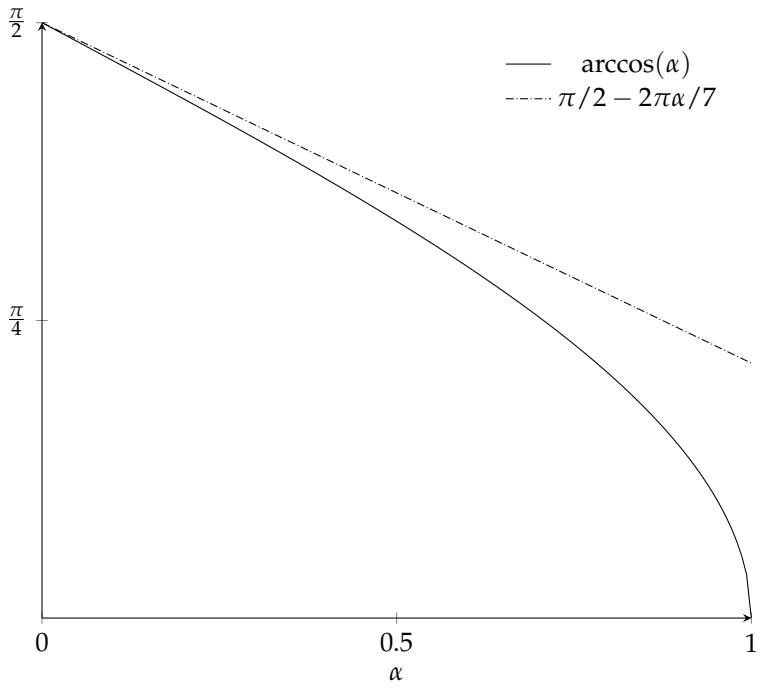


Figure 2.2: $\arccos(\alpha) \leq \pi/2 - 2\pi\alpha/7$ for $0 \leq \alpha \leq 1$.

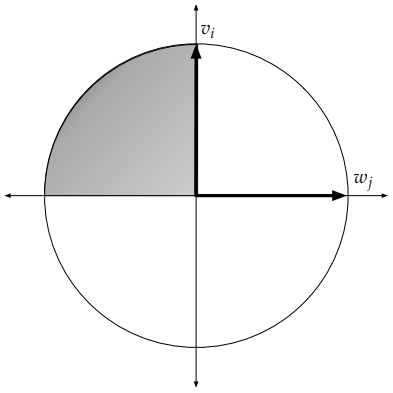


Figure 2.3: The region where all z_k's must fall to ensure that $(i,j) \in R$, when $M_{i,j} = 0$.

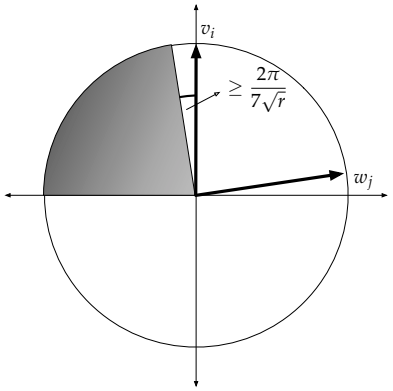


Figure 2.4: The region where all z_k's must fall to ensure that $(i,j) \in R$, when $M_{i,j} = 1$.

For a fixed (i,j) and k the probability that $\langle v_k, z_k \rangle > 0$ and $\langle w_k, z_k \rangle < 0$ is exactly $\frac{1}{4} - \frac{\pi/2 - \theta_{i,j}}{2\pi}$. So we get

$$\Pr_R[(i,j) \in R] \begin{cases} = \left(\frac{1}{4}\right)^t & \text{if } M_{i,j} = 0, \\ \leq \left(\frac{1}{4} - \frac{1}{7\sqrt{r}}\right)^t & \text{if } M_{i,j} = 1. \end{cases}$$

Let R_1 denote the number of 1's in R and R_0 denote the number of 0's. Set $t = 7\sqrt{r}\log r$. By what we have just argued,

$$\begin{aligned} \mathbb{E}[R_0] &\geq \frac{mn/2}{4^{7\sqrt{r}\log r}} \\ &= \frac{mn}{2} \cdot 2^{-14\sqrt{r}\log r}, \\ \mathbb{E}[R_1] &\leq \frac{mn/2}{4^{7\sqrt{r}\log r}} \left(1 - \frac{4}{7\sqrt{r}}\right)^{7\sqrt{r}\log r} \\ &\leq \frac{mn}{2} \cdot 2^{-14\sqrt{r}\log r} \cdot e^{-\frac{4}{7\sqrt{r}} 7\sqrt{r}\log r} \\ &= \frac{mn}{2} \cdot 2^{-14\sqrt{r}\log r} \cdot r^{-4\log e}. \end{aligned}$$

Fact: $1 - x \leq e^{-x}$ for $x \geq 0$.

Now let $Q = R_0 - r^4 R_1$. By linearity of expectation, we have

$$\begin{aligned} \mathbb{E}[Q] &\geq \frac{mn}{2} \cdot 2^{-14\sqrt{r}\log r} \cdot (1 - 1/r) \\ &\geq mn \cdot 2^{-16\sqrt{r}\log r}. \end{aligned}$$

since $r > 1$.

Thus there must be some rectangle R realizing this value of Q. Only $1/r^3$ fraction of such a rectangle can correspond to 1 entries of the matrix. We have shown:

Claim 2.18. *If at least half of the matrix is 0's, then there is a submatrix T of size at least $mn2^{-16\sqrt{r}\log r}$ such that the fraction of 1's in T is at most $1/r^3$.*

Call a row of T good if it contains at most $2/r^3$ fraction 1's. At least half the rows of T must be good, or else T would have more than $1/r^3$ fraction 1's overall. Let T' be the submatrix obtained by restricting T' to the good rows. Since $\mathsf{rank}(T') = r$, there are r rows $A_1, \ldots, A'_r$ that span all the rows of T'. Since each row A_i can have only $2/r^3$ fraction of 1's, at most $2/r^2 \leq 1/2$ fraction of the columns can contain a 1 in one of these r' rows.

Let T'' be the matrix obtained by restricting T' to the columns that do not have a 0 in the rows $A_1, \ldots, A_r$. Since every row of T'' must be a linear combination of rows that only have 0's in them, we have found a monochromatic matrix of size at least $mn2^{-16\sqrt{r}\log r}/4 \geq mn2^{-18\sqrt{r}\log r}$. This concludes the proof of Lemma 2.14.

Open Problem 2.19. *It would be very nice to find a more direct geometric argument to prove Lemma 2.14.*

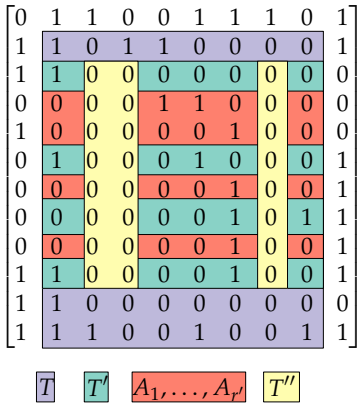


Figure 2.5: Going from a nearly monochromatic rectangle to a monochromatic rectangle.

Non-negative Rank and Covers

ANOTHER WAY TO MEASURE the complexity of a matrix is by measuring its *non-negative* rank. The non-negative rank of a $m \times n$ boolean matrix M is the smallest number r such that $M = AB$, where A, B are matrices with non-negative entries, such that A is an $m \times r$ matrix and B is an $r \times n$ matrix. Equivalently, it is the smallest number of non-negative rank 1 matrices that sum to M. Clearly, we have

Fact 2.20. $\mathsf{rank}(M) \leq \mathsf{rank}_+(M)$.

In general, $\mathsf{rank}(M)$ and $\mathsf{rank}_+(M)$ may be far apart. For example, given a set of numbers $X = \{x_1, \ldots, x_n\}$ of size n, consider the $n \times n$ matrix where $M_{i,j} = (x_i - x_j)^2 = x_i^2 + x_j^2 + 2x_ix_j$. Since M is the sum of three rank 1 matrices, $\mathsf{rank}(M) = 3$. On the other hand, we can show by induction on n that $\mathsf{rank}_+(M) \geq \log n$. Indeed, if $\mathsf{rank}_+(M) = k$, then there must be non-negative rank 1 matrices $R_1, \ldots, R_k$ such that $M = R_1 + \ldots + R_k$. Let the support of R_1 be the rectangle $A \times B$. Then we must have that either $|A| \leq |X|/2$ or $|B| \leq |X|/2$, or else there will be an element $x \in A \cap B$, but $M_{x,x} = 0$. Suppose $A \leq |X|/2$, and let M' be the submatrix that corresponds to the numbers of $X \setminus A$. Then we get that $\mathsf{rank}_+(M') \geq \log(n/2) = \log n - 1$, and so $k - 1 \geq \log n - 1$, proving that $k \geq \log n$.

If $M = R_1 + \ldots + R_r$, where $R_1, \ldots, R_r$ are rank 1 non-negative matrices, then the support of each matrix R_i must be a monochromatic rectangle in M with value 1. Thus, we get a 1-cover of the matrix:

Fact 2.21. *M always has a 1-cover with $\mathsf{rank}_+(M)$ rectangles.*

Moreover, one can prove that if a matrix has both small rank and a small cover, then there is a small communication protocot[11]:

[11] Lovász, 1990

Theorem 2.22. *If M has a 1-cover of size r, then there is a protocol computing M with $O(\log r \cdot \log \mathsf{rank}(M))$ bits of communication.*

Proof. The protocol is similar to the one used to prove Theorem 1.8. For every rectangle R in the cover, we can write

$$M = \begin{bmatrix} R & A \\ B & C \end{bmatrix},$$

and by (2.2), either

$$\mathsf{rank}\left(\begin{bmatrix} R & A \end{bmatrix}\right) \leq (\mathsf{rank}(M) - 3)/2, \tag{2.3}$$

or

$$\mathsf{rank}\left(\begin{bmatrix} R \\ B \end{bmatrix}\right) \leq (\mathsf{rank}(M) - 3)/2. \tag{2.4}$$

So in each step of the protocol, if Alice sees an R that is consistent with her input satisfying (2.3), she announces its name, or if Bob sees a rectangle R in the cover consistent with his input and satisfying (2.4), he announces its name. Both parties then restrict their attention to the appropriate submatrix, which reduces the rank of M by a factor of 2.

This can continue for at most $O(\log \mathsf{rank}(M))$ steps before the rank of the matrix becomes 1. On the other hand, if neither party finds such an R, then there must be be no such R that covers their input, so they can safely output a 1. □

Putting together Fact 2.21 and Theorem 2.22, we get

Corollary 2.23. *The communication of M is at most*

$$O(\log(\mathsf{rank}_+ M) \cdot \log \mathsf{rank}(M)) \leq O(\log^2 \mathsf{rank}_+(M)).$$

Exercise 2.1

Fix a function $f : X \times Y \to \{0,1\}$ with the property that in every row and column of the communication matrix M_f there are exactly t ones. Cover the zeros of M_f using $O(t(log|X| + log|Y|))$ monochromatic rectangles.

Exercise 2.2

Show that Nisan-Widgerson protocol (i.e., the proof of Lemma 2.13) goes though even if we weaken Lemma 2.14 to only guarantee a rectangle with rank at most $r/8$ (instead of rank at most one, or monochromatic).

Exercise 2.3

Recall that for a simple, undirected graph G the chromatic number $\chi(G)$ is the minimum number of colors needed to color the vertices of G so that no two adjacent vertices have the same color. Show that $\log \chi(G)$ is at most the deterministic communication complexity of G's adjacency matrix.

Exercise 2.4

For any symmetric matrix $M \in \{0,1\}^{n \times n}$ with ones in all diagonal entries, show that

$$2^c \geq \frac{n^2}{|M|},$$

where c is the deterministic communication complexity of M, and $|M|$ is the number of ones in M.

Exercise 2.5

For any boolean matrix M, define $\mathsf{rank}_2(M)$ to be the rank of M over $\mathbb{F}_2$, the field with two elements. Exhibit an explicit family of matrices $M \in \{0,1\}^{n \times n}$ with the property that $c \geq \mathsf{rank}_2(M)/10$, where c is the deterministic communication complexity of M. Conclude that this falsifies the analogue of log-rank conjecture for rank_2.

Exercise 2.6

Show that if f has fooling set of size s then $rk(M_f) \geq \sqrt{s}$. *Hint: tensor product.*

3
Randomized Protocols

Access to randomness is an enabling feature in many computational processes, and it is useful in communication protocols as well. We start with some examples of protocols where the use of randomness gives an advantage that cannot be matched by deterministic protocols, before defining randomized protocols formally and proving some basic facts about them. We do not discuss any lower bounds on randomized communication in this chapter. The lower bounds are proved in Chapter 5 and Chapter 6.

Equality Suppose Alice and Bob are each given access to n bit strings x, y, and want to know if these strings are the same or not (1.1). We have shown that at least $n + 1$ bits of communication are required if the communication is deterministic.

However, there is a simple randomized protocol. Alice and Bob sample a random function $h : \{0,1\}^n \to \{0,1\}^k$ and Alice sends $h(x)$ to Bob. If $x = y$, we will have that $h(x) = h(y)$. On the other hand, if $x \neq y$, the probability that $h(x) = h(y)$ is at most 2^{-k}. So the protocol can compute equality with a small probability of failure, even if the communication is a constant number of bits. This protocol may seem dissatisfying, because although the communication is small, the number of shared random bits required is very large (2^{nk}). However, there is a slightly more complicated protocol that uses very few random bits.

Alice and Bob agree on an error correcting code[1] $C : \{0,1\}^n \to \{0,1\}^m$. It can be shown that a random function is a code with high probability, but even explicit constructions of good codes are known. Given the code, Alice can pick k random coordinates of the code and send them to Bob. Bob will check whether these coordinates are consistent with his input. This takes $k \log n$ bits of communication, and now the probability of making an error is at most $2^{-\Omega(k)}$.

Input: Alice knows $x \in \{0,1\}^n$,
Bob knows $y \in \{0,1\}^n$.
Output: Whether or not $x = y$.

Alice and Bob sample a random function $h : \{0,1\}^n \to \{0,1\}^k$;
Alice sends Bob $h(x)$;
Bob announces whether $h(x) = h(y)$;

Figure 3.1: Public-coin Protocol for equality.

Input: Alice knows $x \in \{0,1\}^n$,
Bob knows $y \in \{0,1\}^n$.
Output: Whether or not $x = y$.

Alice and Bob agree on a good code $C : \{0,1\}^n \to \{0,1\}^m$;
Alice picks k coordinates $i_1, \ldots, i_k \in [m]$ at random;
Alice sends Bob $(i_1, C(x)_{i_1}), \ldots, (i_k, C(x)_{i_k})$;
Bob announces whether this is equal to $(i_1, C(y)_{i_1}), \ldots, (i_k, C(y)_{i_k})$;

Figure 3.2: Private-coin Protocol for equality.

[1] This is a function that maps n bits to $m = O(n)$ bits, such that if $x \neq y$, then $C(x)$ and $C(y)$ differ in $\Omega(m)$ coordinates.

Greater-than Suppose Alice and Bob are given numbers $x, y \in [n]$ and want to know which one is greater (1.3). We have seen that any deterministic protocol for this problem requires $\log n + 1$ bits of communication. However, there is a randomized protocol that requires only $O(\log \log n)$ bits of communication.

Here we describe a protocol that requires only $O(\log \log n \cdot \log \log \log n)$ communication. The inputs x, y can be encoded by ℓ-bit binary strings, where $\ell = \log n$. To determine whether $x \geq y$, it is enough to find the most significant bits in x, y where x, y are not the same. We use the randomized protocol for equality, and binary search, to achieve this. In the first step, Alice and Bob will use the protocol for equality to exchange k bits that determine whether the $\ell/2$ most significant bits of x and y are the same. If they are the same, the parties continue with the remaining bits. If not, the parties discard the second half of their strings. In this way, after $\log \ell$ steps, they find the first bit of difference in their inputs. We need to set $k = \log \log \ell$ for this process to work.

The protocol requiring $O(\log \log n)$ bits of communication is described in Exercise 3.1.

k-Disjointness Suppose Alice and Bob are given 2 sets $X, Y \subseteq [n]$ of size at most k, and want to know if these sets intersect or not. We used the rank method to argue that at least $\log \binom{n}{k} \approx k \log(n/k)$ bits of communication are required. Here we give a randomized protocol[2] that requires only $O(k)$ bits of communication[3], which is more efficient when $k \ll n$.

Alice and Bob sample a sequence of sets $R_1, R_2, \ldots \subseteq [n]$, uniformly at random. They exchange 2 bits to announce whether or not their sets are empty. If neither set is empty, Alice announces the index of the first set R_i that contains her set, and Bob announces the index of the first set R_j that contains his set. Now Alice can safely replace her set with $X \cap R_j$, and Bob can replace his set with $Y \cap R_i$. If at any point one of the parties is left with an empty set, they can safely conclude that the inputs were disjoint. We will argue that if the sets are disjoint, this process terminates after $O(k)$ bits of communication.

Assume that X, Y are disjoint. Let us start by analyzing the expected number of bits that will be communicated in the first step. We claim:

Claim 3.1. $\mathbb{E}[i] = 2^{|X|}$, $\mathbb{E}[j] = 2^{|Y|}$.

Proof. The probability that the first set of the sequence contains X is exactly $2^{-|X|}$. In the event that it does not contain X, we are picking the first set that contains X from the rest of the sequence.

Input: Alice knows $x \in \{0,1\}^\ell$,
Bob knows $y \in \{0,1\}^\ell$.
Output: Largest i such that $x_i \neq y_i$, if such an i exists.

Let $J = [n]$;
while $|J| > 1$ **do**
  Let J' be the first $|J|/2$ elements of J;
  Both parties use shared randomness to sample a random function $h : \{0,1\}^{|J'|} \to \{0,1\}^{2 \log \log \ell}$;
  Alice sends h evaluated on the bits in J', $h(x_{J'})$;
  Bob announces whether or not $h(x_{J'}) = h(y_{J'})$;
  if $h(x_{J'}) = h(y_{J'})$ **then**
    Alice and Bob replace $J = J \setminus J'$;
  else
    Alice and Bob replace $J = J'$;
  end
  Both parties announce x_J, y_J;
end

Figure 3.3: Public-coin protocol for greater than.

[2] Håstad and Wigderson, 2007

[3] Later, we show that $\Omega(k)$ bits are required.

Thus:

$$\begin{aligned}\mathbb{E}[i] &= 2^{-|X|} \cdot 1 + (1 - 2^{-|X|}) \cdot (\mathbb{E}[i] + 1) \\ \Rightarrow \mathbb{E}[i] &= 2^{|X|}.\end{aligned}$$

The bound on the expected value of j is the same. □

Since a number of size i can be communicated with at most $2 \log i$ bits, the number of bits communicated to transmit i, is at most[4]

$$\begin{aligned}\mathbb{E}[2 \log i] &\leq 2 \log \mathbb{E}[i] = 2|X| \\ \mathbb{E}[2 \log j] &\leq 2 \log \mathbb{E}[j] = 2|Y| \end{aligned} \tag{3.1}$$

Next we argue that when $X \cap Y = \emptyset$, the above communication process must terminate quickly.

Claim 3.2. *If $X \cap Y = \emptyset$, the expected number of bits communicated by the protocol is at most $6|X| + 6|Y| + 2$.*

Proof. Notice that as the protocol continues, the sets X, Y can only get smaller. So we can prove the bound by induction on the size of the sets X, Y. For the base, case, if X or Y is empty, at most $2 \leq 6(|X| + |Y|) + 2$ bits are communicated. If both X and Y are non-empty, (3.1) shows that the expected number of bits communicated in the first step is $2 + 2|X| + 2|Y|$. By induction, the expected number of bits communicated in the rest of the protocol is $\mathbb{E}\left[6(|X \cap R_j| + |Y \cap R_i|)\right] + 2$. But observe that since X, Y are assumed to be disjoint, $\mathbb{E}\left[|X \cap R_j|\right] = |X|/2$, and $\mathbb{E}[|Y \cap R_i|] = |Y|/2$. Thus the total number of bits communicated is

$$\begin{aligned}&2 + 2|X| + 2|Y| + (6/2)|X| + (6/2)|Y| + 2 \\ &= 6|X| + 6|Y| + 2 - (|X| + |Y| - 2) \\ &\leq 6|X| + 6|Y| + 2,\end{aligned}$$

as required. □

Claim 3.2 means that if X, Y are disjoint, the expected number of step taken by the process above is $6|X| + 6|Y| + 2$. By Markov's inequality, the probability that the protocol communicates more than $10 \cdot (6|X| + 6|Y| + 2)$ bits is at most $1/10$. Thus if we run this process until $120k + 20$ bits have been communicated, the probability of making an error is at most $1/10$.

Input: Alice knows $X \subseteq [n]$, Bob knows $Y \subseteq [n]$.
Output: Whether or not $X \cap Y = \emptyset$

while $|X| > 1$ *and* $|Y| > 1$ *and at most* $120k + 20$ *bits have been communicated so far* **do**
  Alice and Bob use shared randomness to sample random subsets $R_1, R_2, \ldots \subseteq [n]$;
  Alice sends Bob the smallest i such that $X \subseteq R_i$;
  Bob sends Alice the smallest j such that $Y \subseteq R_j$;
  Alice replaces $X = X \cap R_j$;
  Bob replaces $Y = Y \cap R_i$;
end
if $X = \emptyset$ *or* $Y = \emptyset$ **then**
  Alice and Bob conclude that the sets were disjoint;
else
  Alice and Bob conclude that the sets were intersecting;
end

Figure 3.4: Public-coin protocol for k-disjointness.

[4] since log is concave

Variants of Randomized Protocols

A RANDOMIZED PROTOCOL is a deterministic protocol where each party has access to a random string, in addition to the inputs to

the protocol. The random string is sampled independently from the inputs, but may have an arbitrary distribution. We say that the protocol uses *public coins* if all parties have access to a common shared random string. We say that the protocol uses *private coins* if each party samples an independent random string. Every private coin protocol can be simulated by a public coin protocol. There are at least two ways to quantify the errors made by a protocol:

We shall soon see a partial converse: every public coin protocol can be simulated with private coins, with a small increase in the communication

Worst-case We say that a randomized protocol has error ϵ in the *worst-case* if the probability that the protocol makes an error is at most ϵ on every input.

If a randomized protocol never makes an error, we can fix the randomness to obtain a deterministic protocol that is always correct.

Average-case Given a distribution on inputs μ, we say that the 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.

The worst-case error is ϵ if and only if the error is ϵ under *every* distribution on inputs.

When a protocol has error $\epsilon < 1/2$ in the worst case, we can run it several times and take the majority output to reduce the error. If we repeat the protocol k times, and output the most frequent output in all of the runs, there will be an error in the output only if at least $k/2$ of the runs computed the wrong answer. By the Chernoff bound, the probability of error is at most $2^{-\Omega(k(1/2-\epsilon)^2)}$.

If a randomized protocol makes no errors, we can fix the randomness to obtain a deterministic protocol that comptes the function.

Worst-case and average-case errors are related by via Yao's minimax principle:

Theorem 3.3. *The communication complexity of computing a function g in the worst-case with error at most ϵ is equal to the maximum, over all distributions μ, of the communication complexity of computing g with error at most ϵ with respect to μ.*

Theorem 3.3 can be proved by appealing to von Neumann's minimax principle[5]:

[5] von Neumann, 1928

The minimax principle can also be seen as a consequence of linear programming duality.

Theorem 3.4. *Let M be an $m \times n$ matrix. Then*

$$\min_{x \geq 0} \max_{y \geq 0} xMy = \max_{y \geq 0} \min_{x \geq 0} xMy,$$

where x is a $1 \times m$ row vector with $\sum_i x_i = 1$, and y is a $n \times 1$ column vector with $\sum_j y_j = 1$.

Let us see how to prove Theorem 3.3. One direction is easy: if there is a protocol that computes g with error ϵ in the worst case, then the same protocol must compute g with error ϵ in the average case, no matter what the input distribution is.

Now suppose we know that for every distribution μ, there is a c-bit protocol that computes g with error ϵ in the average case. Consider the boolean matrix M where every row corresponds to a deterministic communication protocol, and every column corresponds

to an input to the protocol, such that

$$M_{i,j} = \begin{cases} 1 & \text{if protocol } i \text{ computes } g \text{ correctly on input } j, \\ 0 & \text{otherwise.} \end{cases}$$

A distribution on the inputs corresponds to a choice of $y \geq 0$ such that $\sum_j y_j = 1$. Since a randomized protocol can be thought of as a distribution on deterministic protocols, a randomized protocol corresponds to a choice of $x \geq 0$ such that $\sum_i x_i = 1$. The probability that a fixed randomized protocol x makes an error when the inputs come from the distribution y is exactly xMy. Thus $\max_{y \geq 0} \min_{x \geq 0} xMy \leq \epsilon$. Theorem 3.4 implies that $\min_{x \geq 0} \max_{y \geq 0} xMy \leq \epsilon$ as well, which is exactly what we want to prove. There is a fixed randomized protocol that has error at most ϵ under every distribution on inputs.

Public Coins vs Private Coins

While every private coin protocol can be simulated by a public coin protocol, can every private coin protocol be simulated by a public coin protocol? Such a simulation is possible[6], up to a small additive loss in communication:

[6] Newman, 1991

Theorem 3.5. *If $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ can be computed with c bits of communication, and error ϵ in the worst case, then it can be computed by a private coin protocol with $c + \log(n/\epsilon^2) + O(1)$ bits of communication, and error 2ϵ in the worst case.*

It is known that computing whether or not two n-bit strings are equal requires $\Omega(\log n)$ bits of communication if only private coins are used. This shows that Theorem 3.5 is tight.

Proof. We use the probabilistic method to find the required private coin protocol. Let us pick t independent random strings, each of which can be used as the randomness for the given private-coin protocol.

For any fixed input, some of these t random strings lead to the public coin protocol computing the right answer, and some of the lead to the protocol computing the wrong answer. By the Chernoff bound, the probability that $1 - 2\epsilon$ fraction of the t strings lead to the wrong answer is at most $2^{-\Omega(\epsilon^2 t)}$. We set $t = O(2n/\epsilon^2)$ to be large enough so that this probability is less than 2^{-2n}. Then by the union bound, we get that the probability that $2\epsilon t$ of these strings give the wrong answer for *any* input is less than 1. Thus there must be some fixed strings with this property.

The private coin protocol is now simple. Alice samples one of the t strings and sends its index to Bob, which takes at most $\log(n/\epsilon^2) + O(1)$ bits. Alice and Bob then run the original public coin protocol. □

Nearly Monochromatic Rectangles

MONOCHROMATIC RECTANGLES PROVED to be a very useful concept to understand deterministic protocols. A similar role is played by *nearly monochromatic* rectangles when trying to understand randomized protocols.

Definition 3.6. *Given a distribution μ on inputs, we say that a rectangle R has bias $(1-\epsilon)$ under a function g if there is a constant b so that*

$$\Pr_{\mu}[g(x,y) = b | (x,y) \in R] \geq 1 - \epsilon.$$

Such a rectangle is called $(1-\epsilon)$-monochromatic.

We would like to claim that a protocol with small error ϵ induces a partition of the space into nearly monochromatic rectangles. That is not quite true, but we can claim that the average rectangle must be very close to being monochromatic:

Theorem 3.7. *If there is a c-bit protocol that computes g with error ϵ under a distribution μ, then you can partition the inputs into 2^c rectangles, such that the average bias of a random rectangle from the partition is at least $1-\epsilon$.*

Applying Markov's inequality to this average gives that there must be many large nearly monochromatic rectangles:

Theorem 3.8. *If there is a c-bit protocol that computes g with error ϵ under μ, then for every ℓ, there are disjoint $(1-\ell\epsilon)$-monochromatic rectangles $R_1, R_2, \ldots, R_{2^c}$ such that $\Pr_{\mu}[(x,y) \in \cup_i R_i] \geq 1 - 1/\ell$.*

Theorem 3.8 will be instrumental to prove lower bounds on randomized protocols

Proof. Since we can always fix the randomness of the protocol in the best way, we can assume that the protocol is deterministic. By Theorem 1.7, we know that the protocol induces a partition of the space into 2^c rectangles. Consider all the rectangles that are not $(1-\ell\epsilon)$-monochromatic. If the probability that the input lands in one of these rectangles is bigger than $1/\ell$, the error of the protocol will be bigger than ϵ. Thus the inputs must land in a $(1-\ell\epsilon)$-monochromatic rectangle with probability at least $1 - 1/\ell$. □

As a corollary, we get:

Corollary 3.9. *If there is a c-bit protocol that computes g with error ϵ under μ, then for every ℓ, there is a $(1-\ell\epsilon)$-monochromatic rectangle of density at least $2^{-c}(1-1/\ell)$.*

Exercise 3.1

In this exercise we will develop a randomized protocol for greater-than that requires only $O(\log \log n)$ bits of communication. Let $x, y \in \{0,1\}^\ell$ be two strings. Alice and Bob want to find the smallest i such that $x_i \neq y_i$.

Exercise 3.2

In this exercise, we design a randomized protocol for finding the first difference between two n-bit strings. Alice and Bob are given n bit strings $x \neq y$ and want to find the smallest i such that $x_i \neq y_i$. In class we saw how to accomplish this using $O(\log n \log \log n)$ bits of communication. Here we do it with $O(\log n)$ bits of communication.

Define a rooted tree as follows. Every vertex will correspond to an interval of coordinates from $[n]$. The root corresponds to the interval $I = [n]$. Every internal vertex corresponding to the interval I will have two children, the left child corresponding to the first half of I and the right child corresponding to the right half of I. This defines a tree of depth $\log n$, where the leaves correspond to intervals of size 1 (i.e. coordinates) of the input. At each leaf, attach a path of length $3 \log n$. Every vertex of this path represents the same interval of size 1. The depth of the tree is now $4 \log n$.

1. Fill in the details of the following protocol. Prove an upper bound on the expected number of bits communicated and a lower bound on the success probability.

   The players use their inputs and hashing to start at the root of the tree and try to navigate to the smallest interval that contains the index i that they seek. In each step, the players will either move to a parent or a child of the node that they are at. When the players are at a vertex that corresponds to the interval I, they should first exchange $O(1)$ hash bits to confirm that the first difference does lie in I. If this hash shows that the first difference does not lie in I, they should move to the parent of the current node. Otherwise, they exchange $O(1)$ hash bits and use this to decide on which child of the current node to move to. Once the players reach the nodes of the tree that correspond to intervals of size 1, they use their hashes to either move to a parent or child.

2. Argue that as long as the number of nodes where the protocol made the right choice exceeds the number of nodes where the players made the wrong choice by $\log n$, the protocol you defined does succeed in computing i.

3. Use the Chernoff bound to argue that the number of hashes that gives the right answer is high enough to ensure that the protocol

succeeds with high probability on any input.

Exercise 3.3

Show that if the inputs to greater-than are sampled uniformly and independently, then there is a protocol that communicates only $O(\log(1/\epsilon))$ bits and has error at most ϵ under this distribution.

4
Numbers On Foreheads

The number-on-forehead model[1] of communication is one way to generalize the case of two party communication to the multiparty setting. There are k parties communicating, and the i'th party has an input drawn from the set $\mathcal{X}_i$ written *on their forehead*. Each party can see all of the inputs except the one that is written on their forehead. The fact that each party can see most of the inputs means that parties do not need to communicate as much. It also means that proving lower bounds against this model is particularly hard. Indeed, we do not yet known how to prove optimal lower bounds in the model of computation, in stark contrast to models where the inputs are completely private.

[1] Chandra et al., 1983

When there are only $k = 2$ parties, this model is identical to the model of 2 party communication.

Moreover, optimal lower bounds in this model would have very interesting consequences to the study of circuit complexity.

We start with some examples of clever number-on-forehead protocols.

Equality We have seen that the best protocol for deterministically computing equality in the 2 party setting involves one of the parties revealing their entire input. Suppose 3 parties each have an n bit string written on their foreheads. Then there is a trivial protocol for computing whether all three strings are the same: Alice announces whether or not Bob and Charlie's strings are the same, and Bob announces whether or not Alice and Charlie's strings are the same.

Intersection size Suppose there are k parties, and the i'th party has a subset $X_i \subseteq [n]$ on their forehead. The parties want to compute the size of the intersection $\cap_i X_i$. We shall describe a protocol[2] that requires only $O(k^4 n/2^k)$ bits of communication.

A protocol solving this problem would compute both the disjointness function and the inner product function.

[2] Grolmusz, 1998; and Babai et al., 2003

In Chapter 5, we prove that at least $n/4^k$ bits of communication are required.

We start by describing a protocol that requires only $k^2 \log n$ bits of communication, as long as $n < \binom{k}{k/2}$. It is helpful to think of the input as a $k \times n$ boolean matrix. Each of the parties knows all but one row of this matrix, and they wish to compute the number of all 1's columns. Let $C_{i,j}$ denote the number of

columns containing j 1's that are visible to the i'th party. The parties compute and announce the values of $C_{i,j}$, for each i, j. The communication of the protocol is at most $k^2 \log n$ bits. Let A_j denote the actual number of columns with j ones in them.

Claim 4.1. *If there are two valid solutions $A_k, \ldots, A_0$ and $A'_k, \ldots, A'_0$ that are both consistent with the values $C_{i,j}$, then either $A'_k = A_k$, or for each j, $|A_j - A'_j| \geq \binom{k}{j}$.*

Proof. Suppose $A'_k \neq A_k$. Then $|A_k - A'_k| \geq 1 = \binom{k}{k}$. We proceed by induction. Since a column of weight j is observed as having weight $j-1$ by j parties, and having weight j by $k-j$ parties, we have:

$$\begin{aligned}(k-j)A_j + (j+1)A_{j+1} = \sum_{i=1}^{k} C_{i,j} &= (k-j)A'_j + (j+1)A'_{j+1} \\ \Rightarrow |A_j - A'_j| &\geq \left(\frac{j+1}{k-j}\right) |A_{j+1} - A'_{j+1}| \\ &\geq \left(\frac{j+1}{k-j}\right)\binom{k}{j+1} = \binom{k}{j},\end{aligned}$$

as required. □

Claim 4.1 implies that if $n < \binom{k}{k/2} \approx 2^k/\sqrt{k}$, then there can only be one solution for A_k, since $|A_{k/2} - A'_{k/2}|$ cannot exceed n. To get the final protocol, the parties divide the columns of the matrix into blocks of size at most $\binom{k}{k/2}$, and compute A_k for each such block separately. The total communication is then $\frac{n \cdot k^2 \log \binom{k}{k/2}}{\binom{k}{k/2}} = O(k^4 n/2^k)$.

Exactly n Suppose 3 parties each have a number from $[n]$ written on their forehead, and want to know whether these numbers sum to n or not.A trivial protocol is for one of the parties to announce one of the numbers she sees, which takes $\log n$ bits. Here we use ideas of Behrend[3] to show that one can do it with just $O(\sqrt{\log n})$ bits of communication. Behrend's ideas lead to a coloring of the integers that avoids monochromatic three-term arithmetic progressions:

The randomized communication of exactly n is only a constant, since Alice can use the randomized protocol for equality to check that her forehead is equal to what it needs to be for all numbers to sum to n.

[3] Behrend, 1946

Theorem 4.2. *One can color the set $[n]$ with $2^{O(\sqrt{\log n})}$ colors, such that for any $a, b \in [n]$, if the numbers $a, a+b, a+2b$ are all in $[n]$, they cannot have the same color.*

Proof. For parameters d, r, with $d^r = n$, we write each number $x \in [n]$ in base d, using at most r digits: we express $x = \sum_{i=0}^{r-1} x_i d^i$, where $x_i \in [d-1]$. One can interpret $x \in [n]$ as a vector $v(x) \in \mathbb{R}^r$

whose i'th coordinate is the i'th digit of x. We approximate each of these vectors using a vector where every coordinate is off by at most $d/4$: let $w(x)$ be the vector where the i'th coordinate is the largest number of the form $jd/4$ such that $jd/4 \leq x_i$ and j is an integer.

Color each number $x \in [n]$ by the vector $w(x)$ and the integer $\|v(x)\|^2 = \sum_{i=0}^{r} x_i^2$. The number of choices for $w(x)$ is at most $2^{O(r)}$, and the number of possible values for $\|v(x)\|^2$ is at most $O(rd^2)$, so the total number of possible colors is at most $2^{O(r+\log d)}$. Setting $r = \sqrt{\log n}, d = 2^{\sqrt{\log n}}$ gives the required bound.

It only remains to check that the coloring avoids arithmetic progressions. Suppose $a, b \in [n]$ are such that $a, a+b, a+2b$ all get the same color. Then we must have $\|v(a)\| = \|v(a+b)\| = \|v(a+2b)\|$, so the three vectors $v(a), v(a+b), v(a+2b)$ all lie on the surface of a sphere. We will get a contradiction by proving that $v(a+b) = \frac{v(a)+v(a+2b)}{2}$, so these three vectors are collinear, and no three vectors on the sphere can be collinear.

Let $W(x) = \sum_{i=0}^{r} w(x)_i d^i$. Note that by assumption, $W(a) = W(a+b) = W(a+2b)$. Then we have

$$\begin{aligned} a + 2b + a &= 2(a+b) \\ \Rightarrow a + 2b - W(a+2b) + a - W(a) &= 2(a+b-W(a+b)). \end{aligned}$$

Now observe that the base d representation of $x - W(x)$ is exactly $v(x) - w(x)$, and in this vector, every coordinate is at most $d/4$. This means that the base d representation of $a + 2b - W(a+2b) + a - W(a)$ is exactly $v(a+2b) - w(a+2b) + v(a) - w(a)$. Thus, we conclude

$$\begin{aligned} v(a+2b) - w(a+2b) + v(a) - w(a) &= 2(v(a+b) - w(a+b)) \\ \Rightarrow v(a+2b) + v(a) &= 2v(a+b), \end{aligned}$$

as required.

□

Now we show how the coloring of Theorem 4.2 can be used to get a protocol for the exactly n problem with communication $O(\sqrt{\log n})$. Suppose the three inputs are x, y, z. Alice computes the number $x' = n - y - z$, and Bob computes $y' = n - x - z$. Define $\ell(x, y) = x + 2y$. Then we see that if $x + y + z = n$, then $x' = x$ and $y' = y$. On the other hand, if $x + y + z \neq n$, then $x - x' = y - y' \neq 0$, and $\ell(x, y), \ell(x', y), \ell(x, y')$ form an arithmetic progression. So Alice simply announces the color of $\ell(x', y)$ in the coloring promised by Theorem 4.2. Bob and Charlie just check that this color is the same as the color of $\ell(x, y)$ and $\ell(x, y')$. The

communication required is $O(\sqrt{\log n})$ since there are at most $2^{O(\sqrt{\log n})}$ colors in the coloring.

Cylinder Intersections

THE BASIC BUILDING BLOCKS of protocols in the number-on-forehead model are *cylinder intersections*. They play the same role that rectangles play in the case that the number of parties is 2. Any set $S \subseteq \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ can be described using its characteristic function:

$$\chi_S(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } (x_1, \ldots, x_k) \in S, \\ 0 & \text{otherwise.} \end{cases}$$

We can then define cylinder intersections as:

Definition 4.3. *$S \subseteq \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ is called a* cylinder *if χ_S does not depend on one of its inputs. S is called a* cylinder intersection *if it can be expressed as an intersection of cylinders.*

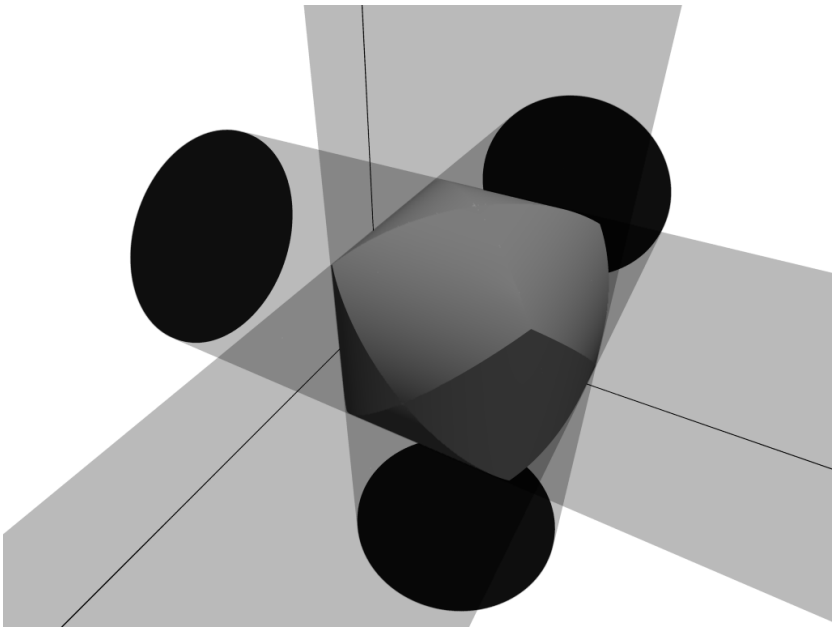


Figure 4.1: A cylinder intersection.

Figure 4.2: Figure 4.1 viewed from above.

If S is a cylinder intersection, we can always express

$$\chi_S(x_1, \ldots, x_k) = \prod_{i=1}^{k} \chi_i(x_1, \ldots, x_k),$$

where χ_i is a boolean function that does not depend on the i'th input.

When $k = 2$, cylinder intersections are the same as rectangles. However, when $k > 2$, they are much more complicated to understand than rectangles.

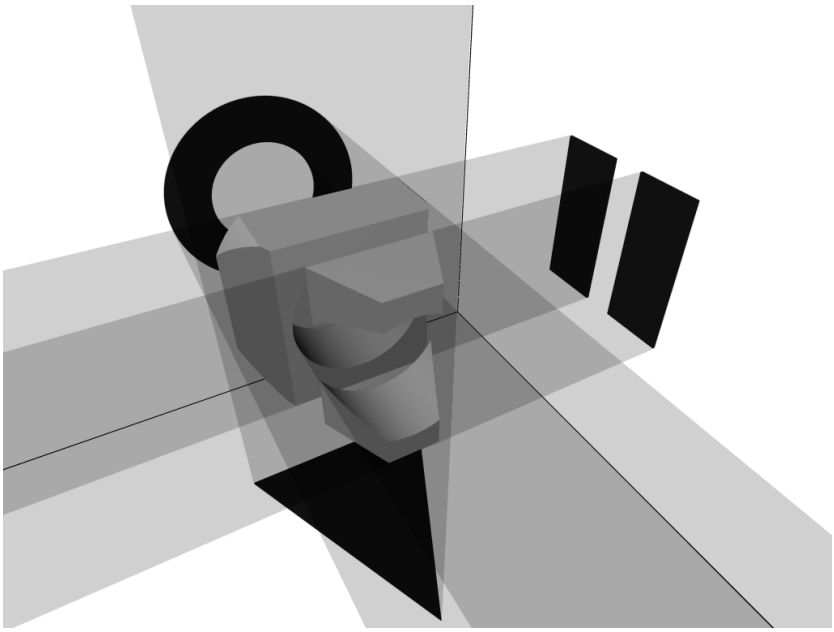


Figure 4.3: A cylinder intersection.

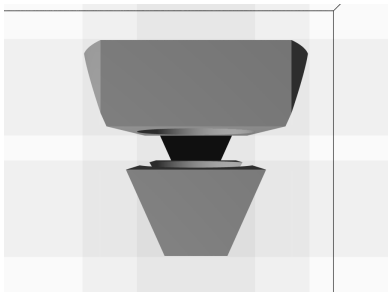


Figure 4.4: Figure 4.3 viewed from above.

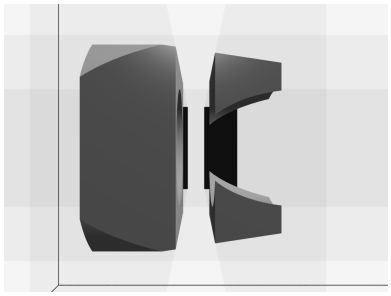


Figure 4.5: Figure 4.3 viewed from the left.

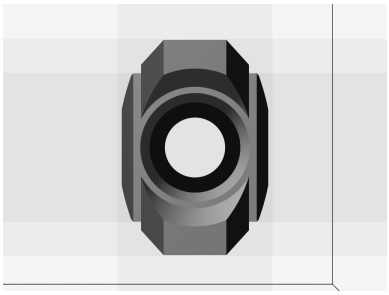


Figure 4.6: Figure 4.3 viewed from the right.

Fact 4.4. *The intersection of two cylinder intersections is also a cylinder intersection.*

Just as for rectangles, we say that a cylinder intersection is *monochromatic* with respect to a function g, if $g(x) = g(y)$ for every two inputs x, y in the cylinder intersection. In analogy with the 2 party case, we have the following theorem:

Theorem 4.5. *If the communication complexity of $g : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \{0,1\}$ is c, then $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ can be partitioned into at most 2^c monochromatic cylinder intersections.*

Indeed, for every outcome of the protocol m, it is easy to verify that the set of inputs that are consistent with that outcome form a cylinder intersection. Moreover, in analogy with Theorem 1.8, one can show[4] a small cover by monochromatic cylinder intersections leads to a protocol computing the same function.

[4] See Exercise **??**.

Lower bounds from Ramsey Theory

One way to prove lower bounds on protocols in the number-on-forehead model is by appealing to arguments from Ramsey Theory.

Let us consider the *Exactly n* problem: Alice, Bob and Charlie are each given a number from $[n]$, written on their foreheads, and want to know if their numbers sum to n. We have shown that there is a

In Chapter 5 we discuss the discrepancy method, which leads to the strongest known lower bounds in the number-on-forehead model.

protocol that computes this function using $O(\sqrt{\log n})$ bits of communication. Here we show that $\Omega(\log\log\log n)$ bits of communication are required[5].

[5] Chandra et al., 1983

Let c_n is the communication of the exactly n problem. Three points of $[n] \times [n]$ form a *corner* if they are of the form $(x, y), (x + d, y), (x, y + d)$. A *coloring* of $[n] \times [n]$ with 2^c colors is a function $g : [n] \times [n] \to [C]$. We say that the coloring avoids monochromatic corners if there is no corner with $g(x, y) = g(x + d, y) = g(x, y + d)$. Let C_n be the minimum number of colors required to avoid monochromatic corners in any coloring of $[n] \times [n]$. We claim that C_n essentially captures the value of c_n:

Claim 4.6. *$c_n \leq 2 + \log C_n$, and $c_n \geq \log C_{n/3}$.*

Proof. For the first inequality, suppose there is a coloring with C colors that avoids monochromatic corners. As in the protocol we saw, Alice can compute $x' = n - y - z$, and Bob can compute $y' = n - x - z$. Alice will then announce the color of (x', y), and Bob and Charlie will say whether this color is consistent with their inputs. These three points form a corner, since $x' - x = n - x - y - x = y' - y$. So if they all have the same color, they must all be the same point.

To prove the second inequality, suppose there is a protocol computing the exactly $n/2$ problem with c bits of communication. Then by Theorem 4.5, every input can be colored by one of 2^c colors that is the name of the corresponding cylinder intersection. This induces a coloring of $[n/3] \times [n/3]$: color (x, y) by the name of the cylinder intersection containing the point $(x, y, n - x - y)$. We claim that this coloring avoid monochromatic corners. Indeed, if $(x, y), (x + d, y), (x, y + d)$ is a monochromatic corner, then $(x, y, n - x - y), (x + d, y, n - x - y - d), (x, y + d, n - x - y - d)$ must all belong to the same cylinder intersection. But then $(x, y, n - x - y - d)$ must *also* be in the same cylinder intersection, since it agrees with each of the three points in two coordinates. That contradicts the correctness of the protocol, since the sum of the points $(x, y, n - x - y)$ is n, and the sum of $(x, y, n - x - y - d)$ is $n - d$. □

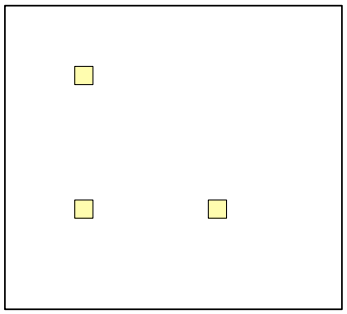


Figure 4.7: A monochromatic corner.

Next we prove[6]:

[6] Graham, 1980; and Graham et al., 1980

Theorem 4.7. $C_n \geq \Omega\left(\frac{\log\log n}{\log\log\log n}\right)$.

Proof. The proof will proceed by induction on the number of colors, but using a stronger structure than monochromatic corners.

A *rainbow-corner* with r colors and center (x, y) is specified by a set of r colors, and numbers $d_1, \ldots, d_{r-1}$, such that $(x + d_i, y)$ and $(x, y + d_i)$ are both colored using the i'th color, and (x, y) is colored by the r'th color.

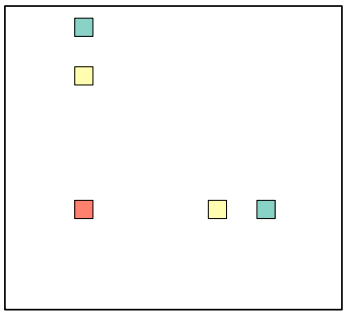


Figure 4.8: A rainbow-corner.

We shall prove by induction that as long as $C > 3$, if $n \geq 2^{C^{2r}}$, then any coloring of $[n] \times [n]$ with C colors must contain either a monochromatic corner, or a rainbow corner with r colors. When $r = C + 1$, this means that if $n \geq 2^{C^{2(C+1)}}$, $[n] \times [n]$ must contain a monochromatic corner, proving that $C_n \geq \Omega\left(\frac{\log\log n}{\log\log\log n}\right)$.

$2(C_n + 1)\log C_n \geq \log\log n$, which cannot happen if $C_n = o(\log\log n / \log\log\log n)$.

For the base case, when $r = 2, n = 4$, two of the points of the type $(x, n - x)$ must have the same color. If $(x, n - x)$ and $(x', n - x')$ have the same color, with $x > x'$, then $(x', n - x), (x, n - x), (x', n - x')$ are either a monochromatic corner, or a rainbow corner with 2 colors.

For the inductive step, if $n = 2^{C^{2r}}$, n contains $m = 2^{C^{2r} - C^{2(r-1)}}$ consecutive disjoint intervals: $[n] = I_1 \cup I_2 \cup \ldots \cup I_m$, each of size exactly $2^{C^{2(r-1)}}$. By induction, each of the sets $I_j \times I_j$ must have either a monochromatic corner, or a rainbow-corner with $r - 1$ colors. If one of them has a monochromatic corner, we are done, so suppose they all have rainbow-corners with $r - 1$ colors. Since a rainbow corner is specified by choosing the center, choosing the colors and choosing the offsets for each color, there are at most $(2^{C^{2(r-1)}})^2 \cdot 2^C \cdot (2^{C^{2(r-1)}})^C$ rainbow-corners in each interval. This number is at most $2^{2C^{2(r-1)} + C + C^{2r-1}} < 2^{C^{2r} - C^{2(r-1)}} = m$, so there must be $j < j'$ that have exactly the *same* rainbow corner with the same coloring. Then we see (Figure 4.9) that these two rainbow corners must induce a monochromatic corner centered in the box $I_j \times I_{j'}$, or a rainbow corner with r colors. □

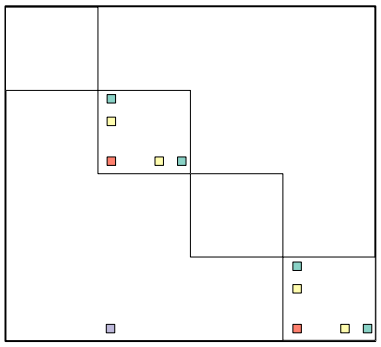


Figure 4.9: A rainbow-corner induced by two smaller rainbow corners.

Exercise 4.1

Define the generalized inner product function GIP as follows. Here each of the k players is given a binary string $x_i \in \{0,1\}^n$. They want to compute $\mathsf{GIP}(x) = \sum_{j=1}^{n} \prod_{i=1}^{k} x_{i,j} \pmod 2$.

This exercise outlines a number-on-forehead GIP protocol using $O(n/2^k + k)$ bits. It will be convenient to think about the input X as a $k \times n$ matrix with rows corresponding to $x_1, \ldots, x_k$.

Each vector x_i can be interpretted as a subset of $[n]$. Our set intersection protocol computes GIP with $O(k^4 n/2^k)$ bits. This improved protocol, by A. Chattopadhyay, slightly improves a famous protocol of V. Grolmusz.

- Fix $z \in \{0,1\}^n$. Assume the first t coordinates of z are ones and the rest are zeros. For $\ell \in \{0, 1, \ldots, k-1\}$ define c_ℓ as the number of columns in X with ℓ ones, followed by either a one or zero, followed by $k - \ell - 1$ zeros. Note that $\mathsf{GIP}(x) = c_k \pmod 2$. Find a protocol to compute $\mathsf{GIP}(x)$ using $O(k)$ bits *assuming the players know* $c_t \pmod 2$.

- Exhibit an overall protocol for GIP by showing that the players can agree upon a vector z and communicate to determine $c_t \pmod 2$ using $O(n/2^k + k)$ bits.

One can extend this protocol to compute any function of the number of all ones rows using $O(n/2^k + k \log n)$ bits.

Exercise 4.2

Given a function $g : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, recall that we define g^r to be the function that computes r copies of g. This exercise explores, in the number-on-forehead model, what we can say about the communication required by g^r, knowing the communication complexity of g. The approach taken in the proof of Theorem 1.29 does not work because cylinder intersections do not tensorize nicely like rectangles do. Fortunately, we can appeal to a powerful result from Ramsey theory called the *Hales-Jewett Theorem*[7] to prove that the communication complexity of g^rmust increase as r increases.

[7] Hales and Jewett, 1963

For an arbitrary set S, the Hales-Jewett Theorem gives insight into the structure of the cartesian product $S^n = S \times S \times \cdots S$ as n grows large. For the precise statement we need the notion of a combinatorial line. The *combinatorial line* specified by a nonempty set of indices $I \subseteq [n]$ and a vector $v \in S^n$ is the set $\{x \in S^n : x_i = v_i, \text{if } i \notin I, \text{and for every } i, j \in I, x_i = x_j\}$. For example, when $S = [3]$ and $n = 4$ then the set $\{1132, 2232, 3332\}$ is a combinatorial line with $I = \{1,2\}$ and $v_3 = 3, v_4 = 2$.

Given a set S and a number t, the Hales-Jewett theorem says that as long as n is large enough, then any coloring of S^n with t colors must contain a monochromatic combinatorial line.

- Set $S = g^{-1}(0)$, so $S \subseteq \mathcal{X} \times \mathcal{Y}$, a subset of the domain of g. Use the Hales-Jewett theorem to argue that there exists an n large enough such that the following holds. Any protocol correctly computing $\wedge_{i=1}^n g(x_i)$ induces a coloring of the domain of g^r that can be used to get a single monochromatic cylinder intersection containing all the points of S.
- Assume that the communication complexity of g is strictly greater than the number of players (that is, $c > k$). Define c_n to be communication required to compute $\wedge_{i=1}^n g(x_i)$. Prove that

$$\lim_{n \to \infty} c_n = \infty.$$

Exercise 4.3

A three player NOF puzzle demonstrates that unexpected efficiency is sometimes possible.

Inputs: Alice has a number $i \in [n]$ on her forehead, Bob has a number $j \in [n]$ on his forehead, and Charlie has a string $x \in \{0,1\}^n$ on his forehead.

Output: On input (i, j, x) the goal is for Charlie to output the bit x_k where $k = i + j \pmod n$.

Question: Find a deterministic protocol such that Bob sends one bit to Charlie, and Alice sends $\lfloor \frac{n}{2} \rfloor$ bits to Charlie. Alice and Bob must

each send Charlie their message *simultaneously*; then Charlie should be able to output the correct answer.

Exercise 4.4

Show that any degree d polynomial over $\mathbb{F}_2$ over the variables $x_1, \ldots, x_n$ can be computed by $d+1$ players with $O(d)$ bits of Number-On-Forehead communication, for any partition of the inputs where each party has $n/(d+1)$ bits on their forehead. (You may assume $d+1$ divides n exactly).

5
Discrepancy

The discrepancy method is a powerful way to prove lower bounds on communication complexity. We will use it here to prove optimal lower bounds on randomized protocols, and tight lower bounds in the number-on-forehead model.

One reason why our previous approaches were insufficient to prove lower bounds on randomized protocols is that the existence of a randomized protocol only guarantees a partition of the space into *nearly* monochromatic rectangles, rather than completely monochromatic rectangles. In order to work with nearly monochromatic rectangles, we need to work with a quantity that is sensitive to the bias of a rectangle. Let g be a boolean function, and let χ_S be the characteristic function of the set S. Then we define the *discrepancy* of S with respect to g to be

$$\left|\mathbb{E}\left[\chi_S(x)\cdot(-1)^{g(x)}\right]\right|,$$

where the expectation is taken over a random input x. A large, nearly monochromatic rectangle (or cylinder intersection) must have high discrepancy:

Fact 5.1. *If R is a $(1-\epsilon)$-monochromatic rectangle (or cylinder intersection) of density δ, then the discrepancy of R must be at least $(1-2\epsilon)\delta$.*

Proof. Only points inside R contribute to its discrepancy. Since $(1-\epsilon)$ fraction of these points have the same value under g, the discrepancy is at least $\delta(1-\epsilon-\epsilon) = \delta(1-2\epsilon)$. □

Let $\pi(x,y)$ denote the output of a protocol π with c bits of communication and error ϵ. Let $R_1,\ldots,R_t$ be the rectangles induced by the

protocol. Then we have

$$\begin{aligned}1-2\epsilon &= \mathop{\mathbb{E}}_{x,y}\left[(-1)^{\pi(x,y)+g(x,y)}\right] \\ &= \mathop{\mathbb{E}}_{x,y}\left[(-1)^{\pi(x,y)}\cdot(-1)^{g(x.y)}\right] \\ &\leq \mathop{\mathbb{E}}_{x,y}\left[\left(\sum_{i=1}^{t}\chi_{R_i}(x,y)\cdot o(R_i)\right)\cdot(-1)^{g(x,y)}\right],\end{aligned}$$

where here $o(R_i)$ is -1 if the protocol outputs 1 in R_i, and it is 1 if the protocol outputs O. We can continue to bound:

$$\begin{aligned}1-2\epsilon &\leq \sum_{i=1}^{t}\left|\mathop{\mathbb{E}}_{x,y}\left[\chi_{R_i}(x,y)\cdot(-1)^{g(x,y)}\right]\right| \\ &\leq 2^c\cdot\max_R\left|\mathop{\mathbb{E}}_{x,y}\left[\chi_R(x,y)\cdot(-1)^{g(x,y)}\right]\right|,\end{aligned}$$

where the maximum is taken over all choices of rectangles. Rearranging, we get

$$2^c \geq \frac{1-2\epsilon}{\max_R\left|\mathbb{E}_{x,y}\left[\chi_R(x,y)\cdot(-1)^{g(x,y)}\right]\right|}.$$

The same calculation also works in the case of cylinder intersections. We have shown:

Theorem 5.2. *If the maximum discrepancy of every rectangle (or cylinder intersection) is at most γ, then every protocol with error ϵ computing the function must have communication at least* $\log\left(\frac{1-2\epsilon}{\gamma}\right)$.

Some Examples Using Convexity in Combinatorics

To bound the discrepancy of communication protocols, we shall use Jensen's inequality. Before applying these ideas to bounding the discrepancy of rectangles and cylinder intersections, we show how to use them to prove some interesting results in combinatorics.

While there are dense graphs that have no 3-cycles (for example the complete bipartite graph), there are no dense graphs that avoid 4-cycles:

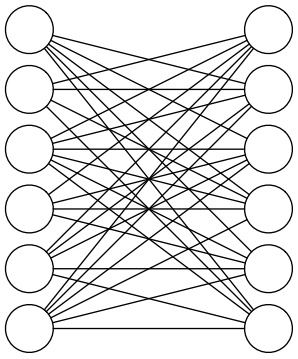


Figure 5.1: A dense graph with no 3-cycles.

Lemma 5.3. *Every n-vertex graph with $\epsilon\binom{n}{2}$ edges has at least $(\epsilon n-1)^4/4$ 4-cycles.*

Proof. Let $\mathbb{1}_{x,y}$ be 1 when there is an edge between the vertices x and y, and 0 otherwise. Then if x, x', y, y' are chosen uniformly at random,

we can count the number of 4-cycles by computing:

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{1}_{x,y}\cdot\mathbb{1}_{x',y}\cdot\mathbb{1}_{x,y'}\cdot\mathbb{1}_{x',y'}\right]\\
&=\mathop{\mathbb{E}}_{x,x'}\left[\mathop{\mathbb{E}}_{y}\left[\mathbb{1}_{x,y}\cdot\mathbb{1}_{x',y}\right]^2\right]\\
&\geq\mathop{\mathbb{E}}_{x,x'}\left[\mathop{\mathbb{E}}_{y}\left[\mathbb{1}_{x,y}\cdot\mathbb{1}_{x',y}\right]\right]^2\\
&=\mathop{\mathbb{E}}_{y}\left[\mathop{\mathbb{E}}_{x}\left[\mathbb{1}_{x,y}\right]^2\right]^2\\
&\geq\mathop{\mathbb{E}}_{x,y}\left[\mathbb{1}_{x,y}\right]^4.
\end{aligned}$$

This last quantity is at least $(\epsilon - 1/n)^4$, since we are picking a random edge as long as x and y are distinct. This gives $(\epsilon n - 1)^4/4$ cycles, since each cycle is counted 4 times. □

We can use similar ideas to prove that every dense bipartite graph must contain a reasonably large bipartite clique. Next we show a slightly different way to prove this:

Lemma 5.4. *If G is a bipartite graph of edge density ϵ, and bipartition A, B, with $|B| = n$, then there exists subsets $Q \subseteq A$, $R \subseteq B$ with $|Q| \geq \frac{\log n}{2\log(e/\epsilon)}$, $|R| \geq \sqrt{n}$, such that every pair of vertices $q \in Q, r \in R$ is connected by an edge.*

Proof. Pick a random subset $Q \subseteq A$ of size $\frac{\log n}{2\log(e/\epsilon)}$, and let R be all the common neighbors of Q. Given any vertex $b \in B$ that has degree d, the probability that b is included in R is exactly

$$\frac{\binom{d}{\frac{\log n}{2\log(e/\epsilon)}}}{\binom{n}{\frac{\log n}{2\log(e/\epsilon)}}} \geq \left(\frac{d}{en}\right)^{\frac{\log n}{2\log(e/\epsilon)}}.$$

Fact: $(\frac{n}{k})^k \leq \binom{n}{k} \leq (\frac{en}{k})^k$.

So if d_i is the degree of the i'th vertex, the expected size of the set R is at least

$$\sum_{i=1}^{n}\left(\frac{d_i}{en}\right)^{\frac{\log n}{2\log(e/\epsilon)}} \geq n\cdot\left(\frac{1}{n}\sum_{i=1}^{n}\frac{d_i}{en}\right)^{\frac{\log n}{2\log(e/\epsilon)}} \geq n\cdot\left(\frac{\epsilon}{e}\right)^{\frac{\log n}{2\log(e/\epsilon)}} = \sqrt{n}.$$

By convexity.

So there must be some choice of Q, R that proves the Lemma. □

Lower bounds for Inner-Product

Say Alice and Bob are given $x, y \in \{0,1\}^n$ and want to compute $\langle x, y\rangle \mod 2$. We have seen that this requires $n+1$ bits of communication using a deterministic protocol. Here we show that it requires $\approx n/2$ bits of communication even using a randomized protocol.

Lemma 5.5. *For any rectangle R, the discrepancy of R with respect to the inner product is at most $2^{-n/2}$.*

Proof. Since R is a rectangle, we can write its characteristic function as the product of two functions A and B. Thus we can write:

$$\begin{aligned}\mathop{\mathbb{E}}_{x,y}\left[\chi_R(x,y)\cdot(-1)^{\langle x,y\rangle}\right]^2 &= \mathop{\mathbb{E}}_{x,y}\left[A(x)\cdot B(y)\cdot(-1)^{\langle x,y\rangle}\right]^2\\ &= \mathop{\mathbb{E}}_{x}\left[A(x)\mathop{\mathbb{E}}_{y}\left[B(y)\cdot(-1)^{\langle x,y\rangle}\right]\right]^2\\ &\le \mathop{\mathbb{E}}_{x}\left[A(x)^2\mathop{\mathbb{E}}_{y}\left[B(y)\cdot(-1)^{\langle x,y\rangle}\right]^2\right],\end{aligned}$$

where the inequality follows from the fact that $\mathbb{E}[Z]^2 \le \mathbb{E}[Z^2]$ for any real valued random variable Z. Now we can drop $A(x)$ from this expression to get:

$$\begin{aligned}\mathop{\mathbb{E}}_{x,y}\left[\chi_R(x,y)\cdot(-1)^{\langle x,y\rangle}\right]^2 &\le \mathop{\mathbb{E}}_{x}\left[\mathop{\mathbb{E}}_{y}\left[B(y)\cdot(-1)^{\langle x,y\rangle}\right]^2\right]\\ &= \mathop{\mathbb{E}}_{x,y,y'}\left[B(y)B(y')\cdot(-1)^{\langle x,y\rangle+\langle x,y'\rangle}\right]\\ &= \mathop{\mathbb{E}}_{x,y,y'}\left[B(y)B(y')\cdot(-1)^{\langle x,y+y'\rangle}\right]\end{aligned}$$

In this way, we have completely eliminated the set A! Moreover, we can eliminate the set B too and write:

$$\begin{aligned}\mathop{\mathbb{E}}_{x,y}\left[\chi_R(x,y)\cdot(-1)^{\langle x,y\rangle}\right]^2 &\le \mathop{\mathbb{E}}_{x,y,y'}\left[B(y)B(y')\cdot(-1)^{\langle x,y+y'\rangle}\right]\\ &\le \mathop{\mathbb{E}}_{y,y'}\left[\left|\mathop{\mathbb{E}}_{x}\left[(-1)^{\langle x,y+y'\rangle}\right]\right|\right] \qquad (5.1)\end{aligned}$$

Now, whenever $y+y'$ is not 0 modulo 2, the expectation is 0. On the other hand, the probability that $y+y'$ is 0 modulo 2 is exactly 2^{-n}. So we can bound (5.1) by 2^{-n}. □

Lemma 5.5 and Theorem 5.2 together imply:

Theorem 5.6. *Any 2-party protocol that computes the inner-product with error at most ϵ over the uniform distribution must have communication at least $n/2 - \log(1/(1-2\epsilon))$.*

Similar ideas can be used to show that the communication complexity of the *generalized inner product* must be large in the number-on-forehead model[1]. Here each of the k players is given a binary string $x_i \in \{0,1\}^n$. They want to compute $\mathsf{GIP}(x) = \sum_{j=1}^{n}\prod_{i=1}^{k} x_{i,j} \mod 2$ We can show:

Lemma 5.7. *For any cylinder intersection S, the discrepancy of S with respect to the inner product is at most $e^{-n/4^{k-1}}$.*

[1] Babai et al., 1989

Each vector x_i can be interpretted as a subset of $[n]$. Then our protocol for computing the set intersection size gives a protocol for computing the inner product with communication $O(k^4 n/2^k)$.

Proof. Since S is a cylinder intersection, its characteristic function can be expressed as the product of k boolean functions $\chi_S = \prod_{i=1}^{k} \chi_i$, where χ_i does not depend on the i'th input. Thus we can write:

$$\begin{aligned}
\mathop{\mathbb{E}}_{x}\left[\chi_S(x)\cdot(-1)^{\mathsf{GIP}(x)}\right]^2 &= \mathop{\mathbb{E}}_{x}\left[\prod_{i=1}^{k}\chi_i(x)\cdot(-1)^{\mathsf{GIP}(x)}\right]^2 \\
&= \mathop{\mathbb{E}}_{x_1,\ldots,x_{k-1}}\left[\chi_k(x)\mathop{\mathbb{E}}_{x_k}\left[\prod_{i=1}^{k-1}\chi_i(x)\cdot(-1)^{\mathsf{GIP}(x)}\right]\right]^2 \\
&\leq \mathop{\mathbb{E}}_{x_1,\ldots,x_{k-1}}\left[\chi_k(x)^2\mathop{\mathbb{E}}_{x_k}\left[\prod_{i=1}^{k-1}\chi_i(x)\cdot(-1)^{\mathsf{GIP}(x)}\right]^2\right],
\end{aligned}$$

where the inequality follows from the fact that $\mathbb{E}[Z]^2 \leq \mathbb{E}[Z^2]$ for any real valued random variable Z. Now we can drop $\chi_k(x)$ from this expression to get:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x}\left[\chi_S(x)\cdot(-1)^{\mathsf{GIP}(x)}\right]^2 \\
&\leq \mathop{\mathbb{E}}_{x_1,\ldots,x_{k-1}}\left[\mathop{\mathbb{E}}_{x_k}\left[\prod_{i=1}^{k-1}\chi_i(x)\cdot(-1)^{\mathsf{GIP}(x)}\right]^2\right] \\
&= \mathop{\mathbb{E}}_{x_1,\ldots,x_k,x'_k}\left[\prod_{i=1}^{k-1}\chi_i(x)\chi_i(x')\cdot(-1)^{\sum_{j=1}^{n}(x_k+x'_k)\prod_{i=1}^{k-1}x_{i,j}}\right]
\end{aligned}$$

In this way, we have completely eliminated the function χ_k! Repeating this trick $k-1$ times gives the bound

$$\mathop{\mathbb{E}}_{x}\left[\chi_S(x)\cdot(-1)^{\mathsf{GIP}(x)}\right]^{2^{k-1}} \leq \mathop{\mathbb{E}}_{x_2,x'_2,\ldots,x_k,x'_k}\left[\left|\mathop{\mathbb{E}}_{x_1}\left[(-1)^{\sum_{j=1}^{n}x_1\prod_{i=2}^{k}(x_i+x'_i)}\right]\right|\right].$$

Now, whenever $\prod_{i=2}^{k-1}(x_{i,j}+x_{i',j})$ is not 0 modulo 2, at any coordinate j, the expectation is 0. On the other hand, the probability that this expression is 0 modulo 2 is exactly $(1-2^{-k+1})^n$. So we get

$$\mathop{\mathbb{E}}_{x}\left[\chi_S(x)\cdot(-1)^{\mathsf{GIP}(x)}\right]^{2^{k-1}} \leq (1-2^{-k+1})^n < e^{-n/2^{k-1}}.$$

Fact: $1-x<e^{-x}$ for $x>0$.

This proves that

$$\mathop{\mathbb{E}}_{x}\left[\chi_S(x)\cdot(-1)^{\mathsf{GIP}(x)}\right] < e^{-n/4^{k-1}}.$$

□

By Lemma 5.7 and Theorem 5.2:

Theorem 5.8. *Any randomized protocol for computing the generalized inner product in the number-on-forehead model with error ϵ requires $n/4^{k-1} - \log(1/(1-2\epsilon))$ bits of communication.*

Lower bounds for Disjointness in the Number-on-Forehead model

At first it may seem that the discrepancy method is not very useful for proving lower bounds against functions like disjointness, which *do* have large monochromatic rectangles.

Suppose Alice and Bob are given two sets $X, Y \subseteq [n]$ and want to compute disjointness. If we use a distribution on inputs that gives intersecting sets with probability at most ϵ, then there is a trivial protocol with error at most ϵ. On the other hand, if the probability of intersection is at least ϵ, then then there must be some fixed coordinate i such that an intersection occurs in coordinate i with probability at least ϵ/n. Setting $R = \{(X, Y) : i \in X, i \in Y\}$, we get

$$\left| \mathbb{E}\left[\chi_R(X,Y) \cdot (-1)^{\mathsf{Disj}(X,Y)} \right] \right| \geq \epsilon/n,$$

so we cannot hope to prove a lower bound better than $\Omega(\log n)$ this way. Nevertheless, we show that one *can* use discrepancy to give a lower bound on the communication complexity of disjointness [2], even when the protocol is allowed to be randomized, by studying a different expression. In fact, this is the only known method to prove lower bounds on the communication complexity of disjointness in the number-on-forehead model.

[2] Sherstov, 2012; and Rao and Yehudayoff, 2015

Consider the following distribution on sets. Let the universe consist of disjoint sets $I_1, \ldots, I_m$. Alice gets m independently sampled sets $X_1, \ldots, X_m$, where X_i is a random subset of I_i, and Bob gets m random sets of size 1, $Y_1, \ldots, Y_m$, where the i'th set is again drawn from I_i. Let $X = \cup_{i=1}^m X_i$, and $Y = \cup_{i=1}^m Y_i$. We prove:

Lemma 5.9. *For any rectangle R,*

$$\mathbb{E}\left[\chi_R(X,Y) \cdot (-1)^{\sum_{i=1}^m \mathsf{Disj}(X_i,Y_i)} \right] \leq \sqrt{\frac{1}{\prod_{j=1}^m |I_i|}}.$$

Proof. As usual, we express $\chi_R(X, B) = A(X) \cdot B(X)$ and carry out a convexity argument. We get:

$$\begin{aligned}
&\mathbb{E}\left[\chi_R(X,Y) \cdot (-1)^{\sum_{i=1}^m \mathsf{Disj}(X_i,Y_i)} \right]^2 \\
&= \mathbb{E}\left[A(X) \cdot B(Y) \cdot (-1)^{\sum_{i=1}^m \mathsf{Disj}(X_i,Y_i)} \right]^2 \\
&\leq \mathbb{E}\left[A(X)^2 \, \mathbb{E}\left[B(Y) \cdot (-1)^{\sum_{i=1}^m \mathsf{Disj}(X_i,Y_i)} \right]^2 \right] \\
&\leq \mathop{\mathbb{E}}_{X,Y,Y'}\left[B(Y)B(Y') \cdot (-1)^{\sum_{i=1}^m \mathsf{Disj}(X_i,Y_i) + \sum_{i=1}^m \mathsf{Disj}(X_i,Y_i')} \right] \\
&\leq \mathop{\mathbb{E}}_{Y,Y'}\left[\left| \mathop{\mathbb{E}}_{X}\left[(-1)^{\sum_{i=1}^m \mathsf{Disj}(X_i,Y_i) + \sum_{i=1}^m \mathsf{Disj}(X_i,Y_i')} \right] \right| \right]
\end{aligned}$$

For any fixing of Y, Y', the inner expectation is 0 as long as $Y \neq Y'$. The probability that $Y = Y'$ is exactly $1/\prod_{j=1}^{m} |I_i|$. Thus we get that $\mathbb{E}\left[\chi_R(X,Y) \cdot (-1)^{\sum_{i=1}^{m} \mathsf{Disj}(X_i,Y_i)}\right]^2 \leq 1/\prod_{j=1}^{m} |I_j|$, proving the bound. □

Lemma 5.9 may not seem useful at first, because under the given distribution, the probability that X, Y are disjoint is 2^{-m}. However, we can actually use it to give a linear lower bound on the communication of deterministic protocols. Suppose a deterministic protocol for disjointness has communication c. Then there must be at most 2^c monochromatic 1-rectangles $R_1, \ldots, R_t$ that cover all the 1's. Whenever X, Y are disjoint, we have that $\sum_{j=1}^{m} \mathsf{Disj}(X_i, Y_i) = m$. On the other hand, the probability that X, Y are disjoint is exactly 2^{-m}. Thus, we get

$$\begin{aligned} 2^{-m} &\leq \mathbb{E}\left[\sum_{i=1}^{t} \chi_{R_i}(X,Y) \cdot (-1)^{\sum_{j=1}^{m} \mathsf{Disj}(X_i,Y_i)}\right] \\ &\leq \sum_{i=1}^{t} \left|\mathbb{E}\left[\chi_{R_i}(X,Y) \cdot (-1)^{\sum_{j=1}^{m} \mathsf{Disj}(X_i,Y_i)}\right]\right| \\ &\leq 2^c \cdot (1/\prod_{j=1}^{m} \sqrt{|I_j|}). \end{aligned}$$

Setting $|I_i| = 4$, and rearranging gives $c \geq m$, a linear lower bound on the communication complexity of disjointness. While we have already seen several approaches to proving linear lower bounds on disjointness, this approach has a unique advantage: it works even in the number-on-forehead model. Consider the distribution where for each $j = 1, 2, \ldots, m$, $X_{1,j} \subseteq I_i$ is picked uniformly at random, and $X_{2,j}, \ldots, X_{k,j} \subseteq I_i$ are picked uniformly at random, subject to the constraint that their intersection contains exactly 1 element. Set $X_i = \cup_{j=1}^{m} X_{i,j}$. Suppose the i'th player has set X_i written on his forehead. Then we prove:

Lemma 5.10. *For any cylinder intersection S,*

$$\mathbb{E}\left[\chi_S(X) \cdot (-1)^{\sum_{j=1}^{m} \mathsf{Disj}(X_{1,j},\ldots,X_{k,j})}\right] \leq \prod_{j=1}^{m} \frac{2^{k-1}-1}{\sqrt{|I_j|}}.$$

Proof. We prove the lemma by induction on k. When $k = 2$, the statement was already proved in Lemma 5.9.

For ease of notation, we write T_j to denote the input in the j'th interval, $X_{1,j}, \ldots, X_{k,j}$. Suppose $\chi_S(X) = \prod_{i=1}^{k} \chi_i(X)$, where χ_i is the indicator of the i'th cylinder. Then, as usual, we can apply a convexity argument to bound:

$$\begin{aligned}
&\mathbb{E}\left[\chi_S(X)\cdot(-1)^{\sum_{j=1}^m \mathsf{Disj}(X_{1,j},\ldots,X_{k,j})}\right]^2 \\
&\leq \mathop{\mathbb{E}}_{X_1,\ldots,X_{k-1}}\left[\chi_k(X)^2\cdot\mathop{\mathbb{E}}_{X_k}\left[\prod_{i=1}^{k-1}\chi_i(X)\cdot(-1)^{\sum_{j=1}^m \mathsf{Disj}(T_j)}\right]^2\right] \\
&\leq \mathop{\mathbb{E}}_{X_1,\ldots,X_{k-1}}\left[\mathop{\mathbb{E}}_{X_k}\left[\prod_{i=1}^{k-1}\chi_i(X)\cdot(-1)^{\sum_{j=1}^m \mathsf{Disj}(T_j)}\right]^2\right] \\
&= \mathop{\mathbb{E}}_{X_1,\ldots,X_{k-1},X_k,X'_k}\left[\prod_{i=1}^{k-1}\chi_i(X)\chi_i(X')\cdot(-1)^{\sum_{j=1}^m \mathsf{Disj}(T_j)+\mathsf{Disj}(T'_j)}\right], \quad (5.2)
\end{aligned}$$

where here $X = X_1,\ldots,X_k$, $X' = X_1,\ldots,X_{k-1},X'_k$, and $T'_j = X_{1,j},\ldots,X_{k-1,j},X'_{k,j}$. Let v,v' denote the two common intersection points of the last $k-1$ sets.

Now whenever $v = v'$, we have $\mathsf{Disj}(T_j) = \mathsf{Disj}(T'_j)$, and so the j term of the sum is 0 modulo 2. On the other hand, when $v \neq v'$, then any intersection in T_j must take place in the set $X_{k,j} \setminus X'_{k,j}$, and any intersection in T'_j must take place in $X'_{k,j} \setminus X_{j,k}$, so we can fix the intersections of all the sets to $X_k \cap X'_k$ and the $X^c_k \cap X'^c_k$ and use induction to bound the discrepancy.

Let Z_j be the random variable defined as:

$$Z_j = \begin{cases} 1 & \text{if } v = v', \\ \frac{(2^{k-2}-1)^2}{\sqrt{|X_{k,j}\setminus X'_{k,j}||X'_{k,j}\setminus X_{k,j}|}} & \text{otherwise.} \end{cases}$$

Then we get:

$$(5.2) \leq \mathbb{E}\left[\prod_{j=1}^m Z_j\right] \leq \prod_{j=1}^m \mathbb{E}\left[Z_j\right],$$

since the Z_i's are independent of each other. We need a technical claim next:

Claim 5.11. *Suppose a set $Q \subseteq I_j$ is sampled by including a random element $v \in I_j$ and adding every other element to Q independently with probability $\gamma \neq 0$. Then $\mathbb{E}\left[\frac{1}{|Q|}\right] \leq 1/(\gamma|I_j|)$.*

Proof.

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{|Q|}\right] &= \sum_{Q,v}\frac{(1/|I_j|)\cdot\gamma^{|Q|-1}(1-\gamma)^{|I_j|-|Q|}}{|Q|} \\
&= \frac{1}{\gamma|I_j|}\sum_{Q\neq\emptyset}\gamma^{|Q|}(1-\gamma)^{|I_j|-|Q|} \leq \frac{1}{\gamma|I_j|}(1-\gamma+\gamma)^{|I_j|} = \frac{1}{\gamma|I_j|}.
\end{aligned}$$

□

If $X_{2,j} \cap \ldots \cap X_{k-1,j}$ is of size t, then the probability that $v = v'$ is exactly $1/t$. Thus, the probability of this event is exactly the expected size of $1/|Q|$, where Q is the intersection of the first $k-1$ sets. After picking the common intersection point, every other element of I_j is included in Q independently with probability $\frac{1}{2^{k-1}-1}$. So by Claim 5.11, $\Pr[v = v'] = \frac{2^{k-1}-1}{|I_j|}$. When $v \neq v'$, we can bound

$$\begin{aligned} Z_j &= \frac{(2^{k-2}-1)^2}{\sqrt{|X_{k,j} \setminus X'_{k,j}| \cdot |X'_{k,j} \setminus X_{k,j}|}} \\ &\leq \frac{(2^{k-2}-1)^2}{2} \cdot \left(\frac{1}{|X_{k,j} \setminus X'_{k,j}|} + \frac{1}{|X'_{k,j} \setminus X_{k,j}|} \right), \end{aligned}$$

By the Arithmetic mean - geometric mean inequality: $\sqrt{ab} \leq \frac{a+b}{2}$.

Let $Q = X_k \setminus X'_k$. Once again we see that Q is sampled by picking the value of V uniformly, and then every other element is included in Q independently with probability $\frac{2^{k-2}-1}{2(2^{k-1}-1)}$. So by Claim 5.11, $\mathbb{E}\left[\frac{1}{|X_{k,j} \setminus X'_{k,j}|}\right] = \frac{2(2^{k-1}-1)}{(2^{k-2}-1)|I_j|}$. Combining these bounds, we get

$$\begin{aligned} \mathbb{E}\left[Z_j\right] &\leq \Pr[v = v'] + \mathbb{E}\left[\frac{(2^{k-2}-1)^2}{|X_{k,j} \setminus X'_{k,j}|}\right] \\ &\leq \frac{2^{k-1}-1}{|I_j|} + \frac{2(2^{k-1}-1)(2^{k-2}-1)^2}{(2^{k-2}-1)|I_j|} \\ &= \frac{(2^{k-1}-1)^2}{|I_j|}, \end{aligned}$$

as required. □

Lemma 5.10 can be used to prove a linear lower bound on the communication of deterministic protocols. Suppose a deterministic protocol for disjointness has communication c. Then there must be at most 2^c monochromatic 1-cylinder intersections $S_1, \ldots, S_t$ that cover all the 1's. Whenever $X_1, \ldots, X_k$ are disjoint, we have that $\sum_{j=1}^{m} \mathsf{Disj}(X_{1,j}, X_{2,j}, \ldots, X_{k,j}) = m$. On the other hand, the probability that $X_1, \ldots, X_k$ are disjoint is exactly 2^{-m}. Thus, we get

$$\begin{aligned} 2^{-m} &\leq \mathbb{E}\left[\sum_{i=1}^{t} \chi_{S_i}(X_1, \ldots, X_k) \cdot (-1)^{\sum_{j=1}^{m} \mathsf{Disj}(X_{1,i}, \ldots, X_{k,j})}\right] \\ &\leq \sum_{i=1}^{t} \left|\mathbb{E}\left[\chi_{S_i}(X_1, \ldots, X_k) \cdot (-1)^{\sum_{j=1}^{m} \mathsf{Disj}(X_{1,i}, \ldots, X_{k,j})}\right]\right| \\ &\leq 2^c \cdot \left(\prod_{j=1}^{m} \frac{2^{k-1}-1}{\sqrt{|I_j|}}\right). \end{aligned}$$

Setting $|I_j| = 16 \cdot (2^{k-1}-1)^2$, we get that $c \geq m = \frac{n}{16(2^{k-1}-1)^2}$.

Setting $|I_i| = \ell$, for all i, and rearranging gives $c \geq \left(\frac{\sqrt{\ell}}{2 \cdot (2^{k-1}-1)}\right)^{\frac{n}{\ell}} = \left((\ell/a)^{1/\ell}\right)^{n/2}$, where $a = (2 \cdot (2^{k-1} - 1))^2$. The derivative of $(\ell/a)^{1/\ell}$ is $(\ell/a)^{1/\ell} \cdot \left(\frac{1-\ln(\ell/a)}{\ell^2}\right)$, which is 0 when $\ell = e \cdot a$. In this way, one can set ℓ to get a slightly better bound: $c \geq \frac{n \log e}{8e \cdot (2^{k-1}-1)^2}$.

Theorem 5.12. *Any deterministic protocol for computing disjointness in the number-on-forehead model requires* $\frac{n}{16(2^{k-1}-1)^2}$ *bits of communication.*

6
Information

Shannon's seminal work on information theory[1] has had a big impact on communication complexity. Shannon wanted to measure the amount of information (or *entropy*) contained in a random variable X. Shannon's definition was motivated by the observation that the amount of information contained in a message is not the same as the length of the message. Suppose we are working in the distributional setting, where the inputs are sampled from some distribution μ.

[1] Shannon, 1948

- Consider a protocol where Alice's first message to Bob is a c-bit string that is *always* 0^c, no matter what her input is. This message does not convey any information to Bob. We might as well run the protocol imagining that this first message has already been sent, and so reduce the communication of the first step to 0.

  The *entropy* of the message is 0.

- Consider a protocol where Alice's first message to Bob is a random string from a set $S \subseteq \{0,1\}^c$, with $|S| \ll 2^c$. In this case, the parties should use $\log |S|$ bits to index the elements of the set, reducing the communication from c to $\log |S|$.

  The *entropy* of the message is $\log |S|$.

- Consider a protocol where Alice's first message to Bob is the string 0^n with probability $1-\epsilon$, and is a uniformly random n bit string with the remaining probability. In this case one cannot encode every message using fewer than n bits. However, Alice can send the bit 0 to encode the string 0^n, and the the string $1x$ to encode the n bit string x. Although the first message is still quite long in the worst case, the expected length of the message is $1-\epsilon+\epsilon(n+1) = 1+\epsilon n$.

  The *entropy* of the message is $\approx \epsilon n$.

Shannon's definition of entropy gives a general way to compute the length of the smallest encoding of a message. Given a random variable X with probability distribution $p(x)$, define the *entropy* of X to be

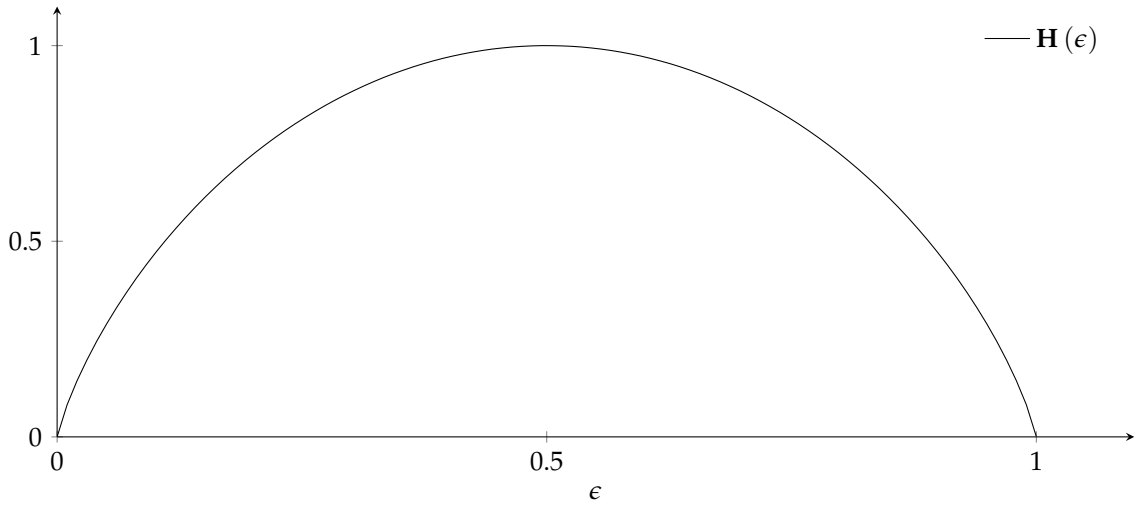


Figure 6.1: The entropy of a bit with $p(1) = \epsilon$.

$$\mathbf{H}(X) = \sum_x p(x) \log(1/p(x)) = \mathop{\mathbb{E}}_{p(x)} \left[\log \frac{1}{p(x)} \right].$$

The entropy of X characterizes the expected number of bits that need to be transmitted to encode X. Intuitively, if there is an encoding of X that has expected length k, then X can be encoded by a string of length $10k$ most of the time, so one would expect that X takes on one of $2^{O(k)}$ values most of the time, and so the expected value of $\log(1/p(x))$ should be $O(k)$. Conversely, if the entropy of X is k, then one can encode X using the positive integers, in such a way that $p(1) \geq p(2) \geq \ldots$. The probability that the sample for X is the i'th integer is at most $1/i$ (since otherwise $\sum_{j=1}^{i} p(i) > 1$), so the expected length of transmitting the number that encodes X should be bounded by $\sum_i p(i) \log i$, which is at most the entropy. Formally, we can prove:

The definition ensures that the entropy is always non-negative.

Theorem 6.1. *X can be encoded using a message whose expected length is at most $\mathbf{H}(X) + 1$. Conversely, every encoding of X has expected length at least $\mathbf{H}(X)$.*

Can you think of an example that shows that the expected length needs to be at least $\mathbf{H}(X) + 1$?

Proof. Without loss of generality, suppose that X is an integer from $[n]$, with $p(i) \geq p(i+1)$. Let $\ell_i = \lceil \log(1/p(i)) \rceil$. We shall encode i with a leaf at depth ℓ_i. Then the expected length of the message will be $\sum_i p(i)\ell_i \leq \sum_i p(i)(\log(1/p(i)) + 1) = \mathbf{H}(X) + 1$. The encoding is done greedily. In the first step, we pick the first vertex of the complete binary tree at depth ℓ_1 and let that vertex represent 1. We delete all of its descendants so that the vertex becomes a leaf. Next we find the first vertex at depth ℓ_2 that has not been deleted, and use it to represent 2. We continue in this way until every element of $[n]$ has been encoded. For $i < j$, the number of vertices at depth ℓ_j that are deleted in the i'th step is exactly $2^{\ell_j - \ell_i}$, so the number of vertices

at depth j that are deleted before the j'th step is

$$\sum_{i=1}^{j-1} 2^{\ell_j - \ell_i} = 2^{\ell_j} \left(\sum_{i=1}^{j-1} 2^{-\ell_i} \right) \leq 2^{\ell_j} \sum_{i=1}^{j-1} p(i) < 2^{\ell_j},$$

so some vertex will be available at the j'th step. This ensures that every step of this process succeeds.

Conversely, suppose X can be encoded in such a way that i is encoded using ℓ_i bits. Then the expected length of the encoding is:

$$\begin{aligned}
\mathop{\mathbb{E}}_{p(i)}[\ell_i] &= \mathop{\mathbb{E}}_{p(i)}[\log(1/p(i))] - \mathop{\mathbb{E}}_{p(i)}\left[\log(2^{-\ell_i}/p(i))\right] \\
&\geq \mathbf{H}(X) - \log\left(\mathop{\mathbb{E}}_{p(i)}\left[2^{-\ell_i}/p(i)\right] \right) && \text{By convexity of the log function,} \\
&= \mathbf{H}(X) - \log\left(\sum_i 2^{-\ell_i} \right). && \mathbb{E}[\log Y] \leq \log(\mathbb{E}[Y]).
\end{aligned}$$

If you pick a random path starting from the root in the protocol tree, you hit the leaf encoding i with probability $2^{-\ell_i}$. The probability that you hit one of the leaves encoding a number from $[n]$ is thus $\sum_i 2^{-\ell_i} \leq 1$. Thus $\log\left(\sum_i 2^{-\ell_i}\right)$ is at most 0, and the entropy is at most the expected length of the encoding. □

Entropy, Divergence and Mutual Information

THE CONCEPTS OF DIVERGENCE AND MUTUAL INFORMATION are closely related to the concept of entropy. They provide a toolbox that helps to understand the flow of information in different situations. The *divergence* between two distributions $p(x)$ and $q(x)$ is defined to be

$$\overline{\overline{\frac{p(x)}{q(x)}}} = \sum_x p(x) \log \frac{p(x)}{q(x)} = \mathop{\mathbb{E}}_{p(x)}\left[\log \frac{p(x)}{q(x)}\right].$$

The divergence is a measure of distance the two distributions. Clearly, $\overline{\overline{\frac{p(x)}{p(x)}}} = 0$.

Fact 6.2. $\overline{\overline{\frac{p(x)}{q(x)}}} \geq 0$.

Proof.

$$\begin{aligned}
\overline{\overline{\frac{p(x)}{q(x)}}} &= \mathop{\mathbb{E}}_{p(x)}\left[\log \frac{p(x)}{q(x)}\right] \\
&= -\sum_x p(x) \log \frac{q(x)}{p(x)} \geq -\log \sum_x p(x) \frac{q(x)}{p(x)} = \log 1 = 0. && \text{The inequality follows from the convexity of the log function.}
\end{aligned}$$

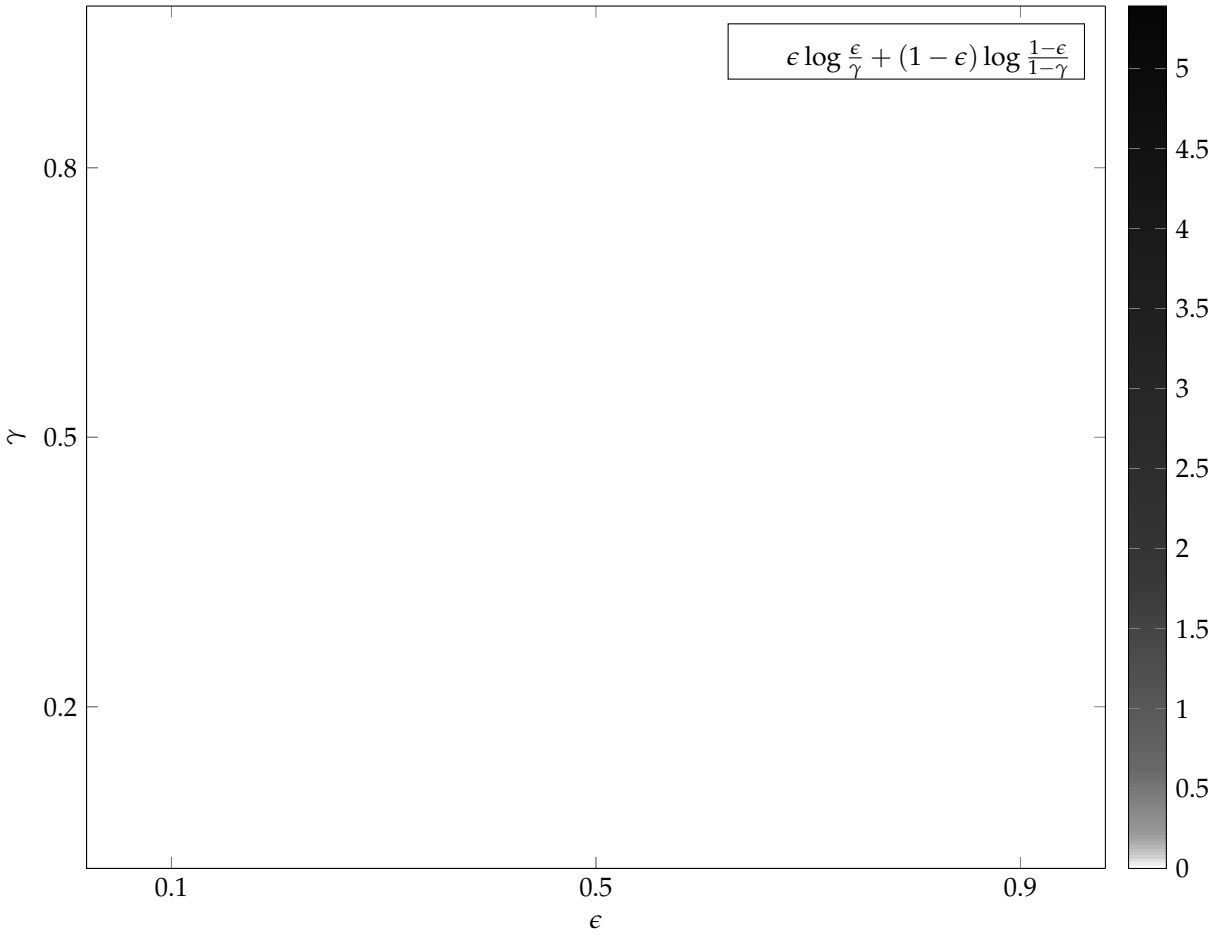


Figure 6.2: The divergence between two bits.

□

However, the divergence is not symmetric: $\dfrac{p(x)}{q(x)} \neq \dfrac{q(x)}{p(x)}$ in general. Moreover, the divergence can be infinite, for example if p is supported on a point that has 0 probability under q. If X is an ℓ-bit string, we see that:

$$\mathbf{H}(X) = \mathop{\mathbb{E}}_{p(x)}[\log(1/p(x))] = \ell - \mathop{\mathbb{E}}_{p(x)}\left[\log \frac{p(x)}{2^{-\ell}}\right] = \ell - \frac{p(x)}{q(x)},$$

where $q(x)$ is the uniform distribution on ℓ-bit strings. So we see that the entropy of a string is just a way to measure the divergence from uniform. In particular, since the divergence is non-negative (Fact 6.2), the uniform distribution has maximum entropy of all the distributions on a set.

In our context, the divergence is most often measured between two distributions that arise from the same probability space. For example, if $\mathcal{E}$ is an event in a probability space containing x, we have

Fact 6.3. $\dfrac{p(x|\mathcal{E})}{p(x)} \leq \log \frac{1}{p(\mathcal{E})}$.

Proof.

$$\begin{aligned}\frac{p(x|\mathcal{E})}{p(x)} &= \mathop{\mathbb{E}}_{p(x|\mathcal{E})}\left[\log \frac{p(x|\mathcal{E})}{p(x)}\right] \\ &= \mathop{\mathbb{E}}_{p(x|\mathcal{E})}\left[\log \frac{p(\mathcal{E}|x)}{p(\mathcal{E})}\right] \\ &\leq \log \frac{1}{p(\mathcal{E})}.\end{aligned}$$

□

The entropy, mutual information and divergence are all expectations over the universe of various log-ratios.

We can use divergence to quantify the dependence between two random variables. If $p(a,b)$ is a joint distribution, we define the *mutual information* between a and b to be

$$\mathbf{I}(A:B) = \mathop{\mathbb{E}}_{p(a,b)}\left[\log \frac{p(a,b)}{p(a)p(b)}\right] = \mathop{\mathbb{E}}_{p(a,b)}\left[\log \frac{p(b|a)}{p(b)}\right] = \mathop{\mathbb{E}}_{p(a)}\left[\frac{p(b|a)}{p(b)}\right].$$

We have that $\mathbf{I}(A:B) = \mathbf{H}(A) + \mathbf{H}(B) - \mathbf{H}(AB)$. The mutual information of any random variable with itself is the same as its entropy $\mathbf{I}(A:A) = \mathbf{H}(A)$. On the other hand, if A, B are independent, $\mathbf{I}(A:B) = 0$. In general, the mutual information is always a number between these two quantities: $0 \leq \mathbf{I}(A:B) \leq \mathbf{H}(A)$. The first inequality follows from Fact 6.2, and the second by observing:

$$\begin{aligned}\mathbf{H}(A) - \mathbf{I}(A:B) &= \mathop{\mathbb{E}}_{p(a,b)}\left[\log \frac{1}{p(a)} - \log \frac{p(a,b)}{p(a)p(b)}\right] \\ &= \mathop{\mathbb{E}}_{p(a,b)}\left[\log \frac{p(b)}{p(a,b)}\right] \geq 0.\end{aligned}$$

Chain Rules

Chain rules allow one to relate bounds on the information of a collection of random variables to the information associated with

each variable. Suppose $p(a,b)$ and $q(a,b)$ are two distributions. Then we have

$$\begin{aligned}\mathrm{D}\left(\frac{p(a,b)}{q(a,b)}\right) &= \mathop{\mathbb{E}}_{p(a,b)}\left[\log\frac{p(a)\cdot p(b|a)}{q(a)\cdot q(b|a)}\right] \\ &= \mathop{\mathbb{E}}_{p(a,b)}\left[\log\frac{p(a)}{q(a)}\right] + \mathop{\mathbb{E}}_{p(a,b)}\left[\log\frac{p(b|a)}{q(b|a)}\right] \\ &= \mathrm{D}\left(\frac{p(a)}{q(a)}\right) + \mathop{\mathbb{E}}_{p(a)}\left[\mathrm{D}\left(\frac{p(b|a)}{q(b|a)}\right)\right].\end{aligned}$$

In words, the total divergence is the sum of the divergence from the first variable, plus the expected divergence from the second variable.

Similar chain rules hold for the entropy and mutual information. Suppose A, B are two random bits that are always equal. Then $\mathbf{H}(AB) = 1 \neq \mathbf{H}(A) + \mathbf{H}(B)$, so the entropy does not add in general. Nevertheless, a chain rule does exist for entropy. Denote

$$\mathbf{H}(B \mid A) = \mathop{\mathbb{E}}_{p(a,b)}\left[\log\frac{1}{p(b|a)}\right].$$

Then we have the chain rule[2]: $\mathbf{H}(AB) = \mathbf{H}(A) + \mathbf{H}(B \mid A)$.

[2] $\mathbf{H}(AB) = \mathbb{E}_{p(a,b)}\left[\log\frac{1}{p(a)p(b|a)}\right] = \mathbb{E}_{p(a,b)}\left[\log\frac{1}{p(a)} + \log\frac{1}{p(b|a)}\right] = \mathbf{H}(A) + \mathbf{H}(B \mid A)$.

Suppose A, B, C are three random bits that are all equal to each other. Then $\mathbf{I}(AB : C) = 1 < 2 = \mathbf{I}(A : C) + \mathbf{I}(B : C)$. On the other hand, if A, B, C are three random bits satisfying $A + B + C = 0 \mod 2$, we have $\mathbf{I}(AB : C) = 1 > 0 = \mathbf{I}(A : C) + \mathbf{I}(B : C)$. Nevertheless, a chain rule does hold for mutual information, after we use the right definition. Denote:

$$\mathbf{I}(B : C \mid A) = \mathop{\mathbb{E}}_{p(a,b,c)}\left[\log\frac{p(b,c|a)}{p(b|a)p(c|a)}\right].$$

Then we have the chain rule[3]: $\mathbf{I}(AB : C) = \mathbf{I}(A : C) + \mathbf{I}(B : C \mid A)$.

[3] $\mathbf{I}(AB : C) = \mathbb{E}_{p(a,b,c)}\left[\log\frac{p(a,c)\cdot p(b|a,c)}{p(a)p(b|a)\cdot p(c)}\right] = \mathbf{I}(A : C) + \mathbb{E}_{p(a,b,c)}\left[\log\frac{p(b|a,c)}{p(b|a)}\right] = \mathbf{I}(A : C) + \mathbb{E}_{p(a,b,c)}\left[\log\frac{p(b,c|a)}{p(b|a)\cdot p(c|a}\right] = \mathbf{I}(A : C) + \mathbf{I}(B : C \mid A)$.

Subadditivity

Each of the definitions we have seen so far satisfies the property that conditioning on variables can either only increase the quantity or only decrease the quantity, a property that we loosely refer to as *subadditivity*. We start with the divergence. Suppose $p(a,b), q(a,b)$ are two distributions. Then:

$$\begin{aligned}\mathop{\mathbb{E}}_{p(b)}\left[\mathrm{D}\left(\frac{p(a|b)}{q(a)}\right)\right] &= \mathop{\mathbb{E}}_{p(a,b)}\left[\log\frac{p(a)}{q(a)} + \log\frac{p(a|b)}{p(a)}\right] \\ &= \mathrm{D}\left(\frac{p(a)}{q(a)}\right) + \mathbf{I}(A : B) \geq \mathrm{D}\left(\frac{p(a)}{q(a)}\right).\end{aligned}$$

One consequence of this last inequality is:

Fact 6.4. *If $q(x_1, \ldots, x_n)$ is a product distribution, then for any p,*

$$\frac{p(x_1, \ldots, x_n)}{q(x_1, \ldots, x_n} \geq \sum_{i=1}^{n} \frac{p(x_i)}{q(x_i)}.$$

When it comes to entropy, we have:

$$\mathbf{H}(AB) = \mathbf{H}(A) + \mathbf{H}(B) - \mathbf{I}(A : B) \leq \mathbf{H}(A) + \mathbf{H}(B).$$

This last inequality also implies that

$$\mathbf{H}(A) \geq \mathbf{H}(AB) - \mathbf{H}(B) = \mathbf{H}(A \mid B).$$

We have already seen that conditioning on a random variable can both decrease, or increase the mutual information. Nevertheless, when A, B are independent, we can prove[4]:

$$\mathbf{I}(AB : C) \geq \mathbf{I}(A : C) + \mathbf{I}(B : C).$$

Shearer's Inequality

A useful consequence of subadditivity is Shearer's inequality:

Lemma 6.5. *Suppose $X = X_1, \ldots, X_n$ is a random variable and $S \subseteq [n]$ is a set sampled independently of X. Then if $p(i \in S) \geq \epsilon$ for every $i \in [n]$, we have $\mathbf{H}(X_S \mid S) \geq \epsilon \cdot \mathbf{H}(X)$.*

Proof. Suppose $S = \{a, b, c\}$, with $a < b < c$. Then we can express

$$\begin{aligned}\mathbf{H}(X_S) &= \mathbf{H}(X_a) + \mathbf{H}(X_b \mid X_a) + \mathbf{H}(X_c \mid X_a, X_b) \\ &\geq \mathbf{H}(X_a \mid X_{<a}) + \mathbf{H}(X_b \mid X_{<b}) + \mathbf{H}(X_c \mid X_{<c}),\end{aligned}$$

by subadditivity. In general, we get that

$$\begin{aligned}\mathbf{H}(X_S \mid S) &\geq \mathop{\mathbb{E}}_{S}\left[\sum_{i \in S} \mathbf{H}(X_i \mid X_{<i})\right] \\ &= \sum_{i=1}^{n} p(i \in S)\mathbf{H}(X_i \mid X_{<i}) \geq \epsilon \cdot \mathbf{H}(X).\end{aligned}$$

□

Pinsker's Inequality

Pinsker's inequality bounds the statistical distance between two distributions in terms of the divergence between them.

Lemma 6.6. $\frac{p(x)}{q(x)} \geq \frac{2}{\ln 2} \cdot |p - q|^2$.

Proof of Fact 6.4.

$$\begin{aligned}&\frac{p(x_1, \ldots, x_n)}{q(x_1, \ldots, x_n} \\ &= \sum_{i=1}^{n} \mathop{\mathbb{E}}_{p(x_{<i}} \left[\frac{p(x_i|x_{<i})}{q(x_i|x_{<i})}\right] \\ &= \sum_{i=1}^{n} \mathop{\mathbb{E}}_{p(x_{<i}} \left[\frac{p(x_i|x_{<i})}{q(x_i)}\right] \\ &\geq \sum_{i=1}^{n} \frac{p(x_i)}{q(x_i)}.\end{aligned}$$

□

[4] $\mathbf{I}(AB : C) - \mathbf{I}(A : C) - \mathbf{I}(B : C) = \mathbf{I}(B : C \mid A) - \mathbf{I}(B : C) = \mathbf{H}(B \mid A) - \mathbf{H}(B \mid AC) - \mathbf{H}(B) + \mathbf{H}(B \mid C) \geq 0$, since $\mathbf{H}(B \mid AC) \leq \mathbf{H}(B \mid C)$, and $\mathbf{H}(B \mid A) = \mathbf{H}(B)$.

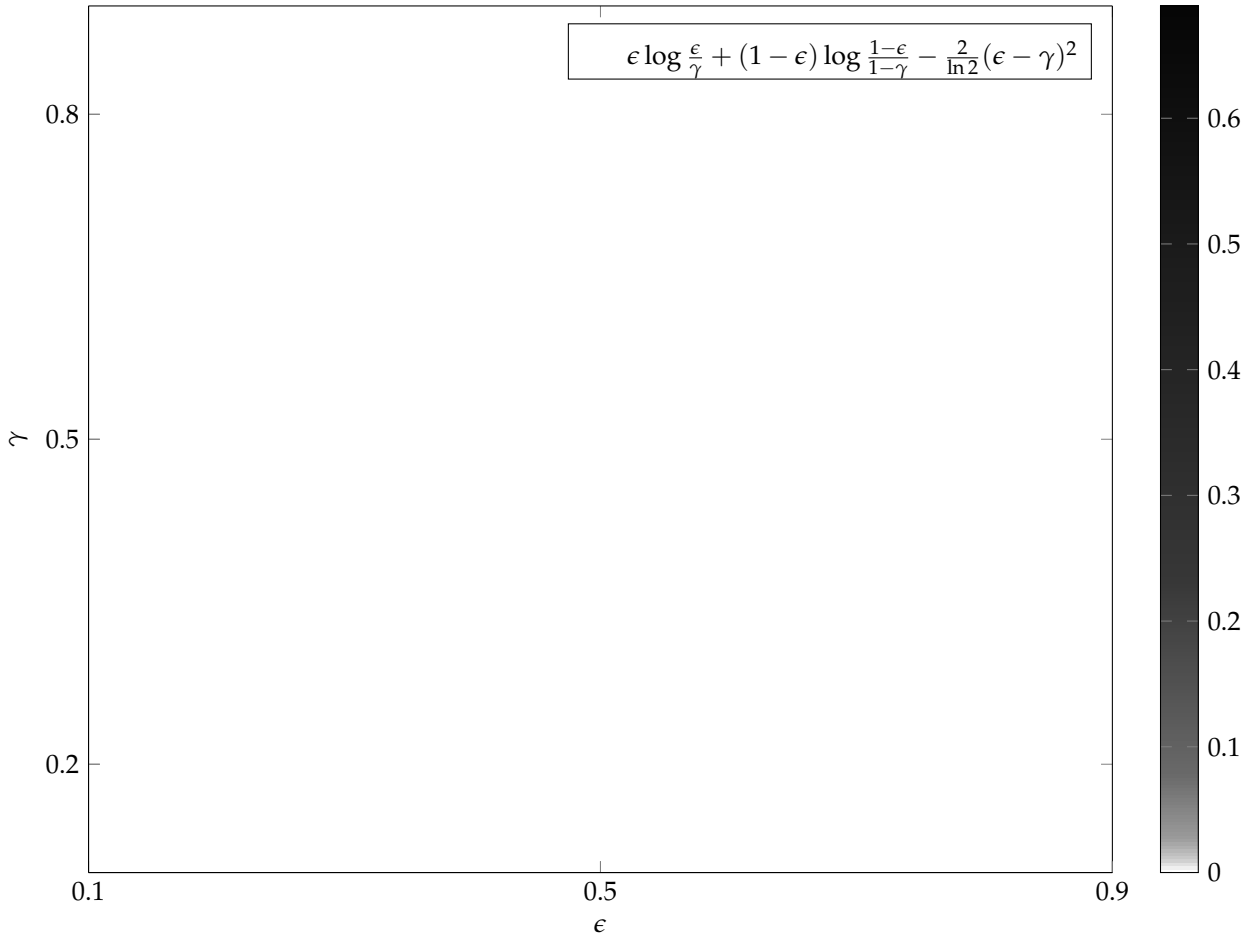


Figure 6.3: Pinsker's Inequality

Proof. Let T be the set that maximizes $p(T) - q(T)$, and define

$$x_T = \begin{cases} 1 & \text{if } x \in T, \\ 0 & \text{otherwise.} \end{cases}$$

Then, $|p - q| = p(T) - q(T) = p(x_T = 1) - q(x_T = 1)$. We shall prove:

$$\begin{aligned} \frac{p(x)}{q(x)} &\geq \frac{p(x_T)}{q(x_T)} \\ &\geq \frac{2}{\ln 2} \cdot (p(x_T = 1) - q(x_T = 1))^2 = \frac{2}{\ln 2} \cdot |p - q|^2. \end{aligned}$$

The first inequality follows from the chain rule for divergence. It only remains to prove the second inequality. Suppose $p(x_T = 1) = \epsilon \geq q(x_T = 1) = \gamma$. Then we shall show that

$$\epsilon \log \frac{\epsilon}{\gamma} + (1 - \epsilon) \log \frac{1 - \epsilon}{1 - \gamma} - \frac{2}{\ln 2} \cdot (\epsilon - \gamma)^2 \tag{6.1}$$

is always non-negative. (6.1) is 0 when $\epsilon = \gamma$, and its derivative with respect to γ is

$$\begin{aligned} &\frac{-\epsilon}{\gamma \ln 2} + \frac{1 - \epsilon}{(1 - \gamma) \ln 2} - \frac{4(\gamma - \epsilon)}{\ln 2} \\ &= \frac{\gamma - \epsilon\gamma - \epsilon + \epsilon\gamma}{\gamma(1 - \gamma) \ln 2} - \frac{4(\gamma - \epsilon)}{\ln 2} \\ &= \frac{(\gamma - \epsilon)}{\ln 2} \left(\frac{1}{\gamma(1 - \gamma)} - 4 \right). \end{aligned}$$

Since $\frac{1}{\gamma(1-\gamma)}$ is always at most 4, the derivative is non-positive when $\gamma < \epsilon$, and non-negative when $\gamma > \epsilon$. This proves that (6.1) is always non-negative, as required. □

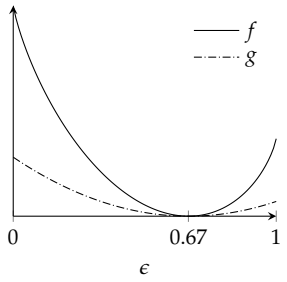


Figure 6.4: $f = \epsilon \log \frac{\epsilon}{2/3} + (1 - \epsilon) \log \frac{1-\epsilon}{1/3}, g = \frac{2}{\ln 2}(\epsilon - 2/3)^2$.

Pinsker's inequality implies that two variables that have low information with each other cannot affect each other's distributions by much:

Corollary 6.7. *If A, B are random variables then on average over b, $p(a|b) \overset{\epsilon}{\approx} p(a)$, where $\epsilon = \sqrt{\frac{\ln 2 \cdot \mathbf{I}(A:B)}{2}}$.*

Another useful corollary is that conditioning on a low entropy random variable cannot change the distribution of many other independent random variables:

Corollary 6.8. *Let $A_1, \ldots, A_n$ be independent random variables, and B be jointly distributed. Let $i \in [n]$ be uniformly random and independent of all other variables. Then on average over b, i, $p(a_i|b) \overset{\epsilon}{\approx} p(a_i)$, where $\epsilon \leq \sqrt{\frac{\mathbf{H}(B) \ln 2}{2n}}$.*

Proof. By subadditivity, we have:

$$\begin{aligned}\mathbf{H}(B)/n &\geq \mathbf{I}(A_1,\ldots,A_n : B)/n \\ &\geq (1/n)\sum_{j=1}^{n}\mathbf{I}(A_j : B).\end{aligned}$$

Thus we get that for a uniformly random coordinate i,

$$\mathbb{E}[\mathbf{I}(A_i : B)] \leq \mathbf{H}(B)/n.$$

The bound then follows from Corollary 6.7. □

Some Examples from Combinatorics

THE ENTROPY FUNCTION has found many applications in combinatorics, where it can be used to give simple proofs. Here we give a few examples that illustrate its power.

On the Size of Projections

Let S be a set of n^3 points in $\mathbb{R}^3$, and let S_{xy}, S_{yz}, S_{xz} denote the projections of S onto the xy, yz, xz planes.

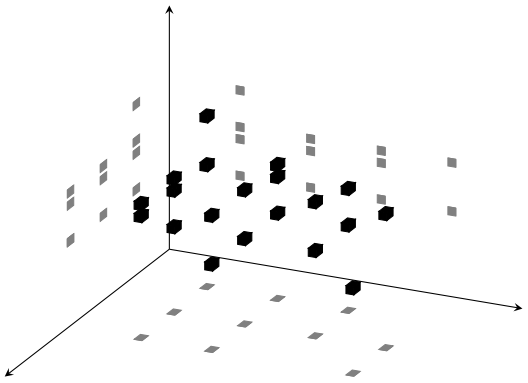


Figure 6.5: A set in $\mathbb{R}^3$ projected to the three planes.

Claim 6.9. *One of the three projections must have size at least n^2.*

Proof. Let X, Y, Z be the coordinates of a uniformly random point from S. By Shearer's inequality,

$$\frac{\mathbf{H}(XY)+\mathbf{H}(YZ)+\mathbf{H}(XZ)}{3} \geq \frac{2}{3}\cdot\mathbf{H}(XYZ) = 2\log n,$$

so one of the first three terms must be at least $2\log n$, proving that the projection must be of size at least n^2. □

On the Size of Triangle Intersecting Graphs

Suppose $\mathcal{F}$ is a family of subsets of $[n]$ such that any two sets from $\mathcal{F}$ intersect. Then we claim:

Claim 6.10. $|\mathcal{F}| \leq 2^{n-1}$.

Proof. For any set $T \in \mathcal{F}$, its complement cannot be in $\mathcal{F}$. So only half of all the sets can be in $\mathcal{F}$. □

Let $\mathcal{G}$ be a family of graphs on n vertices such that every two graphs intersect in a triangle[5]. Such a family can be obtained by choosing a fixed triangle, which gives $2^{\binom{n}{2}}/8$ graphs. This bound is known to be tight[6], but here we give a simple argument that provides a partial converse[7]:

Theorem 6.11. $|\mathcal{G}| \leq 2^{\binom{n}{2}}/4$.

Proof. Let G be a uniformly random graph from the family. G can be described by a binary vector of length $\binom{n}{2}$, where each bit indicates whether a particular edge is present or not. Let S be a random subset of $n/2$ of the vertices, and let G_S denote the graph obtained by deleting all edges that go from S to the complement of S. Since the probability that any particular edge is retained is exactly $1/2$, Shearer's inequality gives $\mathbb{E}_S[\mathbf{H}(G_S \mid S)] \geq \mathbf{H}(G)/2$.

Now any two graphs G, G' in the family intersect in a triangle, so we must have that G_S, G'_S must share an edge in common, no matter what S is, because at least one of the edges of the triangle will not be thrown away in the above process. But this means that the number of such projections is at most half of all possible projections, by Claim 6.10. Writing $e(S) = \binom{|S|}{2} + \binom{n-|S|}{2}$ for the total number of edges possible in the graph G_S, this means that $\mathbf{H}(G_S) + 1 \leq e(S)$. In expectation exactly half of the edges contribute to $e(S)$, so we get:

$$\frac{1}{2} \cdot \binom{n}{2} = \mathop{\mathbb{E}}_S[e(S)] \geq \mathop{\mathbb{E}}_S[\mathbf{H}(G_S \mid S)] + 1 \geq \frac{1}{2} \cdot \mathbf{H}(G) + 1,$$

and so $\mathbf{H}(G) \leq \binom{n}{2} - 2$, which implies that $|\mathcal{G}| \leq 2^{\binom{n}{2}-2} = 2^{\binom{n}{2}}/4$. □

An Isoperimetric Inequality in the Hypercube

The *hypercube* is the graph who vertex set is $\{0,1\}^n$ and the edges connect two vertices that disagree in exactly one coordinate. The hypercube contains 2^n vertices and $2^n n/2$ edges. Here we give a tight bound on the number of edges in any subset of the vertices[8]:

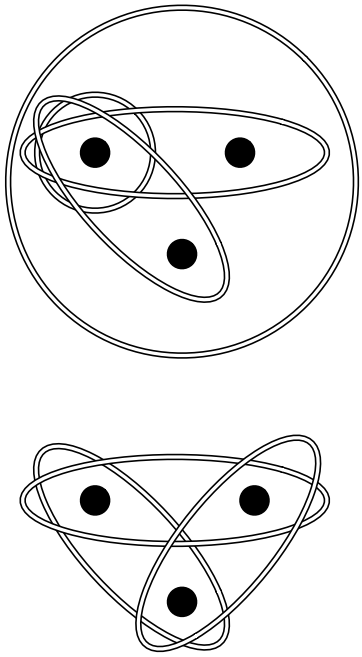


Figure 6.6: Two intersecting families of sets on a universe of size 3.

[5] A cycle of length 3

[6] Ellis et al., 2010

[7] Chung et al., 1986

Very similar ideas can be used to show that any family of graphs that intersects in an r-clique can be of size at most $2^{\binom{n}{2}}/2^{r-1}$. See Exercise 6.4.

[8] Samorodnitsky. Ref?

Theorem 6.12. *If $S \subseteq \{0,1\}^n$, the number of edges contained in S is at most $\frac{|S| \log |S|}{2}$.*

Proof. Let $e(S)$ denote the number of edges in S. Let X be a uniformly random element of S. Then for any vertex $x \in S$ and y such that x, y is an edge of the hypercube where x, y disagree in the i'th coordinate, we have

Here X_{-i} denotes $X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$.

$$\mathbf{H}(X_i \mid X_{-i} = x_{-i}) = \begin{cases} 1 & \text{if } (x,y) \text{ is an edge that is contained in } S, \\ 0 & \text{otherwise.} \end{cases}$$

So $\sum_{x \in S, i \in [n]} \mathbf{H}(X_i \mid X_{-i} = x_{-i}) = 2e(S)$, since each edge is counted twice. By subadditivity,

$$\log |S| = \mathbf{H}(X) = \sum_{i=1}^{n} \mathbf{H}(X_i \mid X_{<i}) \geq \sum_{i=1}^{n} \mathbf{H}(X_i \mid X_{-i}) = \frac{2e(S)}{|S|},$$

proving that $e(S) \leq \frac{|S| \log |S|}{2}$. □

Lower bound for Indexing

WE NOW HAVE ENOUGH information theory tools to prove some lower bounds in communication complexity. Suppose Alice has a random n bit string x, and Bob is given a random index $i \in [n]$. The goal of the players is to compute the i'th bit, x_i, but the protocol must start with a message from Alice to Bob, and then Bob must output the answer. We prove that $\Omega(n)$ bits of communication are necessary, even if the parties are only looking for an average-case protocol.

If Bob could tell Alice i in the first step, that would give a $\log n$ bit protocol.

Proving a deterministic lower bound for this problem is easy: after Alice's message, Bob must know the entire n-bit string. So Alice must send n bits.

Suppose there is a protocol for this problem where Alice sends a message M that is ℓ bits long. Then by Corollary 6.8, on average over the choice of m and a random coordinate i, $p(x_i|m) \overset{\epsilon}{\approx} p(x_i)$, with $\epsilon = \sqrt{\frac{\ell \ln 2}{2n}}$. Since $p(x_i)$ is uniform for each i, the probability that Bob makes an error in the i'th coordinate must be at least $1/2 - |p(x_i|m) - p(x_i)|$. So the probability that Bob makes an error is at least $1/2 - \sqrt{\frac{\ell \ln 2}{2n}}$, proving that at least $\Omega(n)$ bits must be transmitted if the protocol has a small probability of error.

The square-root dependence is tight: If Alice sends the majority of all her bits, that bit is equal to a random coordinate with probability $1/2 + \Omega(1/\sqrt{n})$. See Exercise 6.2.

Note that if Alice has a random set from a family of sets of size $2^{\Omega(n)}$, the lower bound for indexing would still hold.

Randomized Communication of Disjointness

ONE OF THE TRIUMPHS of information theory is its ability to prove optimal lower bounds on the randomized communication complexity of functions like disjointness[9], which we do not know how to prove any other way.

[9] Kalyanasundaram and Schnitger, 1992; Razborov, 1992; Bar-Yossef et al., 2004; and Braverman and Moitra, 2013

This result is especially impactful because many other lower bounds in other models (more in Part II) are consequences of Theorem 6.13.

Theorem 6.13. *Any randomized protocol that computes disjointness function with error $1/2 - \epsilon$ must have communication $\Omega(\epsilon^2 n)$.*

Obstacles to Proving Theorem 6.13

The natural way to prove lower bounds on randomized protocols is to find a *hard distribution* on the inputs, such that any protocol with low communication must make an error a significant fraction of the time. This is the approach we took when we proved lower bounds on the inner-product function (Theorem 5.6), and the same distribution works to understand the pointer-chasing problem (Theorem 6.16). In those cases, the uniform distribution on inputs is a hard distribution. But the uniform distribution is not a hard distribution for disjointness: two uniformly random sets X, Y will intersect with very high probability, so the protocol can output 0 without communicating and still have very low error. In fact, it can be shown that *any* distribution where X and Y are independent cannot be used to prove a strong lower bound. So we must use a hard distribution where X, Y are correlated.

By Theorem 3.3, Theorem 6.13 is equivalent to the existence of such a hard distribution.

A natural distribution to use, given these constraints, is a convex combination of two uniformly random disjoint sets, and two sets that intersect in exactly one element. Once we restrict our attention to such a distribution, we have a second challenge: the events $i \in X \cap Y$ and $j \in X \cap Y$ are not independent for $i \neq j$. This makes arguments involving subadditivity much harder to carry out. The subtleties in the proof arise from coming up with technical ideas that allow us to circumvent these obstacles.

Intuitively this is because if for typical i, if the enropy $\mathbf{H}(X_i Y_i \mid X_{<i} Y_{<i}) << 1$ then Alice can encode the relevant coordinates of her set (those where there is a good chance of an intersection) with much less than n bits and send it to Bob. On the other hand, if this entropy is typically close to 1, then the sets will intersect with high probability.

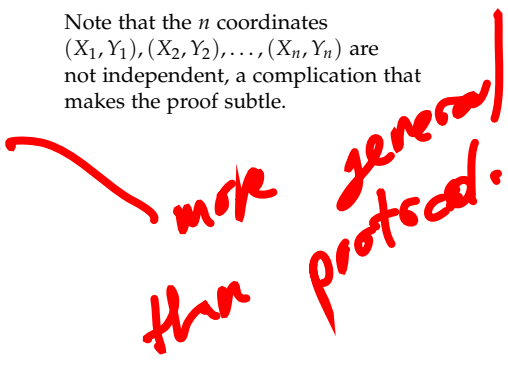
Proving Theorem 6.13

Given a randomized protocol with error $1/2 - \epsilon$, one can make the error an arbitrarily small constant by repeating the protocol $O(1/\epsilon^2)$ times and outputting the majority outcome. So to prove the lower bound, it suffices to show that any protocol with error $< \frac{1}{32}$ must have communication $\Omega(n)$.

We start by defining a hard distribution on inputs. View the sets X, Y as n-bit strings, by setting $X_i = 1$ if and only if $i \in X$. Pick an index $T \in [n]$ uniformly at random, and let X_T, Y_T to be random and independent bits. For $i \neq T$, sample (X_i, Y_i) to be one of $(0,0), (0,1), (1,0)$ with equal probability, and independent of all other pairs (X_j, Y_j). X and Y intersect in at most 1 element, and they intersect with probability $\frac{1}{4}$.

Note that the n coordinates $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ are not independent, a complication that makes the proof subtle.

Let M denote the messages of a deterministic protocol of communication ℓ whose error is at most $1/32$. Let Q denote the random variable $T, X_{<T}, Y_{>T}$. Observe that $X - MQ - Y$, after you fix MQ the



distribution of XY become independent. Moreover, after Q is fixed, the n tuples $(X_1, Y_1), \ldots, (X_n, Y_n)$ become independent.

After fixing Q, X, Y become indepenent: for every q, $p(ab|q) = p(a|q) \cdot p(b|q)$. Since fixing M restricts the inputs to a rectangle, X, Y remain independent after fixing M: for every q, s, $p(ab|qs) = p(a|qs) \cdot p(b|qs)$.

For any q, s, let α_{qq} denote the statistical distribution of $p(x_t|qs)$ from uniform, and let β_{qs} denote the distance of $p(y_t|qs)$ from uniform. Then the probability that the protocol makes an error conditioned on q, s is at least $\frac{1}{4} - \alpha - \beta$, since the probability that the sets will be disjoint is within $\alpha_{qs} + \beta_{qs}$ of $\frac{1}{4}$.

We shall prove that the protocol learns a significant amount of information about x_t (or y_t), even conditioned on the event that $y_t = 0$ (or $x_t = 0$), by showing the following claim:

Claim 6.14. $\mathbb{E}_{p(q,s|y_t=0)}\left[\alpha_{qs}\right] + \mathbb{E}_{p(q,s|x_t=0)}\left[\beta_{qs}\right] \geq \frac{1}{16}$.

Before proving Claim 6.14, we use the subadditivity of information and Pinsker's inequality to show it implies that a linear number of bits were communicated.

We need a technical lemma:

Lemma 6.15. *Let $X = X_1, \ldots, X_n$ and $Y = Y_1, \ldots, Y_n$ be random variables such that the n tuples $(X_1, Y_1), \ldots, (X_n, Y_n)$ are mutually independent. Let M be an arbitrary random variable. Then*

$$\sum_{i=1}^{n} \mathbf{I}\left(X_i : M \mid X_{<i} Y_{\geq i}\right) \leq \mathbf{I}\left(X : M \mid Y\right),$$
$$\sum_{i=1}^{n} \mathbf{I}\left(Y_i : M \mid X_{\leq i} Y_{>i}\right) \leq \mathbf{I}\left(Y : M \mid X\right).$$

Proof. Using the chain rule repeatedly:

$$\begin{aligned}
\sum_{i=1}^{n} \mathbf{I}\left(X_i : M \mid X_{<i} Y_{\geq i}\right) &\leq \sum_{i=1}^{n} \mathbf{I}\left(X_i : M Y_{<i} \mid X_{<i} Y_{\geq i}\right) \\
&= \sum_{i=1}^{n} \mathbf{I}\left(X_i : Y_{<i} \mid X_{<i} Y_{\geq i}\right) + \mathbf{I}\left(X_i : M \mid X_{<i} Y\right) \\
&= \sum_{i=1}^{n} \mathbf{I}\left(X_i : M \mid X_{<i} Y\right) = \mathbf{I}\left(X : M \mid Y\right). && \text{Since } \mathbf{I}\left(X_i : Y_{<i} \mid X_{<i} Y_{\geq i}\right) = 0.
\end{aligned}$$

The second bound is proved similarly. □

Let $\mathcal{D}$ denote the event that X, Y are disjoint. Then we see that $p(xy|\mathcal{D})$ satisfies the assumptions of Lemma 6.15. Moreover T is independent of X, Y, M, once we have conditioned on $\mathcal{D}$, so Lemma

6.15 gives:

$$\begin{aligned}
\frac{\ell}{n} &\geq \mathbf{I}(X : M \mid Y) && \text{Since } M \text{ has at most } \ell \text{ bits.}\\
&\geq \mathbf{I}(X_T : M \mid TX_{<T}Y_{\geq T}\mathcal{D}) && \text{By Lemma 6.15}\\
&= \mathbf{I}(X_T : M \mid QY_T\mathcal{D})\\
&\geq p(x_t = 0|\mathcal{D}) \cdot \mathbf{I}(X_T : M \mid Q, Y_T = 0, \mathcal{D})\\
\Rightarrow \frac{3\ell}{2n} &\geq \mathbf{I}(X_T : M \mid Q, Y_T = 0, \mathcal{D}) && \text{Since } p(x_t = 0|\mathcal{D}) = 2/3\\
&= \mathbf{I}(X_T : M \mid Q, Y_T = 0). && \text{Since } Y_T = 0 \text{ implies } \mathcal{D}.
\end{aligned}$$

By Pinsker's inequality (Corollary 6.7) we get:

$$\begin{aligned}
\sqrt{\frac{3\ell \ln 2}{4n}} &\geq \mathop{\mathbb{E}}_{p(qs|y_t=0)}\left[|p(x_t|qs, y_t = 0) - p(x_t|q, y_t = 0)|\right]\\
&= \mathop{\mathbb{E}}_{p(qs|y_t=0)}\left[|p(x_t|qs) - p(x_t|q)|\right] = \mathop{\mathbb{E}}_{p(qs|y_t=0)}\left[\alpha_{qs}\right]. && \text{Since } p(x_t|q) \text{ is uniform.}
\end{aligned}$$

Using exactly the same argument, we get

$$\sqrt{\frac{3\ell \ln 2}{4n}} \geq \mathop{\mathbb{E}}_{p(qs|x_t=0)}\left[\beta_{qs}\right],$$

and combining this with Claim 6.14 proves that $\sqrt{\frac{3\ell \ln 2}{4n}} \geq \frac{1}{32}$, proving that $\ell \geq \Omega(n)$, as required. It only remains to prove Claim 6.14:

Proof of Claim 6.14. Since the error of the protocol is at most $1/32$, we have

$$\begin{aligned}
\frac{1}{32} &\geq \mathop{\mathbb{E}}_{p(q,s)}\left[\frac{1}{4} - \alpha_{qs} - \beta_{qs}\right]\\
&\geq p(x_t = 0 = y_t) \cdot \mathop{\mathbb{E}}_{p(q,s|x_t=0=y_t)}\left[\frac{1}{4} - \alpha_{qs} - \beta_{qs}\right]\\
&= \frac{1}{4} \cdot \mathop{\mathbb{E}}_{p(q,s|x_t=0=y_t)}\left[\frac{1}{4} - \alpha_{qs} - \beta_{qs}\right]
\end{aligned}$$

which implies that $\mathbb{E}_{p(q,s|x_t=0=y_t)}\left[\alpha_{qs} + \beta_{qs}\right] \geq \frac{1}{8}$. Then

$$\begin{aligned}
&\mathop{\mathbb{E}}_{p(q,s|y_t=0)}\left[\alpha_{qs}\right] + \mathop{\mathbb{E}}_{p(q,s|x_t=0)}\left[\beta_{qs}\right]\\
&\geq p(x_t = 0|y_t = 0) \cdot \mathop{\mathbb{E}}_{p(q,s|x_t=0=y_t)}\left[\alpha_{qs}\right]\\
&+ p(y_t = 0|x_t = 0) \cdot \mathop{\mathbb{E}}_{p(q,s|x_t=0=y_t)}\left[\beta_{qs}\right]\\
&\geq \frac{1}{16}.
\end{aligned}$$

□

Lower bound for Number of Rounds

ARE INTERACTIVE PROTOCOLS MORE POWERFUL than protocols that do not have much interaction?[10] Here we show that a protocol with more rounds can have significantly less communication than a protocol with fewer rounds.

[10] Yao, 1983; Duris et al., 1987; Halstenberg and Reischuk, 1993; and Nisan and Wigderson, 1993

In the k step pointer-chasing problem, Alice and Bob each have a string $x, y \in [n]^n$. Define $1 = z_0, z_1, z_2, \ldots$ using the rule

$$z_i = \begin{cases} x_{z_{i-1}} & \text{if } i \text{ is odd,} \\ y_{z_{i-1}} & \text{if } i \text{ is even.} \end{cases}$$

The goal of the parties is to output whether or not $z_k > n/2$.

There is an obvious deterministic protocol that takes k rounds and $k \log n$ bits of communication: in each step one of the players announces $z_1, z_2, \ldots, z_k$. There is a randomized protocol with $k-1$ rounds and $O((k + n/k) \log n)$ bits of communication. In the first step, Alice and Bob each announce the values of x_i, y_i, for $i \leq 10n/k$. Alice and Bob then continue to use the deterministic protocol, but do not communicate if one of the values they need has already been announced. In expectation, this protocol will have $k + 1 - 10$ rounds[11].

Since the information about the number of rounds is lost once we move to viewing a protocol as a partition into rectangles, it seems hard to prove a separation between a few rounds and many rounds using the techniques we have seen before. A protocol with low communication will have a large rectangle, so we cannot bound the size of rectangles to get a separation between interactive protocols and non-interactiveprotocols.

[11] It can be shown that this randomized protocol will have $< k$ rounds with high probability. Indeed, the probability that none of the announced values help to save a round of communication is exponentially small in k, as long as $\Omega(k)$ of the values z_i are distinct. For a uniformly random input, most of the z_i's will be distinct with high probability.

We shall prove that any randomized or deterministic protocol with $k-1$ rounds must have much more communication.

Let $x, y \in [n]^n$ be uniformly distributed, and let $m_{\leq k-1}$ denote the first $k-1$ messages of a protocol computing whether or not $z_k > n/2$, and the communication complexity of the protocol is ℓ. The key idea here is quite similar to the lower bound for the indexing problem. We will try to argue by induction that z_k remains random even after conditioning on $m_{<k}$. Suppose k is even. Then intuitively, if Alice sends the message m_{k-1}, we will have shown by induction that z_{k-1} is close to random conditioned on $m_{<k-1}$, but now m_{k-1} is independent of z_k after fixing $m_{<k-1}$. So $p(z_k|m_{<k})$ is distributed like a random coordinate of $y|m_{<k-1}$, which is likely to be close to uniform. On the other hand, if Bob sends m_{k-1}, then z_{k-1} is again close to uniform conditioned on $m_{<k-1}$ by induction, and now z_{k-1} is independent of y, m_{k-1}, so $p(z_k|m_{<k})$ is distributed like a random coordinate of $y|m_{<k}$, which is again close to uniform.

Theorem 6.16. *Any randomized $k-1$ round protocol for the k-step pointer chasing problem that is correct with probability $1/2 + \epsilon$ requires $\frac{\epsilon^2 n}{(k-1)^2} - k \log n$ bits of communication.*

Theorem 6.16 actually proves that the communication is at least $\Omega(n/k^2)$ in the randomized setting with $k-1$ rounds. This is because when $k < \sqrt[3]{n/\log n}$, $\frac{\epsilon^2 n}{4(k-1)^2} - k \log n = \Omega(n/k^2)$, and when $k \geq \sqrt[3]{n/\log n}$, the communication must be at least k which is again $\Omega(n/k^2)$.

Proof. The proof will proceed by induction. We shall show that z_k remains close to uniformly random, even conditioned on the

messages that have been sent in the first $k-1$ rounds. Initially, z_1 is uniform (when we do not condition on any of the messages).

Let r_k denote the random variable $m_1, \ldots, m_k, z_1, \ldots, z_k$. We shall prove by induction on k that on average over r_{k-1}, $p(z_k|r_{k-1})$ is ϵ-close to uniform, with $\epsilon \leq (k-1)\sqrt{\frac{\ell+\log n}{n}}$. Rearranging, this would imply that $\ell \geq \frac{\epsilon^2 n}{(k-1)^2} - k\log n$.

The case when $k = 1$ is trivial. Suppose $k \geq 2$ and k is even[12]. Since r_{k-2}, m_{k-1} contains at most $\ell + k\log n$ bits of information, Corollary 6.8 implies that if i is a uniformly random coordinate independent of all other variables, then on average over i, r_{k-2}, m_{k-1},

$$p(y_i|r_{k-2}) \overset{\epsilon'}{\approx} p(y_i) \overset{\epsilon'}{\approx} p(y_i|m_{k-1}, r_{k-2}),$$

where $\epsilon' = \sqrt{\frac{\ell+k\log n}{n}}$. There are two cases to consider:

Bob sends the message m_{k-1} In this case, after fixing r_{k-2}, z_{k-1} is independent of y_i for every i. By induction, $p(z_{k-1}|r_{k-2})$ is ϵ-close to uniform, with $\epsilon = (k-2)\sqrt{\frac{\ell+k\log n}{n}}$. So on average over r_{k-1}, i,:

$$p(z_k|r_{k-1}) = p(y_{z_{k-1}}|m_{k-1}, r_{k-2}) \overset{\epsilon}{\approx} p(y_i|m_{k-1}, r_{k-2}) \overset{\epsilon'}{\approx} p(y_i).$$

Alice sends the message m_{k-1} In this case, $p(y_i|r_{k-1}) = p(y_i|r_{k-2})$, since after fixing r_{k-2}, y_i is independent of m_{k-1}, z_{k-1}. So on average over r_{k-1}, i:

$$p(z_k|r_{k-1}) = p(y_{z_{k-1}}|r_{k-2}) \overset{\epsilon}{\approx} p(y_i|r_{k-2}) \overset{\epsilon'}{\approx} p(y_i).$$

Both of these bounds imply that $p(z_k|r_{k-1})$ is $(k-1)\sqrt{\frac{\ell+k\log n}{n}}$-close to uniform, as required. □

Very similar intuitions can be used to show that the deterministic communication of the pointer-chasing problem is $\Omega(n)$ if fewer than k rounds of communication are used.

Theorem 6.17. *Any $k-1$ round deterministic protocol that computes the k-step pointer-chasing problem requires $\frac{n}{16} - k$ bits of communication.*

Proof. Consider any $k-1$ round deterministic protocol with communication complexity $\ell \leq \frac{n}{16} - k$, and let $m_1, \ldots, m_{k-1}$ denote the messages of the protocol. Let r_i denote $z_0, z_1, \ldots, z_i, m_1, \ldots, m_i$. Let p denote the uniform distribution on inputs to the protocol. We shall show by induction on i that there is a fixed value of r_i such that

- $z_0, z_1, \ldots, z_i$ are all distinct.
- $p(z_{i+1}|r_i)$ is ϵ-close to uniform, with $\epsilon = 2\sqrt{\frac{\ell+k}{n}} \leq 1/4$.

[12] The proof is exactly the same when k is odd.

Fact: If i, j, a are independent, and $p(i) \overset{\gamma}{\approx} p(j)$, we have $p(a_i) \overset{\gamma}{\approx} p(a_j)$. See the Conventions chapter of the book for a proof.

- $p(m_{\leq i}|z_{\leq i}) \geq 2^{-|m_{\leq i}|-i}$.

The first property applied to $i = k-1$ shows that the protocol cannot be correct, since r_{k-1} contains all the messages in the first k rounds, and so $p(z_k|r_{k-1})$ cannot be close to uniform.

When $i = 0$, the claims are trivially satisfied. Now suppose $i > 0$ is even[13], so $z_{i+1} = x_{z_i}$. By induction, there exists a setting of r_{i-1} that satisfies the given conditions. We only need to show that there exists a setting of values for z_i, m_i to append to r_{i-1} to obtain the setting of r_i that we want. There are two cases:

[13] The proof is symmetric when i is odd.

Alice sends the $i+1$'st message In this case, fixing r_{i-1} leaves m_i and z_i independent. Pick m_i by greedily setting each bit of m_i in such a way that the probability of that bit is maximized conditioned on r_{i-1} and all previous bits. This ensures that

$$p(m_i|r_{i-1}) \geq 2^{-|m_i|}.$$

To choose z_i, define

$$B_1 = \{z_0, z_1, \ldots, z_{i-1}\}$$
$$B_2 = \left\{ j : \frac{p(x_j|m_i, r_{i-1})}{p(x_j)} > 4 \cdot \frac{\ell + k}{n} \right\}$$
$$B_3 = \left\{ j : \frac{p(Z_i = j|r_{i-1})}{p(Z_i = j|z_{\leq i-1})} < 1/2 \right\}$$

We shall prove:

Claim 6.18. $|B_1 \cup B_2 \cup B_3| < n$.

Proof. Obviously, $|B_1| \leq k - 1 < n/16 - \ell \leq n/16$.

We have $|B_3| \leq 2\epsilon n \leq n/2$, or else we would have

$$p(Z_i \in B_3|z_{\leq i-1}) - p(Z_i \in B_3|r_{i-1}) > 2\epsilon - \epsilon = \epsilon,$$

contradicting the fact that $p(z_i|r_{i-1})$ is ϵ-close to uniform.

We shall prove that $|B_2 - B_1| \leq n/4$. Observe that:

$$\begin{aligned} |B_2 - B_1| \cdot 4 \cdot \frac{\ell + k}{n} &\leq \sum_{j \in B_2 - B_1} \frac{p(x_j|m_i, r_{i-1})}{p(x_j)} \\ &= \sum_{j \in B_2 - B_1} \frac{p(x_j|m_{\leq i}, z_{\leq i-1})}{p(x_j|z_{\leq i-1})} \\ &\leq \frac{p(x_{[n]-B_1}|m_{\leq i}, z_{\leq i-1})}{p(x_{[n]-B_1}|z_{\leq i-1})}. \end{aligned}$$

Since x_j is independent of $z_{\leq i-1}$ for all $j \notin B_1$.

By Fact 6.4. Here $x_{[n]-B_1}$ denotes x projected to the coordinates that are not in B_1.

By the choice of m_i, we have

$$\begin{aligned} p(m_{\leq i}|z_{\leq i-1}) &= p(m_i|r_{i-1}) \cdot p(m_{\leq i-1}|z_{\leq i-1}) \\ &\geq 2^{-|m_i|} \cdot 2^{-|m_{\leq i-1}|-i+1} \geq 2^{-\ell-k}, \end{aligned}$$

So we can apply Fact 6.3 to conclude that

$$\frac{p(x_{[n]-B_1}|m_{\leq i}, z_{\leq i-1})}{p(x_{[n]-B_1}|z_{\leq i-1})} \leq \ell + k,$$

giving that $|B_2 - B_1| \leq n/4$. Thus $|B_1 \cup B_2 \cup B_3| < n/16 + n/2 + n/4 = n$. □

Set z_i to be an arbitrary element outside of $B_1 \cup B_2 \cup B_3$. This completes the description of r_i. Since $z_i \notin B_1$, $z_0, \ldots, z_i$ are distinct. Since after fixing m_i, r_i, x is independent of y, the distribution of $p(x_{z_i}|r_i)$ is the same as the distribution of $p(x_{z_i}|m_i r_{i-1})$. Thus it is $2\sqrt{\cdot \frac{\ell+k}{n}}$-close to uniform by Pinsker's inequality and the fact that $z_i \notin B_3$.

Finally, we have:

$$\begin{aligned} p(m_{\leq i}|z_{\leq i}) &= p(m_i|r_{i-1}) \cdot p(m_{\leq i-1}|z_{\leq i}) \\ &\geq 2^{-|m_i|} \cdot \frac{p(z_i|m_{\leq i-1}, z_{\leq i-1})}{p(z_i|z_{\leq i-1})} \cdot p(m_{\leq i-1}|z_{\leq i-1}) \\ &\geq 2^{-|m_{\leq i}|-i} \cdot (1/2) = 2^{-|m_{\leq i}|-(i+1)} \end{aligned}$$

by the choice of m_i, and the fact that $z_i \notin B_2$.

$$\begin{aligned} & p(m_{\leq i-1}|z_{\leq i}) \\ &= \frac{p(m_{\leq i-1}, z_i|z_{\leq i-1})}{p(z_i|z_{\leq i-1})} \\ &= \frac{p(z_i|m_{\leq i-1}, z_{\leq i-1})}{p(z_i|z_{\leq i-1})} \cdot p(m_{\leq i-1}|z_{\leq i-1}) \end{aligned}$$

Bob sends the $i+1$'st message In this case, we pick z_i first. Define the sets:

$$\begin{aligned} B_1 &= \{z_0, z_1, \ldots, z_{i-1}\} \\ B_2 &= \left\{ j : \frac{p(x_j|r_{i-1})}{p(x_j)} > 4 \cdot \frac{\ell+k}{n} \right\} \\ B_3 &= \left\{ j : \frac{p(Z_i = j|r_{i-1})}{p(Z_i = j|z_{\leq i-1})} < 1/2 \right\} \end{aligned}$$

Analogous to Claim 6.18, we have

Claim 6.19. $|B_1 \cup B_2 \cup B_3| < n$.

Proof. $|B_1| \leq n/16$ and $|B_3| \leq n/2$, as proved in Claim 6.18.

We shall prove that $|B_2 - B_1| \leq n/4$. Observe that:

$$\begin{aligned}
|B_2 - B_1| \cdot 4 \cdot \frac{\ell + k}{n} &\leq \sum_{j \in B_2 - B_1} \frac{p(x_j|r_{i-1})}{p(x_j)} \\
&= \sum_{j \in B_2 - B_1} \frac{p(x_j|m_{\leq i-1}, z_{\leq i-1})}{p(x_j|z_{\leq i-1})} && \text{Since } x_j \text{ is independent of } z_{\leq i-1} \text{ for all } j \notin B_1. \\
&\leq \frac{p(x_{[n]-B_1}|m_{\leq i-1}, z_{\leq i-1})}{p(x_{[n]-B_1}|z_{\leq i-1})} && \text{By Fact 6.4.} \\
&\leq \ell + k, && \text{Using } p(m_{\leq i-1}|z_{\leq i-1}) \geq 2^{-\ell-k}, \text{ and Fact 6.3.}
\end{aligned}$$

giving that $|B_2 - B_1| \leq n/4$. Thus $|B_1 \cup B_2 \cup B_3| < n/16 + n/2 + n/4 = n$. □

We let z_i be an element that is not in $B_1 \cup B_2 \cup B_3$, and pick m_i by greedily setting each bit of m_i in such a way that the probability of that bit is maximized conditioned on r_{i-1}, z_i and all previous bits.

Clearly, $z_0, \ldots, z_i$ are all distinct. $p(x_{z_i}|r_i)$ has the same distribution as $p(x_{z_i}|r_{i-1})$, which is $2\sqrt{\frac{\ell+k}{n}}$-close to uniform by Pinsker's inequality and the fact that $z_i \notin B_2$.

Finally, we have

$$\begin{aligned}
p(m_{\leq i}|z_{\leq i}) &\geq p(m_i|r_{i-1}, z_i) \cdot p(m_{\leq i-1}|z_{\leq i}) \\
&\geq 2^{-|m_i|} \cdot \frac{p(z_i|r_{\leq i-1})}{p(z_i|z_{\leq i-1})} \cdot p(m_{\leq i-1}|z_{\leq i-1}) \\
&\geq 2^{-|m_{\leq i}|-(i+1)},
\end{aligned}$$

as required.

□

Lower bounds on Non-Negative Rank

Exercise 6.1

Show that for anys two joint distributions $p(x, y), q(x, y)$ with same support, we have

$$\mathop{\mathbb{E}}_{p(y)} \left[\frac{p(x|y)}{p(x)} \right] \leq \mathop{\mathbb{E}}_{p(y)} \left[\frac{p(x|y)}{q(x)} \right].$$

Exercise 6.2

Suppose n is odd, and $x \in \{0,1\}^n$ is sampled uniformly at random from the set of strings that have more 1's than 0's. Use Pinsker's

inequality to show that the expected number of 1's in x is at most $n/2 + O(\sqrt{n})$.

Exercise 6.3

Let X be a random variable supported on $[n]$ and $g : [n] \to [n]$ be a function. Prove that

$$\Pr[X \neq g(X)] \geq \frac{\mathbf{H}(X \mid g(X)) - 1}{\log n}.$$

Use this bound to show that if Alice has a uniformly random vector $y \in [n]^n$, and Bob has uniformly random input $i \in [n]$, and Alice sends Bob a message M with that contains ℓ bits, the probability that Bob guesses y_i is at most $\frac{1+\ell/n}{\log n}$.

Use the fact that $\alpha \log \alpha \geq \frac{-\log e}{e} \geq -1$, for $\alpha > 0$.

Exercise 6.4

Let $\mathcal{G}$ be a family of graphs on n vertices, such that every two vertices in the graph share a clique on r vertices. Show that the number of graphs in the family is at most $2^{\binom{n}{2}}/2^{r-1}$.

Hint: Partition the graph into r parts uniformly at random and throw away all edges that do not stay within a part. Analyse the entropy of the resulting distribution on graphs from the family.

7
Compressing Communication

When X and Y are the inputs to a communication protocol, and M denotes the messages of the protocol, $\mathbf{I}(X : M \mid Y)$ measures the amount of information the messages of the protocol reveal to Bob about X. Similarly $\mathbf{I}(Y : M \mid X)$ measures the amount of information the messages of the protocol reveal to Alice about Y. Indeed, we have:

Lemma 7.1. *If X, Y are inputs to a deterministic protocol and M denotes the messages of the protocol, then $\mathbf{I}(X : M \mid Y)$ is at most the expected number of bits sent by Alice in the protocol, and $\mathbf{I}(Y : M \mid X)$ is at most the expected number of bits sent by Bob in the protocol.*

Proof. We prove the first inequality, since the second has the same proof. By the chain rule, we can write:

$$\mathbf{I}(X : M \mid Y) = \sum_i \mathbf{I}(X : M_i \mid Y M_{<i}),$$

where here $M_1, M_2, \ldots$ are the bits of M. Now for any fixing of $m_{<i}$,

$$\mathbf{I}(X : M_i \mid Y m_{<i}) \leq \begin{cases} 1 & \text{if } M_i \text{ is to be transmitted by Alice,} \\ 0 & \text{if } M_i \text{ is to be transmitted by Bob,} \end{cases}$$

since in the first case the bit can have at most 1 bit of information, and in the second case M_i is independent of X once we fix $M_{<i}$ and Y. Taking the expectation over $m_{<i}$ gives

$$\mathbf{I}(X : M_i \mid Y M_{<i}) \leq \Pr[M_i \text{ is transmitted by Alice}],$$

proving that $\mathbf{I}(X : M \mid Y)$ is at most the expected number of bits transmitted by Alice, by linearity of expectation. □

Part II

Applications

8
Circuits and Branching Programs

Although communication complexity studies the amount of communication needed between two parties that are far apart, it has had quite an impact in understanding local computation. In this chapter, we illustrate some of the salient work in this direction by focussing on two models: boolean circuits and branching programs.

Boolean Circuits

A *boolean circuit* is a directed acyclic graph whose vertices (called gates) are associated with boolean operators or input variables. Every gate with in-degree 0 corresponds to an input variable or its negation, and all other gates compute either the AND or the OR of the inputs that feed into them. Usually the fan-in of the gates is restricted to being at most 2, and in the rest of the discussion we adopt the convention that the fan-in of each gate is at most 2 unless we explicitly state otherwise. The circuit computes a function $f : \{0,1\}^n \to \{0,1\}$ if some gate in the circuit evaluates to f. A *formula* is a circuit whose underlying graph is a tree. The size of a circuit is the number of gates, and the depth of the circuit is the length of the longest path in the graph. When the circuit does not use any negated variables, it is called a *monotone* circuit.

It is well known that every (monotone) function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a (monotone) circuit of depth n and size at most $O(2^n/n)$.

The importance of understanding boolean circuits stems from the fact that they are a universal model of computation. Any function that can be computed by an algorithm in $T(n)$ steps can also be computed by circuits of size $\tilde{O}(T(n))$. Thus to prove lower bounds on the time complexity of algorithms, it is enough to prove that there are no small circuits that can carry out the computation. However, we know

It also makes sense to consider circuits where every gate has fan-in 2 and computes an arbitrary function of its inputs. This only changes the size and depth of the circuit by a constant factor, since any function on 2 bits can be computed by a small circuit using only AND and OR gates.

The size of the circuit captures the total number of basic operations needed to evaluate it. The depth captures the number of *parallel* steps needed to evaluate it: if a circuit has size s and depth d, then it can be evauated by s processors in d time steps.

A super-polynomial lower bound on the circuit size of an NP problem would imply that P $\neq$ NP, resolving the most famous open problem in computer science.

of no explicit function (even outside NP) for which we can prove a super-linear lower bound, highlighting the difficulty in proving lower bounds on algorithms. In contrast, counting arguments imply that almost every function requires circuits of exponential size.

The number of circuits of size s can by bounded by $2^{O(s \log s)}$, while the number of functions f is 2^{2^n}, so if $s \ll 2^n/n$, one cannot hope to compute every function with a circuit of size s.

Karchmer-Wigderson Games

EVERY BOOLEAN FUNCTION DEFINES a communication problem via its *Karchmer-Wigderson* game[1]. The game defined by $f : \{0,1\}^n \to \{0,1\}$ is the communication problem where Alice gets $x \in f^{-1}(0)$, and Bob gets $y \in f^{-1}(1)$. The goal of Alice and Bob is to compute an index i such that $x_i \neq y_i$. When f is *monotone* (namely $f(y) \geq f(x)$ whenever $x \geq y$ coordinate by coordinate), one can define the *monotone* Karchmer-Wigderson game to be the problem where the inputs are x, y as before, but now the players are required to output i such that $x_i < y_i$.

[1] Karchmer and Wigderson, 1990

If there is a circuit computing f of depth d, then the cost of the associated game is at most d. If f is computed as $f = g \wedge h$, then either $g(x) = 0$ or $h(x) = 0$, while $g(y) = h(y) = 1$. Alice can announce whether $g(x)$ or $h(x)$ is 0, and the parties can continue the protocol using g or h. Similarly if $f = g \vee h$, then either $g(y) = 1$ or $h(y) = 1$, while $g(x) = h(x) = 0$. Bob can announce whether $g(x) = 1$ or $h(x) = 1$. The parties then continue with either g or h. After at most d steps, the parties will have identified an index i for which $x_i \neq y_i$. When the circuit is monotone (has no negations), the above simulation finds an index i such that $x_i = 0, y_i = 1$.

Indeed every AND gate corresponds to a node in the protocol tree where Alice speaks, and every OR gate corresponds to a node where Bob speaks. Moreover, a circuit of size s gives a protocol which has at most s vertices.

Conversely, we have:

Lemma 8.1 proves that any protocol that solves the game on the input sets A, B can be viewed as solving the game on sets A', B' such that $A \subseteq A', B \subseteq B'$, and A', B' partition $\{0,1\}^n$.

Lemma 8.1. *If $A, B \subseteq \{0,1\}^n$ are disjoint non-empty sets and π is a protocol that solves the (monotone) Karchmer-Wigderson game on A, B, such that every node of π is reachable by some input in $A \times B$. Then there is a (monotone) boolean function $f : \{0,1\}^n \to \{0,1\}$ such that $f(A) = 0$, $f(B) = 1$, and a circuit whose underlying graph can be obtained by replacing every node of π where Alice speaks with an* AND *gate, every node where Bob speaks with an* OR *gate and every output node with the corresponding variable or its negation.*

Proof. We prove the lemma by induction on the number of nodes in the protocol.

The protocol has only one node with output i, we must have that

$x_i \neq y_i$ (or $x_i = 0, y_i = 1$ in the monotone case) for every $x \in A, y \in B$. Thus setting f to be the i'th variable or its negation works.

For protocols with more nodes, suppose without loss of generality that Alice speaks first. Then her message partitions the set A into two disjoint sets $A = A_0 \cup A_1$. Both sets must be non-empty, since every node in the protocol is reachable by assumption. By induction the two children of the root correspond to boolean functions f_0 and f_1. Consider the circuit that takes the AND of the two gates obtained inductively, and denote the function it computes by f. Then for all $y \in B$, $f(y) = f_0(y) \wedge f_1(y) = 1 \wedge 1 = 1$. For all $x \in A$, either $x \in A_0$ or $x \in A_1$. In either case $f(x) = f_0(x) \vee f_1(x) = 0$. □

One immediate consequence of Lemma 8.1 is regarding the circuit depth required to compute the majority and parity functions. In Section 1 and Exercise 1.4, we proved that solving the Karchmer-Wigderson games for these functions requires at least $2\log n - O(1)$ bits of communication. This shows that if the fan-in is at most 2, both of these functions requires circuits of depth $2\log n - O(1)$.

Karchmer-Wigderson Games in few Rounds

Protocols that solve Karchmer-Wigderson games in a few rounds have a particularly nice structure. Suppose we have a game with input sets $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$.

If the game can be solved without communication, then the output i must have the feature that $x_i \neq y_i$. The only way this can happen is if there is a $b \in \{0,1\}$ such that $\mathcal{X} = \{x : x_i = b\}, \mathcal{Y} = \{y : y_i \neq b\}$.

A *restriction* of the input is a string $\rho \in \{0,1,*\}$. Its size $|\rho| = |\{i : \rho_i \neq *\}|$ is the number of coordinates which it *restricts*, namely the number of coordinates that are 0 or 1. An input $x \in \{0,1\}^n$ can be thought of as a restriction of size n. Given two restrictions ρ, α, we write $\rho \parallel \alpha$ if the two are consistent: for every $i \in [n]$, either $\alpha_i = \rho_i$, or $\alpha_i = *$, or $\rho_i = *$. We denote

$$S_\rho = \{x : \rho \parallel x\}$$

If the game can be solved with 1 round of communication, we can show:

Lemma 8.2. *If the Karchmer-Wigderson game can be solved with* 1 *round of communication, where Alice speaks first, then there is* $\rho \in \{0,1,*\}$ *such that* $\mathcal{Y} \subseteq S_\rho$*, and* $\mathcal{X}$ *is disjoint from* S_ρ*.*

Proof. Let $a \in \mathcal{Y}$ be arbitrary, and define the restriction ρ by

$$\rho_i = \begin{cases} a_i & \text{if } i \text{ can be output by the protocol,} \\ * & \text{otherwise.} \end{cases}$$

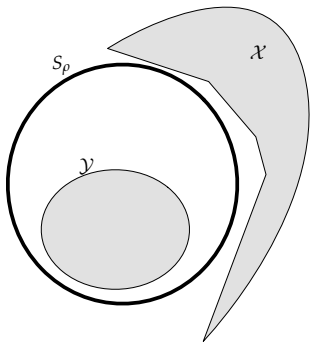


Figure 8.1: Lemma 8.2

We claim that $\mathcal{Y} = \{y : \rho \parallel y\}$. Indeed, if $y \parallel \rho$, then it cannot be in $\mathcal{X}$, otherwise the protocol must make an error on inputs (y, a). So it must be in $\mathcal{Y}$. Conversely, if $y \nparallel \rho$, because say $y_i \neq a_i$ and $\rho_i \neq *$, then when the protocol outputs i it must make an error either when Bob has a as input or Bob has y as input. □

If the game can be solved with a 2-round protocol where Alice speaks first we can show:

Lemma 8.3. *If the Karchmer-Wigderson game can be solved with* 2 *rounds of communication that begins with Alice sending a k-bit message, then there is a set of restrictions R with* $|R| \leq 2^k$*, such that*

$$\mathcal{X} \subseteq \cup_{\rho \in R} S_\rho, \qquad \mathcal{Y} \text{ is disjoint } \cup_{\rho \in R} S_\rho.$$

Proof. For each message m that Alice sends, let $\mathcal{X}_m$ denote the set of inputs that are consistent with that message. Then by Lemma 8.2, we get that there is a restriction ρ^m such that $\mathcal{X}_m \subseteq S_{\rho^m}$ but $\mathcal{Y}$ is disjoint from S_{ρ^m}. Since every $x \in \mathcal{X}$ is consistent with some message m, the sets S_ρ obtained in this way must cover $\mathcal{X}$. □

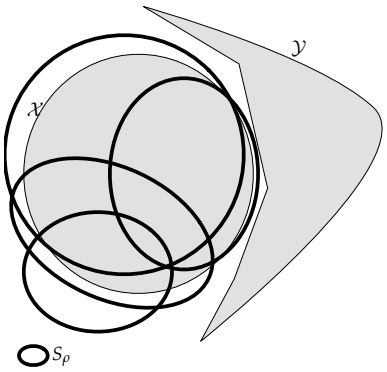


Figure 8.2: Lemma 8.3

Lowerbounds on the Depth of Monotone Circuits

One of topics we do not yet understand in circuit complexity is whether polynomial sized circuits of small depth are strictly weaker than polynomial circuits of larger depth.

We do know, via counting arguments, that there is a constant ϵ such that the set of functions computable by size $s \log s$ circuits is strictly larger than the set of functions computable by size ϵs circuits. Similarly, we know that circuits of depth d compute a bigger set of functions than those computable in depth ϵd.

Open Problem 8.4. *Can every function that is computable using circuits of size polynomial in n be computed by circuits of depth* $O(\log n)$*?*

However, we do know how to prove interesting results when the underlying circuits are monotone.

One of the most well studied combinatorial problems is the problem of finding the largest *matching* in a graph. A matching is a set of disjoint edges. Today, we know of several polynomial time algorithms that can find the matching of largest size in a given graph. This translates to polynomial sized circuits for computing whether or not a graph has a matching of any given size.

Given a graph G on n vertices, define

$$\mathsf{Match}(G) = \begin{cases} 1 & \text{if } G \text{ has a matching of size at least } n/3 + 1, \\ 0 & \text{otherwise.} \end{cases}$$

Since there are polynomial time algorithms for finding matchings, one can obtain polynomial sized circuits that compute Match. However, we do not know of any logarithmic depth circuits that compute

Match, and here we show that there are no such circuits that are also monotone[2]:

Theorem 8.5. *Every monotone circuit computing* Match *has depth* $\Omega(n)$.

Recall Theorem **??**.

Proof. It is enough to prove a lower bound on the communication complexity of corresponding monotone Karchmer-Wigderson game. In the game, Alice gets a graph G which has a matching of size $n/3+1$ and Bob gets a graph H that does not have a matching of size $n/3+1$. Their goal is to compute an edge which is in G, but not in H.

Set $m = n/3$. We shall show that if the parties can solve the monotone Karchmer-Wigderson game using c bits of communication, then they can get a randomized protocol for computing disjointness on a universe of size m. Since any such randomized protocol requires linear communication complexity[3], this gives a communication lower bound of $\Omega(n)$.

[3] By Theorem 6.13.

Suppose Alice and Bob get inputs $X \subseteq [m]$ and $Y \subseteq [m]$. Alice constructs the graph G_X on the vertex set $[3m+2]$ such that for each i, G_X contains the edge $\{3i, 3i-1\}$ if $i \in X$, and has the edge $\{3i, 3i-2\}$ if $i \notin X$. In addition, G_X contains the edge $\{3m+1, 3m+2\}$. Alice's graph consists of $m+1$ disjoint edges. Bob uses Y to build a graph H_Y on the same vertex set as follows. For each $i \in [m]$, Bob connects $3i-2$ to *all* the other $3m+1$ vertices of the graph if $i \in Y$. If $i \notin Y$, Bob connects $3i$ to all the other vertices. Since every edge of H_Y contains exactly one of $\{3i, 3i+1\}$.

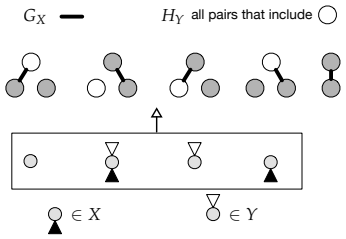


Figure 8.3: The graphs G_X and H_Y

Alice and Bob permute the vertices of the graph randomly and run the protocol promised by the monotone Karchmer-Wigderson game on G_X and H_Y. If X and Y are disjoint, the outcome of the protocol must be the edge corresponding to $\{3m+1, 3m+2\}$. On the other hand, if X and Y intersect in k elements, then the outcome of the protocol is equally likely to be one of the edges that corresponds to these k elements, so the probability that it is $\{3m+1, 3m+2\}$'th edge is at most $1/2$. If Bob sees that the i'th edge is output, then Bob knows that $i \in X \cap Y$. Thus, repeating this experiment a few times, Bob will know whether the sets are disjoint or not with high probability.

This proves that the communication for the Karchmer-Wigderson game must be at least $\Omega(n)$, as required.

□

CONNECTIVITY?

Lowerbounds on Circuits with few Alternations

ANOTHER SPECIAL KIND OF boolean circuit is a circuit where the number of *alternations* in the circuit, namely the number of switches between AND and OR gates on input-output paths is small. Let us say that the circuit has d alternations if there are d such switches.

The number of alternations has a very natural interpretation in terms of communication protocols: it corresponds to the number of rounds of communication generated in the corresponding protocol for the Karchmer-Wigderson game. One can prove that if the number of alternations is small, then the circuit must have exponential size even to compute simple functions like majority and parity[4]:

[4] HÅstad, 1987

Theorem 8.6. *If a circuit with d alternations computes $\sum_{i=1}^{n} x_i \mod 2$, where $x \in \{0,1\}^n$, then it must have $2^{\Omega(n^{\frac{1}{d-1}})}$ gates.*

In Section 6, we saw that protocols with few rounds are in general much weaker than protocols with many rounds. To prove Theorem 8.6, we prove an exception to this moral: we show that if we are allowed to *restrict* the input, then one can simulate any protocol for a Karchmer-Wigderson game with fewer rounds. This is the technical heart of the proof of Theorem 8.6.

Suppose Alice is given an input from a set $\mathcal{X} \subseteq \{0,1\}^n$, and Bob is given an input from the set $\mathcal{Y} \subseteq \{0,1\}^n$. Let $\alpha \in \{0,1,*\}$ be a restriction. Applying the restriction gives us new sets of inputs:

Note that the sets $\mathcal{X}_\alpha, \mathcal{Y}_\rho$ may become empty.

$$\mathcal{X}_\alpha = \{x \in \mathcal{X} : x \parallel \alpha\}, \qquad \mathcal{Y}_\alpha = \{y \in \mathcal{Y} : y \parallel \alpha\}.$$

In Lemma 8.3, we showed that every 2 round protocol is associated with a set of restrictions R such that $\cup_{\alpha \in R} S_\alpha$ covers the inputs. Given a restriction α for which $\mathcal{X}_\alpha, \mathcal{Y}_\alpha$ are not empty, we write π_α to denote the protocol operating on the inputs $\mathcal{X}_\alpha, \mathcal{Y}_\alpha$. It is the protocol obtained by deleting all nodes in π that are not reachable via the inputs $\mathcal{X}_\alpha \times \mathcal{Y}_\alpha$. Let α be a uniformly random restriction of size $\ell = (1-\epsilon)n$.

Claim 8.7. *If $t \leq n/2$, and $\epsilon \leq 1/8$, if γ is a restriction with $|\gamma| \geq t$, $\Pr_\alpha[\gamma \parallel \alpha] \leq (7/8)^t$.*

Proof. The probability that the first coordinate set in γ is consistent with α is at most $\frac{\ell}{2n} + 1 - \frac{\ell}{n}$. Given any fixing of $< t$ coordinates of α, the probability that a new coordinate i that is set in γ is consistent

with α is at most

$$\underbrace{\frac{1}{2}}_{\Pr[\alpha_i=\gamma_i|\alpha_i\neq *]} \cdot \underbrace{\frac{\ell-t}{n}}_{\Pr[\alpha_i\neq *]} + \underbrace{1-\frac{\ell-t}{n}}_{\Pr[\alpha_i=*]} = 1-\frac{\ell-t}{2n}$$

$$\leq \frac{2n-(1-\epsilon)n+n/2}{2n} \qquad \text{since } t < n/2$$

$$= \frac{3}{4}+\frac{\epsilon}{2} \leq \frac{7}{8}. \qquad \text{since } \epsilon \leq 1/8$$

□

The heart of the proof of Theorem 8.6 is the following lemma. Say that a two round protocol where Alice speaks first has restrictions of size t, if every restriction sent by Alice has size at most t.

Lemma 8.8. *If a two-round protocol π has restrictions of size t, then π_α can be simulated by a two-round protocol which has restrictions of size t, where Bob speaks first, except with probability $(8\epsilon t)^t$.*

Before we give the proof of Lemma 8.8, let us use it to prove Theorem 8.6. Set $t = n^{\frac{1}{d-1}}/16$. Suppose the circuit has $s < (8/7)^{n^{\frac{1}{d-1}}/16}/d = (8/7)^t/d$ gates, and is of depth d. Now consider the corresponding Karchmer-Wigderson protocol. We claim:

Claim 8.9. *There is a k-restriction β, with $k < n - n^{\frac{1}{d-1}}/2$, such that π_β can be simulated by a two-round protocol of size t.*

To prove the claim, first apply a random $n/2$-restriction, and then $d-2$ restrictions that set $(1-\epsilon)$ fraction of the variables, with $\epsilon = n^{\frac{-1}{d-1}}$. After $d-2$ such rounds of applying restrictions, the number of variables left alive is $\epsilon^{d-2}n/2 = n^{\frac{1}{d-1}}/2$. There are at most s two round protocols that are executed in π, and by Claim 8.7, the probability that any of them remains of size $> t$ after the first restriction is at most $s(7/8)^t < 1/d$. The probability that any of the subsequent restrictions generates a two-round protocol that has restrictions of size $> t$ is at most $(d-2)s \cdot (8\epsilon t)^t < (4/7)^t$. Thus the probability that we are left with a two-round protocol with restrictions of size t is at least $1 - 1/d - (4/7)^t > 0$. This proves that some choice of restriction gives the claim.

Now any two-round protocol computing solving the Karchmer-Wigderson game after the restriction must use restrictions of size $> t$, since any restriction that the first player sends must completely determine the input in order to determine the parity of his bits. So the original protocol cannot have computed the Karchmer-Wigderson game. This completes the proof of Theorem 8.6.

Proof of Lemma 8.8. Let $R_\alpha \subseteq R$ be the subset of restrictions that are consistent with α. On input $y \in \mathcal{Y}_\alpha$, define ρ^y to be the smallest restriction that distinguishes y from all of the restrictions in R:

$$\rho^y = \underset{\{\rho : \rho \| y, \forall \gamma \in R_\alpha, \rho \nparallel \gamma\}}{\arg\min} |\rho|.$$

By definition, it is enough for Bob to send Alice ρ^y in order to solve the Karchmer-Wigderson game. Once Alice has ρ^y, she can find $\gamma \in R_\alpha$ that is consistent with her input x, which allows her to solve the game, since $x \parallel \gamma \nparallel \rho^y$. We only need to show that ρ^y is of size at most t with high probability.

Claim 8.10. *If $\rho^y_i \neq *$, there is a $\beta \in R_\alpha$ such that $\beta_i \neq *$.*

Proof. If not, we could set $\rho^y_i = *$ to obtain a smaller restriction that is inconsistent with every $\gamma \in R_\alpha$, contradicting the minimality of ρ^y. □

Given ρ^y, α , if $|\rho^y| > t$, we shall compute an $\ell + t$-restriction ρ and *blame* it. ρ will have the property that if $\rho_i \neq *$, then $\rho^y_i \neq *$ or $\alpha_i \neq *$.

To compute ρ, let $\rho = \alpha$ to begin with, so $|\rho| = \ell$. We need to set t more coordinates of ρ to bit values. Whenever $\rho^y_i \neq *$, we must have $\alpha_i = *$, or else ρ^y can be made smaller. We will set t coordinates of ρ, and they will all be coordinates that are set to bits in ρ^y, using the algorithm shown in Figure 8.6. We then blame the resulting restriction ρ.

We shall prove that the probability that a specific restriction ρ is blamed is very small.

Claim 8.11. *For any restriction ρ, the number of restrictions α that could lead to ρ being blamed is at most $(2t)^t$.*

Proof. Given ρ, one can immediately identify the lexicographically first restriction $\beta \in R_\alpha$: it is the first restriction in R that is consistent with ρ. There are then at most t options for the first coordinate that was set in ρ. Given the first coordinate that was set in ρ, there at most $2t$ options for the next coordinate of ρ that was set: it is either one of the coordinates of β, or a coordinate in the next restriction in R that is consistent with ρ after setting the first coordinate to $*$. In this way, we see that there are at most $(2t)^t$ choices for the set of coordinates that were set by the algorithm for computing ρ from α. □

Claims 8.11 implies that the probability that any ρ is blamed when R_α contains no restriction of size bigger than t is at most

$$\binom{n}{\ell+t} \cdot 2^{\ell+t} \cdot \frac{(2t)^t}{\binom{n}{\ell} \cdot 2^\ell} \leq \left(\frac{n-\ell}{\ell+t}\right)^t \cdot 2^t \cdot (2t)^t$$
$$= (8t\epsilon)^t .$$

Input: Alice knows $x \in \mathcal{X}_\rho$, Bob knows $y \in \mathcal{Y}_\rho$.
Output: i such that $x_i \neq y_i$.

Bob sends Alice the name of ρ^y;
Alice outputs an index $i \in [n]$ such that $x_i \neq \rho^y \neq *$;

Figure 8.4: Generic 2-round protocol τ for Karchmer-Wigderson games.

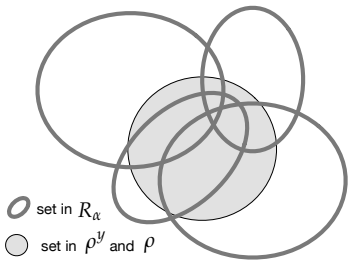


Figure 8.5: The restrictions from R must cover ρ^y.

Recall that by Lemmas 8.2 and 8.3, S_{ρ^y} must be disjoint from S_β, for every $\beta \in R_\alpha$.

Input: $\alpha, \rho^y \in \{0,1,*\}^n$, with $|\alpha| = \ell$, $|\rho^y| > t$.
Output: $\rho \in \{0,1,*\}$ of size $t + \ell$.

Set $\rho = \alpha$;
while $|\rho| < \ell + t$ **do**
  Let $\beta \in R_\alpha$ be the lexicographically first restriction such that $\exists i \in [n]$ with $\beta_i \neq * \neq \rho^y_i$, yet $\rho_i = *$;
  Set $\rho_i = \beta_i$.
end
Output ρ;

Figure 8.6: Algorithm for computing ρ from α, ρ^y.

Since the number of $\ell + t$ -restrictions is $\binom{n}{\ell+t} \cdot 2^{\ell+t}$, and the number of ℓ-restrictions is $\binom{n}{\ell} \cdot 2^\ell$.

□

Monotone Circuit Depth Hierarchy

We can use the connection to communication to show that monotone circuits of larger depth are strictly more powerful than circuits of larger depth. Throughout this section, we work with circuits of arbitrarily large fan-in.

We do now how to prove a similar result for general circuits.

Let k be an even number, and consider the formula F of depth k, where every gate at odd depth is an OR gate, every gate at even depth is an AND gate, and every gate has fan-in exactly n. Every input gate is labeled by a distinct unnegated variable. The formula has size $O(n^k)$. We prove:

Theorem 8.12. *Any circuit of depth $k-1$ that computes F must have size at least $2^{\frac{n}{16k}-1}$.*

Proof. To prove Theorem 8.12, it is enough to find show that any protocol computing the associated Karchmer-Wigderson game has communication at least $\Omega(n/k^2 - k\log n)$, the lower bound follows, since if the size of the circuit is at most s, the communication of a $k-1$ round protocol for the Karchmer-Wigderson game can be at most $(k-1)\log s$, so we get that $\log s \geq \Omega(n/k^3 - \log n)$, as required.

We prove that the Karchmer-Wigderson game has large communication by reducing the problem to the pointer-chasing problem. Here Alice and Bob are given $x, y \in [n]^n$ and what to compute z_k, where $1 = z_0, z_1, z_2, \ldots$ are defined using the rule

$$z_i = \begin{cases} x_{z_{i-1}} & \text{if } i \text{ is odd,} \\ y_{z_{i-1}} & \text{if } i \text{ is even.} \end{cases}$$

Note that every variable in the formula can be described by a string in $v \in [n]^k$. We say that v is consistent with x if

$$v_i = \begin{cases} x_1 & \text{when } i = 1, \\ v_{v_{i-1}} & \text{when } i \text{ is odd and not 1,} \end{cases}$$

We say that v is consistent with y if $v_i = y_{v_{i-1}}$ when i is even. Then note that there is a unique v that is consistent with both x and y, and that's when $v = z$, the intended path in the pointer-chasing problem.

Alice sets all the coordinates of $x' \in \{0,1\}^{[n]^k}$ that are consistent with her input to be 0, and all other coordinates to be 1. Bob sets all the coordinates of $y' \in \{0,1\}^{[n]^k}$ that are consistent with her input to be 1, and all other coordinates to be 0. Clearly, there is only one

coordinate where x' is 1 and y' is 0, and that is the coordinate that is consistent with both x, y.

Now we claim that every gate in the formula that corresponds to a path that is consistent with Alice's input evaluates to 0. This clearly true for the gates at depth k, since that is how we set the variables in x'. For the gates at depth $d < k$, if the gate is an AND gate, then it must be true because one of its inputs will be consistent with x and so evaluate to 0. On the other hand, if the gate is an OR gate, then all of its inputs correspond to paths that must be consistent with Alice's input, so they must all evaluate to 0. Thus $F(x') = 0$, since the gate at the root is certainly consistent with Alice's input. Similar arguments prove that $F(y') = 1$.

Thus any protocol for the monotone Karchmer-Wigderson game gives a protocol solving the pointer-chasing problem, and so by Theorem 6.17, we get that the communication of the game must be at least $n/16 - k$, as required.

□

Branching Programs

BRANCHING PROGRAMS MODEL COMPUTATIONS that do not use a lot of memory. A branching program of length ℓ and width w is a layered directed graph whose vertices are a subset of $[\ell + 1] \times [w]$. All the vertices in the u'th layer $(u, \cdot)$ are associated with a variable from $x_1, \ldots, x_n$.

Every vertex (u, v), with $u < \ell + 1$ has exactly two edges coming out of it, and both go to a vertex $(u + 1, \cdot)$ in the next layer. Every vertex in the last layer $(\ell + 1, \cdot)$ is labeled with an output of the program. On input $x \in \{0,1\}^n$, the program is executed by starting at the vertex $(1, 1)$ and reading the variables associated with each layer in turn. These variables define a path through the program. The program outputs the label of the last vertex on this path.

Every function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a branching program of width 2^n, and most functions require exponential width. Here we define an explicit function that requires that $\ell \cdot \log w \geq \Omega(n \log^2 n)$, for any branching program that computes it.

To prove the lower bound, we shall first show that any branching program can be simulated (at least, in a sense) by an efficient communication protocol in the number-on-forehead model (Chapter 4). Let $g : (\{0,1\}^r)^k \to \{0,1\}$ be an arbitrary function that k-parties wish to compute in the number-on-forehead model. Then define $g' : \{0,1\}^{r \log t + t} \to \{0,1\}$ to be $g'(S, x) = g(x_S)$, where here the first part of g''s input is a set $S \subseteq [t]$ of size r, and the second part of g''s

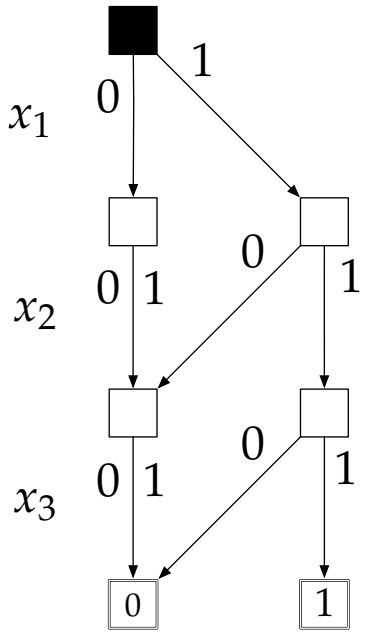


Figure 8.7: A branching program that computes the logical AND, $x_1 \wedge x_2 \wedge x_3$.

In the literature, the programs we define here are referred to as *oblivious* branching programs. In general branching programs, every vertex of the program can be associated with a different variable, vertices in a particular need not read the same variable.

input is a string $x \in \{0,1\}^t$. $g(x_S)$ is the output obtained when x is projected to the coordinates in S.

We claim [5]:

[5] Babai et al., 1989; and Hrubes and Rao, 2015

Theorem 8.13. *If g' can be computed by a length $\ell \ll n \log^2 n$, width w branching program, then g can be computed by $\ll \log n$ players with communication $\log w \cdot \log^2 n$ in the number-on-forehead model.*

Setting g to be the generalized inner-product function, Theorem 8.13 and Theorem 5.8 imply that any program with length $\ell \ll n \log^2 n$ that computes g' must have width that is at least $2^{n^{\Omega(1)}}$.

The proof of Theorem 8.13 relies on Lemma 5.4. Given a branching program of length ℓ and width w, partition the layers of the program into $O(\ell/n)$ sets of size $n/2$ each. Consider the bipartite graph where every vertex on the left corresponds to an element of the partition, and every vertex on the right corresponds to a variable. We connect two vertices if the variable does not occur in the corresponding partition of the program.

Repeatedly applying Lemma 5.4, we find disjoint sets $Q_1, \ldots, Q_{(1/2)\log n}$ on the left and $R_1, \ldots, R_{(1/2)\log n}$ on the right, such that $|Q_i| = \frac{\log n}{2 \log 3e}$, $|R_i| = \sqrt{n}$, and Q_i, R_i form a bipartite clique. The parties simply pick the set S in such a way that the i'th players input can be mapped to R_i. This proves the theorem.

Neciporuk for formulas and branching programs

9
Distributed Computing

The study of distributed computing has become increasingly relevant with the rise of the internet. In a distributed computing environment, there are n parties that are connected together by a communication network, yet no single party knows exactly what the network is.

More formally, the network is defined by an undirected graph on n vertices, where each of the vertices represents one of the parties. Each protocol begins with each party knowing their own name. In each round, each of the parties can send a message to all of their neighbors.

In general, networks may be asynchronous, and one can either charge or not charge for the length of each message. Moreover, one can also study the model in which some of the parties do not follow the protocol. In this chapter we stick to the model of synchronous networks, where we count both the number of rounds of communication and the total communication. We also assume that all parties execute the protocol correctly.

Coloring Problem

Suppose the parties in a distributed environment want to properly color themselves so that no two neighboring parties have the same color. Let us start with the example that n parties are connected together so that every party has at most 2 neighbors.

Here is a simple protocol[1] that finds a coloring in $\log^* n$ rounds of communication. Initially, the parties color themselves with their names. This takes n colors, and the coloring is proper. In each round, the parties send all of their neighbors their current color. If $a \in \{0,1\}^t$ denotes the color of one of the parties in a round, and $b, c \in \{0,1\}^t$ denote the colors assigned to her neighbors, the party sets i to be the smallest number such that $a_i \neq b_i$, and j to be the smallest number such that $a_j \neq c_j$. Her new color is then i, j, a_i, c_j. In this way, the number of colors has been reduced from t to $2^{2\log t + 2}$. After $\log^* n$ rounds, the number of colors will be a constant.

[1] Cole and Vishkin, 1986

The coloring protocol can be generalized to handle arbitrary graphs of degree d. Any graph of degree d can be colored using $d+1$ colors. Here we give a protocol[2] that uses $\log^* n$ rounds to find a

[2] Linial, 1992

coloring using d^2 colors.

The protocol relies on the following combinatorial lemma:

Lemma 9.1. *There is a family of t subsets of $[5d^2 \log t]$, such that for any $d+1$ sets $S_1, \ldots, S_{d+1}$ in the family, S_1 is not contained in the union of $S_2, \ldots, S_{d+1}$.*

Proof. Pick the t sets at random from $[5d^2 \log t]$, where each element is included in each set independently with probability $1/d$. Then for a particular choice of $S_1, \ldots, S_{d+1}$, the probability that some element $j \in S_1$ but j is not in any of the other sets is

$$(1 - 1/d)^d / d \geq 2^{-2}/d = \frac{1}{4}d.$$

The probability that there is no such j is at most $(1 - \frac{1}{4}d)^{5d^2 \log t} \leq e^{-d \log t}$.

The number of choices for $d+1$ such sets from the family is $\binom{t}{d+1} \leq t^{d+1} \leq 2^{d \log t}$. Thus, by the union bound, the probability that the family does not have the property we need is at most $e^{-d \log t} 2^{d \log t} < 1$. □

In each round of the protocol, all parties send their current color to all the other parties. If there are t colors in a particular round, each party looks at the d colors she received and associates each with a set from the family promised by Lemma 9.1. She picks a color by picking an element that belongs to her own set but not to any of the others. Thus, the next round will have at most $5d^2 \log t$ colors. Continuing in this way, the number of colors will be reduced to $O(d^2 \log d)$ in $\log^* n$ rounds.

On computing the Diameter

Suppose the parties in a network want to compute the diameter of the graph: namely the length of the longest path. Here we show that in any distributed algorithm for computing the diameter of an n vertex graph, there must n links that transmit at least $\Omega(n^2)$ bits in total[3], even if it is just trying to distinguish whether the diameter of the graph is at most 4 or bigger than 4. We do this by a reduction to the 2 party communication complexity of disjointness.

[3] Frischknecht et al., 2012; and Holzer and Wattenhofer, 2012

Let $X, Y \subseteq [n] \times [n]$ be two subsets. For every such pair of sets, we shall define a graph $G_{X,Y}$, and then show that if there is an efficient distributed algorithm for computing the diameter of the graphs $G_{X,Y}$, then there must be a an efficient communication protocol for computing whether or not X, Y are disjoint.

Let A, B, C, D be disjoint cliques of size n. Let v be a vertex that is connected to all the vertices of A, B and w be a vertex that is connected to all the vertices of C, D. We connect v and w with an edge as well. For each i, we connect the i'th vertex of A to the i'th vertex of C, and the i'th vertex of B to the i'th vertex of D.

Finally we connect the i'th vertex of A to the j'th vertex of B if and only if $(i, j) \notin X$. Similarly, we connect the i'th vertex of C to the j'th vertex of D if and only if $(i, j) \notin Y$.

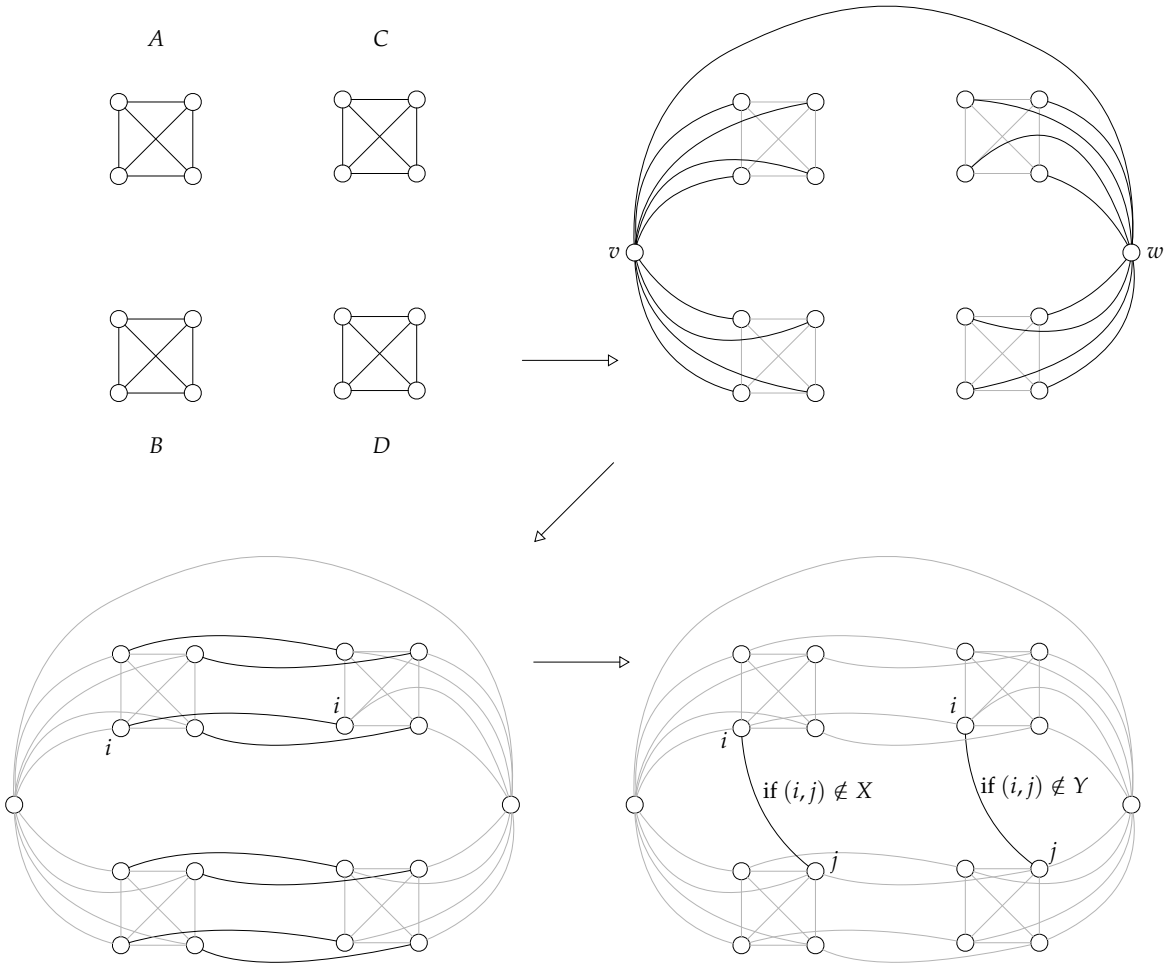


Figure 9.1: $G_{X,Y}$ for $n = 4$.

$G_{X,Y}$ is clearly connected. Moreover, the distance of v from every vertex is at most 2, and the distance of w from every vertex is at most 2. Similarly the distance of all the vertices of A, B, C, D from the rest

of the graph is always at most 3. When $(i,j) \notin X \cap Y$, we also have that the distance of the i'th vertex in A from the j'th vertex in D is at most 2. This is because, if say $(i,j) \notin X$, we can take the path from the i'th vertex of A to the j'th vertex of B to the j'th vertex of D. On the other hand, if $(i,j) \in X \cap Y$, the distance of the i'th vertex of A from the j'th vertex of D is at least 3. To summarize:

Claim 9.2. *The diameter of $G_{X,Y}$ is 2 if X, Y are disjoint, and 3 if X, Y are not disjoint.*

Consider the protocol obtained when Alice simulates all the vertices close to A, B, and Bob simulates all the nodes close to C, D. This protocol must solve the disjointness problem, and so has communication at least $\Omega(n^2)$. This proves that the $O(n)$ links that cross from the left to the right in the above network must carry at least $\Omega(n^2)$ bits of communication to compute the diameter of the graph.

Detecting Triangles

[4] Drucker et al., 2014

Another basic measure associated with a graph is its *girth*, which is the length of the shortest cycle in the graph. Here we show that any distributed protocol for computing the girth of an n vertex graph must involve at least $\Omega(n^2 2^{-O(\sqrt{\log n})}))$ bits of communication.

We prove[4] this by showing that any such protocol can be used to compute disjointness in the number-on-forehead model with 3 parties, and a universe of size $\Omega(n^2 2^{-O(\sqrt{\log n})}))$. Applying Theorem 5.12 gives the lower bound.

Suppose Alice, Bob and Charlie have 3 sets $X, Y, Z \subseteq U$ written on their foreheads, where U is a set that we shall soon specify. Let A, B, C be 3 disjoint sets of size $2n$. We shall define a graph $G_{X,Y,Z}$ on the vertex set $A \cup B \cup C$, that will have a triangle (namely a cycle of length 3) if and only if $X \cap Y \cap Z$ is non-empty.

To construct $G_{X,Y,Z}$ we need the coloring promised by Theorem 4.2. This is a coloring of $[n]$ with $2^{O(\sqrt{\log n})}$ colors such that there are no monochromatic arithmetic progressions. Since such a coloring exists, there must be a subset $Q \subseteq [n]$ of size $n2^{-O(\sqrt{\log n})}$ that does not contain any non-trivial arithmetic progressions.

Now define a graph G on the vertex set $A \cup B \cup C$, where for each $a \in A, b \in B, c \in C$,

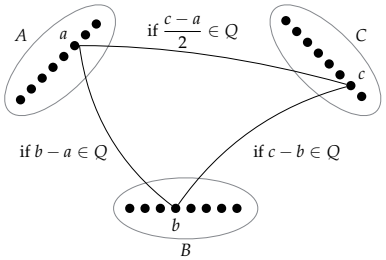


Figure 9.2: The graph G.

$$(a,b) \in G \Leftrightarrow b - a \in Q,$$
$$(b,c) \in G \Leftrightarrow c - b \in Q,$$
$$(a,c) \in G \Leftrightarrow \frac{c-a}{2} \in Q.$$

Claim 9.3. *The graph G has at least $n|Q| = \Omega(n^2 2^{-O(\sqrt{\log n})})$ triangles, and no two triangles in G share an edge.*

Proof. For each element $q \in Q$, the vertices $a \in A, a+q \in B, a+2q \in C$ certainly form a triangle, as long as $a + 2q \leq 2n$. No two of these triangles share an edge, since any edge determines a, q. We claim that there are no other triangles. Indeed, if a, b, c was a triangle in the graph, then $b - a = q_1 \in Q, c - b = q_2 \in Q$, $\frac{q_1+q_2}{2} = \frac{c-b+b-a}{2} = \frac{c-a}{2} = q_3 \in Q$, so we have $q_1, q_3, q_2 \in Q$ form an arithmetic progression. The only way this can happen is if $q_1 = q_2 = q_3$, so that the progression is trivial. But in this case we recover one of the triangles above. □

Let U denote the set of triangles in G. The graph $G_{X,Y,Z}$ will be defined as follows:

$$(a,b) \in G \Leftrightarrow \text{a triangle of } U \text{ containing } (a,b) \text{ is in } Z,$$
$$(b,c) \in G \Leftrightarrow \text{a triangle of } U \text{ containing } (b,c) \text{ is in } X,$$
$$(a,c) \in G \Leftrightarrow \text{a triangle of } U \text{ containing } (a,b) \text{ is in } Y.$$

Given sets X, Y, Z as input, Alice, Bob and Charlie build the network $G_{X,Y,Z}$ and execute the protocol for detecting triangles, with Alice, Bob, Charlie simulating the behavior of the nodes in A, B, C of the network. Each of the players knows enough information to simulate the behavior of these nodes. By Theorem 5.12, the total communication of triangle detection must be at least $\Omega(n^2 2^{-O(\sqrt{\log n})})$, as required.

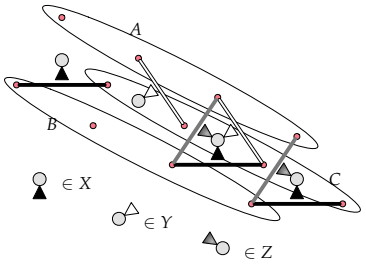


Figure 9.3: The graph $G_{X,Y,Z}$.

Verifying a Spanning Tree

10
Data Structures and Sketching

A DATA STRUCTURE is a way to efficiently maintain access to data. Many algorithmic problems, rely on efficient data structures to give their strongest performance. We start with some examples of efficient data structures, before proving some lower bounds for data structure problems.

For example, interesting data structures are used in Dijkstra's algorithm for finding the shortest path in directed graphs, and for computing the minimum spanning tree of undirected graphs.

Minimum of n Numbers Suppose we want to maintain a list of numbers from $[n]$ so that you can quickly add new numbers, and delete the minimum of the numbers. A trivial solution is to store the n numbers in a list. Then adding a number is fast, but finding the minimum might take as long as n steps. A better solution is to maintain the numbers in a *heap*: a balanced binary tree, with the property that every node is at most as large as the value of its children. One can add a number to the heap by adding it at a leaf, and bubbling it up the tree. One can delete the minimum by deleting the number at the root, inserting one of the numbers at a leaf into the root, and bubbling down the number. This takes only $O(\log n)$ time for each operation. See Figure 10.1 for an example.

These are basic operations that need to be carried out efficiently in the execution the fastest algorithms for computing the shortest path in graphs.

Connectivity in Graphs Suppose we want to store a graph on n vertices using a small amount of space such that we can later quickly answer whether two given vertices are connected or not. A trivial solution is to store the adjacency matrix of the graph, and then perform a breadth first search using this matrix. A better solution is to store a vector in $[n]^n$ which stores the name of the connected component that each vertex belongs to. This can be stored with n *words* of size $\log n$, and now connectivity for two vertices can be computed by two queries to the data structure.

There are also clever solutions to this problem that allow for dynamic operations on the graph. For example, if we want to maintain a graph while allowing edges to be added, we can do

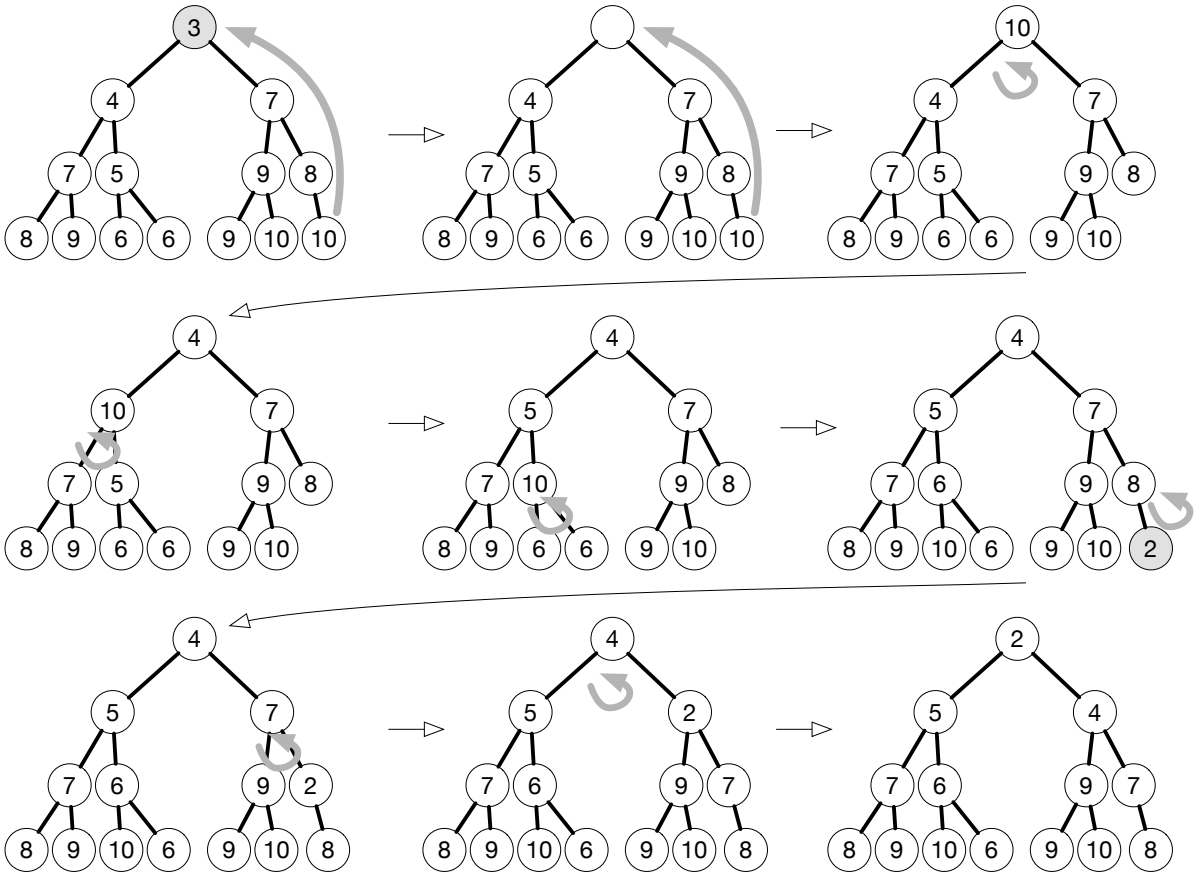


Figure 10.1: Deleting the minimum (3) from a heap, and adding a new number (2).

this using the *union-find* data structure. Each connected component is represented by a balanced tree of pointers. To check if two vertices are in the same component, we only need to check if the roots of the trees are the same, which takes at most $O(\log n)$ steps if the trees are balanced. When an edge is added, if the edge is contained in a connected component, nothing needs to be done. Otherwise the two connected components are merged by adding a pointer from the root of the shallow component to the root of the deeper component, which ensures that the trees remain balanced. This gives a simple data structure of size $O(n \log n)$ where each operation takes $O(\log n)$ time.

A related problem is the problem of computing short *sketches*. A sketch is the same as a data structure, but this time we do not care about the time it takes to answer the queries we care about. We just want to minimize the space required to store the input so that all queries can be answered.

Static Data Structures

Open Problem Deterministic Dictionaries

2d range counting Chazelle

11
Extension Complexity of Polytopes

Polygons

Independent Set Polytope

Correlation Polytope

Matching Polytope

Bibliography

A. Aho, J. Ullman, and M. Yannakakis. On notations of information transfer in VLSI circuits. In *Proc. 15th Ann. ACM Symp. on Theory of Computing*, pages 133–139, 1983.

Noga Alon and Alon Orlitsky. Repeated communication and ramsey graphs. *IEEE Transactions on Information Theory*, 41(5):1276–1289, 1995.

László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols and logspace-hard pseudorandom sequences. In *Proceedings of the 21st Annual Symposium on Theory of Computing (STOC)*, pages 1–11, 1989.

László Babai, Anna Gál, Peter G. Kimmel, and Satyanarayana V. Lokam. Communication complexity of simultaneous messages. *SIAM J. Comput*, 33(1):137–166, 2003.

Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004. URL http://dx.doi.org/10.1016/j.jcss.2003.11.006.

Felix A. Behrend. On the sets of integers which contain no three in arithmetic progression. *Proc. Nat. Acad. Sci.*, 1946.

Mark Braverman and Ankur Moitra. An information complexity approach to extended formulations. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC*, pages 161–170. ACM, 2013. ISBN 978-1-4503-2029-0. URL http://dl.acm.org/citation.cfm?id=2488608.

Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multiparty protocols. In *Proceedings of the fifteenth annual ACM Symposium on Theory of Computing*, pages 94–99. ACM Press, 1983.

Fan R. K. Chung, Ronald L. Graham, Peter Frankl, and James B. Shearer. Some intersection theorems for ordered sets and graphs. *Journal of Combinatorial Theory, Ser. A*, 43(1):23–37, 1986.

Richard Cole and Uzi Vishkin. Deterministic coin tossing and accelerating cascades: micro and macro techniques for designing parallel algorithms. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 206–219, 28–30 May 1986.

Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In Magnús M. Halldórsson and Shlomi Dolev, editors, *PODC*, pages 367–376. ACM, 2014. ISBN 978-1-4503-2944-6. URL http://dl.acm.org/citation.cfm?id=2611462.

Pavol Duris, Zvi Galil, and Georg Schnitger. Lower bounds on communication complexity. *Information and Computation*, 73(1):1–22, 1987.

David Ellis, Yuval Filmus, and Ehud Friedgut. Triangle-intersecting families of graphs, 2010. URL http://arxiv.org/abs/1010.4909.

Tomàs Feder, Eyal Kushilevitz, Moni Naor, and Noam Nisan. Amortized communication complexity. *SIAM Journal on Computing*, 24(4): 736–750, 1995. Prelim version by Feder, Kushilevitz, Naor FOCS 1991.

Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In *SODA*, pages 1150–1162. SIAM, 2012.

Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:50, 2015. URL http://eccc.hpi-web.de/report/2015/050.

Ronald L. Graham. *Rudiments of Ramsey theory*. Number 45 in Regional Conference series in mathematics. American Mathematical Society, 1980.

Ronald L. Graham, Bruce L. Rothschild, and Joel H. Spencer. *Ramsey theory*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Chichester-New York-Brisbane-Toronto-Singapore, 1980.

Vince Grolmusz. Circuits and multi-party protocols. *Computational Complexity*, 7(1):1–18, 1998.

Alfred. W. Hales and Robert. I. Jewett. On regularity and positional games. *Trans. Amer. Math. Soc.*, 106:222–229, 1963.

Bernd Halstenberg and Rüdiger Reischuk. Different modes of communication. *SIAM Journal on Computing*, 22(5):913–934, 1993.

Johan HÅstad. *Computational limitations of small-depth circuits*. The MIT Press, Cambridge(MA)-London, 1987.

Johan Håstad and Avi Wigderson. The randomized communication complexity of set disjointness. *Theory of Computing*, 3(1):211–219, 2007. URL http://dx.doi.org/10.4086/toc.2007.v003a011.

Stephan Holzer and Roger Wattenhofer. Optimal distributed all pairs shortest paths and applications. In Darek Kowalski and Alessandro Panconesi, editors, *PODC*, pages 355–364. ACM, 2012. ISBN 978-1-4503-1450-3. URL http://dl.acm.org/citation.cfm?id=2332432.

Pavel Hrubes and Anup Rao. Circuits with medium fan-in. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33, pages 381–391, 2015.

Fritz John. Extremum problems with inequalities as subsidiary conditions. pages 187–204, 1948.

Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, November 1992. ISSN 0895-4801 (print), 1095-7146 (electronic).

Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, May 1990.

Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

László Lovász. Communication complexity: A survey. Technical report, 1990.

László Lovász and Michael E. Saks. Lattices, Möbius functions and communication complexity. In *FOCS*, pages 81–90. IEEE Computer Society, 1988.

Shachar Lovett. Communication is bounded by root of rank. In David B. Shmoys, editor, *STOC*, pages 842–846. ACM, 2014. ISBN 978-1-4503-2710-7.

Peter Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57:37–49, 1 1998.

Ilan Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67–71, 31 July 1991.

Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. *SIAM Journal on Computing*, 22(1):211–219, 1993.

Noam Nisan and Avi Wigderson. On rank vs. communication complexity. *Combinatorica*, 15(4):557–565, 1995.

Anup Rao and Amir Yehudayoff. Simplified lower bounds on the multiparty communication complexity of disjointness. In *Conference on Computational Complexity*, volume 33, pages 88–101, 2015.

Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992. URL http://doi.acm.org/10.1145/146637.146684.

Razborov. On the distributed complexity of disjointness. *TCS: Theoretical Computer Science*, 106, 1992.

Thomas Rothvoß. A direct proof for lovett's bound on the communication complexity of low rank matrices. *CoRR*, abs/1409.6366, 2014. URL http://arxiv.org/abs/1409.6366.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, 1948. Monograph B-1598.

Alexander A. Sherstov. The multiparty communication complexity of set disjointness. In *STOC*, pages 525–548, 2012.

John von Neumann. Zur Theorie der Gesellschaftsspiele. (German) [On the theory of games of strategy]. *Mathematische Annalen*, 100: 295–320, 1928. ISSN 0025-5831.

M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *JCSS: Journal of Computer and System Sciences*, 43, 1991.

Andrew Chi-Chih Yao. Some complexity questions related to distributive computing. In *STOC*, pages 209–213, 1979.

Andrew Chi-Chih Yao. Lower bounds by probabilistic arguments (extended abstract). In *FOCS*, pages 420–428. IEEE Computer Society, 1983.