1 Homework problem: Tribes

We will solve the third question in the homework. The goal is to show that the nondeterministic complexity \( N^1(f), N^0(f) \) of TRIBES\(_n\) function is \( O(\sqrt{n}) \) and deterministic complexity is \( \Omega(n) \). Hence the gap between \( N^1(f), N^0(f) \) and \( D(f) \) is essentially tight. The tribes function is given by

\[
\text{TRIBES}_n(x, y) = \bigwedge_{i=1}^{\sqrt{n}} \bigvee_{j=1}^{\sqrt{n}} x_{ij} \land y_{ij}
\]

Let’s calculate the nondeterministic complexity \( N^1(f) \) first. It is enough to specify positions in all the \( \sqrt{n} \) clauses where both \( x_{ij}, y_{ij} \) take the value one. It takes \( \log(\sqrt{n}) \) bits to specify a position in \( \sqrt{n} \) long clause. Since there are \( \sqrt{n} \) such clauses \( N^1(f) \leq \sqrt{n} \log \sqrt{n} = O(\sqrt{n} \log n) \). In the article [1] an upper bound of \( O(\sqrt{n}) \) is claimed. The log factor, however, is actually required. One can create a fooling set \( F = \{(x, x) : x \text{ has exactly one one in each block}\} \). There are \( \sqrt{n} \sqrt{n} \) many such inputs and they all satisfy \( \text{TRIBES}_n(x, x) = 1 \). For \( (x, x), (x', x') \in F \) where \( x \neq x' \), however, we have \( \text{TRIBES}_n(x, x') = 0 \).

One can show that \( N^0(f) = O(\sqrt{n}) \). In case the function evaluates to zero the prover can give the index of an input to of the AND gate which is not satisfied with \( \log(\sqrt{n}) \) bits, and then the players can verify with \( O(\sqrt{n}) \) bits of communication that their inputs do not intersect on this block.

For the deterministic complexity lower bound, we will use ”tensor product” trick. First notice that every clause is solving a set intersection problem. i.e., given two strings, find if their is a common index (position) where both are one. Set intersection problem is the negation of the disjointness problem. The disjointness problem is

\[
\text{DISJ}_n = \bigwedge_{i=1}^{n} \neg(x_i \land y_i)
\]

The matrix for \( \text{DISJ}_1 \), the disjointness function on one bit, is given by

\[
\begin{pmatrix}
1 & 1 \\
1 & 0
\end{pmatrix}
\]

We claim that \( \text{DISJ}_n = \otimes_{i=1}^{n} \text{DISJ}_1 \). Given two matrices \( A, B \), their tensor product is

\[
(A \otimes B)[(x_1, x_2), (y_1, y_2)] = A(x_1, y_1) \times B(x_2, y_2)
\]
So when the matrices have entries from \{0, 1\}, the tensor product $A \otimes B$ expresses the AND of all pairs from $A$ and $B$. Two $n$-bit strings $x, y$ are disjoint if and only if $x_1, \ldots, x_{\lfloor n/2 \rfloor}, y_1, \ldots, y_{\lfloor n/2 \rfloor}$ are disjoint AND $x_{\lfloor n/2 \rfloor+1}, \ldots, x_n, y_{\lfloor n/2 \rfloor+1}, \ldots, y_n$ are disjoint, we see that by expanding this expression recursively that $\text{DISJ}_n = \otimes_{i=1}^n \text{DISJ}_1$.

Also for the tensor product we know $\text{rk}(A \otimes B) = \text{rk}(A) \times \text{rk}(B)$. Notice that the matrix for $\text{DISJ}_1$ is full rank, so the rank of $\text{DISJ}_n$ is $2^n$. Since set intersection is the negation of disjointness problem, $\text{DISJ}_n = J - \text{SI}_n$, where $J$ is the all 1’s matrix. By the subadditivity of rank

$$\text{rk}(\text{DISJ}_n) \leq \text{rk}(J) + \text{rk}(\text{SI}_n) \Rightarrow \text{rk}(\text{SI}_n) \geq 2^n - 1$$

The TRIBES$_n$ function is the AND of $\sqrt{n}$ many set intersection problems. Thus the communication matrix for TRIBES$_n$ is the tensor product of $\sqrt{n}$ copies of the communication matrix for SI$\sqrt{n}$. Again by multiplicativity of rank under tensor product we have.

$$\text{rk}(\text{TRIBES}_n(x, y)) \geq (2^{\sqrt{n}} - 1)^{\sqrt{n}},$$

and so $D(\text{TRIBES}_n) \geq \log(\text{rk}(\text{TRIBES}_n)) = \Omega(n)$.

## 2 Randomized Communication Complexity

In the last lecture, we defined randomized communication complexity. There are two versions, public coin and private coin. We will focus on public coin randomized complexity (recall that by Newman’s theorem the two models are equivalent, up to an additive logarithmic factor). A public coin protocol is a probability distribution over deterministic protocols $\{P_r\}$. A protocol has error at most $\epsilon$ if $\Pr_r[P_r(x, y) = f(x, y)] \geq 1 - \epsilon$ for all $(x, y)$.

The cost of a randomized protocol is $\max_r D(P_r)$. As usual the randomized complexity $R_\epsilon(f)$ is the minimum cost of a protocol which computes $f$ with error at most $\epsilon$.

### 2.1 Lower bounds on randomized communication complexity

Consider a randomized protocol $P$ which with probability $p_r$ runs the deterministic protocol $P_r$. We identify $P_r$ with the Boolean matrix whose $(x, y)$ entry is the output of $P_r$ on $(x, y)$. Let $M_P$ be the matrix whose $(x, y)$ entry represents the probability that the protocol outputs 1. In other words, $M_P = \sum_r p_r P_r$.

Consider a function with communication matrix $F$. If the protocol $P$ computes $F$ with error at most $\epsilon$, then we have that $|F - M_P|_{\infty} \leq \epsilon$.

The fact that a randomized protocol is a convex combination of deterministic protocols allows us to quite generally transform a lower bound technique for deterministic complexity with nice properties into a lower bound for randomized complexity.

**Theorem 1.** Suppose there exist a function $\Phi : |X| \times |Y| \rightarrow \mathbb{R}$, which satisfies the following properties

1. $\Phi(F) \leq C^P(F)$
2. $\Phi(x + y) \leq \Phi(x) + \Phi(y)$

3. $\Phi(cx) = |c|\Phi(x)$

Let $\Phi^\epsilon(F) = \min_G\{\Phi(G) : |G - F|_\infty \leq \epsilon\}$ (remember this notation, we will use it for different functions also). Then $R_\epsilon(F) \geq \log \Phi^\epsilon(F)$

Proof: The matrix for protocol $P$ is $\epsilon$ away from the matrix for $F$. So we get $\Phi^\epsilon(F) \leq \Phi(M_P)$. Now we compute $\Phi(M_P)$.

$$\Phi(M_P) = \Phi\left(\sum_r p_r P_r\right)$$
$$\leq \sum_r p_r \Phi(P_r)$$
$$\leq \sum_r p_r C^P(P_r)$$
$$\leq \sum_r p_r 2^{D(P_r)}$$
$$\leq \sum_r p_r 2^{D(P_{r^*})} \text{ (where } D(P_r) \text{ is maximum for } r^*)$$
$$\leq 2^{D(P_{r^*})}$$
$$\leq 2^{R_\epsilon(F)}$$

So we get $\Phi^\epsilon(F) \leq 2^{R_\epsilon(F)}$. In other words $R_\epsilon(F) \geq \log \Phi^\epsilon(F)$.

### 2.2 Examples for Theorem [1]

Let's take example of $\text{rk}(F)$ and see if it fits in the framework of Theorem 1. We know $\text{rk}(F) \leq C^P(F)$ and also that $\text{rk}(x + y) \leq \text{rk}(x) + \text{rk}(y)$. But the third property doesn’t work because $\text{rk}(cx) = \text{rk}(x)$. Let's look at rank as an optimization program and try to relax it to some other combinatorial quantity which will fit into theorem’s framework. Notice that

$$C^D(F) = \min_{\alpha_i} \alpha_i$$

such that $F = \sum_i \alpha_i x_i y_i$ where $x_i, y_i \in \{0, 1\}^n$

$$\alpha_i \in \{0, 1\}$$
Rank can be written as a relaxation of this program

\[ \text{rk}(F) = \min_i \alpha_i \]

such that \( F = \sum_i \alpha_i x_i y_i^t \) where \( x_i, y_i \in \mathbb{R}^{2^n} \)

\[ \alpha_i \in \{0, 1\} \]

We see that rank relaxes the constraint that the \( x_i, y_i \) are Boolean vectors and allows them to be real vectors, but keeps the constraint that \( \alpha_i \in \{0, 1\} \), a quadratic constraint.

Let us relax \( C^D(F) \) in a different way.

\[ \mu(F) = \min_i |\alpha_i| \]

such that \( F = \sum_i \alpha_i x_i y_i^t \) where \( x_i, y_i \in \{0, 1\}^{2^n} \)

\[ \alpha_i \in \mathbb{R} \]

Here we keep \( x_i, y_i \) to be Boolean vectors, but relax the constraint on \( \alpha_i \). This quantity can now be written as a linear program, albeit a huge one as there are \( 2^{2n+1} \) many variables, one for each rectangle.

It is easy to verify that \( \mu \) satisfies the properties of theorem \( \square \) and hence \( R_\epsilon(F) \geq \log \mu^\epsilon(F) \). While not obvious from the definition, it actually turns out \( \mu(F)^2 = O(\text{rk}(F)) \) for a Boolean matrix \( F \).

We can even consider a tighter relaxation of \( C^D \). Call it \( \mu_+ \).

\[ \mu_+(F) = \min_i |\alpha_i| \]

such that \( F = \sum_i \alpha_i x_i y_i^t \) where \( x_i, y_i \in \{0, 1\}^{2^n} \)

\[ \alpha_i \geq 0. \]

Notice that \( \mu_+ \) is only defined for nonnegative matrices (otherwise we can think of it as taking the value \(+\infty\)). Because of this it does not strictly satisfy condition (3) from the Theorem \( \square \) However, it does satisfy this for nonnegative constants \( c \) and one can easily see that this is all that is needed for the theorem to go through, thus we also have \( R_\epsilon(F) \geq \log \mu_+^\epsilon(F) \).

### 2.3 Dual of a norm

Any function \( \Phi : |X| \times |Y| \rightarrow \mathbb{R} \) which satisfies these two conditions is called a seminorm.

1. \( \Phi(A + B) \leq \Phi(A) + \Phi(B) \)
2. \( \Phi(cA) = |c|\Phi(A) \)
Notice the similarity with conditions in Theorem 1. The dual of a seminorm $\Phi$ is defined as

$$\Phi^*(A) = \max_B |\langle A, B \rangle| = \max_{\Phi(B)=1} |\langle A, B \rangle|.$$ 

Here $|\langle A, B \rangle|$ is the inner product of $A, B$ thought of as long vectors. We can use these norms to give lower bound on randomized communication complexity.

Example of norms and dual norms:
- $\ell_1(v) = \sum_i |v_i|$. The dual norm is $\ell_1^*(v) = \max \sum_i |u_i| \geq \sum_i v_i u_i = \ell_\infty(v)$ (the best $u$ is the one which puts all its weight on the coordinate of $v$ of largest magnitude).
- $\ell_2(v) = \sqrt{\sum_i v_i^2} \Rightarrow \ell_2^*(v) = \ell_2(v)$.
- In general $\ell_p$ is dual to $\ell_q$ if $\frac{1}{p} + \frac{1}{q} = 1$.
- Now consider the dual norm of $\mu$. This norm is defined as

$$\mu^*(A) = \max_{B: \mu(B)=1} |\langle A, B \rangle|.$$ 

If $\mu(B) = 1$, then we can express $B = \sum_i \alpha_i R_i$ where $\sum_i |\alpha_i| = 1$ and each $R_i$ is a rank one Boolean matrix. Then we see that

$$|\langle A, \sum_i \alpha_i R_i \rangle| \leq \sum_i |\alpha_i| |\langle A, R_i \rangle| \leq \max_i |\langle A, R_i \rangle|. $$

This last value is achievable by setting $\alpha_i = 1$ where $R_i$ realizes this maximum. Thus we see that $\mu^*(A) = \max_i |\langle A, R_i \rangle|$.

Expressing $\Phi$ in terms of its dual norm can be useful for showing lower bounds as it is phrased as a maximization problem. This to lower bound $\Phi(A)$ we just have to exhibit a vector $B$ which has non-negligible inner product with $A$ and small dual norm.

For this framework we will also need the fact that the dual of a dual norm is again the original norm.

**Theorem 2.** $\Phi^{**}(v) = \Phi(v)$ for any norm $\Phi$.

For example, this theorem gives that

$$\mu(A) = \max_B \frac{|\langle A, B \rangle|}{\mu^*(B)}.$$ 

We will use this formulation next to show a lower bound on the randomized complexity of inner product.
3 Approximate norms

From Theorem 1 the idea of an approximate norm naturally arose. Recall that
\[
\Phi^\epsilon(A) = \min \{ \Phi(B) : |A - B|_\infty \leq \epsilon \}.
\]
Note that \( \Phi^\epsilon \) is not itself a norm in general.

Let us see how we can lower bound \( \Phi^\epsilon \). We have
\[
\Phi^\epsilon(A) = \min_{\Delta : |\Delta|_\infty \leq \epsilon} \max_B \frac{\langle A + \Delta, B \rangle}{\Phi^*(B)} \geq \max_B \frac{\langle A, B \rangle - \epsilon \ell_1(B)}{\Phi^*(B)}
\]
In fact it turns out that \( \Phi^\epsilon \) is equal to this expression, as we will discuss in the next lecture. This means that we do not lose any power by showing a lower bound in this fashion.

Next we will use this expression with the norm \( \mu \) to see a lower bound on the inner product function.

3.1 Lower bound on Inner Product

The inner product function is defined as
\[
IP_n(x, y) = (-1)^{\langle x, y \rangle}
\]
The matrix for \( IP_1 \) is given by
\[
\begin{pmatrix}
1 & 1 \\
1 & -1
\end{pmatrix}
\]
This is the familiar 2-by-2 Hadamard matrix, an orthogonal matrix (i.e. has orthogonal rows) with entries from \{-1, +1\}.

We can again use the tensor product to express the communication matrix for inner product. As we are working over \{-1, +1\} the tensor product now takes the xor of the respective entries. This means that \( IP_n = IP_1^\otimes n \) and is also orthogonal.

Now we want to lower bound
\[
\mu^{2\epsilon}(IP_n) \geq \max_B \frac{\langle IP_n, B \rangle - 2\epsilon \ell_1(B)}{\mu^*(B)}.
\]
(The transformation from \( \{0, 1\} \) valued matrices to \{-1, +1\} means we have to work with \( 2\epsilon \) approximations rather than \( \epsilon \).)

We choose \( B = IP_n \). Then the numerator is \((1 - 2\epsilon)2^n\). It remains to upper bound \( \mu^*(IP_n) \).
\[
\mu^*(IP_n) = \max_{x, y \in \{0, 1\}^n} x^t IP_n y \\
\leq \max_{y \in \{0, 1\}^n} 2^{n/2} \|IP_n y\|
\]
by the Cauchy-Schwarz inequality. Let us now evaluate \( \|IP_n y\|^2 = y^t IP_n IP_n y = 2^n y^t y \leq 2^{2n} \).

Putting everything together we have \( \mu^2(IP_n) \geq (1 - 2\epsilon)2^{n/2} \). This shows that \( R_\epsilon(IP_n) \geq n/2 + \log(1 - 2\epsilon) \).

References

