CS 2429 - Foundations of Communication Complexity

Lecture #8: 3 November 2009

Lecturer: Matei David

Scribe Notes by: Matei David and Tasos Zouzias

1 The generalized inner product function

The generalized inner product function is defined as follows: $GIP(x_1, \ldots, x_k) = (\#$ of all-1 columns of A) mod 2, where A is an $k \times n$ matrix whose i^{th} row equals x_i for every $i = 1, \ldots, k$.

Definition 1 (Multilinear Function)**.** *A function F is* multilinear *if $\forall i,\ \forall x_1, x_2, \ldots, x_k, x_i'$ the following equality holds*

$$F(x_1, x_2, \ldots, x_i, \ldots x_k) \cdot F(x_1, x_2, \ldots, x_i', \ldots x_k) = F(x_1, x_2, \ldots, x_i + x_i', \ldots x_k).$$

Fact 1. *The GIP is a multilinear function.*

We will use the following lemma.

Lemma 2. *If F is a multilinear function, then*

$$\mathbb{E}_{\bar{x}^0, \bar{x}^1}[\prod_{u \in \{0,1\}^k} F(\bar{x}^u)] = E_{\bar{x}}[F(\bar{x})] := \Delta(F).$$

Fact 3 (Eric Vee"s Phd Thesis)**.**

$$\Delta(GIP) \leq e^{-n/2^k}.$$

This implies that $disc(GIP) \leq (e^{-n/2^k})^{1/2^k}$ then $R(GIP) = \Omega(n/4^k)$.

2 Matrix multiplication

Let the matrix multiplication function $MM : (\{0,1\}^{n \times n})^k \to \{-1, +1\}$. Player i gets an $n \times n$ matrix A_i over $GF(2)$. They want to compute the $(1,1)$ entry of the product, i.e.,

$$MM(A_1, A_2, \ldots, A_k) := (-1)^{A_1 \cdot A_2 \cdot \cdots \cdot A_k})_{11}.$$

Remark: Babai, Hayes, Ki "Missing bit"
Instead of taking the $(1,1)$ entry take the trace of the output matrix.

A protocol for computing GIP

We start by restating the protocol from the lecture notes. Let $\alpha \in \{0,1\}^k$. Let us define

$$c(\alpha) := \{ \text{ \# of columns in the input that equals to the } \alpha \text{ vector}\}.$$

The goal of the players for the GIP function is to compute the value $c(\mathbf{1}^k)$, where $\mathbf{1}^k$ is the all one vector of dimension k.

We will need the following definition for the analysis of the protocol. Let $\beta \in \{0,1\}^k$, $i \in [k]$, and $b \in \{0,1\}$. We define as

$$\text{flip}(\beta, i, b) := (\beta_1, \beta_2, \ldots, \beta_{i-1}, b, \beta_{i+1}, \ldots, \beta_{k-1}).$$

Our analysis will based on the following two facts.

Fact 4. *For every β, $\forall i$, player i sees $c(\textit{flip}(\beta, i, 0)) + c(\textit{flip}(\beta, i, 1))$.*

Proof. Fix β, and i. Player i clear can see all the columns of the matrix A except row i, therefore he/she can compute $c(\text{flip}(\beta, i, 0)) + c(\text{flip}(\beta, i, 1))$. □

Fact 5. *If **all** players know $c(\alpha)$ for some vector α, then they can compute GIP with additional communication of $k \log n$ bits.*

The protocol is the following:

- Player 1 finds the most unfrequent column $\gamma \in \{0,1\}^{k-2}$ in the submatrix $A_{3:k}$, where $A_{3:k}$ is the submatrix that corresponds to the players 3 up to k.
- First we want to bound the number of occurences of γ. It is easy to see that there are 2^{k-2} possible different columns and in the worse case all of them occur the same number of times, but we have n columns. This implies that the number of columns in $A_{3:k}$ that are equal to γ is at most $n/2^{k-2}$.
- Player 1 sends to all other players the vector γ. And then for every column j such that the j column of $A_{3:k}$ equals γ sends the bit $A(2, j)$.
- Player 2 sends the value of $c(00\gamma)$.

The number of bits communicated is : $k-2$ bits for sending the vector $\gamma \in \{0,1\}^{k-2}$. Then $n/2^{k-2}$ for all the bits $A_{2,j}$ of the least occuring (sub)-column γ. And finally $k \log n$ using Fact 5.

In conclusion, the protocol uses $O(k + n/2^{k-2} + k \log n)$ bits are needed.

Remark: The same protocol works for the *Set Disjointness* function.

3 Randomized vs. Non-deterministic Protocols

Now we will give seperation between randomized and non-deterministic NOF complexity classes.

[4, 1] show that:

$$NP^{cc} \not\subseteq BPP^{cc}_k,$$

for every $k < \log \log n$.

The main goal of this section is to prove that:

Theorem 6 ([2])**.**

$$NP_k^{cc} \not\subseteq BPP_k^{cc},$$

for every $k \leq \delta \log n$ *and every* $\delta < 1$.

Definition 2 (Correlation)**.**

$$corr_\mu(f, g) = \mathbb{E}_{x\sim\mu}[f(x)g(x)]$$

For a class of functions G, we take the maximum over all the elements of G, i.e.,

$$corr_\mu(f, G) := \max_{g\in G} corr_\mu(f, g).$$

Fact 7. *Let* $f, g : \{0,1\}^m \to \{-1,+1\}$. *Then*

$$corr_\mu(f, g) := 1 - 2 \Pr_{x\sim\mu}[f(x) \neq g(x)].$$

Fact 8. $d(f, g) = 1 - corr(f, g)$ *is a distance function over the set of functions with domain* $\{0,1\}^m$ *and range* $\{-1,+1\}$, *in particular, it satisfies trinagle inequality.*

Definition 3. *Let* Π^c *denote the set of all deterministic NOF protocols with communication cost at most* c.

For a protocol π, we slightly abuse notation and identify π with the function that it is computing, so we write $\pi(x)$ for the output of protocol π on input x.

Fact 9. $\pi \in \Pi^c$ *computes* F *with error at most* ε *under* $\lambda \iff corr_\lambda(F, \pi) \geq 1 - 2\varepsilon$.

Combining the fact above with the usual connection between distributional and randomized communication complexity [3, Theorem 3.20], we get

Fact 10. $\exists$ *a distribution* λ *such that* $corr_\lambda(F, \Pi^c) < 1/3 \iff R^{1/3}(f) > c$.

In what follows, we start with a certain $F \in NP_k^{cc}$, and our goal is to find a distribution λ and a constant α such that $corr_\lambda(F, \Pi^{n^\alpha}) < 1/3$, implying that $R(F) > n^\alpha$, and hence, that $F \notin RP_k^{cc}$.

In our proof, we bound the correlation of a function with efficient protocols by its discrepancy. The following lemma relates these two measures.

Lemma 11 ([7])**.**

$$corr_\lambda(F, \Pi^c) \leq 2^c disc_\lambda(F).$$

We can now begin the proof of the main theorem.

Definition 4 (Lifted function)**.** *Let* $m := n^\epsilon$ *for a small constant* ϵ *determined later. Given a* base function $f : \{0,1\}^m \to \{-1,+1\}$, *and a* selection function $\phi : \{0,1\}^{kn} \to \binom{[n]}{m}$, *we define the* lifted function $Lift(f, \phi) : \{0,1\}^{(k+1)n} \to \{-1,+1\}$ *by*

$$Lift(f, \phi)(x, y_1, y_2, \ldots, y_k) := f(x|\phi(y_1, \ldots, y_k)).$$

We want to compute $Lift(f, \phi)$ *using* $(k+1)$*-player NOF protocols, where player 0 has* x *on its forehead, and player* $i > 0$ *has* y_i *on its forehead.*

Observe that, for every ϕ, $Lift(OR, \phi)$ has an efficient non-deterministic protocol: P_0 sees $y_1, \ldots, y_k$ and computes $\phi(y_1, \ldots, y_k)$; P_0 guesses $i \in \phi(y_1, \ldots, y_k)$ and communicates it; P_1 checks that $x_i = 1$. Hence,

$$\forall \phi, \qquad N(Lift(OR, \phi) \leq \log n + 1.$$

In order to obtain a lower bound for $R(Lift(OR, \phi))$, we use the following connection between approximate degree and correlation with Fourier characters of small sets.

Lemma 12 ([6])**.** *If $f : \{0,1\}^m \to \{-1,+1\}$ has ε-approximate degree d then there exists a function $g : \{-1,+1\}^m \to \{-1,+1\}$ and a distribution μ on $\{-1,+1\}^m$ such that:*

- *$corr_\mu(f, g) \geq \varepsilon$.*
- *$corr_\mu(g, \chi_S) = 0$ for all S with $|S| < d$,*

where χ_S is the Fourier character function for the set $S \subset [m]$.

We also have the following lower bound for the approximate degree of the OR function:

Fact 13 ([5])**.** *The 5/6-approximate degree of the OR function is $\Omega(\sqrt{m})$.*

By the two facts above, let γ be a constant, let $g : \{0,1\}^m \to \{-1,+1\}$ be a function, and let μ be a distribution on $\{0,1\}^m$ such that:

- $corr_\mu(OR, g) \geq 5/6$.
- $corr_\mu(g, \chi_S) = 0$ for all S with $|S| < \gamma\sqrt{m}$.

The functions OR and g are highly correlated under μ. We consider the following distribution under which the associated lifted functions are also highly correlated.

Definition 5. *Let λ be the distrbution on $\{0,1\}^{(k+1)n}$ defined as follows:*

- *Select $y_1, \ldots, y_k$ uniformly at random.*
- *Select $x|\overline{\phi(y_1, y_2, \ldots, y_k)}$ uniformly at random.*
- *Select $x|\phi(y_1, y_2, \ldots, y_k)$ according to the distribution μ.*

It is easy to check the following facts:

Fact 14.

$$\lambda(x, y_1, \ldots, y_k) = \frac{2^m \mu(x|\phi(y_1, \ldots, y_k))}{2^{(k+1)n}}.$$

Fact 15.

$$corr_\lambda(Lift(OR, \phi), Lift(g, \phi)) = corr_\mu(OR, g).$$

References

[1] A. Chattopadhyay and A. Ada, *Multiparty communication complexity of disjointness*, Electronic Colloquium on Computational Complexity (ECCC), 15 (2008). 2

[2] M. David, T. Pitassi, and E. Viola, *Improved separations between nondeterministic and randomized multiparty communication*, ACM Trans. Comput. Theory, 1 (2009), pp. 1–20. 3

[3] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press, first ed., 1996. 3

[4] T. Lee and A. Shraibman, *Disjointness is hard in the multi-party number-on-the-forehead model*, in 23rd Annual Conference on Computational Complexity (CCC), Washington, DC, USA, 2008, IEEE Computer Society, pp. 81–91. 2

[5] N. Nisan and M. Szegedy, *On the degree of boolean functions as real polynomials*, Computational Complexity, 4 (1994), pp. 301–313. 4

[6] A. Sherstov, *The pattern matrix method for lower bounds on quantum communication*, in 40th Annual Symposium on the Theory of Computing (STOC), ACM Press, 2008, pp. 85–94. 4

[7] E. Viola and A. Wigderson, *Norms, XOR lemmas, and lower bounds for GF(2) polynomials and multiparty protocols*, Theory of Computing, 4 (2008), pp. 137–168. 3