16. Direct Sum III

Lecturer: Prahladh Harsha *Scribe: Karteek Sreenivasaiah*

16.1 Small information cost

In the previous lecture we showed the following lemma:

Lemma 16.1. *$\exists C > 0, \forall$ protocol $\pi, \forall \mu, \forall$ error $\varepsilon, \exists$ protocol τ such that the following holds:*

- $|\tau| \leq C\sqrt{|\pi|\, IC_\mu(\pi)}\frac{\log(\pi/\varepsilon)}{\varepsilon}$
- $Pr[\pi_A(X, \tau(X,Y) \neq \pi(X,Y)]] \leq 1 - \varepsilon$
- $Pr[\pi_A(X, \tau(X,Y)) \neq \pi_B(X, \tau(X,Y))] \leq \varepsilon$

Today we show the following direct sum like result:

Claim 16.2. *For every function f, $\forall \mu, \forall \rho, \varepsilon > 0$,*

$$\mathrm{D}_\rho^{\mu^n}(f^n) \log\left(\mathrm{D}_\rho^{\mu^n}(f^n)/\varepsilon\right) = \Omega(\varepsilon\sqrt{n}\,\mathrm{D}_{\rho+\varepsilon}^{\mu}(f))$$

Proof. A direct application of lemma 16.1 from last class on the protocol obtained from the following lemma proves the claim.

Lemma 16.3. *$\forall \mu, f, \varepsilon, \exists$ protocol π computing f on distribution μ such that:*

- $|\pi| \leq \mathrm{D}_\rho^{\mu^n}(f^n)$
- $IC_\mu(\pi) \leq \frac{2\,\mathrm{D}_\rho^{\mu^n}(f^n)}{n}$

Now we proceed to prove the above lemma itself:

Proof. Let Π be the protocol that achieves $\mathrm{D}_\rho^{\mu^n}(f^n)$. Inputs to Π look like $\langle (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\rangle$ and Π computes the answer to each pair (x_i, y_i). Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ denote random variables distributed according to μ^n. Define $W_1, W_2, \ldots, W_n$ to be random variables which take values in $X \cup Y$ such that for each i, $W_i = X_i$ with probability $1/2$ and $W_i = Y_i$ with probability $1/2$. Let W denote $W_1, W_2, \ldots, W_n$ and W_{-i} denote $W_1, \ldots, W_{i-1}, W_{i+1}, \ldots, W_n$.

Protocol π that computes f on (x, y) is as follows:

- Alice and Bob together pick a random coordinate $j \in J$ using public randomness and set $x_j \leftarrow x$ and $y_j \leftarrow y$.
- Alice and Bob together pick a $w_{-j} \in W_{-J}$ using public randomness.
- For each $i \neq j$, Alice samples X_i conditioned on w_{-j}.

- For each $i \neq j$, Bob samples Y_i conditioned on w_{-j}.
- Run Π on $\langle (x_1, y_1), (x_2, y_2) \ldots, (x_n, y_n) \rangle$. Output the j^{th} answer.

Note that since Π works well on μ^n, π will work well when input (x, y) is drawn from μ. Hence π has error at most that of Π. And since the above protocol π just runs Π on some suitable constructed input, $|\pi| \leq |\Pi| = \mathrm{D}_{\rho}^{\mu^n}(f^n)$ - which is the first part of the claim. Now we proceed to prove the second part:

$$\begin{aligned}
D^{\mu^n}(f^{(n)}) &\geq I[XY : \Pi | W] \\
&\geq \sum_j I[X_j Y_j : \Pi | W] \\
&= nI[X_J Y_J : \Pi | W J] \\
&= n(I[X_J Y_J : \Pi | J W_J W_{-J}] + I[X_J Y_J : J W_{-J}]) \\
&= n(I[X_J Y_J : \Pi W_J J | W_J]) \\
&= n(I[XY : \Pi W_J J | W_J]) \\
&= n\left(\frac{I[XY : \Pi W_J J | X_J] + I[XY : \Pi W_J J | Y_J}{2}\right) \\
&= n\left(\frac{I[Y : \Pi W_J J] + I[X : \Pi W_J J]}{2}\right) \\
&= (n/2)\,\mathrm{IC}_{\mu}(\pi)
\end{aligned}$$

□

□

If we had a better protocol compression procedure, we would have got the original direct sum result. So we ask: For what distributions can we get good protocol compression? In the following section we show that for product distributions we can get good protocol compression.

16.2 Product distributions

The following is what we prove in this section:

Lemma 16.4. $\exists c > 0, \forall$ *protocols* $\Pi, \forall$ *product distributions* $\mu, \forall \varepsilon, \exists$ *another protocol* τ *such that:*

- $|\tau| \leq c\, IC_{\mu}(\pi) \frac{polylog(\pi/\varepsilon)}{\varepsilon}$
- $Pr[\pi_A(X, \tau(X, Y) \neq \pi(X, Y)]] \leq 1 - \varepsilon$
- $Pr[\pi_A(X, \tau(X, Y)) \neq \pi_B(X, \tau(X, Y))] \leq \varepsilon$

Remark 16.5. *If for a function f, the worst distribution happens to be a product distribution, then by the above lemma, f will completely respect direct sum.*

The main tool we will use is "rejection sampling". Suppose we have a source distribution Q and a sequence of samples $x_1, \ldots, x_n$ drawn from Q. The goal of rejection sampling is to output an index i_* (smaller the better) such that x_{i_*} looks like it was drawn according to P. For eg: A sequence of coin tosses from a biased source with probability of heads 2/3 and probability of tails 1/3. We want to sample this sequence so that we eliminate the bias.

In general a rejection sampler looks like:

For $i \leftarrow 1$ to ∞

- Read sample x_i
- Accept x_i with probability $a(x_i)$

where $a : \mathcal{U} \to [0, 1]$. Suppose the input distribution to the above procedure was Q and we wanted the output distribution to be P (say), then intuitively setting $a(x) = P(x)/Q(x)$ should work. However this is not always possible because we want $a(x) \in [0, 1]$. So we scale this by $L = \max_x \frac{P(x)}{Q(x)}$ to get $a(x) = \frac{1}{L}\frac{P(x)}{Q(x)}$.Let T be a random variable denoting the number of iterations that occur before termination of the procedure. Then we have:

$$\begin{aligned} Pr[T = 1] &= \sum_{x \in \mathcal{U}} Q(x)a(x) \\ \text{Suppose } Pr[T = i | T \geq i] &= s \text{ then,} \\ \mathbb{E}[T] &= \frac{1}{s} \\ P(x) &= \sum_{i=1}^{\infty} (1-s)^{i-1} Q(x)a(x) \\ &= \underline{\underline{\frac{Q(x)a(x)}{s}}} \end{aligned}$$

We will use rejection sampling in the following way: Suppose Alice gets x drawn from a distribution μ and sends a message M to Bob. The message sent can be seen as a random variable $M(X)$ (where X is the random variable associated with Alice's input). Now if Alice's input was a string x, then Alice and Bob can view the public randomness as a sequence $m_1, m_2, \cdots \sim M(X)$ (since Bob knows the distribution from which Alice's inputs are drawn). And Alice can now do rejection sampling on this sequence to output according to $M(x)$. The following theorem gives an upper bound on the expected size of i_*:

Theorem 16.6. $\mathsf{E}[\mathit{length}(i_*)] \leq 2I[M : X] + O(1)$

A drawback of the above theorem is the $O(1)$ in the RHS. The problem arises if in a protocol Π, each message has only $o(1)$ information, in which case, there will be a large blowup when Alice tries to communicate i_* to Bob. To tackle this problem, we modify the protocol Π to wait and accumulate enough messages till there is large enough information to be communicated.

As a first step: we make sure that there is no message that contains too much information. i.e., we make every message have information content $\leq O(\beta)$ for some $\beta < 1$. We achieve this using the following idea: instead of sending bit b, send many independent random bits - each equal to b with a probability of $1/2 + \beta$. The other party can take a majority vote on the bits he/she sees to obtain the value b correctly with high probability.

Lemma 16.7. $\exists \Pi'$, *such that* $\forall x, y, v, j$ *we have:*

$$\begin{aligned} Pr[\Pi'(x,y)_{j+1} = 1 | \Pi(x,y)_{\leq j} = v_{\leq j}] &\in (1/2 - \beta, 1/2 + \beta) \\ |\Pi'| &\leq O\left(\frac{|\Pi| \log(|\Pi/\varepsilon)}{\beta^2}\right) \end{aligned}$$

We shall prove this lemma in the next lecture.
[**?**, **?**, **?**]

References

[CR11] Amit Chakrabarti and Oded Regev. *An optimal lower bound on the communication complexity of gap-Hamming-distance.* In *Proc. 43rd ACM Symp. on Theory of Computing (STOC)*, pages 51–60. 2011. arXiv:1009.3460, eccc:TR10-140, doi:10.1145/1993636.1993644.

[PD04] Mihai Patrascu and Erik D. Demaine. *Tight bounds for the partial-sums problem.* In *Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 20–29. 2004. doi:10.1145/982792.982796.

[She11] Alexander A. Sherstov. *The pattern matrix method.* SIAM J. Computing, 40(6):1969–2000, 2011. (Preliminary version in *40th STOC*, 2008). arXiv:0906.4291, doi:10.1137/080733644.