17. Direct Sum(Part IV): Conclusion

Lecturer: Prahlad Harsha *Scribe: Sajin Koroth*

In this last lecture of the four part seris on direct sum we will be focusing on protocol compression for product distributions and proving a tighter direct sum result for product distributions. In the past three lectures we proved that protocol compression along with analysis of information content of a protocol for f^n restricted to f gave direct sum results. Today we will do a different protocol compression technique which crucially uses the fact that inputs come from a product distribution to prove a direct sum result which is almost tight (except for a slight increase in error and poly-logarithmic factors). We will prove the following protocol compression theorem for product distributions :

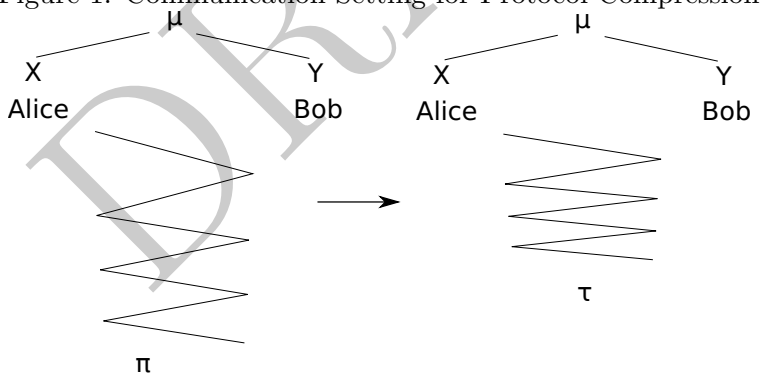
Theorem 17.1. *[?]*

$$\mathrm{D}_{\rho}^{\mu^n}(f^n)\mathrm{polylog}\left(\frac{\mathrm{D}_{\rho}^{\mu^n}(f^n)}{\alpha}\right) = \Omega\left(\mathrm{D}_{\rho+\alpha}^{\mu}(f)\alpha^2 n\right)$$

17.1 Protocol Compression for Product Distributions

We will consider a randomized protocol π using both public and private coins. The inputs of Alice and Bob come from a distribution μ which is a product distribution. This setting is illustrated in the figure 17.1.

Figure 1: Communication Setting for Protocol Compression



Protocol compression is about compressing a communication protocol to its information content as far as possible. When we refer to information content of a protocol it is either of the following definitions of information that we are referring to.

- External Information : This is the amount of information revealed to an external entity watching the protocol. This is defined as the mutual information between

the inputs, X, Y and the message transcript including the public randomness used, $\pi(X, Y)$.

$$\mathrm{IC}^o_\mu(\pi) \triangleq \mathrm{I}(XY; \pi(X, Y))$$

- Internal Information : This is the amount of information one party involved in communication learns about the other parties information.

$$\mathrm{IC}^i_\mu(\pi) \triangleq \mathrm{I}(X; \pi(X, Y)|Y) + \mathrm{I}(Y; \pi(X, Y)|X)$$

Since each party knows his/her own input, the information revealed by the protocol is the mutual information between the input of the other party and the transcript of the protocol including the public randomness used. When the inputs are drawn from a product distribution (that is the inputs are independent) then external information equals the internal information.

One way to achieve protocol compression is to do message compression, that is compress the messages send by each party in each round to the amount of information contained in the message(information theory guarantees message compression schemes which will achieve this). But if you do message compression on a general protocol then it is clear that for each round of the protocol you have to spend at least $O(1)$ bits, but it could be the case that the global information in the original message is $<<1$. This scheme would hence fail as the compressed protocols length would be at least the number of rounds and this in general could be much more than the information content in the original protocol. Hence you need to ensure that there is uniformity in the amount of information in each message you send. For achieving this, suppose if there is a bit transmitted in some round which carries a lot of information, in the modified protocol we would send a lot of bits such that the majority of the bits is slightly biased towards the value of the original bit, but each individual bit is almost unbiased. For doing this we will use the idea of rejection sampling introduced in the last lecture. To analyze the compressed protocol we will use the protocol tree view of a private coins protocol. The compression scheme which we will detail later on is : first fix the public randomness, so that the protocol is now a private coins protocol. Now use rejection sampling to transform the protocol such that in each node of the protocol tree, the bit transmitted is almost unbiased. Now in the protocol tree both Alice and Bob will choose a frontier such that all the paths from roots to a node in the frontier has at least β information. Now both of the will use rejection sampling with public randomness to obtain a node w in the frontier. Repeat this until the current frontier reaches the leaf of the protocol tree and then output the value of the protocol at that leaf.

17.2 Spreading the information uniformly

Recall that a private coins protocol tree is rooted, where each internal node v, the owner of the node has a probability distribution on its children based on the input, i.e. for node owned by Alice there is a collection of probability distributions $P_{v,x}$ supported on the children of v. Similarly for the nodes owned by Bob there is collection $P_{v,y}$. Each path from root to an internal nodes at depth j is labeled by the messages send out by the parties according to the protocol and hence can be represented by a binary string of length j. So here after

when we say node v bear in mind that there is an associated binary representation which has j bits where j is the level of v in the protocol tree. For the protocol transcript $\pi(x, y)$ which also contains the public randomness, we denote by $\pi(x, y)_j$ the jth bit communicated by the protocol on input x, y and by $\pi(x, y)_{\leq j}$ the j bit prefix of the transcript on input (x, y) along with the public randomness.

The first step, after fixing the public randomness is transform the private randomness protocol π_R to $\pi_R^{'}$ such that $\pi^{'}$ has the following property

$$\forall x, y, v, j \Pr\left[\pi^{'}(x, y)_{j+1} \mid \pi^{'}(x, y)_{\leq j} = v\right] \in \left[\frac{1}{2} - \beta, \frac{1}{2} + \beta\right]$$

But in doing so we want the error of $\pi^{'}$ not to increase more than $\gamma + \varepsilon$, where ε is the error of π. For that we need a blowup of $\frac{\log|\pi|/\gamma}{\beta^2}$ for each bit in π so that we can use Chernoff bound to prove that the majority of the extra bits will be equal to the original bit with probability at least $1 - \gamma$.

17.3 Accumulating enough information a.k.a building the frontier

Now let us analyze the information, divergence at each node. Recall that the divergence between two distributions P, Q was defined as

$$D(P||Q) = \Sigma p_i \log \frac{p_i}{q_i}$$

And we saw that it is related to information by the following equation.

$$I[X; Y] = \mathop{\mathbb{E}}_{y \leftarrow Y}\left[D\left[X_{Y=y} \mid\mid X\right]\right]$$

Let us take a node of the protocol tree, say v at depth j from root. And let us denote by $\pi(x, y)_{j+1} \mid_{v \leq j}$ the probability distribution on the $j+1$th bit transmitted by the protocol given that $j$bit prefix of the protocol transcript is v(remember that v is the binary representation of the path from root to node v in the protocol tree). We define $D^{\pi}_{x,j}(v)$ and $D^{\pi}_{y,j}(v)$ as the following.

$$D^{\pi}_{x,j}(v) = D\left[\pi(x, Y)_{j+1} \mid_{v \leq j} || \ \pi(X, Y)_{j+1} \mid_{v \leq j}\right]$$
$$D^{\pi}_{y,j}(v) = D\left[\pi(X, y)_{j+1} \mid_{v \leq j} || \ \pi(X, Y)_{j+1} \mid_{v \leq j}\right]$$

If v is owned by Alice then $D^{\pi}_{x,j}(v)$ as defined above is exactly equal to the amount of information the $j+1$th bit, send by Alice will reveal to Bob about Alice's input.

Observation 17.2. *If v is owned by Alice then*

$$D^{\pi}_{x,j} \geq 0$$
$$D^{\pi}_{y,j} = 0$$

This is because, if v is owned by Alice then the next bit transmitted does not depend on Bob's input y and also $X_x = X$ since we are guaranteed that the distribution on inputs is a product distribution. Similarly

Observation 17.3. *If v is owned by Bob then*

$$D^{\pi}_{x,j} = 0$$
$$D^{\pi}_{y,j} \geq 0$$

Now we prove the relation between divergence at a node and the information content of the bit transmitted by the node.

$$\begin{aligned}\mathop{\mathbb{E}}_{x,v\leftarrow\pi(X,Y)}\left[D^{\pi}_{x,j}(v)\right] &= \mathop{\mathbb{E}}_{x}\left[D\left[\pi(x,Y)_{j+1}\mid_{v\leq j}||\ \pi(X,Y)_{j+1}\mid_{v\leq j}\right]\right] \\ &= I\left[\pi(X,Y)_{j+1}; X \mid \pi(X,Y)_{\leq j}; Y\right]\end{aligned}$$

This proves that expected value of $D^{\pi}_{x,j}(v)$ for Alice(for Bob its always zero) is the mutual information of $j+1$th bit of $\pi(X,Y)$ and the input X conditioned on the mutual information between the transcript upto j bits and the input Y. That is exactly the amount of information revealed to Bob by $j+1$th bit about Alice's input. Also note that

$$\Sigma_{j=0}^{|\pi|-1}\mathop{\mathbb{E}}_{x,y,\pi}\left[D^{\pi}_{x,y}(v)\right] = I[X;\pi|Y]$$

This leads to the following observation.

Observation 17.4.

$$\mathbb{E}\left[\Sigma_{j=0}^{|\pi|-1}D^{\pi}_{x,j}(v) + \Sigma_{j=0}^{|\pi|-1}D^{\pi}_{y,j}(v)\right] = \mathrm{IC}_{\mu}(\pi)$$

Definition 17.5 (Frontier $B_{v,xy}(\beta)$)**.** *Given a node v and inputs x, y frontier $B_{v,xy}(\beta)$ contains all nodes w such that w is a descendant of v satisfying the following properties : For all prefixes $w^{'}$ of w :*

$$\max\left\{\Sigma_{j=|v|}^{|w^{'}|-1}D^{\pi}_{x,j}(w^{'}), \Sigma_{j=|v|}^{|w^{'}|-1}D^{\pi}_{y,j}(w^{'})\right\} < \beta$$

and

$$\max\left\{\Sigma_{j=|v|}^{|w|-1}D^{\pi}_{x,j}(w), \Sigma_{j=|v|}^{|w|-1}D^{\pi}_{y,j}(w)\right\} \geq \beta$$

or w is a leaf node.

Let us define the following probability distributions for a $w \in B_{v,xy}(\beta)$

Definition 17.6.

$$\begin{aligned}P_{v,xy}(w) &\triangleq \Pr\left[\pi(x,y)|_{|w|} = w \mid \pi(x,y)|_{|v|} = v\right] \\ P_{v,x}(w) &\triangleq \Pr\left[\pi(x,Y)|_{|w|} = w \mid \pi(x,Y)|_{|v|} = v\right] \\ P_{v,y}(w) &\triangleq \Pr\left[\pi(X,y)|_{|w|} = w \mid \pi(X,y)|_{|v|} = v\right] \\ P_{v}(w) &\triangleq \Pr\left[\pi(X,Y)|_{|w|} = w \mid \pi(X,Y)|_{|v|} = v\right]\end{aligned}$$

Where X and Y are the average inputs according to the distribution μ.

For any element w in the frontier $B_{v,xy}(\beta)$,

$$\begin{aligned}
P_{v,xy}(w) &= \Pr\left[\pi(x,y)_{|w|} = w | \pi(x,y)_{|v|} = v\right] \\
&= \prod_{j=|v|}^{|w|-1} \Pr\left[\pi(x,y)_{j+1} = w_{j+1} | \pi(x,y)_{\leq j} = w_{\leq j}\right] \\
&= \prod_{j \in A_w} \Pr\left[\pi(x,y)_{j+1} = w_{j+1} | \pi(x,y)_{\leq j} = w_{\leq j}\right] \prod_{j \in B_w} \Pr\left[\pi(x,y)_{j+1} = w_{j+1} | \pi(x,y)_{\leq j} = w_{\leq j}\right] \\
&= \prod_{j \in A_w} P_{v,x}(w) \prod_{j \in B_w} P_{v,y}(w) \\
&= P^A_{v,x}(w) \times P^B_{v,y}(w)
\end{aligned}$$

Where A_w denotes nodes of Alice in the path from v to w and B_w Bob's nodes. Similarly you can prove that

$$\begin{aligned}
P_{v,x}(w) &= P^A_{v,x}(w) \times P^B_{v,x}(w) \\
P_{v,y}(w) &= P^A_{v,y}(w) \times P^B_{v,y}(w) \\
P_v(w) &= P^A_v(w) \times P^B_v(w)
\end{aligned}$$

Claim 17.7.

$$\begin{aligned}
P^B_{v,x} &= P^B_v \\
P^B_{v,xy} &= P^B_{v,y} \\
P^A_{v,y} &= P^B_v \\
P^A_{v,xy} &= P^B_{v,x}
\end{aligned}$$

Proof. $P^B_{v,x} = P^B_v$ because at Bob's node the next bit depends only on the transcript so far (v) and Bob's input y. But Bob's input y is independent of Alice's input x as we are assuming that the inputs come from a product distribution. The other three claims also follow from a similar argument. Note that this the only crucial place that we are using the fact that the inputs come from a product distribution. □

17.4 One Round Compression Protocol

Let us first assume that both Alice and Bob know the frontier. They use rejection sampling to find a node w in the current frontier.

Protocol $\tau_{v,t,\beta}$:

1. Sample $w \in B_{v,xy}(\beta)$ using the public coins with probability $P_v(w)$.
2. Alice accepts w with probability $\min\left\{\frac{P_{v,x}(w)}{tP_v(w)}, 1\right\}$.

3. Bob accept w with probability $\min\left\{\frac{P_{v,x}(w)}{tP_v(w)}, 1\right\}$.

4. Alice and Bob send a bit each to notify each other if they have accepted the current w. They stop if both accept w, otherwise they goto step 1.

We know that $P_{v,xy}(w) = P^A_v(w)P^B_v(w)$. The above protocols accepts a w when its accepted by both Alice and Bob. Suppose w is owned by Alice, then (assuming no min's)

$$\begin{aligned} Q(w)a(w) &= P_{|v|}(w) \times \frac{P^A_{v,x}(w)}{tP_v(w)} \frac{P^B_{v,y}}{tP_v(w)} \\ &= P_{|v|}(w) \times \frac{P^A_{v,x}(w)}{tP_v(w)} \times \frac{1}{t} \\ &= P^A_{v,x}(w) \times \frac{1}{t^2} = P_{v,xy}(w)\frac{1}{t^2} \end{aligned}$$

Similarly when Bob owns the node, the same result can be obtained. Hence the above rejection sampling stopping probability for a specific w in the first round is given by probability $\frac{1}{t^2}P_{v,xy}$. We showed in the last lecture that if we denote by s $Pr[T = 1]$, then the expected number of rounds is $\mathbb{E}[T] = \frac{1}{s}$ and if the target distribution is $P(x)$ then $s = \frac{Q(x)a(x)}{P(x)}$. Since the target distribution is $P_{v,xy}(w)$ we get that $s = \frac{1}{t^2}$. Our assumption avoiding the min can be ensured by choosing a large enough t such that

$$t \geq \frac{P_{v,x}(w)}{P_v(w)} \frac{P_{v,y}(w)}{P_v(w)}$$

17.5 Compressed Protocol $\tau_{\beta,t}$

1. Use public randomness $r \leftarrow R$ to obtain a protocol π_r.

2. $v \leftarrow$root(π_r)

3. **While** v is not a leaf

   (a) $v \leftarrow \tau_{\beta,t}(v)$

4. Output v

The working of the protocol is illustrated by figure 2.

But there is an issue in the implementation of the algorithm, so far we have assumed that both Alice and Bob know the frontier. But the calculation of frontier depends on both Alice's input and Bob's input. To tackle this issue instead of sampling a node from the frontier, sample a leaf v of the protocol tree with probability P_v using public randomness. Now Alice can compute the first point where her divergence crosses β and Bob can compute the first point where his divergence crosses β on the path v. Use binary search to find the first point on the path where divergence crosses β(the binary search will ask in each round Alice and Bob whether their first point lies in the first half of the range or not, and if it is the case that Alice's point lies there and Bob's doesn't, then we know that at Alice's point

Bob's divergence is strictly less than β)and let us call this node w. The probability of such w being picked is not reduced by picking a leaf v because we can express the probability of w being picked as a sum over probabilities that one of the leaf nodes originating from w is picked.

Recall that the expected number of iterations of the one round compression protocol is, $\mathbb{E}[T] = t^2$. Each one round compression protocol can be implemented using $O(t^2 \log |\pi|)$ expected bits of communication(the $\log |\pi|$ coming from the binary search for w).

17.6 Analysis of the Protocol

So it remains to analyze the number of steps of the compression protocol $\tau_{t,\beta}$ samples elements from successive frontiers until it hits a leaf node.

We will not be proving the following claim, but we will give a proof sketch and the remaining details can be worked out.

Claim 17.8. *For all x, y, t, β,*

$$\Pr_{w \leftarrow B^{\beta}_{v,xy}} \left[\frac{P_{v,x}(w)}{P_v(w)} \geq t \right] \leq \exp\left(-\Omega\left(\frac{\log t - O(\beta^2)}{\beta} \right) \right) \triangleq \delta_\beta(t)$$

Proof Sketch. Let $P = \frac{P_{v,x}(w)}{P_v(w)}$, then we get

$$P = \prod \frac{\Pr\left[\pi(x,Y)_{j+1} = w_{j+1} \mid \pi(x,Y)_{\leq j} = w_{\leq j}\right]}{\Pr\left[\pi(X,Y)_{j+1} = w_{j+1} \mid \pi(X,Y)_{\leq j} = w_{\leq j}\right]}$$

Let Z_j equal the log of jth term in the above product. We get $\log P = \Sigma Z_j$.

$$\mathbb{E}\left[Z_i \mid_{w_{\leq i-1}}\right] = D^{\pi}_{x,i}(w)$$

Furthermore since we transformed π to π' such that the probability of each bit transmitted assuming a value is in $[\frac{1}{2} - \beta, \frac{1}{2} + \beta]$ we get for each i, $|Z_i| = O(\beta)$. And $T_i = Z_i - \mathbb{E}\left[Z_i \mid Z_1 \ldots Z_{i-1}\right]$. You can check that $T_1, T_2, \ldots$ is a martingale with bounded increments. And you can also check that $\mathbb{E}\left[T_i | T_1 \ldots T_{i-1}\right] = 0$. □

Now let us analyze the probability that the compressed protocol $\tau_{\beta,t}$ stops in a single step using the above claim. Let us denote by $\min_1(w) \triangleq \min\left\{\frac{P_{v,x}(w)}{tP_v(w)}, 1\right\}$ and $\min_2(w) \triangleq \min\left\{\frac{P_{v,x}(w)}{tP_v(w)}, 1\right\}$.

$$\begin{aligned} Pr[T = 1] &= \Sigma_w \min_1(w) \min_2(w) \\ &\geq \sum_{w \text{ good for } \min_1, \min_2} \frac{1}{t^2} P_{v,xy} \\ &\geq \frac{1}{t^2}\left(1 - 2\delta_\beta(t)\right) \end{aligned}$$

Hence we get that expected number of iterations of $\tau_{\beta,t}(v)$ is $O(\frac{t^2}{1-2\delta_\beta(t)})$.

We will now prove that no sample gets undue attention, i.e.,

Claim 17.9.

$$\frac{P'_{v,xy}(w)}{P_{v,xy}(w)} \leq 1 + 2\delta_t(\beta)$$

Where P' is the distribution obtained by rejection sampling in one round compression protocol.

Proof. For all w

$$P'_{v,xy}(w) = \frac{P_v(w)a(w)}{\frac{1}{t}\left(1 + \delta_{t,\beta}\right)}$$

□

Now let us fix the parameters such that we get the direct sum result for product distributions. Let us fix β such that $\frac{1}{k\log(|\pi|/\varepsilon)}$. And let t be a small positive constant such that $\delta_\beta(t) = \exp\left(-\Omega(\frac{1}{\beta})\right) = \frac{\varepsilon}{|\pi|}$. Now the total communication by the protocol is

$$\mathbb{E}_l\left[\frac{D_x^\pi(l) + D_y^\pi(l)}{\beta}\frac{t^2\log|\pi|}{1-2\delta_\beta(t)}\right] = O\left(\mathrm{IC}_\mu(\pi)\mathrm{polylog}\left(\frac{\pi'}{\varepsilon}\right)\right)$$

And for all leaves l,

$$\frac{\Pr\left[\tau \text{ outputs } l\right]}{\Pr\left[\pi(x,y) = l\right]} \leq \left(1 + 2\delta_\beta\right)(t))^{\frac{|l|}{\beta}} \leq \exp\left(O(\varepsilon)\right)$$

References

[BBCR10] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. *How to compress interactive communication.* In *Proc. 42nd ACM Symp. on Theory of Computing (STOC)*, pages 67–76. 2010. doi:10.1145/1806689.1806701.

Figure 2: Illustration of the compressed protocol. The grey nodes represent nodes in a given frontier. The protocol progresses by choosing a frontier node and calculating a frontier from that node and repeating by choosing a node from the new frontier