**Communication Complexity** 17 Oct, 2011 (@ IMSc)

# Lec. 15: Direct Sum (Part II) - protocol compression

*Lecturer: Prahladh Harsha* *Scribe: Raja S*

**Summary**

In this lecture, we continued our discussion on directsum. We gave a protocol compression strategy over general distribution, in which given a protocol $\Pi$ with information content $IC_\mu(\Pi)$, produces another protocol $\tau$ such that expected length of $\tau$ is upper bounded by $c\sqrt{|\Pi| * IC_\mu(\Pi)} * (\frac{\log(\frac{|\Pi|}{\varepsilon})}{\varepsilon})$

**Public + Private coins protocol tree:** We recall that we viewed communication between Alice and Bob as a protocol tree $T$ where each internal node of $T$ is owned exactly by one party (Alice or Bob) and also for each internal node $v$ of $T$ they have probability distribution $P_{v,z}$ by which they select either left or right subtree, where $z$ is either $x$ or $y$ depends on whether $v$ is owned by Alice or Bob respectively. We viewed public coin protocol as a distribution over such protocol trees (i.e., both Alice and Bob use public randomness to select one of such trees and proceed).

**Node Representation:** We define representation of a node $v$ of $T$ as concatenation of binary bits on the unique path between root and $v$ in the protocol tree (i.e., transcript (in binary) that would have led to node $v$).

**Notations:** Let $j_v$ is the length of the node representation of $v$. Here probability is over $R, R_A, R_B$.
(1) $P_{v,xy}(w) = P[\Pi(x,y)_{j_v+1} = w | \Pi(x,y)_{\leq j_v} = v]$
(2) $P_{v,xY}(w) = P_Y[\Pi(x,Y)_{j_v+1} = w | \Pi(x,Y)_{\leq j_v} = v]$
(3) $P_{v,Xy}(w) = P_Y[\Pi(X,y)_{j_v+1} = w | \Pi(X,y)_{\leq j_v} = v]$
(4) $P_{v,XY}(w) = P_{X,Y}[\Pi(X,Y)_{j_v+1} = w | \Pi(X,Y)_{\leq j_v} = v]$

**Observation:** If node $v$ in $T$ is owned by Alice then $\forall y\ P_{v,xy} = P_{v,x}$. Similarly, if node $v$ in $T$ is owned by Bob then $\forall x\ P_{v,xy} = P_{v,y}$

The overall algorithm is first both Alice and Bob select a protocol tree using public randomness and then each sample a path $P_A$ and $P_B$ (from root to a leaf) respectively from this tree (independently). Note that, in selecting $P_A$, for nodes owned by Bob, Alice has to select one of its subtrees, but she does not have information about $P_{v,y}$, so she will choose based on $P_{v,xY}$. Similarly, Bob also selects $P_B$. Let $P$ be a path selected if original protocol $\Pi$ is followed. There may be disagreement(s) between $P_A$ and $P_B$. So, they will try to find and fix these disagreement(s), for this they will use the public coin protocol for $EQ_n$. Note that, $P_A$ and $P$ may differ in locations in which corresponding node in the protocol tree is owned by Bob. Similarly for $P_B$ and $P$. Note that, once all disagreement(s) are fixed then $P_A = P_B = P$.

DRAFT

## 15.1 Preliminaries

### 15.1.1 FIND-FIRST-DISAGREEMENT

Given two $n$ bits binary string $x$ and $y$, the task is, if $x \neq y$, then to find the first location $i$, from msb, such that $x_i \neq y_i$. We know how to check if $x = y$ or not, using $O(1)$-bits of communication. They both check if $x \neq y$ then they check whether first half of $x$ and $y$ is equal or not. If it is not equal, they recursively sovle this sub-problem and if it is equal then they solve the other subproblem. Using $O(\log n)$-bits of communication, they can both find location $i$ such that $x_i \neq y_i$.

*Error analysis:* We know that $R^{pub}_{\frac{1}{3}} = O(1)$. We repeat each equality for $O(\log \log n)$ times and output equality only if each time we get a equality. This new protocol gives $R^{pub}_{\frac{1}{3 \log n}} = O(\log \log n)$ for each subproblem. Since there are $O(\log n)$ subproblems (i.e., equality check) where each subproblem needs $O(\log \log n)$ bits of communication, total communication is $O(\log n \log \log n)$ bits with error at most $\frac{1}{3}$ (using union bound).

If original error is $\varepsilon$ then we repeat each equality for $\log(\frac{\log n}{\varepsilon})$, which gives $R^{pub}_{\frac{\varepsilon}{\log n}} = O(\log(\frac{\log n}{\varepsilon}))$ and thus total communication is $O(\log n \log(\frac{\log n}{\varepsilon}))$. But the following result is known,

**Theorem 15.1.** *$\forall \varepsilon > 0$, there exists a $O(\log(\frac{n}{\varepsilon}))$ bits public coin protocol that finds the first disagreement, if one exists, with error probability at most $\varepsilon$.*

### 15.1.2 Correlated sampling

Both Alice and Bob use public randomness to obtain $k_w \in_{u.a.rr} [0, 1]$, for each internal node $w$ of the protocol tree.

If $Pr[\Pi_r(X, Y)$ reaches left child $\mid \Pi_r(X, Y)$ reaches $w, X = x] > k_w$ then set $c_x(w) =$ left child of $w$. Otherwise, set $c_x(w) =$ right child of $w$

### 15.1.3 Handling public coins

Both Alice and Bob choose a protocol tree using public randomness. We use $IC_\mu(\Pi)$, where $\Pi$ is public + private coins protocol, to denote information content of $\Pi$ over distribution $\mu$. We use $IC_\mu(\Pi_r)$, where $\Pi_r$ is the private coins protocol with public coin is fixed to $r$, to denote information content in $\Pi_r$. We now show that $IC_\mu(\Pi) = E_r[IC_\mu(\Pi_r)]$.

DRAFT

**Lemma 15.2.** $IC_\mu(\Pi) = E_r[IC_\mu(\Pi_r)]$

*Proof.*

$$\begin{aligned}
IC_\mu(\Pi) &= I[X : \Pi_r(X,Y)R|Y] + I[Y : \Pi_r(X,Y)R|X] \\
&= I[X : R|Y] + I[Y : R|X] + I[X : \Pi_r(X,Y)|R,Y] + I[Y : \Pi_r(X,Y)|R,X] \\
&\textit{( using the chain rule for information } I[X : Z_1Z_2] = I[X : Z_1] + I[X : Z_2|Z_1]\textit{)} \\
&= I[X : \Pi_r(X,Y)|R,Y] + I[Y : \Pi_r(X,Y)|R,X] \\
&= E_r[IC_\mu(\Pi_r)]
\end{aligned}$$

□

## 15.2 Compressed Protocol $\tau_{\beta,\gamma}$

(1)*(public randomness)*Alice and Bob use public randomness $r$ to fix private coin protocol $\Pi_r$
(2)*(correlated sampling)*For each internal node $v$ in the protocol tree of $\Pi_r$, both Alice and Bob use public randomness to pick $\kappa \in_{u.a.r} [0,1]$ and set
For Alice

$$C^x(v) = \begin{cases} 0 & \text{if } \kappa_v \leq P_{v,xY} \\ 1 & \text{otherwise} \end{cases}$$

For Bob

$$C^y(v) = \begin{cases} 0 & \text{if } \kappa_v \leq P_{v,Xy} \\ 1 & \text{otherwise} \end{cases}$$

(3)*(path)* Alice's path is $V_x = v_x^0, v_x^1, ..., v_x^{|\Pi_r|}$ where $v_x^0 =$ root of the protocol tree and $v_x^{l+1} = C^x(v_x^i)$. Similarly Bob's path is also defined.

Note that now both Alice and Bob have an individual path and also they have not yet communicated each other.

We also define the correct path as follows, $V = v^0, v^1, ..., v^{|\Pi_r|}$ where $v^0 =$ root of the protocol tree and

$$v^{i+1} = \begin{cases} C^x(v^i) & \text{if } v^i \text{ is owned by Alice} \\ C^y(v^i) & \text{if } v^i \text{ is owned by Bob} \end{cases}$$

(4)*(path fixing phase)*

For $i = 1$ to $\frac{T}{\gamma}$
Run FIND-FIRST-DISAGREEMENT with error $\beta$ and fix the disagreements

*Error analysis:* $P[\text{Alice output} \neq \text{Bob output} \neq V]$
The source of errors are,
(1) Number of disagreements are $> \frac{T}{\gamma}$, which can be upper bounded by $\gamma$ using Markov inequality
(2)One of FIND-FIRST-DISAGREEMENT errors, which is upper bounded by $\beta.\frac{T}{\gamma}$ (using

DRAFT

union bound)

Thus, total error is upper bounded by $\gamma + \beta.\frac{T}{\gamma} = \frac{\gamma^2+\beta T}{\gamma}$.

By setting, $\gamma^2 = \beta T$ and $\beta = \frac{\varepsilon^2}{4T}$, we have $\frac{\gamma^2+\beta T}{\gamma} = 2\sqrt{\beta T} = \varepsilon$. $(\gamma = \frac{\varepsilon}{2})$

Total comunication is $\frac{T}{\gamma}.O(\log\frac{n}{\beta}) = \frac{T*O(\log\frac{n}{\beta})}{\varepsilon}$

We now calculate expected number of disagreements $T$ in a protocol tree selected by public coins.

Let $E_i$ be an indicator random variable which is set to 1 if there is a disagreement at level $i$ in the protocol tree and 0 otherwise. Note that disagreement happens at level $i$ exactly when $C^x(v_i) \neq C^y(v_i)$.

$$\begin{aligned}
E[E_i] &= E_{X,Y,V}[|V^i|_{xv_{<i}} - V^i|_{yv_{<i}}|] \\
&= E_{X,Y,V}[\Delta(V^i|_{XV_{<i}}, V^i|_{XYV_{<i}}) + \Delta(V^i|_{YV_{<i}}, V^i|_{XYV_{<i}})] \\
&\leq E_{X,Y,V}[\sqrt{D(V^i|_{XYV_{<i}}||V^i|_{XV_{<i}}) + D(V^i|_{XYV_{<i}}||V^i|_{YV_{<i}})}] \text{(using the relation between } \Delta \text{ and } D) \\
&\leq \sqrt{E_{X,Y,V}[D(V^i|_{XYV_{<i}}||V^i|_{XV_{<i}}) + D(V^i|_{XYV_{<i}}||V^i|_{YV_{<i}})]} \text{ (using Jensen inequality)} \\
&\leq \sqrt{I[V^iY|XV_{<i}] + I[V^iX|YV_{<i}]} \text{ (using the relation between } I \text{ and } D)
\end{aligned}$$

Total number of disagreements are,

$$\begin{aligned}
Z &= \sum_i Z_i \\
E[Z] &= \sum_i E[Z_i] \\
&\leq \sum_i \sqrt{I[V^iY|XV_{<i}] + I[V^iX|YV_{<i}]} \\
&\leq \sqrt{|\Pi| * \sum_i I[V^iY|XV_{<i}] + I[V^iX|YV_{<i}]} \quad \text{(using Cauchy-Schwarz inequality)} \\
&= \sqrt{|\Pi| * (I[V:Y|X] + I[V:X|Y])} \quad \text{(using chain rule for information)}
\end{aligned}$$

Now, we calculate the expected number of disagreements $T$ over public coins.

DRAFT

$$
\begin{aligned}
E_r[\#\text{ disagreements}] &\le E[\sqrt{|\Pi| * \sum_i I[V:Y|X] + I[V:X|Y]]} \\
&\le \sqrt{E[|\Pi| * (I[V:Y|X] + I[V:X|Y])]} \qquad \text{(using } \sqrt{x} \text{ is concave and Jensen inequality)} \\
&= \sqrt{E[|\Pi| * IC_\mu(\Pi)]}
\end{aligned}
$$

We have the following theorem,

**Theorem 15.3.** *There exists a contant $c > 0$, $\forall\mu\ \forall\varepsilon\ \forall\Pi$, there exists another protocol $\tau$ and functions $\Pi_A, \Pi_B$ such that the following are true*
*(1) $|\tau| \le c\sqrt{|\Pi| * IC_\mu(\Pi)} * (\frac{\log(\frac{|\Pi|}{\varepsilon})}{\varepsilon})$*
*(2)$Pr[\Pi_A(X, \tau(X,Y)) \ne \Pi(X,Y)] \le \varepsilon$*
*(3)$Pr[\Pi_A(\tau(X,Y) \ne \Pi_B(Y, \tau(X,Y))] \le \varepsilon$*

## 15.3 Appendix

### 15.3.1 Divergence / Relative Entropy / Kullback-Leibler Distance

The relative entropy or Kullback-Leibler distance between two probability mass funciton $p(x)$ and $q(x)$ is defined as, $D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} = E_p[\log \frac{p(X)}{q(X)}]$

**Entropy:** Entropy $H$ of a distribution $P$ is defined as $H(P) = \sum_i p_i \log \frac{1}{p_i}$

Intuitively, $H(P)$ is expected number of bits required to encode a random variable distributed according to $P$.

Intuitively, $D(P||Q)$ is expected number of extra bits used when elements in distribution $P$ are encoded using encoding for distribution $Q$.

#### 15.3.1.1 Properties of $D(P||Q)$

(1)$D(P||Q) \ge 0$

$$
\begin{aligned}
D(P||Q) &= \sum_i p_i \log \frac{p_i}{q_i} \\
&= -\sum_i p_i \log \frac{q_i}{p_i} \\
&\ge -\log \sum_i p_i \frac{q_i}{p_i} \text{ (using } \log x \text{ is concave and Jensen inequality)} \\
&= 0
\end{aligned}
$$

(2) $D(P||Q) < \infty$ iff $supp(P) \subseteq supp(Q)$

DRAFT

(3) $\Delta^2(P,Q) \leq D(P||Q)$ ($\Delta(.,.)$ is total variation)

(4) $I[X:Y] = E_{x \leftarrow X}[D(Y_x||Y)]$

$$\begin{aligned} I[X:Y] &= \sum_{x,y} p_{xy} \log \frac{p_{xy}}{p_x . p_y} \\ &= \sum_x p_x \sum_y p_{y|x} \log \frac{p_x p_{y|x}}{p_x . p_y} \\ &= \sum_x p_x D(Y_x||Y) \\ &= E_{x \leftarrow X}[D(Y_x||Y)] \end{aligned}$$

### 15.3.2 Correlated sampling

Let Alice has a distribution $P$ with $P[head] = p$ and $P[tail] = 1 - p$. Let Bob has a distribution $Q$ with $P[head] = q$ and $P[tail] = 1 - q$. In correlated sampling, they do the following.
(1) they use public coins to pick $\kappa \in_{u.a.r} [0,1]$
(2) Alice output head if $\kappa \leq p$ and tail otherwise. Similarly, Bob output head if $\kappa \leq q$ and tail otherwise.

**Facts:**
(1) Alice output is distributed according to $P$
(2) Bob output is distributed according to $Q$
(3) $Pr[\text{Alice output} \neq \text{Bob output}] = |P - Q|_1$

DRAFT