Communication Complexity 16 Sep, 2011 (@ TIFR)

10. Pointer Chasing

Lecturer: Jaikumar Radhakrishnan *Scribe: Sagnik Mukhopadhyay*

In today's lecture, we will continue the discussion on index function problem and we will introduce another problem, namely, pointer chasing problem.

10.1 Index Function Problem

The index function problem is defined as follows. Alice has a string x distributed uniformly over $\{0,1\}^n$. Bob has an index i distributed uniformly over $[n]$. The goal for Bob is to guess x_i when only one round of communication is allowed, i.e., Alice can send only one message to Bob.

The naive protocol for this problem is that Alice sends all her bits to Bob. Then the message length is n. Our goal is to figure out whether we can reduce the message length using randomization and if so, how much we can reduce. Formally we want to have a lower bound on the message length. We will show this lower bound using information theoretic argument.

Let us fix a deterministic protocol $\mathcal{P}$ that computes the index function. Suppose the error made by the deterministic protocol $\mathcal{P}$ is $\frac{1}{2} - \delta$ on uniform distribution over input.

Let M is a random variable that represent the message send by Alice, which is determined by Alice's input $X = X_1 X_2 .... X_n$ where $X_i s$ are independent.

$$
\begin{aligned}
\text{Cost}(P) &\geq \log(\text{number of distict messages}) \\
&= \log(|\text{support}(M)|) \\
&\geq H(M) && \text{(By claim 9.7)} \\
&\geq H(M) - H(M|X) \\
&= I(X : M) && \text{(By the definition of Mutual information)} \\
&\geq \sum_i I(X_i : M) && \text{(By claim 9.15)} \qquad (10.1.1)
\end{aligned}
$$

Let $\Pr\left[\mathcal{P} \text{ errors} \mid \text{Alice sends message } m \text{ and Bob has input } i\right] = r_i^m$

We know that

$$
\begin{aligned}
\frac{1}{2} - \delta & \geq \Pr\left[\mathcal{P}\text{errors}\right] \\
& = \sum_i \sum_m \Pr\left[\text{Alice sends } m,\ \text{Bob has } i \text{ and } \mathcal{P} \text{ errors}\right] \\
& = \sum_i \sum_m \Pr\left[\mathcal{P} \text{ errors}|\text{Alice sends } m,\ \text{Bob has } i\right].\Pr\left[\text{Alice sends } m,\ \text{Bob has } i\right] \\
& = \mathbb{E}_{i,m}\left[r_i^m\right] \qquad (10.1.2) \\
I(X_i : M) & = H(X_i) - H(X_i|M) \\
& = H(\tfrac{1}{2}) - \mathbb{E}_m\left[H(X_i|M = m)\right] \\
& = 1 - \mathbb{E}_m\left[H(r_i^m)\right] \\
& \geq 1 - H\left[\mathbb{E}_m(r_i^m)\right] \qquad (\because H \text{ is concave}) \\
\text{Cost}(P) & \geq \sum_i I(X_i : M) \\
& \geq \sum_i \left(1 - H\left[\mathbb{E}_m(r_i^m)\right]\right) \\
& \geq n - n\sum_i \frac{1}{n} H\left[\mathbb{E}_m(r_i^m)\right] \\
& \geq n - n\mathbb{E}_i\left[H\left[\mathbb{E}_m(r_i^m)\right]\right] \\
& \geq n - nH\left[\mathbb{E}_{i,m}(r_i^m)\right] \qquad (\because H \text{ is concave}) \\
& \geq n\left[1 - H(1-\tfrac{1}{2})\right] \qquad (\text{due to } (10.1.2)) \\
& = n\left(1 + \left(\frac{1}{2} + \delta\right)\log\left(\frac{1}{2} + \delta\right) + \left(\frac{1}{2} - \delta\right)\log\left(\frac{1}{2} - \delta\right)\right) \\
& = n\left(\left(\frac{1}{2} + \delta\right)\log\left(\frac{\frac{1}{2} + \delta}{\frac{1}{2}}\right) + \left(\frac{1}{2} - \delta\right)\log\left(\frac{\frac{1}{2} - \delta}{\frac{1}{2}}\right)\right) \\
& = n\log e.K_{\frac{1}{2}+\delta,\frac{1}{2}} \qquad (K_{p,q} \text{ is Kullback Leibler distance}^1) \\
& = n\log e.2\delta^2 \qquad [\text{by Pinsker's inequality}]^1 \qquad (10.1.3)
\end{aligned}
$$

Corollary 10.1. *In any one round randomized protocol with error* $(1/2 - \delta)$ *Alice must send Bob at least* $2\log e\delta^2 n$ *bits.*

This is because of the fact that for any randomized protocol there is a deterministic protocol which can be derived from the randomized protocol by fixing its random coin tosses. The question is how good is this lower bound, i.e., can we design a randomized protocol that achieves this bound?

[1] Let P and Q be probability distributions, $P(1) = p, P(0) = 1 - p$ and $Q(1) = q, Q(0) = 1 - q$. The Kullback Leibler-distance is defined to be $K_{p,q} = p\ln\frac{p}{q} + (1-p)\ln\frac{1-p}{1-q}$
Pinsker's inequality: $K_{p,q} \geq 2(p-q)^2$

Let us consider the natural protocol for the index function problem where Alice send Bob a sample of $(2\delta)n$ bits for her input. Bob either finds the bit he needs or guesses it. Clearly, the error probability in this case is $\frac{1}{2}(1-2\delta) = \frac{1}{2} - \delta$. But the lower bound claims that only $\Omega(\delta^2 n)$ bits are needed to be transferred which is still lesser than what is tranferred in this easy protocol.

Can we come up with a better protocol? It turns out that the answer is yes. Under the uniform distribution on $\{0,1\}^n$, let us consider a ball of radius $(1/2-\delta)n$ centered at origin. Let x be the string chosen uniformly from $\{0,1\}^n$. It can be shown that

$$Pr[x \in Ball(\overline{0}, (1/2-\delta)n)] \sim e^{-c\delta^2 n}$$

Alice and Bob use shared randomness as sequence of random strings from $\{0,1\}^n$. Alice points to the first string that falls in the ball of radius $(1/2-\delta)n$ around her input, i.e., within $Ball(x, (1/2-\delta)n)$ where Alice's input is x. The expected communication is $O(\log e^{c\delta^2 n}) = O(\delta^2 n)$ (If some event happens with some probability, then the waiting time of that event is inverse of that probability, and the index is the log of that inverse).

10.1.1 Variant of Index Function Problem

If Bob is allowed to send $\log n$ bits to Alice in the course of the protocol, then the situation is simple and Alice can only send the bit x_i to Bob. We now look at a harder situation where Bob is allowed to send only k bits where $k << \log n$. If Bob is allowed to send only 1 bit, then depending on that bit, Alice can decide whether the index is on the left part of her input or on the right part. Then Alice only needs to send $n/2$ bits depending on which part Bob's input index lies. By similar argument, we can colclude that Alice needs to send $\frac{n}{2^k}$ bits to Bob if Bob is allowed to send k bits to Alice and no error is allowed. We now ask how many bits must Alice send in order for Bob to guess x_i with probability $(\frac{1}{2}+\delta)$. In this case, Bob sends the k bits as before. Alice, after knowing the bits sent by Bob, perform the one round protocol described in the previous section on $\frac{n}{2^k}$ bits of interest of Bob. This gives an upper bound of $O(\frac{n}{2^k}\delta^2)$ on the communication complexity. Here the shared randomness is the sequence of strings drawn independently from $\{0,1\}^{n/2^k}$.

Is this a tight bound? For answering this question we need to show a lower bound on the communication of this protocol. The driving idea for showing this lower bound is the following claim.

Claim 10.2. *If there is a protocol where Bob sends k bits and error is at most $(\frac{1}{2}-\delta)$, then there exists a randomized protocol with k bits of randomness where Bob sends nothing, Alice sends the same number of bits as before and the error is at most $(\frac{1}{2}-\frac{\delta}{2^k})$.*

This claim gives a lower bound on Alice's communication, i.e., it implies that Alices communication is at least $\frac{n\delta^2}{2^{2k}}$.

Proof. This proof actually converts the former protocol to the latter one. For doing so, Alice guesses the trancript of Bob by tossing k fair coins, which is the shared randomness. Consulting this guess as Bob's actual transcript, Alice simulates the protocol on her side and sends Bob her part of the transcript. Bob checks whether Alice's guess is right by simulating the protocol with Alices transcript. If Alice's guess is wrong at some point, Bob gives up and tosses a fair coin to guess x_i. If Alice's guess is correct, Bob sends Alice the output of the protocol.

Now, as Alice has to guess k bits, the probability that she would guess all the bits correctly is only $\frac{1}{2^k}$. So with probability $(1 - \frac{1}{2^k})$ Alice guesses wrong and Bob has to toss a fair coin to guess x_i. On the other hand, if Alice guesses correctly, then with probability $(\frac{1}{2} - \delta)$ Bob guesses correctly. Hence the total probability of error is at most $\frac{1}{2}(1 - \frac{1}{2^k}) + (\frac{1}{2} - \delta)\frac{1}{2^k} = (\frac{1}{2} - \frac{\delta}{2^k})$ □

Comment: Instead gussing k bits in the above proof, if Alice sends information for all 2^k choices the error probability can be reduced to $\frac{1}{2} - \delta$, but the cost will be 2^k times the cost of protocol where Bob sends k bits. This gives us a lower bound of $\Omega(\frac{\delta^2 n}{2^k})$ for the protocol where Bob sends k bits.

10.2 Pointer Chasing Problem

The pointer chasing problem gives a tradeoff between number of rounds of a protocol and amount of communication needed to perform the protocol. The motivation of this problem comes from the following question: are there problems for which k-round protocol and $(k-1)$-round protocol makes a big difference in the communication complexity? The pointer chasing problem for k-round, $Pointer_k$, is defined as follows.

There are $k+1$ layers of vertices: $L_0, ..., L_k$. The top layer L_0 has only one vertex, v_0, and other layers have n vertices, and each vertex in one layer has exactly one directed edge leading to a vertex in the next layer. The edges coming out of the even layers known only to Alice and the edges coming out of the odd layers are known only to Bob. Clearly, the graph has a unique path $v_0, ..., v_k$ from layer L_0 to L_k. Alice and Bob seek to exchange messages in order to find v_k.

The problem can be solved with $k \log n$ bit messages if Bob starts. In each round, Alice and Bob exchange the value of the pointer, which is of length $O(\log n)$. As there are k rounds, the amount of communication is $k \log n$.

We now ask the question: Can $Pointer_k$ be solved if $(k-1)$ rounds are allowed and Alice starts the protocol? Answering this question gives us an upper bound on the communication complexity of $Pointer_k$. We will consider a few base cases and we will conclude the upper bound for k-round protocol by similar argument.

- CASE 1 : k = 2
  For this case, if Alice starts, Bob just needs to know what value Alice has on the vertex indexed by v_0 depending on which Bob can easily determine the value of v_3. As only one round of communication (from Alice) is allowed, this case is the same as index function problem. Here Alice just sends all her bits to Bob and hence the communication has complexity $\Omega(nlog\ n)$.

- CASE 2 : k = 3
  In this case, let us assume that Alice sends rn bits of her input to Bob which is first r bits of her n vertices. Hence, by the similar argument we did in Section 10.1.1, Bob needs to send $\frac{n}{2^r} \log n$ bits to Alice for Alice to know the value of v_3. We choose $r = \log \log n$ for which the communication cost is $O(n \log \log n)$.

- $\mathsf{CASE\ 3:\ k = 4}$
  As before, Alice sends rn bits of her input. In return Bob sends two things: he sends $r.2^r$ bits per vertex for the $\frac{n}{2^r}$ potential vertices that Alice might point to. He also sends the value in v_0. On the last round, Alice sends $\frac{n}{2^{r.2^r}} \log n$ bits of her input which is of Bob's interest. If we choose $r = n \log\log\log n$, then we get an upper bound of $O(\log\log\log n)$ on the communication cost.

By similar argument, we can show an upper bound of $O(n \log^{k-1} n)$ on the communication cost of $Pointer_k$, where $\log^{k-1} n$ means taking repeated logarithm of n, $(k-1)$ times.