Lecture : Randomized Decision Tree Complexity

Scribe: Vatsal Pramod Jha   Lecture: Rajat Mittal

IIT Kanpur

1 Randomized Decision Trees

Recall that a randomized decision tree, say A, computing a boolean function f is a probability distribution over the Deterministic Decision trees.

Formally, we say that A computes f with bounded error ϵ if for every x we have:

$$\Pr_{D\sim A}(D(x) = f(x)) \geq 1 - \epsilon,$$

Here the probability is taken over all deterministic trees D having non-zero probability in the distribution defined by A. For simplicity, we will use A to represent both the randomized decision tree as well as the distribution defined by it.

The complexity or cost of a randomized decision tree A for an input x, denoted as $cost(A, x)$, has two alternative definitions:

- The first approach is of defining $cost(A, x)$ to be the expected cost of computing x by a deterministic tree D, where the expectation is taken according to the the probability distribution represented by A. Formally,

$$cost(A, x) := \mathop{\mathbf{E}}_{D\sim A}[cost(D, x)]$$

- The second approach is of defining the $cost(A, x)$ to be the worst-case cost i.e.:

$$cost(A, x) := \max_{D\in A:\Pr(D)\neq 0} cost(D, x).$$

We will see that for the bounded error case the above two definitions of $cost(A, x)$ are equivalent, in the sense that they differ by a constant factor.

Similar to the Deterministic tree complexity, the randomized decision tree complexity of a function f with bounded error ϵ is defined as:

$$\mathrm{R}_\epsilon(f) := \min_{A:A \text{ computes } f} \max_x cost(A, x).$$

For brevity of notation, $\mathrm{R}^1_\epsilon(f)$ will denote the worst-case cost while $\mathrm{R}^2_\epsilon(f)$ will denote the expected cost. We now relate the two measures for randomized costs.

Proposition 1. $\mathrm{R}^1_\epsilon(f) = \theta(\mathrm{R}^2_\epsilon(f))$.

Proof. In order to prove the above proposition it is necessary and sufficient to show that $\mathrm{R}^1_\epsilon(f) = O(\mathrm{R}^2_\epsilon(f))$ and $\mathrm{R}^2_\epsilon(f) = O(\mathrm{R}^1_\epsilon(f))$.

For simplicity let us assume $\epsilon = 1/3$.

By the definition of $\mathop{\mathbf{E}}_{D\sim A}[cost(D, x)]$, we obtain:

$$\mathop{\mathbf{E}}_{D\sim A}[cost(D, x)] = \sum_{D\in A:\Pr(D)\neq 0} cost(D, x) \Pr(D) \leq \max_{D\in A:\Pr(D)\neq 0} cost(D, x).$$

As the choice of the randomized decision tree A was arbitrary, we get:

$$\mathrm{R}^2_{1/3}(f) = O(\mathrm{R}^1_{1/3}(f)).$$

Now it remains to show that $\mathrm{R}^1_{1/3}(f) = O(\mathrm{R}^2_{1/3}(f))$.

For a randomized decision tree A and for a particular input x let:

$$\mathop{\mathbf{E}}_{D\sim A}[cost(D,x)] = d.$$

We now devise a randomized decision tree $A^{'}$, using A, in the following manner:

As soon as the $cost(D,x)$, for $D \in A$, becomes $\geq 100d$ the algorithm $A^{'}$ answers arbitrarily.

The rational behind defining $A^{'}$ in such a way is the observation: $\max_{D\in A'} cost(D,x) \leq 100d$.

Now we proceed to to show that $A^{'}$ computes f.

To prove the aforementioned claim we give an upper bound for the fraction of trees with depth $\geq 100d$ using Markov's inequality. This is done as follows:

$$\begin{aligned}
d = \mathop{\mathbf{E}}_{D\sim A}[cost(D,x)] &= \sum_{D\in A} cost(D,x)\Pr(D) \\
&= \sum_{D\in A: cost(D,x)<100d} cost(D,x)\Pr(D) + \sum_{D\in A: cost(D,x)\geq 100d} cost(D,x)\Pr(D) \\
&\geq \sum_{D\in A: cost(D,x)\geq 100d} cost(D,x)\Pr(D) \\
&\geq 100d \sum_{D\in A: cost(D,x)\geq 100d} \Pr(cost(D,x))
\end{aligned}$$

This implies,

$$\frac{1}{100} \geq \Pr(cost(D,x) \geq 100d).$$

In other words, trees with depth $\geq 100d$ occur with probability at most $1/100$.

Now as $\Pr_{D\in A}(D(x) = f(x)) \geq \frac{2}{3}$, hence we obtain:

$$\begin{aligned}
\Pr_{D\in A}(D(x) = f(x)) &= \Pr_{D\in A}(D(x) = f(x),\ cost(D,x) < 100d) + \Pr_{D\in A}(D(x) = f(x),\ cost(D,x) \geq 100d) \\
&= \Pr_{D\in A'}(D(x) = f(x)) + \Pr_{D\in A}(D(x) = f(x),\ cost(D,x) \geq 100d) \geq \frac{2}{3}
\end{aligned}$$

Now assuming that in the worst-case $A^{'}$ answers wrong whenever $cost(D,x) \geq 100d$, we get the following inequality:

$$\Pr_{D\in A'}(D(x) = f(x)) \geq \frac{2}{3} - \frac{1}{100} > \frac{1}{2}$$

Note 1. You can verify that the success probability here is greater than $1/2 + \epsilon$ for a constant ϵ. It is not enough to show that probability $> 1/2$.

The success probability of A' stated above can be improved by iterating A' on x for a suitable number of iterations (determined by the Chernoff bound).

This implies A' computes f with bounded error 1/3 such that for every input x, we have:

$$\max_{D:D\in A'} cost(D,x) \leq (100c)d,$$

where c is the factor introduced due to error reduction.

Hence $\max_{D\in A'}(cost(D,x)) = O(\mathop{\mathbf{E}}_{D\sim A}[cost(D,x)])$.

As the choice of the randomized decision tree A was arbitrary hence:

$$\mathrm{R}^1_{1/3}(f) = O(\mathrm{R}^2_{1/3}(f)).$$

This completes the proof of the present Proposition.

□

Generally, the complexity of a randomized decision tree A is taken to be its worst-case cost. Following the same convention henceforth, by $\mathrm{R}_\epsilon(f)$ we would refer to $\mathrm{R}^1_\epsilon(f)$. Also, for simplicity we fix ϵ to be equal to $\frac{1}{3}$. In fact any $\epsilon < \frac{1}{2}$ would have sufficed because of the Chernoff bound.

Note 2. It is sufficient that the error probability here is less than $1/2 - \epsilon$ for a constant ϵ. It is not enough to show that probability $< 1/2$.

1.1 Randomized Decision tree for OR

We now consider the randomized tree complexity of OR. The following proposition gives an upper bound on $\mathrm{R}_{1/3}(\mathrm{OR})$:

Proposition 2. $\mathrm{R}_{1/3}(\mathrm{OR}) \leq \frac{2n}{3}$

Proof. Consider the randomized algorithm, A, in which we randomly pick $\frac{2n}{3}$ indices from $[n]$ and check if one of the queried indices is a 1 or not. If one of the queries is 1 then we output 1 else 0.

Now if we show that A computes OR with bounded error $\leq \frac{1}{3}$ and has cost$\leq \frac{2n}{3}$ then we will be done with the proof.

Let us assume that the subsets of size $\frac{2n}{3}$ are picked uniformly. As there are $\binom{n}{\frac{2n}{3}}$ subsets of size $\frac{2n}{3}$ in $[n]$, the probability of picking a subset of size $\frac{2n}{3}$ is $\frac{1}{\binom{n}{\frac{2n}{3}}}$.

We now convert the above combinatorial picture to that of distribution over deterministic decision trees. Let us consider our randomized decision tree or equivalently the distribution, A, to be over those deterministic trees which query a string till either we get a 1 or a total of $\frac{2n}{3}$ queries are not made.

Clearly, the devised A has the worst-case complexity of $\frac{2n}{3}$.

It remains to show that A computes f with bounded error $\frac{1}{3}$.

It can be argued that the inputs for which A is most likely to fail will be the strings $x \in \{0,1\}^n$ having hamming weight 1.

Without loss of generality consider the input $x = (1, 0, 0..., 0)$ i.e. 1 with trailing zeros. The trees D in A which output a wrong answer for the given input are the ones which do not query x_1 i.e. the first bit of x. The number of such trees are $\binom{n-1}{2n/3}$.

Therefore the probability of failure of A on the given input x is equal to $\binom{n-1}{\frac{2n}{3}}/\binom{n}{\frac{2n}{3}} = \frac{1}{3}$.

Here we highlight that the order in which queries are made on x has been ignored but even if we did take the order of queries into account then we would have got an extra factor of $\left(\frac{2n}{3}\right)!$ which eventually gets cancelled in probability calculation.

By the above arguments it is implied that A computes OR with error $\leq \frac{1}{3}$ and has (worst-case) complexity equal to $\frac{2n}{3}$.

Hence $\mathrm{R}_{1/3}(\mathrm{OR}) \leq \frac{2n}{3}$. $\square$

The previous proposition gives us a non-trivial upper bound on $\mathrm{R}_{1/3}(\mathrm{OR})$ but we still have not made any similar claims about the lower bound for $\mathrm{R}_{1/3}(\mathrm{OR})$.

Intuitively, it seems that a randomized decision tree for OR might need to query almost all the input bits in order to determine the output i.e. $\mathrm{R}_{1/3}(\mathrm{OR})$ should be $\theta(n)$.

Indeed this is the case and we prove this formally in the following section.

2 Lower Bound Techniques for Randomized Decision Trees

Recall that the lower bounds for the Deterministic tree complexity of a boolean function f, i.e. $D(f)$, is calculated using $deg(f)$ or by the adversary argument.

Clearly, the degree argument fails for the randomized tree complexity as seen from the example for OR where $\mathrm{R}_{1/3}(\mathrm{OR}) \leq \frac{2n}{3}$ and $deg(\mathrm{OR}) = n$. Interestingly, the adversary argument works even for the randomized case.

We will be using the adversary argument to prove $\mathrm{R}_{1/3}(\mathrm{OR}) = \Omega(n)$.

But before that we need to review the notion of distributional complexity and Yao's minimax lemma which will be used in the proof of $\mathrm{R}_{1/3}(\mathrm{OR}) = \Omega(n)$ by adversary argument.

2.1 Yao's Minimax Lemma

In this subsection we review the Yao's Minimax Lemma along with its proof.

Yao's Minimax lemma provides a general strategy for lower bounding randomized algorithms. It relates the distributional complexity for a boolean function f to $\mathrm{R}_\epsilon(f)$.

Distributional Complexity The distributional complexity of a boolean function f with respect to a distribution μ on the inputs $\{0,1\}^n$ is defined as:

$$D_\mu(f) := \min_{D\colon D \text{ computes } f \text{ accdg } \mu} \max_{x \sim \mu} cost(D, x),$$

where a decision tree D is said to compute f according to μ if $\Pr_{x \sim \mu}(D(x) = f(x)) \geq \frac{2}{3}$.

Lemma 1. *(Yao's Minimax Lemma) Let f be a boolean function and let μ be a distribution over the inputs. Then Yao's Minimax Lemma states that:*

$$\max_\mu D_\mu(f) = \mathrm{R}_{1/3}(f).$$

Proof. We first show that $\max_\mu D_\mu(f) \leq \mathrm{R}_{1/3}(f)$.

Let A be an optimal randomized decision tree computing f i.e. $cost(A) = \mathrm{R}_{1/3}(f)$ with the associated distribution to be λ. Further, consider a distribution μ over the inputs $\{0,1\}^n$.

Our aim will be to show that there exists a decision tree D with non-zero probability in λ such that D computes f according to μ. The argument used will be the probabilistic version of equating the column sums and row sums

Let $\mathbb{1}_{(x,D)}$ denote the indicator function that $D(x) = f(x)$. This implies for all x we have:

$$\mathop{\mathbf{E}}_{D\sim\lambda}[\mathbb{1}_{(x,D)}] = \sum_{D\sim\lambda} \mathbb{1}_{(x,D)} \Pr(D) = \sum_{D:D\sim\lambda,\ D(x)=f(x)} \Pr(D) \geq \frac{2}{3}.$$

Now, consider the following expression:

$$\mathop{\mathbf{E}}_{x\sim\mu,\ D\sim\lambda}[\mathbb{1}_{(x,D)}] = \sum_{x\sim\mu,\ D\sim\lambda} \mathbb{1}_{(x,D)} \Pr(x,D).$$

As picking x and picking D are independent of each other we obtain:

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim\mu,\ D\sim\lambda}[\mathbb{1}_{(x,D)}] &= \sum_{x\sim\mu,\ D\sim\lambda} \mathbb{1}_{(x,D)} \Pr(x)\Pr(D) \\ &= \sum_{x\sim\mu} \Pr(x) \sum_{D\sim\lambda} \mathbb{1}_{(x,D)} \Pr(D)\end{aligned}$$

Now using $\sum_{D\sim\lambda} \mathbb{1}_{(x,D)} \Pr(D) \geq \frac{2}{3}$ we obtain the following inequality:

$$\mathop{\mathbf{E}}_{x\sim\mu,\ D\sim\lambda}[\mathbb{1}_{(x,D)}] \geq \frac{2}{3} \sum_{x\sim\mu} \Pr(x) = \frac{2}{3}.$$

Now taking the inner sum over x instead of D in the above double summation gives us:

$$\mathop{\mathbf{E}}_{x\sim\mu,\ D\sim\lambda}[\mathbb{1}_{(x,D)}] = \sum_{D\sim\lambda} \Pr(D) \sum_{x\sim\mu} \Pr(x)\mathbb{1}_{(x,D)} = \sum_{D\sim\lambda} \Pr(D) \sum_{x\sim\mu,\ D(x)=f(x)} \Pr(x).$$

We claim that there exists a D in λ satisfying:

$$\mathop{\mathbf{E}}_{x\sim\mu}[\mathbb{1}_{(x,D)}] = \sum_{x\sim\mu,\ D(x)=f(x)} \Pr(x) \geq \frac{2}{3}.$$

If it is not the case then we have a contradiction to the observation $\mathop{\mathbf{E}}_{x\sim\mu, D\sim\lambda}[\mathbb{1}_{(x,D)}] \geq \frac{2}{3}$.

The above claim is equivalent to claiming that there exists a D in λ which computes f according to the distribution μ.

As the choice of μ was arbitrary this implies:

$$\max_{\mu} D_\mu(f) \leq \mathrm{R}_{1/3}(f).$$

Now to prove the equality, it suffices to show that there exists a distribution μ over the inputs such that $D_\mu(f) = \mathrm{R}_\mu(f)$.

The existence of such a μ follows from duality theorems in linear programming. This completes the proof of Yao's Minimax lemma. □

Having proved the Yao's Minimax lemma we now proceed to obtain a lower bound on $\mathrm{R}_{1/3}(\mathrm{OR})$ using the adversary argument.

2.2 Lower bound for $\mathbf{R_{1/3}(OR)}$

Proposition 3.

$$\mathrm{R}_{1/3}(\mathrm{OR}) \geq \frac{n}{2}.$$

Proof. If we are able to prove that $\max_{\mu} D_\mu(\mathrm{OR}) \geq \frac{n}{2}$ then by Yao's Minimax Lemma we have our result.
To show that $\max_{\mu} D_\mu(\mathrm{OR}) \geq \frac{n}{2}$ we will use the adversary argument.
Assume that $\max_{\mu} D_\mu(\mathrm{OR}) < \frac{n}{2}$. This implies for every distribution μ over the inputs we have $D_\mu(\mathrm{OR}) < \frac{n}{2}$.

Now consider a decision tree D having depth $< \frac{n}{2}$ which computes OR according to the distribution, say μ, defined as follows:

$$\Pr(X = x) = \begin{cases} \frac{1}{3} + \frac{1}{n}, & \text{if } |x| = 0 \\ \\ \frac{2}{3n} - \frac{1}{n^2}, & \text{if } |x| = 1 \end{cases}.$$

For the all zero branch of D if the output on that branch is 1 then the error probability of D is $\frac{1}{3}+\frac{1}{n} > 1/3$.

While if the output is 0 then for more than half the inputs x having $|x| = 1$, the output will be 0 and the error probability of D in this case will be:

$$\left(\frac{n}{2} + 1\right)\left(\frac{2}{3n} - \frac{1}{n^2}\right) = \frac{1}{3} + \frac{1}{6n} - \frac{1}{n^2} > \frac{1}{3},$$

for $n > 6$.

This clearly contradicts the assumption that D computes OR according to the given distribution μ.

As our choice of D was arbitrary, by the above argument we have proved that there cannot exist a D with depth $< n/2$ that computes OR according to the distribution μ defined above.

Hence,

$$\mathrm{R}_{1/3}(\mathrm{OR}) = \max_{\mu} D_\mu(\mathrm{OR}) \geq \frac{n}{2}.$$

□

It can be observed that the key step in the adversary argument provided above was to identify the "hard" distribution μ, in order to apply Yao's lemma. Hence Yao's lemma evidently provides an adversary strategy for lower bounding $\mathrm{R}_\epsilon(f)$.