18 Lecture 18, December 3, 2014

18.1 Randomized and distributional complexity, error-free ("Las Vegas"): the minimax lower bound method

Let L be a computational problem, restricted to inputs of "size" n (however that is measured).

M_L = set of correct algorithms for L. (Regardless of run-time etc.)

X_L = set of inputs to L.

$$T : M_L \times X_L \to \mathbb{N}$$

$$T(r, x) = \text{ complexity of algorithm } r \text{ on input } x$$

The quantity we are usually interested in is:

Randomized complexity of L:

$$\text{Rand}(L) = \inf_{\substack{p \\ \text{dist. on } M_L}} \max_{x \in X_L} E_{r \in p} T(r, x)$$

(We write $r \in p$ to denote that r is sampled from distribution p.)

Giving an upper bound on Rand is, in principle, a straightforward task: devise an algorithm and bound its complexity. Well, it's not always easy, but it's clear "what" you have to do.

Lower bounds are another matter. How do you show that *no* algorithm can beat a certain complexity? Fortunately, there is an interesting way of going about this. We need the following definition.

Distributional complexity of L:

$$\text{Dist}(L) = \max_{\substack{q \\ \text{dist. on } X_L}} \inf_{r \in M_L} E_{x \in q} T(r, x)$$

Key facts are:

Weak LP duality:

$$\text{Rand}(L) \geq \text{Dist}(L)$$

Strong LP duality:

$$\text{Rand}(L) = \text{Dist}(L)$$

We need here only the easy part, weak LP duality, which is proven as follows: Take the optimum distributions p and q for each side. Writing T as an $|M_L| \times |X_L|$ matrix, p and q as column vectors, and denoting a singleton column vector by e_j, we have

$$\text{Rand}(L) = \max_x p^\dagger T e_x \geq p^\dagger T q \geq \min_r e_r^\dagger T q = \text{Dist}(L)$$

What this means is that instead of lower bounding the complexity of randomized algorithms on worst-case inputs, it is enough (the strong version tells us its actually equivalent) to lower bound the complexity of deterministic algorithms on random inputs. And even if we can't find the ideal distribution on inputs, we still get weaker lower bounds from suboptimal distributions so long as we can analyze them. This is a very powerful fact.

18.2 Using the minimax lower bound method

Let's apply this lower bound method to the query complexity of binary NAND trees.

It is very natural to try simply picking iid input bits. It turns out that if we do that, it's quite easy to analyze how efficient deterministic algorithms can be, because we have a very helpful theorem of Tarsi:

Theorem 77 (Tarsi [69]) *Let T be a finite NAND tree that is "spherically symmetric"—that is, all nodes at a common level have the same degree (in particular, all leaves are at the same level). Let the input bits be chosen iid. Then right-to-left DFS is an optimal algorithm in terms of the expected number of leaf queries.* (Proof omitted.)

Necessity of the spherical symmetry assumption: Consider the tree in Fig. 8, in which the triangle represents a large complete subtree of degree-2 NANDs Select input bits iid in such a manner that evey node in the tree retains probability bounded away from 0 and 1 of equallying 1. This can be accomplished by setting each input to 0 with probability p satisfying $p = (1-p)^2$, which happens to solve to $p = \frac{3-\sqrt{5}}{2}$. With this distribution, all vertices in a binary NAND tree, regardless of its structure, have probability p of equalling 0. Now, in the tree in the Figure, if x_n happens to be a 0, the formula value is set to 0. So the expected number of queries is minimized by checking this vertex first, before if necessary tackling the large complete subtree, which will certainly require a large number of queries.

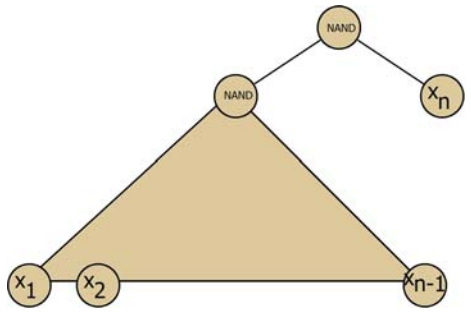

Figure 8: A non-spherically-symmetric tree

Applying Tarsi's theorem: We may now obtain a lower bound on the randomized query complexity of evaluating the complete binary NAND tree by evaluating the left-to-right DFS algorithm. Letting $\dot{T}_n^b$ be the complexity of this algorithm on this distribution, conditioned on the output value being b, we see that

$$\dot{T}_n^0 = 2\dot{T}_{n-1}^1$$

The more interesting case is next, when we examine the three different ways, totalling to $2p - p^2$, that one or both of the children can evaluate to 0. The first term here accounts for the children being $(0,1)$ or $(1,0)$; the second term accounts for the children being $(0,0)$:

$$\begin{aligned}\dot{T}_n^1 &= \frac{2p(1-p)}{2p-p^2}(\dot{T}_{n-1}^0 + \frac{1}{2}\dot{T}_{n-1}^1) + \frac{p^2}{2p-p^2}\dot{T}_{n-1}^0 \\ &= \dot{T}_{n-1}^0 + \frac{1-p}{2-p}\dot{T}_{n-1}^1\end{aligned}$$

Now we have the vector recursion

$$\begin{pmatrix}\dot{T}_n^0 \\ \dot{T}_n^1\end{pmatrix} = \begin{pmatrix}0 & 2 \\ 1 & \frac{1-p}{2-p}\end{pmatrix}\begin{pmatrix}\dot{T}_{n-1}^0 \\ \dot{T}_{n-1}^1\end{pmatrix}$$

When we substitute the above value of p, we find eigenvalues of $\frac{1+\sqrt{5}}{2}$ and $\frac{-4}{1+\sqrt{5}}$; the former is larger in absolute value, so we find that $\dot{T}_n^0, \dot{T}_n^1 \in \Theta((\frac{1+\sqrt{5}}{2})^n) \cong 1.618^n$, which is to say $N^{\lg \frac{1+\sqrt{5}}{2}} \cong N^{0.694}$ where $N = 2^n$ is the number of leaves.

This leaves a gap to the upper bound of $N^{\lg \frac{1+\sqrt{33}}{4}} \cong N^{0.7537}$. The weakness is with the lower bound. In the NAND tree if both inputs to a vertex are 0, no reasonable algorithm will read leaves of both sub-trees of that vertex. So, in order to prove the best lower bound, we have to choose a distribution on inputs that

precludes the possibility that both inputs to a node will be 0. Thus, a "hard" distribution will allow only the possibilities 01, 10 or 11 at the inputs to any node.

It is no longer possible to apply Tarsi's theorem to such distributions, since the inputs are not iid. However, a careful analysis by Saks and Wigderson [62] shows that the upper bound of $N^{\lg \frac{1+\sqrt{33}}{4}}$ is tight.

18.3 Randomized and distributional complexity with error ("Monte Carlo"):

We consider now randomized algorithms which are permitted to err but still have a worst-case guarantee: for all inputs x, the probability of error must be at most λ. A randomized algorithm can be described by a probability distribution p on seeds (strings of coin tosses) r. Each such seed may be correct on some x's and incorrect on other x's; so the guarantee that we have error probability at most λ on any input x is a constraint on the allowable distributions p. It is understood that our expressions are functions of the size parameter n, and that for any n, there are only finitely many inputs x. However there may be infinitely many seeds r.

The randomized complexity of problem L, with error parameter λ, is

$$\mathrm{Rand}_\lambda(L) = \inf_{\substack{p: \\ \text{err. prob.} \le \lambda \\ \text{for all } x}} \max_x E_{r \in p} T(r,x)$$

where $T(r,x)$ is the complexity (run-time, query complexity, etc.) of the algorithm using seed r on input x.

The distributional complexity of L is stated in terms of a distribution q on inputs x. Since there are only finitely many x of given size n, the space of distributions q is compact, so we write max rather than sup.

The error constraint in distributional complexity is that every seed r that is allowed, must err with probability at most λ on the distribution q.

$$\mathrm{Dist}_\lambda(L) = \max_q \inf_{\substack{r: \\ \text{err. prob.} \le \lambda \\ \text{on } q}} E_{x \in q} T(r,x)$$

We have the following extension of weak LP duality to the Monte Carlo case.

Theorem 78 (Yao) *Let a, b be a "conjugate pair": $a, b > 1$, $1/a + 1/b = 1$. Then* $\mathrm{Rand}_\lambda \ge \frac{1}{a} \mathrm{Dist}_{b\lambda}$. *In particular* $\mathrm{Rand}_\lambda \ge \frac{1}{2} \mathrm{Dist}_{2\lambda}$.

Proof: Fix any $\varepsilon > 0$. Let p_ε be a distribution on seeds r s.t.

$$\forall x: \quad E_{p_\varepsilon}(T(r,x)) \le \mathrm{Rand}_\lambda + \varepsilon/a \tag{36}$$

$$\forall x: \quad \Pr_{p_\varepsilon}(\text{Error on } x) \le \lambda \tag{37}$$

Let q be an arbitrary distribution on x. Set

$$T_q = E_{r \in p_\varepsilon, x \in q}(T(r,x)). \qquad \text{Note } T_q \le \mathrm{Rand}_\lambda + \varepsilon/a \quad \text{by 36}$$

$$\lambda_q = \Pr_{r \in p_\varepsilon, x \in q}(r \text{ errs on } x). \qquad \text{Note } \lambda_q \le \lambda \quad \text{by 37}$$

Let

$$K = \{r : E_{x \in q}(T(r,x)) > a T_q\}$$

$$L = \{r : \Pr_{x \in q}(r \text{ errs on } x) > b \lambda_q\}$$

Picture two tables:

$$\begin{array}{c} \\ r \end{array}\begin{array}{c} x \\ \begin{bmatrix} \dots & \dots & \dots \\ \dots & T(r,x) & \dots \\ \dots & \dots & \dots \end{bmatrix} \end{array} \qquad \begin{array}{c} \\ r \end{array}\begin{array}{c} x \\ \begin{bmatrix} \dots & \dots & \dots \\ \dots & [\![r \text{ errs on } x]\!] & \dots \\ \dots & \dots & \dots \end{bmatrix} \end{array}$$

K is a set of "bad" (high expectation over q) rows in the first table, L is a set of "bad" (high expectation over q) rows in the second table. Now by the Markov ineq.,

$$\Pr_{r \in p_\varepsilon}(K) < 1/a$$

$$\Pr_{r \in p_\varepsilon}(L) < 1/b.$$

We conclude by a simple union bound that for every distribution q, there is an r s.t.

$$\Pr_{x \in q}(r \text{ errs on } x) \leq b\lambda_q \leq b\lambda$$

$$E_{x \in q}(T(r,x)) \leq aT_q \leq a\,\mathrm{Rand}_\lambda + \varepsilon$$

Consequently,

$$\mathrm{Dist}_{b\lambda} \leq a\,\mathrm{Rand}_\lambda + \varepsilon$$

and since this holds for all $\varepsilon > 0$,

$$\mathrm{Dist}_{b\lambda} \leq a\,\mathrm{Rand}_\lambda\,.$$

□

18.4 An application of the Monte Carlo minimax lower bound

We consider two-party communication problems defined by boolean functions f. The setting is that Alice receives an input $x \in X$, Bob receives an input $y \in Y$, and they wish to jointly compute $f(x,y)$; the cost measure is the total number T of bits exchanged before Alice announces the output. There are various possible costs:

$$\text{deterministic } D(f) \geq \mathrm{Rand}^{\text{private}}(f) \text{ with private coins } \geq \mathrm{Rand}^{\text{public}}(f) \text{ with public coins.}$$

For the randomized complexities we may also consider their variants allowing λ probability of error,

$$\mathrm{Rand}^{\text{private}}_\lambda(f) \geq \mathrm{Rand}^{\text{public}}_\lambda(f).$$

Today we focus on $\mathrm{Rand}^{\text{public}}_\lambda(f)$. From the previous section we have

$$\mathrm{Rand}^{\text{public}}_\lambda(f) \geq \frac{1}{2}\,\mathrm{Dist}_{2\lambda}(f)$$

where

$$\begin{aligned} \mathrm{Dist}_{2\lambda}(f) &= \max_q \min_{\substack{r: \\ \text{err. prob.} \leq 2\lambda \\ \text{on } q}} E_{(x,y)\in q} T(r,(x,y)), \quad \text{equivalently} \\ \mathrm{Dist}_{2\lambda}(f) &= \max_q \mathrm{Dist}^q_{2\lambda}(f) \quad \text{for} \quad \mathrm{Dist}^q_{2\lambda}(f) = \min_{\substack{r: \\ \text{err. prob.} \leq 2\lambda \\ \text{on } q}} E_{(x,y)\in q} T(r,(x,y)) \end{aligned}$$

Here q is a probability distribution on the set $X \times Y$, r is a deterministic communication protocol achieving error probability $\leq 2\lambda$ on distribution q, and $T(r,(x,y))$ is the communication complexity of r on input (x,y).

(NB, the usual inf over r has been replaced by a min here because there are only finitely many possible communication protocols, given that X and Y are finite sets.)

It is worth emphasizing that Dist is defined by maximization over all *joint distributions* on the pair x, y. If one maximizes only over product distributions, one generally gets a much lower value. (Although in the example we'll give today, it just so happens that the distribution we'll use is a product distribution.)

So now we need a technique for lower bounding the communication complexity of protocols which can tolerate error. The classic method for this is the *discrepancy method.*

A "rectangle" is a set $R = A \times B$ for $A \subseteq X, B \subseteq Y$. The significance of rectangles is that at the end of any communication protocol, Alice and Bob have identified that their input belongs to some rectangle. The whole protocol partitions $X \times Y$ into such rectangles. Given q, the best strategy for the players upon reaching a rectangle $A \times B$ is to announce the output 0 or 1 depending on which of $q(R \cap f^{-1}(0))$ or $q(R \cap f^{-1}(1))$, respectively, is larger. The protocol is relatively likely to err on that rectangle if the two quantities are close. A protocol which achieves low error must, generally (w.r.t. distribution q), wind up in rectangles where the two quantities are not close.

This motivates the following definition. The discrepancy of rectangle R for boolean function f and distribution q is

$$\Delta(q, f, R) = |q(R \cap f^{-1}(0)) - q(R \cap f^{-1}(1))|$$

The overall discrepancy of f for distribution q is

$$\Delta(q, f) = \max_R |q(R \cap f^{-1}(0)) - q(R \cap f^{-1}(1))|$$

Theorem 79 *For $\varepsilon > 0$, $\mathrm{Dist}^q_{1/2-\varepsilon}(f) \geq \frac{\varepsilon}{2} \lg(\frac{\varepsilon}{\Delta(q,f)})$.*

Corollary 80 *For $\lambda < 1/2$, $\mathrm{Rand}^{public}_{\lambda}(f) \geq \max_q \frac{1/4-\lambda/2}{1/4+\lambda/2} \mathrm{Dist}^q_{1/4+\lambda/2}(f) \geq \max_q \frac{(1/4-\lambda/2)^2}{1/2+\lambda} \lg \frac{1/4-\lambda/2}{\Delta(q,f)}$.*

So ultimately to get a good lower bound on the Monte Carlo public coin complexity we'll want to choose a distribution q that achieves low discrepancy on large rectangles. That is of course problem-specific and can be interesting. Here we just prove Theorem 79 and then cite one good example.

18.4.1 This proof omitted in class:

Proof: of Theorem 79

Fix a deterministic protocol r computing f with error probability $\leq 1/2 - \varepsilon$ and $c \leq \mathrm{Dist}^q_{1/2-\varepsilon}(f)$ expected bits of communication on distribution q. By the Markov inequality, there is probability only $\varepsilon/2$ that r uses more than $2c/\varepsilon$ bits of communication, so we may abbreviate r to a protocol that $\bar{r}$ never uses more than $2c/\varepsilon$ bits of communication, and achieves error at most $1/2 - \varepsilon/2$.

Let $\bar{r}$ partition $X \times Y$ into rectangles $R_1, \ldots, R_C$ for $C \leq 2^{2c/\varepsilon}$. We may write $\bar{r}$ as a boolean function on the rectangles it defines, $\bar{r}(R_\ell)$, and also as a boolean function on individual inputs, $\bar{r}(x, y) = \bar{r}(R_\ell)$ if $(x, y) \in R_\ell$. Let $[\![\bar{r}(x, y) = f(x, y)]\!] = \{(x, y) : \bar{r}(x, y) = f(x, y)\}$. The guarantee that $\bar{r}$ errs with probability

$\leq 1/2 - \varepsilon/2$ implies that:

$$\begin{aligned}
\varepsilon &\leq q([\![\bar{r}(x,y) = f(x,y)]\!]) - q([\![\bar{r}(x,y) \neq f(x,y)]\!]) \\
&= \sum_\ell (q(R_\ell \cap [\![\bar{r}(x,y) = f(x,y)]\!]) - q(R_\ell \cap [\![\bar{r}(x,y) \neq f(x,y)]\!])) \\
&\leq \sum_\ell |q(R_\ell \cap [\![\bar{r}(x,y) = f(x,y)]\!]) - q(R_\ell \cap [\![\bar{r}(x,y) \neq f(x,y)]\!])| \\
&= \sum_\ell |q(R_\ell \cap [\![\bar{r}(x,y) = 0]\!]) - q(R_\ell \cap [\![\bar{r}(x,y) = 1]\!])| \\
&= \sum_\ell \left| q(R_\ell \cap \bar{r}^{-1}(0)) - q(R_\ell \cap \bar{r}^{-1}(1)) \right| \\
&= \sum_\ell \Delta(q, f, R_\ell) \\
&\leq 2^{2c/\varepsilon} \Delta(q, f)
\end{aligned}$$

So $2c/\varepsilon \geq \lg \frac{\varepsilon}{\Delta(q,f)}$ or $c \geq \frac{\varepsilon}{2} \lg \frac{\varepsilon}{\Delta(q,f)}$. □

18.4.2 Application to the inner product function

A good example of the application of this method is to the communication complexity of the *inner product function:* $X = Y = (\mathbb{Z}/2)^n$ and $\mathrm{IP}(x,y) = \sum x_i y_i$ (note, addition is mod 2). It can be shown (see [45] Ch. 3) that for $q =$ the uniform distribution, $\Delta(q, \mathrm{IP}) = 2^{-n/2}$. Applying Corollary 80 we now find: For $\lambda < 1/2$,

$$\mathrm{Rand}_\lambda^{\mathrm{public}}(\mathrm{IP}) \geq \frac{(1/4 - \lambda/2)^2}{1 + 2\lambda}(n - 2 \lg \frac{4}{1 - 2\lambda})$$

This implies a linear communication complexity for any error rate strictly below $1/2$.