0368.4163 Randomness and Computation June 17, 2009

Lecture 13

Lecturer: Ronitt Rubinfeld *Scribe: Cagan Roy, Estrin Michael*

1 Overview

1.1 Last Lecture: Boosting Weak Learners Into Strong Learners

In the last lecture, we showed that if any concept class can be weakly learned, then it can also be strongly learned. Our proof used a Boosting Algorithm that could boost any weak PAC-learner (an algorithm that could learn to do slightly better than average) into a strong PAC-learner (an algorithm that could learn to do a lot better than average).

1.2 This Lecture: Hard Functions, Uniformity, and Derandomization

The Boosting Algorithm we used was based on what was originally a complexity theory result. (It took the Machine Learning theorists a while to realize that it gave them a boosting algorithm.)

In this lecture, we mention the original complexity result (on the existence of a "hard" set of inputs). We use it to prove Yao's XOR Lemma, an analogous theorem for boosting hardness of functions: from any function that is *slightly*-hard-to-compute using small circuits, we can produce a *very*-hard-to-compute function.

1.3 Notations (reminder)

1. $R_c(x) = \begin{cases} +1 & \text{if } f(x) = c(x) \\ -1 & \text{o.w.} \end{cases}$ gives +1 if (weak) hypothesis c is right on example x
2. $N_i(x) = \sum_{1 \le j \le i} R_{c_j}(x)$ is the number of right c's exceeding the wrong ones (i.e. #right-#wrong)
3. $M_i(x) = \begin{cases} 1 & \text{if } N_i(x) \le 0 \\ 0 & \text{if } N_i(x) \ge \frac{1}{\varepsilon\gamma} \\ 1 - \varepsilon\gamma N_i(x) & \text{o.w.} \end{cases}$
   is a "Probabilty filter keeps sample x"- a "measure" which upper bounds the error of hypothesis $c = \text{Maj}(c_1, \ldots, c_i)$ on example x.
4. $|M_i| = \sum_x M_i(x)$ is the total "mass" of all examples according to "measure" M_i.
5. $\mu(Mi) = \frac{|M_i|}{2^n}$ is the normalized size of the "measure" given M_i.
6. $error(c_{i+1}) \equiv Pr_{x \in uniform}[c(x) \neq f(x)] \le \frac{|M_i|}{2^n}$
7. $D_{M_i}(x) = \frac{M_i(x)}{|M_i|}$ is a distribution over x
8. $\text{Adv}_c(M) = \sum_x R_c(x)M(x)$ is the advantage of c on M which gives an indication on the number of inputs for which c is correct. (Random guessing gives 0.)

   Note: $\text{Adv}_c(M) \ge \gamma|M|$ iff $\Pr_{x \in D_M}[c(x) = f(x)] \ge \frac{1}{2} + \frac{\gamma}{2}$

1.4 Non-Uniform Complexity Classes

Definition 1 *Let C be a class of languages, and take any function $a : \mathbb{N} \to \mathbb{N}$. The class C* with advice *a, C/a, is defined as the set of languages L such that $L \in C/a$ if and only if there is some $L' \in C$ such that for all n, there exists $\alpha_n \in \{0,1\}^*$ with $|\alpha_n| \leq a(n)$, and $x \in L$ if and only if $(x, \alpha_{|x|}) \in L'$.*

Example: $\mathrm{P/poly} = \bigcup \mathrm{P}/n^c$ - the class of languages recognizable in polynomial time with polynomial-sized advice. It can be shown that P/poly is also the set of languages computable by a polynomial-sized (non-uniform) circuit. (The circuits correspond exactly to the advice.)

It can be shown that $\mathrm{RP} \subseteq \mathrm{P/poly}$ and $\mathrm{BPP} \subseteq \mathrm{P/poly}$.

2 Yao's XOR Lemma

2.1 Goal

We would like to take a problem, which is hard on some inputs, and make it hard for all inputs.

2.2 Definition of Hard

Definition 2 *A function $f : \{\pm 1\}^n \to \{\pm 1\}$ is* δ-hard on distribution D for size g *if for any Boolean circuit C with at most g gates* $\Pr_{x \in_D \{\pm 1\}^n}[C(x) = f(x)] \leq 1 - \delta$.

In other words, f is δ-hard if there is some δ-fraction of the inputs that is hard to compute.

Definition 3 *Let M be a measure. If $\mathrm{Adv}_C(M) < \epsilon|M|$ for any circuit C of size g, we say that f is* ϵ-hard-core on M for size g.

(Note that if M is the characteristic function of a set S, then $Adv_C(M)$ ~ fraction of inputs on which a circuit of size g is correct)

Theorem 4 (Impagliazzo Hard-Core Set Theorem) *Let f be δ-hard for size g on the uniform distribution on n-bit strings, and let $0 < \epsilon < 1$. Then:*

1. *There is a measure M with $\mu(M) \geq \delta$ such that f is ϵ-hard-core on M for size $g' = \frac{\epsilon^2 \delta^2 g}{4}$*
2. *There is a 2ϵ-hard-core set S for f, for size g' and $|S| \geq \delta 2^n$*

Proof Idea

1. Assume not true. Then, for every measure M with $\mu(M) \geq \delta$, there is a circuit C of size g' that does well ($\mathrm{Adv}_C(M) \geq \epsilon|M|$) The existence of such a circuit (a "weak learner") implies the existence of a "strong learner", which is a circuit built by pasting together the "weak learner" outputs to get a circuit of size $\leq g$, which is correct on $> 1 - \delta$ fraction of the inputs. The existence of this "strong learner" is a contradiction to the assumption that f is δ-hard for size g.

2. The number of circuits of size g' is $\ll \frac{e^{2^n \epsilon^2 \delta^2/2}}{4}$. Pick S randomly from M (on D_m) and use the Chernoff bound to show that Pr[any circuit C of size g' has good advantage] is small.

■

2.3 Yao's XOR Lemma

We now state Yao's XOR Lemma, which says that for any hard function f, the XOR of many copies of f is even harder. Intuitively, this is true because computing XOR requires knowing either the value of f at each input or how many times we are wrong.

Definition 5 *For any function f, define a new function*

$$f^{\oplus k}(x_1, \ldots, x_k) = f(x_1) \oplus f(x_2) \cdots \oplus f(x_k)$$

Theorem 6 (Yao's XOR Lemma) *Assume f is $2\epsilon'$-hard-core for size g' on a set H s.t. $|H| \geq \delta 2^n$. Then $f^{\oplus k}$ is $2(\epsilon' + (1-\delta)^k)$-hard-core over $\{\pm 1\}^n$ for size g' according to the uniform distribution.*

Proof Suppose that f is $2\epsilon'$-hard-core on a set H, $|H| \geq \delta 2^n$, for size g'. For contradiction, assume also that there exists a circuit C of size at most g' which satisfies
$\Pr_{x_1,\ldots,x_k \in \{\pm 1\}^n}[C(x_1, \ldots, x_k) = f^{\oplus k}(x_1, \ldots, x_k)] \geq \frac{1}{2} + \epsilon' + (1-\delta)^k$.

Plan: For all H s.t. $|H| \geq \delta 2^n$, use C to produce a new circuit C' that computes f such that $|C'| \leq g'$ and $\Pr_{x \in H}[C'(x) = f(x)] \geq \frac{1}{2} + \epsilon'$, contradicting our assumption that f is $2\epsilon'$-hard-core on H.

Implementation: Given H such that $|H| > \delta 2^n$, construct C' :
Let A_m denote the event that exactly m of $x_1, \ldots, x_k$ are in H. (Exactly m of the x_i are in the "hard part" and exactly $k-m$ are in the "easy part".) Then the probability that none of the x_i are in H is at most $\Pr[A_0] = \Pr[\text{none of the } x_i \text{ are in } H] \leq (1-\delta)^k$,
So: $\Pr_{x_1,\ldots,x_k}[C(x_1, \ldots, x_k) = f^{\oplus k}(x_1, \ldots, x_k) | A_m \text{ for some } m > 0] \geq \frac{1}{2} + \epsilon'$.
This implies that there exists $m > 0$ s.t. $\Pr_{x_1,\ldots,x_k}[C(x_1, \ldots, x_k) = f^{\oplus k}(x_1, \ldots, x_k) | A_m] \geq \frac{1}{2} + \epsilon'$.
Now suppose we are given some random $x \in H$, we will compute $f(x)$ as follows:

1. Pick $x_1, \ldots, x_{m-1} \in_R H$
2. Pick $y_{m+1}, \ldots, y_k \in_R \overline{H}$
3. Permute $(x_1, \ldots, x_{m-1}, x, y_{m+1}, \ldots, y_k)$ via π - a random permutation on k elements
4. Xor result with "known" bit - Gives circuit of size $\leq g$

So we have:

$$\Pr_{x, x_i\text{'s}, y_j\text{'s}, \pi}[C(\pi(x_1, \ldots, x_{m-1}, x, y_{m+1}, \ldots, y_k)) = f^{\oplus k}(x_1, \ldots, x_{m-1}, x, y_{m+1}, \ldots, y_k)] \geq \frac{1}{2} + \epsilon'$$

Using an averaging argument, this means there is a choice of $x_1, \ldots, x_{m-1}, y_{m+1}, \ldots, y_k, \pi$ such that

$$\Pr_x[C(\pi(x_1, \ldots, x_{m-1}, x, y_{m+1}, \ldots, y_k)) = f^{\oplus k}(x_1, \ldots, x_{m-1}, x, y_{m+1}, \ldots, y_k)] \geq \frac{1}{2} + \epsilon'$$

Notice that $f^{\oplus k}(x_1, \ldots, x_{m-1}, x, y_{m+1}, \ldots, y_k)] = f(x) + [\bigoplus f(x_i) + \bigoplus f(y_j)]$

Thus, we have a circuit C' of size at most g' such that $\Pr_x[C'(x) = f(x)] \geq \frac{1}{2} + \epsilon'$, which is a contradiction to the assumption that f is hard-core for size g' on the set H. ■