Derandomization (I)

Yijia Chen

Shanghai Jiao Tong University

November 11, 2012

Randomized Computation

Definition

A *probabilistic Turing machine* (PTM) is a Turing machine with two transition functions δ_0 and δ_1. To execute a PTM $\mathbb{M}$ on an input x, we choose in each step independently with probability $1/2$ to apply the transition function δ_0 and with probability $1/2$ to apply δ.

The machine only outputs 1 (Accept) or 0 (Reject). $\mathbb{M}(x)$ is the random variable corresponding to the value $\mathbb{M}$ writes at the end of this process.

For a function $T : \mathbb{N} \to \mathbb{N}$, we say that $\mathbb{M}$ runs in $T(n)$-time if for any input x, $\mathbb{M}$ halts on x within $T(|x|)$ steps regardless of the random choices it makes.

BPTIME and BPP

Definition

For $T : \mathbb{N} \to \mathbb{N}$ and $L \subseteq \{0,1\}^*$, a PTM $\mathbb{M}$ decides L in time $T(n)$, if for every $x \in \{0,1\}^*$, the machine $\mathbb{M}$ halts in $T(|x|)$ steps (i.e., $\mathbb{M}$ runs in $T(n)$-time), and

$$\Pr[\mathbb{M}(x) = L(x)] \geq \frac{2}{3},$$

where $L(x) = 1$ if $x \in L$ and $L(x) = 0$ if $x \notin L$.

Then $\mathbf{BPTIME}(T(n))$ is the class of languages decided by PTMs in $O(T(n))$ time and

$$\mathbf{BPP} := \bigcup_{d \in \mathbb{N}} \mathbf{BPTIME}(n^d).$$

Theorem
$\mathbf{P} \subseteq \mathbf{BPP}$.

Remark. *It is open whether* $\mathbf{BPP} \subseteq \mathbf{NP}$ *or* $\mathbf{NP} \subseteq \mathbf{BPP}$.

Conjecture
$\mathbf{P} = \mathbf{BPP}$.

Theorem

$L \in \mathbf{BPP}$ if and only if there exists a polynomial-time TM $\mathbb{M}$ and a polynomial $p \in \mathbb{N}[X]$ such that for every $x \in \{0,1\}^*$,

$$\Pr_{r \in \{0,1\}^{p(|x|)}} \left[\mathbb{M}(x, r) = L(x) \right] \geq \frac{2}{3}.$$

Definition

An *n-variable algebraic circuit* is a directed acyclic graph with the sources labeled by a variable name from the set $x_1, \ldots, x_n$, and each non-source node has in-degree two and is labeled by an operator from the set $\{+, -, \times\}$. There is a single sink in the graph, i.e., the *output* node.

Definition

$$\text{ZEROP} = \{ C \mid C \text{ an algebraic circuit that always outputs zero} \}.$$

Why ZEROP looks difficult?

The polynomial

$$\prod_{i\in[n]}(1+x_i)$$

can be computed using a circuit of size $2 \cdot n$ but has 2^n terms in its coefficient representation.

Schwartz-Zippel Lemma

Lemma

Let $p(x_1, x_2, \ldots, x_n)$ *be a polynomial of total degree at most* d *and* S *a finite set of integers. When* $a_1, a_2, \ldots, a_n$ *are randomly chosen with replacement from* S*, then*

$$\Pr\left[p(a_1, a_2, \ldots, a_n) \neq 0\right] \geq 1 - \frac{d}{|S|}.$$

A naive algorithm

A circuit of size m on n variables defines a polynomial of degree at most 2^m.

1. Choose n random numbers $x_1, \ldots, x_n$ from 1 to $10 \cdot 2^m$ (this requires $O(n \cdot m)$ random bits).
2. Evaluate the circuit C on $x_1, \ldots, x_n$ to obtain an output y.
3. Accept if $y = 0$, and reject otherwise.

Problematic: intermediate values as large as

$$(10 \cdot 2^m)^{2^m}.$$

ZEROP $\in$ **BPP**

1. Choose n random numbers $x_1, \ldots, x_n$ from 1 to $10 \cdot 2^m$.
2. Choose a random number $k \in [2^{2 \cdot m}]$ uniformly at random.
3. Evaluate the circuit C on $x_1, \ldots, x_n$ *modulo k* to obtain an output $y \mod k$ where $y = C(x_1, \ldots, x_n)$.
4. Accept if $y \mod k = 0$, and reject otherwise.

The correctness of the algorithm

Trivially $\Pr[\mathbb{M} \text{ accepts } C] = 1$, if $C = 0$. So assume $C \neq 0$, then we will show

$$\Pr[\mathbb{M} \text{ rejects } C] \geq \delta,$$

where $\delta = 1/(4 \cdot m)$.

Let $S := \left\{p_1, \ldots, p_\ell\right\}$ be the distinct prime factors of y.
By the Prime Number Theorem,

$$\Pr_{k \in [2^{2 \cdot m}]} \left[k \text{ is prime}\right] \geq \frac{1}{2 \cdot m} = 2 \cdot \delta.$$

y can have at most $\log y \leq 5 \cdot m \cdot 2^m$ distinct factors,

$$\Pr[k \in S] \leq \frac{5 \cdot m \cdot 2^m}{2^{2 \cdot m}} < \delta$$

Hence, $\Pr\left[k \text{ does not divide } y\right] \geq \Pr[k \text{ is a prime not in } S] \geq 2 \cdot \delta - \delta = \delta.$

Pseudorandom Generators

Definition

Let R be a distribution over $\{0,1\}^m$, $S \in \mathbb{N}$, and $\varepsilon > 0$. Then R is an (S, ε)-*pseudorandom distribution* if for every circuit C of size at most S,

$$\Pr[C(R) = 1] - \Pr[C(U_m) = 1]| < \varepsilon,$$

where U_m is the uniform distribution over $\{0,1\}^m$.

Let $S : \mathbb{N} \to \mathbb{N}$ be a function. A 2^n-time computable function $G : \{0,1\}^* \to \{0,1\}^*$ is an $S(\ell)$-*pseudorandom generator* if $|G(z)| = S(|z|)$ for every $z \in \{0,1\}^*$ and for every $\ell \in \mathbb{N}$ the distribution $G(U_\ell)$ is $(S(\ell)^3, 1/10)$-pseudorandom.

Derandomize **BPP**

Theorem

If there exists a $2^{\lceil \ell/a \rceil}$*-pseudorandom generator for some constant* $a \in \mathbb{N}$ *then* $\mathbf{BPP} = \mathbf{P}$.

Proof (1)

Let $L \in \mathbf{BPP}$. Assume that there is an algorithm $\mathbb{A}$ that on input $x \in \{0,1\}^n$ runs in time $n^d = 2^{d \cdot a \cdot \log n / a}$ for some constant $d \in \mathbb{N}$, such that

$$\Pr_{r \in \{0,1\}^{n^d}} \left[\mathbb{A}(x, r) = L(x)\right] \geq \frac{2}{3}.$$

Consider the *deterministic algorithm* $\mathbb{B}$:

> *On input $x \in \{0,1\}^n$, go over all $z \in \{0,1\}^{d \cdot a \cdot \log n}$, compute $\mathbb{A}(x, G(z))$ and output the majority answer.*

$\mathbb{B}$ runs in time $2^{O(d \cdot a \cdot \log n)} = n^{O(1)}$.

Proof (2)

Claim: Let $n \in \mathbb{N}$ and $x \in \{0,1\}^n$

$$\Pr_{z \in \{0,1\}^{d \cdot a \cdot \log n}} \left[\mathbb{A}(x, G(z)) = L(x)\right] \geq \frac{2}{3} - 0.1.$$

Assume otherwise, then

$$\Pr_{r \in \{0,1\}^{n^d}} \left[\mathbb{A}(x, r) = L(x)\right] - \Pr_{z \in \{0,1\}^{d \cdot a \cdot \log n}} \left[\mathbb{A}(x, G(z)) = L(x)\right] > 0.1.$$

Consider the circuit C defined by

$$C(r) \mapsto \mathbb{A}(x, r).$$

If $L(x) = 1$, then $\Pr\left[C(U_{n^d}) = 1\right] - \Pr\left[C(G(U_{d \cdot a \cdot \log n})) = 1\right] > 0.1$,
If $L(x) = 0$, then $\Pr\left[C(G(U_{d \cdot a \cdot \log n})) = 1\right] - \Pr\left[C(U_{n^d}) = 1\right] > 0.1$.

Thank you