CSE200 Lecture Notes – Classes of problems with probabilistic algorithms

Lecture by Russell Impagliazzo
Notes by Jiawei Gao

March 3-8, 2016

1 Complexity classes

- BPP
  - $x \in L \implies \text{Prob}[\text{accept}] \geq 3/4$
  - $x \notin L \implies \text{Prob}[\text{reject}] \geq 3/4$
- RP
  - $x \in L \implies \text{Prob}[\text{accept}] \geq 3/4$
  - $x \notin L \implies \text{Prob}[\text{reject}] = 1$

  The accepting run is a "witness" that $x \in L$. So RP $\subseteq$ NP.
- co-RP
  - $x \in L \implies \text{Prob}[\text{accept}] = 1$
  - $x \notin L \implies \text{Prob}[\text{reject}] \geq 3/4$

  The rejecting run is a "witness" that $x \notin L$. So co-RP $\subseteq$ co-NP.
- ZPP
  - $x \in L \implies \text{Prob}[\text{accept}] \geq 3/4$ and $\text{Prob}[\text{reject}] = 0$
  - $x \notin L \implies \text{Prob}[\text{reject}] \geq 3/4$ and $\text{Prob}[\text{accept}] = 0$

- PrBPP (Promise-BPP, a class of problems, instead of a class of languages)
  - $x \in$ yes instances, then $\text{Prob}[\text{accept}] \geq 3/4$
  - $x \in$ no instances, then $\text{Prob}[\text{reject}] \geq 3/4$

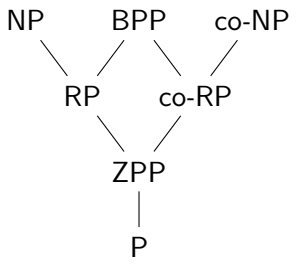

Figure 1: relations between classes

Theorem 1. $\mathsf{ZPP} = \mathsf{RP} \cap \text{co-RP}$.

Theorem 2. If $\mathsf{P} = \mathsf{NP}$, then $\mathsf{BPP} \subseteq \mathsf{P}$.

Later on we will show BPP is in the second level of PH. Thus if P = NP then PH collapses to P so BPP = P.

2 BPP problems have polynomial size circuits

2.1 Circuit complexity

For Boolean function $f : \{0,1\}^n \to \{0,1\}$, we use $\mathsf{Size}(f)$ to denote the minimum number of gates in a Boolean circuit that computes f.

For function $f : \{0,1\}^* \to \{0,1\}$, define f_n as f restricted to inputs of length n. We look at the growth of $\mathsf{Size}(f_1), \ldots, \mathsf{Size}(f_n), \ldots$ We say f is in the class $\mathsf{P/poly}$ if $\mathsf{Size}(f_n) \in O(n^k)$ for some k.

Every polynomial time algorithm can be simulated by a polynomial size circuit, so $\mathsf{P} \subseteq \mathsf{P/poly}$. A function f is exponentially hard if $\mathsf{Size}(f_n) \in 2^{\Omega(n)}$. A classical result by Riordan, Shannon and Lupanov is that the circuit complexity of any function is at most $2^n/n$.

For probabilistic algorithm $A(x, r)$ (where x is the input and r is the random bits used by A), on each fixed input length n', we can construct circuit $C_{n'}(x, r)$ in polynomial time. To decide if $x \in L$, we need to compute the fraction of r's that satisfy $C_{n'}(x, r)$. Suppose we fix x and define circuits $C_{x,n'}(r) = C_{n'}(x, r)$.

The *circuit estimation problem* is to estimate the number of r's that satisfy $C_{x,n'}(r)$. Given a circuit $C(r)$, estimate $\mathrm{Prob}_r[C(r) = 1]$ to within additive error $\pm 1/4$.

If we can solve this problem, then we can solve any BPP problem. For $L \in \mathsf{BPP}$, to decide if $x \in L$, we estimate $\mathrm{Prob}_r[C(r) = 1]$. If it's greater than $1/2$, then $x \in L$, otherwise $x \notin L$.

2.2 Pseudorandomness

The number of different r's is $2^{|r|}$, which is exponentially large, so we cannot afford exhaustive search on all r's. Instead, we use a pseudorandom generator that takes in a seed s and outputs a pseudorandom string r. The length of s is much shorter than r. Then we can exhaustive search over seed s to produce the pseudorandom strings in much less time than exaustive searching over true random strings.

Let $R = \{r_1, \ldots, r_{2^{|s|}}\}$ be the pseudorandom strings. We assume $2^{|s|} \ll 2^{|r|}$.

Define

$$Est_R(C) = \frac{|\{r \in R \mid [C(r) = 1]\}|}{|R|}.$$

If R is a collection of n bit strings, we say R is *ϵ-pseudorandom for size m* if for all circuits C with at most m gates,

$$\left|Est_R(C) - \text{Prob}_{r\in\{0,1\}^n}[C(r)=1]\right| \leq \epsilon.$$

In other words, strings selected from R behave like random strings, so we can use R to "fool" circuits with at most m gates.

Lemma 3 (Chernoff bound)**.** For $i \in \{1,\ldots,T\}$, if X_i are the independent events that occur with probability p_i, and $X = \sum_i X_i$ is the total number of events that occur, then

$$\text{Prob}[|X - E[X]| \geq k] \leq e^{-k^2/2T}.$$

In a circuit of m gates, for each gate we have constant number of choices for the operator, and m^2 choices for input gates, so there are $(cm^2)^m = m^{O(m)}$ different circuits. Thus,

Lemma 4. There are at most $m^{O(m)}$ circuits with m gates.

Let $T = O(m \log m)$, and consider picking strings $r_1,\ldots,r_T$ at random and let $R = \{r_1,\ldots,r_T\}$. Fix C with at most m gates. And let $p = \text{Prob}_r[C(r)=1]$.

Let variable X_i be

$$X_i = \begin{cases} 1, \text{if } C(r_i)=1 \\ 0, \text{otherwise} \end{cases}$$

X_i's are independent random variables with probability p each.

Circuit C can distinguish whether r is selected from R or from $\{0,1\}^{|R|}$ iff $\left|\frac{1}{T}\sum_i X_i - p\right| \geq \epsilon$ iff $\left|\sum_i X_i - pT\right| \geq \epsilon T$. By Chernoff bound, $\text{Prob}_R\left[\left|\sum_i X_i - pT\right| \geq \epsilon T\right] \leq e^{-(Te)^2/2T} = e^{-\epsilon^2 T/2}$.

By Lemma 4, $\text{Prob}_R[\exists$ circuit of size m so that our estimation is off by $\epsilon] \leq e^{-\epsilon^2 T/2} \cdot m^{O(m)}$. Since $m^{O(m)} = e^{O(m\log m)}$ and $T \gg O(\frac{m\log m}{\epsilon^2})$, the probability is less than 1. So these exists a set R of size $O(m\log m/\epsilon^2)$, so that any circuit C of m gates cannot distinguishing whether r is selected from R or from $\{0,1\}^{|r|}$.

This method is called the probabilistic method. We just proved the existence of such an R, but how to construct such an R is still an open problem. If we can explicitly construct R, then we can derandomize any problem in BPP with deterministic algorithms.

Theorem 5.

$$\mathsf{BPP} \subseteq \mathsf{P/poly}.$$

We get back to probabilistic algorithm $A(x,r)$, and the circuit $C(x,r)$ that simulates A. Let m be the size of $C_x(r)$, and Fix $R = r_1,\ldots,r_T$ to be the set that fools size m circuits.

Construct circuit $C'(x,r_1,\ldots,r_T)$ which accepts iff $\frac{1}{T}\sum_{i=1}^{T} C(x,r_i) \geq \frac{1}{2}$.

3 BPP is in PH

Theorem 6.

$$\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}}.$$

Proof idea. Let there be a 2-player game, where player 1 selects the set $R_1 = \{r_{1,1}, \ldots, r_{T,1}\}$ and player 2 selects the set $R_2 = \{r_{1,2}, \ldots, r_{T,2}\}$. We define a $T \times T$ matrix, where the $(i,j)^{\text{th}}$entry is $C(r_{i,1} \oplus r_{j,2})$.

$$\begin{array}{c} \\ r_{1,2} \\ \vdots \\ r_{j,2} \\ \vdots \\ r_{T,2} \end{array} \begin{array}{c} \begin{array}{cccccc} r_{1,1} & \cdots & r_{i,1} & \cdots & r_{T,1} \end{array} \\ \begin{pmatrix} C(r_{i,1} \oplus r_{1,2}) & & & & \\ & & & & \\ & & C(r_{i,1} \oplus r_{j,2}) & & \\ & & & & \\ & & & & C(r_{T,1} \oplus r_{T,2}) \end{pmatrix} \end{array}$$

Suppose the probabilities of C accepting on each row are $p_1, \ldots, p_T$, and on each column are $q_1, \ldots, q_T$.

For fixed r', let circuit $C_{r'} = C(r \oplus r')$. For random string r, $r \oplus r'$ is also random, so $\text{Prob}_r[C_{r'}(r) = 1] = \text{Prob}_r[C(r) = 1]$. Therefore, if p is the probability that $C(r) = 1$, then each p_i and each q_i should be $p \pm \epsilon$. Thus each p_i (or q_i) are all within 2ϵ of each other.

After player 1 having selected the $r_{i,1}$'s and player 2 having selected the $r_{i,2}$'s, a referee checks:

- If any two rows differ by more than 2ϵ, then player 2 wins.
- If any two columns differ by more than 2ϵ, then player 1 wins.

The game is symmetric, so each player must have a strategy to prevent the opponent from winning. Then it means for all rows and columns, the probabilities are consistent to each other. Also, because in a matrix, the sum of column equals the sum of row, the estimate of each row (and column) should be equal to the average value, which is the answer of circuit estimation problem.

So the problem becomes deciding $\exists R_1 \forall R_2 \left(\bigwedge_{i=1}^{T} q_i = p \pm \epsilon \right) \wedge A(x, R_1)$, where A is an algorithm using R_1 to derandomize the BPP problem as in Theorem 5.

The details of this proof will be given in Tuesday's class. □

4 Updated on March 8

For a fixed string s, define circuit $C_s(x) = C(x \oplus s)$. Then $\text{Prob}_r[C_s(r) = 1] = \text{Prob}_r[C(r \oplus s) = 1] = \text{Prob}_r[C(r) = 1]$.

If $\text{Prob}[C(r)] \geq 0.75$, we can pick R to be an 0.01-pseudorandom set. $\forall r'$, $Est_R(C_{r'}) \geq \text{Prob}_r[C_{r'}(r) = 1] - 0.01 \geq 0.74$.

If $\text{Prob}[C(r) = 1] \leq 0.25$, fix $R = \{r_1, \ldots, r_T\}$. Pick $r' \in_u \{0.1\}^n$. $\text{E}[Est_R(C_{r'})] = \text{E}[\frac{1}{T} \sum_{i=1}^{T} C(r' \oplus r_i)] = \frac{1}{T} \cdot T\text{Prob}[C(r') = 1]$. So $\forall R \exists r' \in_u \{0,1\}$ $Est_R(C_{r'}) \leq 0.25$.

Let R be a 0.01-pseudorandom set. If $\text{Prob}[C(r) = 1] \geq 3/4$, then $\exists R = \{r_1, \ldots, r_T\}$ so that

$\forall r'\ Est_R(C_{r'}) \geq 0.7$. If Prob$[C(r) = 1] < 1/4$, then $\forall R = \{r_1, \ldots, r_T\} \exists r'\ Est_R(C_{r'}) \leq 0.25$. We set $T = O(m \log m/\epsilon^2)$. so that the size of R is polynomial.

For a language L in BPP computed by probabilistic algorithm $A(x, r)$ or circuit $C(r)$ on fixed x, we decide whether $x \in L$ by deciding if $\exists R \forall r' Est_R(C_{r'}) \geq 0.7$. The size of R and the time for computing Est_R are polynomials. So $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}}$. Because BPP is closed under complementation, also $\mathsf{BPP} \subseteq \Pi_2^{\mathsf{P}}$.

An open problem is whether $\mathsf{BPP} \subseteq \mathsf{P}^{\mathsf{NP}}$.

If $\mathsf{P} = \mathsf{NP}$, then $\mathsf{P} = \mathsf{BPP}$.

If $\mathsf{NP} \subseteq \mathsf{BPP}$, then $\mathsf{PH} \subseteq \mathsf{BPP}$. (Corollary: If $\mathsf{NP} \subseteq \mathsf{BPP}$ then $\mathsf{PH} \subseteq \Sigma_2^{\mathsf{P}}$.)