Lecture 11

Lecturer: Madhu Sudan *Scribe: Colin Jia Zheng*

1 Recap

We defined **RP** as the class of languages accepted by PPT machine with one-sided error bounded below 1/3, **BPP** with two-sided error with gap 1/3. **RP** was shown to be robust in the following sense.

Define $\mathbf{RP}_e$ such that $L \in \mathbf{RP}_e$ if for some poly-time TM M and random bits y,

$$x \in L \Rightarrow \Pr[M(x,y) \text{ rejects}] \leq e(|x|)$$

$$x \notin L \Rightarrow \Pr[M(x,y) \text{ accepts}] = 0$$

Then $\mathbf{RP}_{1-1/\text{poly}(n)} = \mathbf{RP} = \mathbf{RP}_{1/2^{\text{poly}(n)}}$ (the two poly's may be different polynomials), yet $\mathbf{RP}_{1-1/2^n} = \mathbf{NP}$.

We will see that **BPP** is robust in the similar sense. Define $\mathbf{BPP}_{c,s}$ such that $L \in \mathbf{BPP}_{c,s}$ if for some poly-time TM M and random bits y,

$$x \in L \Rightarrow \Pr[M(x,y) \text{ accepts}] \geq c(|x|)$$

$$x \notin L \Rightarrow \Pr[M(x,y) \text{ accepts}] \leq s(|x|)$$

Let us assume that, as often necessary, that s is "nice", ie fully time constructible.

(Quick note: If $c \leq s$ then $\mathbf{BPP}_{c,s}$ would contain every language. While it is not required that $c(n) \geq 0.5$ and $s(n) \leq 0.5$, one can shift the probability by proper amount so that c, s do straddle 0.5.)

2 Amplification for BPP

Using Chernoff bound we will see that $\mathbf{BPP}_{f(n)+1/\text{poly}(n), f(n)-1/\text{poly}(n)} = \mathbf{BPP} = \mathbf{BPP}_{1-2^{-\text{poly}(n)}, 2^{-\text{poly}(n)}}$.

Theorem 1 *(Chernoff bound) Let $X_1, \ldots, X_k \in [0,1]$ be independent random variables and $X = \sum_i X_i/t$. Then $\Pr[|X - E[X]| \geq \epsilon] \leq e^{-(k\epsilon^2/2)}$.*

Suppose some poly-time TM M places L in $\mathbf{BPP}_{f(n)+1/p(n), f(n)-1/p(n)}$ where p is a polynomial, and f a "nice" function. Intuitively if one runs M for k times (with different random bits) and output according to whether the average of k answers exceeds $f(n)$, the error probability should decrease.

By how much? Let random variable X_i denote the output of ith run. For $x \in L$, error occurs if $\sum_i X_i/k < f(n)$ ie at least $p(n)$ off expectation, thus with probability $O(e^{-k/2p(n)^2})$ by Chernoff bound. With k polynomial in n, this can be as small as $2^{-q(n)}$ for any polynomial q. Likewise for $x \notin L$.

Here we have used polynomially many more random bits to reduce error. Can we do with fewer? The state-of-art, using ideas from pseudorandomness (ie expanders), is that $O(k)$ extra random bits *can* reduce error from 1/3 to 2^{-k}.

3 BPP $\subseteq$ P/poly

In advice (ie non-uniform) classes, one piece of (short) advice is expected to help *all* 2^n computations on length n input. This might seem weak at first, but often times randomization is not more powerful than non-uniformity. In particular Adleman showed that $\mathbf{BPP} \subset \mathbf{P}/\text{poly}$.

Suppose machine M places L in **BPP** with error probability below $2^{-p(n)}$, $p(n) > n$ (okay due to amplification). Is there a random string y good for all 2^n inputs of length n, ie $M(x,y) = L(x)$ for each $x \in \{0,1\}^n$? Indeed, for each x only a $2^{-p(n)}$ fraction of all random strings are bad; summing over all 2^n possible x this fraction is still below 1! Thus some advice works for *all* inputs as random tape.

4 $\mathbf{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$

How about some uniform class upper bounding **BPP**? It is clear that $\mathbf{BPP} \subseteq \mathbf{PSPACE}$; it is unclear how **BPP** is related to **NP**. Nevertheless we can show something intermediate: $\mathbf{BPP} \subseteq \Sigma_2^p$. (Which implies $\mathbf{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$ as **BPP** is closed under complementation.)

As before, suppose some machine M places L in **BPP** with error probability below 2^{-n}. Let x be a length n input, and M uses m random bits on x.

(Note that letting $\exists$-player show a set of polynomially many strings good for x, as evidence, is not enough. To decide L by a 2-round debate one must ensure some kind of "fairness", eg say one might let $\exists$-player to produce first half bits of y, $\forall$-player second half, and see if $M(x, y)$ accepts. This is still too crude to work, but illustrates the point.)

The idea is we do let $\exists$-player show a set of polynomially many strings good for x, *and* the $\forall$-player tries to find some bijection mapping *all* of them to strings *bad* for x. The bijections allowed are $\oplus y'$ for any y'. Intuitively, for $x \in L$ it is hard for $\forall$-player to come up with such bijection that works on *all* good strings, and for $x \notin L$ it is easy (and "easy" in a stronger sense than it is hard in the $x \in L$ case).

Formally, one claims

$$L = \{x : \exists y_1, \ldots, y_m \forall y' [\bigvee_{1 \le i \le m} M(x, y_i \oplus y') = 1]\}$$

Proof. Suppose $x \in L$. Imagine one picks $y_1, \ldots, y_m$ at random. Probability that $\bigwedge_i M(x, y_i \oplus y') = 0$, for each y', is below 2^{-mn}; union bound over all possible y' the probability is still below 1, ie *some* $y_1, \ldots, y_m$ make this false for all y'.

Now suppose $x \notin L$. Imagine one picks y' at random. Probability that $\bigvee_i M(x, y_i \oplus y') = 1$, for each $y_1, \ldots, y_m$, is at most $m2^{-n} < 1$, ie *some* y' makes this false for all $y_1, \ldots, y_m$.

This very idea can also be used to show $\mathbf{promiseBPP} \subseteq \mathbf{promiseRP}^{\mathbf{promiseRP}}$ (ie if $\mathbf{P} = \mathbf{promiseRP}$ then $\mathbf{P} = \mathbf{promiseBPP}$).

5 Next time

We will talk about promise problems, which arise naturally eg when **BPP** has no known complete problem (as we don't know how to enumerate error-bounded PPTs, ie to verify error-bounded-ness) yet **promiseBPP** has complete problems (eg given input (M, x), promised that M is indeed error bounded, does $M(x) = 1$?).

We will talk about the complexity of $\mathrm{UNIQUE} - \mathrm{SAT}$, ie SAT with the promise that the satisfying assignment is either unique or non-existing. Is $\mathrm{UNIQUE} - \mathrm{SAT}$ hard? We shall see that $\mathbf{NP} \neq \mathbf{RP} \Rightarrow \mathrm{UNIQUE} - \mathrm{SAT}$ hard. (This problem arises in cryptography, where we want a mapping easy to compute one-way, but hard to revert. The mapping certainly should be one-to-one, so $\mathrm{UNIQUE} - \mathrm{SAT}$ may be a good candidate.)