3 A Probabilistic Computation

CS 6810 – Theory of Computing, Fall 2012
Instructor: David Steurer
Scribe: Costandino Dufort Moraites (cdm82)
Date: 08/30/2012

3.1 Introduction

This lecture will focus on Probabilistic Computation focusing on:

1. Basics Definitions
2. Some examples and identity testing
3. Error Reduction
4. Undirected Connectivity
5. Random walks and eigenvalue expansion

The idea is that we want to augment the model of computation to allow randomness and see compare the power of the new model of computation that results.

3.2 Probabilistic Turing Machines

There are two main ways to describe adding randomness to the computation of a Turing machine. Either you can say M has more than one transition function and non-deterministically chooses which one to use at a given state of computation or you can say that in addition to an input tape giving x, M has a tape r giving a sequence of random bits to assist in its computation of x. We will follow the latter convention and give the following definition of BPP (Bounded Polynomial Probabilistic). Furthermore we will note that one can look at the probability with which M accepts from a one-sided or a two sided version. The one-sided version is as follows:

Definition 3.1 (One-Sided)**.** A language L is in RP if there exists a polynomial-time TM M and a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that for every $x \in \{0,1\}^*$, $x \in L$ implies $\mathbb{P}_{r \in_R \{0,1\}^{p(|x|)}}[M(x,r) = 1] \geqslant 2/3$ while $x \notin L$ implies $\mathbb{P}[M(x) = 1] = 0$.

Then for a two-sided bound we remove the two restrictions on the probability of what happens on a given x and replace them with the one restriction: $\mathbb{P}[M(x,r) = L(x)] \geqslant 2/3$. A language is in BPP if it satisfies the two-sided bound.

Note that the probability with which $M(x, r)$ accepts $L(x)$ is somewhat arbitrary. In fact, all that is necessary is that $\mathbb{P}[M(x, r) = L(x)] \geqslant \frac{1}{2} + |x|^{-c}$. We will see this when we prove the Error Reduction Theorem for BPP.

BPP is a funny class of languages. There is no canonical BPP machine, no known complete problems and no hierarchy theorems. Further, you can find languages in BPP which cannot be computed by any probabilistic TM.

3.2.1 Polynomial Identity Testing

As an example of a polynomial-time probabilistic algorithm for a problem that has no known efficient deterministic algorithm. In particular, the problem takes as input an *algebraic circuit* which is a circuit whose nodes are $+, -, \times$ rather than $\wedge, \vee, \neg$. Then the inputs to the circuit are then n variables of the polynomial and the circuit computes the polynomial as its output.

Then we define the language $ZERO$ to be $ZERO = \{C | C \equiv 0\}$. Next we see this language has a simple, efficient randomized algorithm.

Theorem 3.2. *$ZERO \in CoRP$.*

Proof. The algorithm relies on the Schwartz-Zippel Lemma which is proved in the appendix of the text. The lemma regards a nonzero polynomial $p(x_1, \ldots, x_m)$ of total degree at most d and a finite set of integers, S. The lemma states that if m random integers, denoted $\{a_i\}$ are sampled from S with replacement, then $\mathbb{P}(p(a_1, a_2, \ldots, a_m) \neq 0) \geqslant 1 - \frac{d}{|S|}$.

Now, for a circuit of size m, the resulting polynomial has degree at most 2^m. Choose n random numbers from 1 to $10 \cdot 2^m$, evaluate the circuit and accept if $y = 0$ and reject otherwise. By the Schwartz-Zippel Lemma, if $C \notin ZERO$ we reject with probability $9/10 > 1/2$ so we are certainly in CoRP.

The only problem is that the degree of the polynomial is as high as 2^m and so y could be double exponentially large, requiring exponentially many bits to write down. To get around this problem, we use the technique called *fingerprinting*, where we compute the value $y \mod k$ for $k \in [2^{2m}]$. If y is 0 then so is $y \mod k$ and if $y = 0$ then $y \mod k = 0$. If $y \neq 0$ but $y \mod k = 0$ then with probability at least $\frac{1}{4m}$, k does not divide y. By repeating the reduction mod k for different k $O(1/\delta)$ times and accepting only if the outcome of each trial is 0, we reduce the probability of asserting $p(x) \equiv 0$ incorrectly to within acceptable bounds. □

3.3 Chernoff Bounds

Chernoff bounds are extremely useful in a number of settings for bounding how likely the sum of a family of random variables is to deviate from its expectation. The bound itself is stated as follows:

Theorem 3.3 (Chernoff bounds)**.** *Let $X_1, \ldots, X_n$ be mutually independent random variables taking values 0 or 1 and let $\mu = \sum_{i=1}^{n} \mathbb{E}(X_i)$. The for every $\delta > 0$,*

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i \geqslant (1+\delta)\mu\right) \leqslant \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}$$

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i \geqslant (1-\delta)\mu\right) \leqslant \left[\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right]^{\mu}$$

In fact, there is another form which will be more useful for proving the error reduction theorem. This form gives an exponential decay for deviating from the expected value of the sum of random variables from the statement of the first theorem, namely:

Theorem 3.4 (Chernoff bounds on difference from Expectation)**.** *Under the conditions of Theorem 3.3, for every $c > 0$ we have:*

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} X_i - \mu\right| \geqslant c\mu\right] \leqslant 2 \cdot e^{-\min\{c^2/4, c/2\}\mu}$$

So the probability on the left hand side is bounded by $2^{-\Omega(\mu)}$ where the constant required for the Ω notation depends on c.

This result gives a bound on how unlikely something is to deviate from its expected value. This result is key to proving the following theorem:

Theorem 3.5. *BPP $\subseteq$ PSIZE.*

3.4 Error Reduction

Now we deal with showing that the constants in the definition of BPP and RP were not important. In fact, the proof is elementary, using the Chernoff bounds just discussed.

Theorem 3.6. *If $\mathbb{P}[L(x) \neq M(n,r)] < \frac{1}{2} - |n|^d$ for all x, then for all $c \geqslant 1$, there exists M' such that $Pr[L(x) \neq M'(x,r)] < 2^{-n^c}$.*

Proof. Define a new machine M' which runs M on $m = \frac{n^2}{\delta^2}$ independent random strings for r and the same input x and outputs the majority answer. This is certainly only polynomial blow up in the runtime, making M' a probabilistic polynomial time TM.

To show that in fact, the probability with which this machine is correct, we apply the Chernoff bounds. Namely, we define a family of random variables $\{X_i\}_{i=1}^{m}$ to equal 1 if in the i-th run of M in the definition of M' we had $M(n, r_i) = L(x)$ and 0 otherwise. Then the family of random variables $\{X_i\}$ are independent Boolean random variables with $\mathbb{E}(X_i) = \mathbb{P}(X_i = 1) \geqslant 1/2 + \delta$. Now applying Theorem 3.4 we get the desired result, namely:

$$\mathbb{P}\left[\left|\sum_{i=1}^{m} X_i - \delta m\right| > k\delta m\right] < e^{-\frac{c^2}{4}\delta m}$$

Letting $\delta = 1/2 + |x|^{-d}$ and taking $k = |x|^{-d}/2$ the probability of being incorrect is bounded by $e^{-\Omega(|x|^d|)} \leqslant 2^{-|x|^d}$ which is the bound we alluded to at the beginning of lecture. We can tweak the value of k to get any desired $c \geqslant 1$ in the decay of the exponent. □

3.5 Undirected Graph Connectivity

We can also consider what happens when we restrict ourself to different space constraints.

Definition 3.7. BPL is the class of Probabilistic Polynomial Turing machines with log space memory on their tapes. Similarly for RL in relation to RP.

Theorem 3.8. *Undirected Graph Connectivity ∈ RL.*

Proof. We are given vertices s, t and we want to decide if they are connected in G. Consider a random walk of polynomial length in G from s. If G is connected, we claim that $\mathbb{P}$[walking s to t in exactly k steps] $\approx \frac{1}{n}$.

To prove the claim let $p^{(k)}$ be the distribution after k steps from s. Now we show $p_t^{(k)} \approx \frac{1}{n}$ for $n = poly(n)$. Let G be represented as a normalized adjacency matrix and then note that $p^{(k)} = G^k 1_s$.

To finish the analysis we will assume that G is d-regular.

Then we want $\langle 1_t, G^k 1_s\rangle \approx \frac{1}{n}$.

Let $1_t = (\alpha_1, \ldots, \alpha_n), 1_s = (\beta_1, \ldots, \beta_n)$ be $1_s, 1_t$ represented in the eigenbasis of G. Then $\langle 1_t, G^k 1_s\rangle = \sum \lambda_i^k \alpha_i \beta_i$.

Notice that G is a stochastic matrix and for G it can proved that $|\lambda_i| \leqslant 1$ and $ł_{max} = \lambda_1 = 1$. Using this fact and Cheegar's inequality, we get that if G is connected then $\lambda_2 < 1 - \frac{1}{n^2}$ and now we know $|\alpha_i|, |\beta_i| \leqslant 1$ and $\alpha_1 = \frac{1}{\sqrt{n}} = \beta_1$. Then we have:

$$\sum \lambda_i^k \alpha_i \beta_i \leqslant 1^k \frac{1}{n} + n(1 - \frac{1}{n^2})^k \approx \frac{1}{n}$$

This algorithm will always output correctly if s, t are connected by exhibiting an explicit path between them. Using polynomial time amplification, we can get the probability of finding such a path up to an acceptable constant, thus showing Graph Connectivity is in RL. □