Randomized Algorithms 2020-1
Lecture 3

Large Deviation Bounds (Chernoff) and BPP Amplification *

Moni Naor

We reviewed the complexity classes ZPP, RP, Co-RP and BPP. Recall that $L \in RP$ if there is an algorithm (Probabilistic Turing Machine M) s.t. for $x \in L$ we have that $Prob[M(x)$ outputs 'yes'$] \geq 1/2$ and for $x \notin L$ we have $Prob[M(x)$ outputs 'no'$] = 1$. We say that $L \in BPP$ if there is a Probabilistic Turing Machine M s.t. for $x \in L$ we have that $Prob[M(x)$ outputs 'yes'$] \geq 2/3$ and for $x \notin L$ we have $Prob[M(x)$ outputs 'no'$] \geq 2/3$ (all the probabilities are over the random tapes). We mentioned that $ZPP = RP \cap Co - RP$

We discussed the question of hitting set for RP. For any language $L \in RP$ a *hitting set* for input size n is a collection $C_n = \{R_1, R_2, \ldots R_m\}$ where for every $x \in L \cap \{0,1\}^n$ there is an $R_i \in C_n$ such when the input x is executed with random tape R_i the result is correct. That is if $M(\cdot, \cdot)$ is the Turing Machine establishing that L is in RP, then $M(x, R_i)$ accepts. The goal was to show that for any language $L \in RP$ there is a hitting set of size $m = n$. (This idea is due to Adleman in 1978 [1].

There are several ways to show this. One suggestion was to give a *probabilistic construction*, i.e. chose the collection at random and show that it is a hitting set with non-zero probability (this proves the existence of a proper collection). Instead, we made the following argument: For each $x \in L \cap \{0,1\}^n$ at least half the R's make $M(x, R)$ accept. This implies that there exists a specific R that for at least half of $\{L \cap \{0,1\}^n\}$ makes the TM M accept when used as the random tape. Call this R_1 and delete from further consideration all x's for which R_1 was accepting. Regarding the remaining x's in $L \cap \{0,1\}^n$, it is still the case that at least half the R's make them accept. So we can find an R_2 that is good for at least half of the remaining and so on for at most n rounds.

How robust is BPP wrt to the probabilities of acceptance? We discussed two possible alterations of the class BPP: Weak-BPP and Strong-BPP. In the latter, the probability of error can be be made, for any polynomial $q(n)$, as small as $2^{-q(n)}$. In the former, the advantage over guessing (being correct with probability 1/2) is $1/p(n)$ for *some* polynomial $p(n)$. The main point is:

Theorem 1. *Weak-BPP=Strong-BP.*

If M is a Turing Machine satisfying the Weak-BPP conditions with a polynomial $p(n)$, for any polynomial $q(n)$ we construct a Turing Machine M' for the Strong-BPP condition. The construction

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. In the interest of brevity, most references and credits were omitted.

is based on running M many times independently and taking the **majority** of the answers as the final answer. How many repetitions t of the original algorithm do we need?

The problem we are faced with is figuring out the probability that $\{0,1\}$ random variables

$$Z_1, Z_2, \ldots, Z_t,$$

each being '1' with probability $1/2 + 1/p$ and '0' otherwise, have a majority of '1's. Note that the probability of M' being correct dominates this probability.

To analyze the probability of this event we used large deviation bounds. We mentioned Markov's inequality and Chebyshev's inequality[1]. They are not sufficient for the task at hand, since we want exponential probability of failure. So we introduced and used one of the Chernoff-Hoefding-Azuma-Bernstein inequalities. specifically, we use:

Lemma 2. *Let $X_1, X_2, \ldots, X_t$ be mutually independent random variables such that $|X_i| \leq 1$ and $E[X_i] = 0$. Then*

$$\Pr[\sum_{i=1}^{t} X_i > a] \leq e^{-a^2/2t}.$$

To use this lemma, set X_i to be $Z_i - 1/2 - 1/p$. Now $E[X_i] = 0$. Since our goal is to get probability of error of the form 2^{-q}, We need $q = a^2/2t$ and we have $a = t/p$. So we Set $t = 2qp^2$ and obtain the desired amplification.

Watch: Chernoff, Hoeffding, etc. bounds, CMU, Lecture 5a,b,c of CS Theory Toolkit (Ryan O'Donnell): `https://www.youtube.com/watch?v=qqHHvOp5N6w`

Derandomizing BPP non-uniformly we used this to argue that BPP is in Non-Uniform P, that is that there exists a fixed advice string for each size n that make a Turing Machine recognizes in polynomial membership in $L \cap \{0,\}^n$. Equivalently, there are polynomial sized circuits for recognizing L. You can read about non-uniformity and machines that take advice in Oded Goldreich's notes [3].

Question: Define the class PP as those languages with a probabilistic Turing machine where for each input x we have that $\Pr[M(x,R)]$ is correct$] > 1/2$. Show that $NP \subset PP$.

Vague Question: Can you argue that taking the majority is the best way to amplify the probability of success of a BPP Algorithm? Or is there some other function, e.g. majority of majorities, that is better?

Recall checking matrix multiplication: given three $n \times n$ matrices A, B and C how do you check that $A \cdot B = C$, say over the finite field $GF[2]$ (or some other finite field $GF[q]$)? To recompute the product $A \cdot B$ is relatively expensive: the asymptotic time it takes is denoted as $O(n^\omega)$ where the current (as of 2014) best value for ω is ≈ 2.3728639). A method suggested in 1977 by Freivalds takes $O(n^2)$ for verification: pick at random a vector $r \in \{0,1\}^n$ and compute (i) $A(Br)$) and (ii) Cr and compare the two resulting vectors. The complexity of these operations is $O(n^2)$ since they are matrix times vector operations. If $AB = C$, then the algorithms always says 'yes'.
Question Prove that if $A \cdot B \neq C$, then the algorithm says 'no' with probability at least 1/2. If

[1] In how many ways can "Chebyshev" be legitimately spelled?

the finite field is $GF[q]$ and the random vector is chosen appropriately, what is the probability of inequality?

References

[1] Leonard Adleman, *Two theorems on random polynomial time*, FOCS 1978.

[2] Noga Alon and Joel Spencer, **The Probabilistic Method**, Appendix A.

[3] Oded Goldreich, Non-uniformity and PH, 2005
`http://www.wisdom.weizmann.ac.il/~oded/PSX/cc-text7.pdf`