BPP and Circuits.

CS254
Chris Pollett
Nov. 8, 2006.

Outline

- The class BPP
- Robustness
- Polynomial size circuits

BPP Motivation

- On Monday we introduced the classes RP, ZPP, and PP.
- Of these, RP and ZPP are realistic models.
- We could imagine using some kind of coin flips to do the nondeterministic choices along one path.
- By running the algorithm repeatedly we could get a good idea if a string was in the language or not.
- PP, on the other hand, has the virtue of having a nice syntactic definition, but it is not realistic.
- The reason is one could imagine situations where x being in languages probability $1/2 + 2^{-p(|x|)}$. It is hard to then distinguish this from the case that only 1/2 of path accept which would be rejecting.

Chernoff Bounds

- To analyze the notion of repeated runs more carefully, it is useful to make use of an inequality called Chernoff Bounds.

Lemma (Chernoff). Suppose $X_1,..,X_n$ are independent random variables taking the values 1 and 0 with probabilities p and 1-p. Let $X = \sum^n_{i=1} X_i$. Then for all $0 \leq c \leq 1$,

$$\text{prob}[X \geq (1+c)pn] \leq e^{-(c^2 pn)/2}.$$

Proof of Lemma

If t is a positive real number, then

$$\text{prob}[X \geq (1+c)pn] = \text{prob}[e^{tX} \geq e^{t(1+c)pn}] \qquad (*)$$

By Markov's Inequality,

$$\text{prob}[e^{tX} \cdot E(e^{tX})] \leq 1/k \text{ for any real } k>0.$$

Taking $k=e^{t(1+c)pn}/[E(e^{tX})]$ and using (*) gives

$$\text{prob}[X \geq (1+c)pn] \leq [E(e^{tX})] \cdot e^{-t(1+c)pn}. \qquad (**)$$

Since $X=\sum^{n}_{i=1} x_i$, we have $E(e^{tX})=[E(e^{tX_1})]^n$ which in turn equals $(1 + p(e^t-1))^n$. Substituting this into (**) gives:

$$\text{prob}[X \geq (1+c)pn] \leq (1 + p(e^t-1))^n \cdot e^{-t(1+c)pn}$$

$$\leq e^{-t(1+c)pn} \cdot e^{pn(e^t-1)}, \text{ since } (1+a)^n \leq e^{an}.$$

Take $t=\ln(1+c)$ to get $\text{prob}[X \geq (1+c)pn] \leq e^{pn(c-(1+c)\ln(1+c))}$.

Taylor expanding $\ln(1+c)$ as $c - c^2/2 + \ldots$ and substituting gives the result. i/e., $e^{pn(c-(1+c)\ln(1+c))} \leq e^{pn(c-(1+c)(c-c^2/2+c^3/3+..))} \leq e^{-(c^2 pn)/2}$

A Corollary

Cor. If $p=1/2 + \varepsilon$ for some $\varepsilon>0$, then the probability that $\sum^{n}_{i=1} X_i \leq n/2$ is at most $e^{-\varepsilon^2 n/4}$

Proof. Take $c = \varepsilon/(1/2+ \varepsilon)$. Q.E.D.

So if an experiment has a biased output we can hope to detect this after $1/\varepsilon^2$ experiments. For a probability like $2^{-p(n)}$ that we need in the case of PP, this is exponentially small and this is why it is not realistic.

BPP

Defn. The class BPP contains those languages L for which there is a p-time NTM N with the property that for all inputs x, if x is in L then at least 3/4 of N’s branches accept and if x is not in L, then 3/4’s of N’s branches reject.

Robustness

- Notice if we had chosen $1/2+\varepsilon$ in the definition for some $0 < \varepsilon < 1/4$, in our definition, then it would not have made a difference.
- Let $k = [4\ln 2/(\varepsilon^2)]$. Run the machine that accepts L according to the probabilities $1/2+\varepsilon$ a total of $2k+1$ times and accept the majority of the outcomes.
- So by Chernoff bounds, the odds that the majority vote of these runs is wrong is at most
  $e^{-\varepsilon\wedge 2(2k+1)/4} \leq e^{-\varepsilon\wedge 2(2k)/4} = e^{-8\ln 2/4} = 2^{-2} = 1/4.$
- Thus, we will accept with the 3/4's probability if its in the languages and reject with 3/4 probability if its not.

Relationships

- Notice by repeating an RP machine a couple of times we get a BPP machine for a language.
- Also any BPP machine for a language is also a PP machine for the same language.
- So RP $\subseteq$ BPP $\subseteq$ PP.
- BPP is a semantic class. This because for a L in BPP accepted by some N, we promise that one of the two possible outcomes for x has a clear majority of the N's branches.

Polynomial Size Circuits

- We have already defined what an Boolean circuit is.
- The *size* of a circuit is the number of gates in it.
- We next would like to define what it means for a family of circuits to recognize a language.

Defn. A family of circuits is an infinite sequence $(C_0, C_1, \ldots)$ of Boolean circuits, where C_n has n input variables. We say a language L has polynomial size circuit, if there is a polynomial p such that size(C_n)$\leq$ p(n) and C_n accepts exactly those strings in L.

P is in P/Poly

- We call the class of languages with polynomial circuits P/poly.

Thm. All languages in P have polynomial size circuits.

Proof. This essentially follows from our proof that CVP is P-complete where rather than encode a particular x into the inputs we instead let its value come from variables.