YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 461b: Foundations of Cryptography *Handout #5*
Yitong Yin *February 26, 2009*

Solutions to Problem Set 1

Originally due Thursday, February 19, 2009.

Problem 1 Chernoff bound – application 1

Consider a biased coin with probability $p = 1/3$ of landing heads and probability $2/3$ of landing tails. Suppose the coin is flipped some number n of times, and let X_i be a random variable denoting the i^{th} flip, where $X_i = 1$ means heads, and $X_i = 0$ means tails. Use the Chernoff bound to determine a value for n so that the probability that more than half of the coin flips come out heads is less that 0.001.

Solution

Let $\mathcal{E}$ represent the event that $\frac{1}{n}\sum_{i=1}^{n} X_i \geq \frac{1}{2}$, i.e. the event thet more than half of the coin flips come out heads.

$$\Pr\left[\mathcal{E}\right] = \Pr\left[\frac{1}{n}\sum_{i=1}^{n} X_i \geq \frac{1}{2}\right] \leq \Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{1}{3}\right| \geq \frac{1}{6}\right].$$

Note that $E[X_i] = 1/3$. According to the Chernoff bound, the last probability measure is no greater than $2\exp(-\frac{n}{16})$.

Solving the inequality that $2\exp(-\frac{n}{16}) < 0.001$, we get that $n \geq \lceil 16\ln(2000)\rceil = 122$.

Problem 2 Chernoff bound – application 2

At the Yale-Harvard hockey game on February 6, Yale made 50 shots on the Harvard goal and scored 5 times, whereas Harvard made 15 shots on the Yale goal and scored only once. Assume that both teams had equally good goalies and that for each team, the probability is $p = 0.1$ of a shot scoring a goal. Clearly, the expected number of goals is np, where n is the number of shots. For Yale, the actual number of goals g exactly matched the expectation, but for Harvard with 15 shots, the actual number of goals (1) fell considerably short of the expected number (1.5).

The purpose of this problem is to assess how unusual it is that Harvard scored fewer than the expected number of goals given the number of shots on the goal and the assumed success probability of each shot. In the following, let $n = 15$, $g = 1$, $p = 0.1$, and $q = \Pr[\text{\# goals after } n \text{ shots} \leq g]$. We wish to find a “good” upper bound on q.

(a) Happy Hacker remembered about the Chernoff bound presented in lecture 1, so he decided to use it to bound q. What values should he use for the parameters ε, n, and p that appear in the formula on the right hand side of the bound? Using these values, compute the value b of the right hand side. What does b tell Happy about q that he didn’t already know?

(b) Clever Clara knew right away that it was a waste of time to compute the Chernoff bound in this case and didn’t bother. How did Clara know that?

(c) Stolid Sean didn’t see the need to think hard about this problem and instead just plunged in and computed q to 4 decimal places using standard probability theory. How could he do this, and what is the answer?

Solution

Let $X_i \in \{0, 1\}$ be the random variable that indicates whether the i^{th} shot of Harvard scores.

(a) Let $\epsilon = \left|\frac{g}{n} - p\right|$. It is easy to verify that when $\frac{g}{n} < p$, the event that $\sum_{i=1}^{n} X_i \leq g$ implies that $\left|\frac{1}{n}\sum_{i=1}^{n} X_i - p\right| \geq \epsilon$.

When $n = 15, g = 1$ and $p = 0.1$, it holds that $\frac{g}{n} < p$ and $\epsilon = \frac{1}{30}$. Applying the Chernoff bound,

$$\begin{aligned} q &= \Pr\left[\sum_{i=1}^{n} X_i \leq g\right] \\ &\leq \Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - p\right| \geq \epsilon\right] \\ &= 2\exp\left(-\frac{\epsilon^2 n}{2p(1-p)}\right) \\ &\leq 1.823. \end{aligned}$$

This bound tells nothing about the distribution, since we already know that q is a probability measure which can not be greater than 1.

(b) The mean value is 0.1, which can be achieved precisely if scoring 1.5 (fractional) goals. The closest integral scores around 1.5 are 1 and 2. For both cases, the deviation is $1/30$, and for all possible number of goals, the deviation is at least $1/30$, thus the deviation bound can tell us nothing about X. Therefore, the Chernoff bound which relies solely on the deviation bound can provide little information.

(c) Note that $\sum_{i=1}^{n} X_i$ follows the binomial distribution. Directly compute the probability q.

$$\begin{aligned} q &= \Pr\left[\sum_{i=1}^{n} X_i \leq 1\right] \\ &= \Pr\left[\sum_{i=1}^{n} X_i = 0\right] + \Pr\left[\sum_{i=1}^{n} X_i = 1\right] \\ &= (1-p)^n + \binom{n}{1} p(1-p)^{n-1} \\ &\approx 0.5490. \end{aligned}$$

Problem 3 Derandomization

Let M be a ppTM that accepts a language L and runs in time $p(n)$ for some polynomial $p(\cdot)$. Let x be an input string of length n and r a random choice string of length $p(n) \gg n$. Let $\delta(x, r) = 1$ if $M(x, r)$, the output of M with coin toss sequence r, gives the correct answer about x's membership in L, and let $\delta(x, r) = 0$ otherwise. Suppose $\Pr[M(U_n, U_{p(n)})$ is correct$] = 1 - 2/2^n$.

How large can we make the success probability of $M(U_n, r)$ by setting the second input of M to a fixed string r? That is, what is the best lower bound on

$$\max_r \Pr[M(U_n, r)]$$

that is implied by the given information, where $\max_r$ is taken over all binary strings of length $p(n)$?

[Note: This problem generalizes a fact used in the proof of Theorem 4, section 11, lecture notes 3.]

Solution

The naïve lower bound is $1 - 2/2^n$, since the maximum can not be smaller than the average. We will show that this is the best general lower bound by constructing a case where $\max_r \Pr[M(U_n, r)$ is correct$] = 1 - 2/2^n$.

Denote $\delta(x, r)$ as a $2^n \times 2^{p(n)}$ 0-1 matrix. Assign to each column of $\delta(x, r)$ exactly 2 zeroes. For every r, it holds that $\Pr[\delta(U_n, r)] = (2^n - 2)/2^n = 1 - 2/2^n$.

Therefore it satisfies the above constraint that that $\Pr[\delta(U_n, U_{p(n)})] = 1 - 2/2^n$. The maximum is $\max_r \Pr[\delta(U_n, r)] = 1 - 2/2^n$.

Problem 4 One-way functions and the $\mathcal{P}$-versus-$\mathcal{NP}$ question

[Textbook, Chapter 2, Exercise 3.]

Solution

The function $f(\pi, \phi, \tau)$, where π represents the first $n/2$ bits and (ϕ, τ) is the input of f_{sat}, is defined as

$$f(\pi, \phi, \tau) = \begin{cases} (0, f_{\text{sat}}(\phi, \tau)) & \text{if } \pi = 0 \cdots 0 \\ (1, \pi, \phi, \tau) & \text{o.w.} \end{cases} .$$

It is obvious that f is polynomial-time-computable because both branches in the definition are polynomial-time-computable and the condition can be verified in linear time. Claim 1 holds.

Suppose that A is such a polynomial-time algorithm that always inverts f. Let B be such an algorithm whose input domain is the range of f_{sat} and we define that $B(y) = A(0, y)$. It is trivial that B is also polynomial-time. For every y in the range of f_{sat}, according to the definition of f, $B(y)$ will output some (π, ϕ, τ) such that $(0, f_{\text{sat}}(\phi, \tau)) = (0, y)$, i.e. $f_{\text{sat}}(\phi, \tau) = y$, thus B is a polynomial-time algorithm which always inverts f_{sat}, which is contra to the assumption that $NP \neq P$. Claim 2 holds.

Let C be such a program that $C(y)$ outputs all but the first bit of y if the first bit of y is 1, and outputs an arbitrary string if otherwise. It is easy to see that C is a polynomial-time algorithm that inverts f when the first bit of y is 1. The probability that C fails to invert f can be written as

$$\begin{aligned} \Pr[C \text{ fails }] &= \Pr[C(f(U_n)) \notin f^{-1}(f(U_n))] \\ &\leq \Pr[\text{the first bit of } f(U_n) \text{ is } 0] \\ &= \Pr[\text{the first } n/2 \text{ bits of } U_n \text{ are all 0s}] \\ &= 2^{-\frac{n}{2}}. \end{aligned}$$

Claim 3 holds.