15-359: Probability and Computing

Assignment 5 Due: February 24, 2012

Problem 1: Not too concentrated (5 pts.) Suppose X is a continuous random variable whose density is upper bounded by B. For all $x \in \mathbb{R}$ and $\epsilon > 0$, upper bound $P(|X - x| \le \epsilon)$.

Problem 2: Mediocristan and Extremistan (10 pts.) Our system processes two different types of independent requests. The size of a request is distributed according to its type:

$$\begin{aligned} \bar{F}_M(m) =& e^{-0.1m} \quad m \ge 0 \\ \bar{F}_X(x) =& x^{-10/9} \quad x \ge 1 \end{aligned}$$

A. What are the means of M and X? What are the names of their distributions?

B. Suppose exactly 1% of the requests are larger than m_0 and x_0, respectively. Calculate m_0 and x_0 with the help of a computer.

C. Let's consider the fraction of size (over time) made up by the largest 1% jobs:

$$\begin{aligned} \rho_M &= \int_{m_0}^{\infty} m f_M(m) dm / E(M) \\ \rho_X &= \int_{x_0}^{\infty} x f_X(x) dx / E(X) \end{aligned}$$

How do ρ_M and ρ_X compare? Why are these so different?

Problem 3: Business is booming (10 pts.) Despite his foibles, the hazardous cook from homework 4 is faring well. (He makes pancakes with probability p and pancake-destroying explosions with probability $q = 1 - p$.) Let R be the number of attempts before a stack of 10 pancakes is created. On homework 4, you determined $E(R)$.

A. Determine $\widehat{R}(z)$ using roughly the same recursive argument as in homework 4.

B. Calculate $\mathrm{Var}(R)$ when $p = 0.8$, ideally with the help of a computer.

Problem 4: Work ethic (10 pts.) The fire department decides to monitor the cook. They hire an intern to record the number of explosions made by the cook. That is, they want (rather demandingly)

$$\sum_{i=1}^{\infty} X_i$$

where $X_i = 1$ if attempt i was an explosion, and 0 otherwise. Unfortunately, the intern is also quite unreliable: after watching an attempt, he leaves with probability ℓ, so he ends up recording

$$S = \sum_{i=1}^{N} X_i$$

for some random N.

A. Prove that $\widehat{S}(z) = \widehat{N}\left(\widehat{X}(z)\right)$ where $X \sim \text{Bernoulli}(q)$. (Hint: think back to the definition of the z-transform and use conditioning.)

B. Write down $\widehat{N}(z)$ and $\widehat{X}(z)$.

C. Determine $\text{Var}(S)$ and $E(S^3)$, which quantify the unreliability of the intern. Calculate their values for $q = 0.2, \ell = 0.1$.

Problem 5: Vanilla search trees (20 pts.)

There are many clever techniques to balance search trees. In an effort to avoid implementation work, we can apply the "randomized incremental" paradigm to construct a reasonable search tree T from given, static data set $A = \{a_1, \ldots, a_n\}$, chosen from some ordered set.

A. Describe a (the?) randomized incremental algorithm to construct a binary search tree.

B. Give a bound for the expected distance of a node in the tree from the root.

Problem 6: Have randomness, will predict (20 pts.) I privately sample two random, distinct, nonnegative integers X and Y. You observe one of them; call that one B and the other one Z. Now you must guess if $B > Z$. Employ a randomized strategy which succeeds with probability (with respect to both my random sampling and yours) $1/2 + \alpha$ for some $\alpha > 0$. (Hint: one random sample suffices. Think about intervals.)

Problem 7: Shooting blanks (25 pts.) Suppose B is an n by n Boolean matrix; determine whether B has a column filled entirely with 0's. We want a BCC algorithm (blank column checker) which solves this problem.

For simplicity, let us describe running time only in terms of the number of lookups: the algorithm may query the bit at $B[i, j]$, $1 \leq i, j \leq n$ which we count as one step. Of course, there is a trivial deterministic BCC that takes n^2 steps in the worst case. Moreover, given a particular choice of deterministic algorithm it is easy for an adversary to design a matrix B that forces the algorithm to spend n^2 steps.

There is an obvious lazy Las Vegas algorithm R:

- First permute the columns of B at random, say, $C_1, \ldots, C_n$.
- Then check the columns in this order, bailing out whenever a 1 is encountered.

As in class, let $\mathcal{A}$ be the class of all deterministic BCC algorithms and let $\mathcal{I}$ be the class of all inputs of size n by n; think of n as being fixed for simplicity. Recall Yao's Minimax Principle:

$$\min_{A \in \mathcal{A}} E(T_A(I_\tau)) \leq \max_{I \in \mathcal{I}} E(T_{A_\sigma}(I))$$

where the subscripts τ and σ indicate the probability distributions on $\mathcal{I}$ and $\mathcal{A}$ used to determine the expected values.

A. Show that the expected number of queries of R is bounded by $n(n+1)/2$.

B. Give a reasonable description of $\mathcal{A}$.

C. Show that for any Las Vegas algorithm A, the expected number of queries in the worst case is no better than $n(n+1)/2$. Hint: for reals α, β it suffices to show that $(1-\epsilon)\alpha \leq \beta$ for all $\epsilon > 0$ in order to show that $\alpha \leq \beta$.