Randomness and Computation **Edinburgh INFR11089, Spring 2013**

Problem Set 1

Due: Tuesday, March 5, 2p.m. at the ITO

---

1. (Due to J. von Neumann) Suppose you are given a coin for which the probability of HEADS, say p, is *unknown*. How can you use this coin to generate unbiased (i.e., $\mathbf{Pr}[\text{HEADS}] = \mathbf{Pr}[\text{TAILS}] = 1/2$) coin-flips? Give a scheme for which the expected number of flips of the biased coin for extracting one unbiased coin-flip is no more than $1/[p(1-p)]$.

2. (a) (Exercise 1.6 from the textbook.) Consider the following balls-and-bin game. We start with one black ball and one white ball in a bin. We repeatedly do the following: choose one ball from the bin uniformly at random, and then put the ball back in the bin with another ball of the same color. We repeat until there are n balls in the bin. Show that the number of white balls is equally likely to be any number between 1 and $n-1$.

   (b) (Exercise 3.19 from the textbook.) Let Y be a nonnegative integer-valued random variable with positive expectation. Prove that

$$\frac{(\mathbf{E}[Y])^2}{\mathbf{E}[Y^2]} \leq \mathbf{Pr}[Y \neq 0] \leq \mathbf{E}[Y].$$

3. (Exercises 2.20 & 3.21 from the textbook.) A permutation on the numbers $[n] = \{1, \ldots, n\}$ can be represented as a function $\pi : [n] \to [n]$, where $\pi(i)$ is the position of i in the ordering given by the permutation. A *fixed point* of a permutation $\pi : [n] \to [n]$ is a value for which $\pi(x) = x$. Consider the following random experiment: We pick a permutation uniformly at random from the set of all permutations from $[n]$ to $[n]$. Let F be the random variable representing the number of fixed points of a permutation chosen uniformly at random.

   (a) Find the expectation of F.

   (b) Find the variance of F.

4. (a) (Exercise 4.14 from the textbook.) Modify the proof of Theorem 4.4 to show the following bound for a weighted sum of Poisson trials. Let $X_1, \ldots, X_n$ be independent Poisson trials such that $\mathbf{Pr}[X_i] = p_i$ and let $a_1, \ldots, a_n$ be real numbers in $[0, 1]$. Let $X = \sum_{i=1}^{n} a_i X_i$ and $\mu = \mathbf{E}[X]$. Then the following Chernoff bound holds: for any $\delta > 0$,

$$\mathbf{Pr}\left(X \geq (1+\delta)\mu\right) \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}.$$

   (b) The *lattice approximation problem* is an extension of the set-balancing problem (Section 4.4 of the textbook). As before, we are given an $n \times n$ matrix A all of whose entries are 0 or 1. In addition, we are given a column vector p with n entries, all of which are in the interval $[0, 1]$. We wish to find a column vector q with n entries, all of which are from the set $\{0, 1\}$, so as to minimize $\|A(p-q)\|_{\infty}$.

   Consider the following randomized algorithm to obtain the desired vector q: For each i independently, set $q_i = 1$ with probability p_i and $q_i = 0$ with probability $1 - p_i$. Derive a bound on $\|A(p-q)\|_{\infty}$.

5. In this problem, we will analyze a simple algorithm to learn an unknown probability distribution from samples.

   A *discrete probability distribution* over the set $[n] = \{1, \ldots, n\}$ can be viewed as a function $p : [n] \to [0, 1]$. The number $p(i)$ represents "the probability the distribution p assigns to point i." Hence, we have that $p(i) \geq 0$ for all $i \in [n]$, and $\sum_{i=1}^{n} p(i) = 1$. For two distributions p, q over $[n]$ the *total variation distance* between p and q is the quantity $\mathrm{d_{TV}}(p, q) := \sum_{i=1}^{n} |p(i) - q(i)|$. ($\mathrm{d_{TV}}(p, q)$ represents a measure of the "closeness" between p and q.)

   In many scenarios we are interested in *learning* an *unknown* probability distribution from *samples*. In more detail, a *learning algorithm* is given access to a *sampling oracle* for p, i.e., a "black-box" with the following property: Every invocation of the oracle (query) yields an output $s \in [n]$ that is a random variable distributed according to p (i.e., $\mathbf{Pr}[s = j] = p(j)$ for all $j \in [n]$) and is independent of all previous outputs. For a given error parameter $0 < \epsilon < 1$, the goal of the learning algorithm is to output a *hypothesis distribution* h over $[n]$ such that with probability at least $2/3$ (over the samples obtained from the oracle) the following condition is satisfied: $\mathrm{d_{TV}}(p, h) \leq \epsilon$.

   Given m independent samples $s_1, \ldots, s_m$, drawn from distribution $p : [n] \to [0, 1]$, the *empirical distribution* $\widehat{p}_m : [n] \to [0, 1]$ is defined as follows: for all $i \in [n]$,

$$\widehat{p}_m(i) = \frac{|\{j \in [m] \mid s_j = i\}|}{m}.$$

   Consider the following algorithm:

   "Draw m samples from the oracle for p and output the distribution $h = \widehat{p}_m$."

   (a) For $i \in [n]$, let $N_i = |\{j \in [m] \mid s_j = i\}|$ denote the number of samples that "land" on point i. What is the distribution of N_i? Show that $\mathbf{Var}[N_i] = mp(i)\left(1 - p(i)\right)$.

   (b) Show that $\mathbf{E}\left[|p(i) - \widehat{p}_m(i)|\right] \leq \sqrt{\frac{p(i)(1-p(i))}{m}}$.
   (Hint: Use (a) along with Jensen's inequality.)

   (c) Show that $\mathbf{E}\left[\mathrm{d_{TV}}(p, \widehat{p}_m)\right] \leq \sqrt{\frac{n}{m}}$.
   (Hint: Use (b) along with the concavity of the function $f(x) = \sqrt{x(1-x)}$.)

   (d) Argue that if $m = \Omega(n/\epsilon^2)$ the above described algorithm satisfies $\mathrm{d_{TV}}(p, h) \leq \epsilon$ with probability at least $2/3$.
   (Hint: you can use (c), even if you did not prove it.)

(Remark: It can be shown that the above algorithm is asymptotically optimal, in the sense that *any* algorithm for this learning problem *information-theoretically* requires $m = \Omega(n/\epsilon^2)$ samples.)