Probability and Computing Hilary 2017

Exercise Sheet 3

Elias Koutsoupias (with thanks to Stefan Kiefer and Stanislav Živný)

1. Suppose you are given a biased coin that has Pr(Heads) $= p$ for some unknown p. Given $\varepsilon > 0$ describe a random experiment that with probability at least 0.99 tells you the value of p to within ε. Justify your answer.

2. Suppose that we can obtain t independent samples $X_1, X_2, \ldots, X_t$ of a random variable X and that we want to use $\sum_{i=1}^{t} X_i/t$ as an estimate of $\mathbf{E}[X]$. We want the estimate to be within $\varepsilon\mathbf{E}[X]$ of the true value with probability at least $1-\delta$. We may not be able to use Chernoff's bound directly if X is not a 0-1 random variable. We develop an alternative approach that requires only a bound on $\mathbf{Var}[X]$. Let $r = \frac{\sqrt{\mathbf{Var}[X]}}{\mathbf{E}[X]}$.

   (a) Show using Chebyshev's inequality that $O\big(\frac{r^2}{\varepsilon^2\delta}\big)$ samples are sufficient to solve the above problem.

   (b) Suppose that we only need a weak estimate that is within $\varepsilon\mathbf{E}[X]$ with probability at least 3/4. Briefly argue that $O\big(\frac{r^2}{\varepsilon^2}\big)$ samples suffice.

   (c) Show that by taking the median of $O(\log(1/\delta))$ weak estimates, we can obtain an estimate within $\varepsilon\mathbf{E}[X]$ of $\mathbf{E}[X]$ with probability at least $1-\delta$. Conclude that we only need $O\big(\frac{r^2\log(1/\delta)}{\varepsilon^2}\big)$ samples.
   [**Hint:** Apply a Chernoff bound to a suitable family of 0-1 random variables.]

3. We prove that the Randomized Quicksort algorithm sorts a set U of n numbers in time $O(n \log n)$ with high probability. Consider the following view of Randomized Quicksort. Every point in the algorithm where it decides on a pivot element is called a *node*. Suppose the size of the set to be sorted at a particular node is s. The node is called *good* if the pivot divides the set into two parts, each of size no more than $2s/3$. Otherwise the node is called *bad*. The nodes can be thought of as forming a tree, following the recursive nature of the algorithm.

   (a) Give an upper bound on the number of good pivot elements along any path in the recursion tree from the root to a leaf.

   (b) Fix an element $k \in U$. Give an upper bound on the length of the path in the recursion tree from the root to the node labelled k that holds with high probability.

   (c) Conclude that the running time of Quicksort is $O(n \log n)$ with high probability.