Homework 4 Solutions

Note: These solutions are not necessarily model answers. Rather, they are designed to be tutorial in nature, and sometimes contain a little more explanation than an ideal solution. Also, bear in mind that there may be more than one correct solution. The maximum total number of points available is 33.

1. (a) By linearity of expectation, $\mathrm{E}[X] = \sum_{i=1}^{n} \mathrm{E}[X_i] = \sum_{i=1}^{n} \frac{1}{i} = H_n$. Since the X_i are independent, $\mathrm{Var}[X] = \sum_{i=1}^{n} \mathrm{Var}[X_i] = \sum_{i=1}^{n} \frac{1}{i}(1 - \frac{1}{i})$. *3pts*

(b) We are interested in upper bounds for $p = \Pr[X \geq 4 \ln n]$. Note that $\mathrm{E}[X] = H_n = \ln n + \Theta(1)$ and $\mathrm{Var}[X] \leq H_n$. Since we are asked only for asymptotic bounds, we will omit lower order terms from our final answers. For functions $f(n)$, $g(n)$, the notation $f(n) \sim g(n)$ means that $f(n)/g(n) \to 1$ as $n \to \infty$.

— Applying Markov's inequality, we have *3pts*

$$p \leq \frac{\mathrm{E}[X]}{4 \ln n} = \frac{H_n}{4 \ln n} \sim \frac{\ln n}{4 \ln n} = \frac{1}{4}.$$

— Applying Chebyshev's inequality, we have *3pts*

$$p \leq \Pr[|X - \mathrm{E}[X]| \geq 4 \ln n - H_n] \leq \frac{\mathrm{Var}[X]}{(4 \ln n - H_n)^2} \leq \frac{H_n}{(4 \ln n - H_n)^2} \sim \frac{\ln n}{(3 \ln n)^2} = \frac{1}{9 \ln n}.$$

— Applying the Chernoff bound, we note that *4pts*

$$p = \Pr[X \geq 4 \ln n] = \Pr[X \geq (1+\delta)\mu] \leq \exp(-\frac{\delta^2}{2+\delta}\mu),$$

where $\mu = \mathrm{E}[X] = H_n$ and $\delta = \frac{4 \ln n - H_n}{H_n}$. We have to be a little more careful with the asymptotics here because the expressions are in the exponent. Note that $\delta\mu = 4 \ln n - H_n \sim 3 \ln n$, and $\frac{\delta}{2+\delta} = \frac{4 \ln n - H_n}{4 \ln n + H_n} \sim \frac{3}{5}$. Plugging these asymptotic expressions into the above bound gives

$$p \leq \exp(-\frac{\delta^2}{2+\delta}\mu) \sim \exp(-\frac{9}{5} \ln n) = n^{-9/5}.$$

Observe how the first bound is constant, the second is inverse logarithmic, and the third is inverse polynomial (which is *exponentially* better than the second bound).

2. (a) Let $X_i, i = 1, 2, \ldots, 10^6$ denote the casino's net loss in the i'th game. We have *4pts*

$$X_i = \begin{cases} 2 & \text{w.p. } \frac{4}{25} \\ 99 & \text{w.p. } \frac{1}{200} \\ -1 & \text{w.p. } \frac{167}{200} \end{cases} \qquad \Rightarrow \qquad e^{tX_i} = \begin{cases} e^{2t} & \text{w.p. } \frac{4}{25} \\ e^{99t} & \text{w.p. } \frac{1}{200} \\ e^{-t} & \text{w.p. } \frac{167}{200} \end{cases}$$

Therefore,

$$\mathrm{E}[e^{tX_i}] = \frac{4}{25}e^{2t} + \frac{1}{200}e^{99t} + \frac{167}{200}e^{-t}.$$

Now, $X = X_1 + X_2 + \cdots + X_{10^6}$ is the casino's net loss in the first million games, and we may compute $\mathrm{E}[e^{tX}]$ as follows:

$$\begin{aligned} \mathrm{E}[e^{tX}] &= \mathrm{E}[e^{t(X_1+X_2+\cdots X_{10^6})}] \\ &= \mathrm{E}[e^{tX_1}] \cdot \mathrm{E}[e^{tX_2}] \cdot \cdots \cdot \mathrm{E}[e^{tX_{10^6}}] \quad \text{since the } X_i \text{ are independent} \\ &= \left(\frac{4}{25}e^{2t} + \frac{1}{200}e^{99t} + \frac{167}{200}e^{-t}\right)^{10^6}. \end{aligned}$$

(b) We are interested in the quantity *4pts*

$$\begin{aligned}\Pr[X \geq 10^4] &= \Pr[e^{tX} \geq e^{10^4 t}] \\ &\leq \frac{\mathrm{E}[e^{tX}]}{e^{10^4 t}} \quad \text{by Markov's inequality} \\ &= \left(\frac{4}{25}e^{2t} + \frac{1}{200}e^{99t} + \frac{167}{200}e^{-t}\right)^{10^6} e^{-10^4 t}.\end{aligned}$$

This bound is valid for any $t > 0$, so we are free to choose a value of t that gives the best bound (i.e., the smallest value for the expression on the right). Plugging in $t = 0.0006$ as suggested in the hint, we get the bound 0.0002. This is very small, suggesting that the casino has a problem with its machines.

Aside: It is interesting to compare the above with a direct application of Markov's inequality. To do this, we need to redefine X to be the amount of money the casino pays out, so that X is now a non-negative r.v. An easy calculation gives $\mathrm{E}[X] = 0.98 \cdot 10^6$, and applying Markov's inequality we get the upper bound $\Pr[X \geq 10^6 + 10^4] \leq \frac{98}{101} \approx 0.97$, which is disastrously weaker than the bound we obtained from Chernoff.

3. (a) First, we process ϕ so that every variable appears at most once in each clause (eliminate repeated occurences of a literal, and delete a clause if both a literal and its negation occur). Let n denote the number of variables, and c_i the number of variables in clause i. *3pts*

- size(S_i): return 2^{n-c_i}. The variables in clause i must be fixed to values that satisfy the clause, and the remaining variables may be assigned any value. This takes time $O(n)$ (to count the variables in clause i).
- select(S_i): fix the variables in clause i to values that satisfy the clause; choose the values of the remaining variables independently and u.a.r. Again, this takes time $O(n)$.
- lowest(x): for $i = 1, 2, \ldots$, test if x satisfies clause i (this test is easy); return the index of the first clause that x satisfies (or "undefined" if it satisfies no clauses). This takes time at most $O(mn)$ (i.e., $O(n)$ time per clause).

(b) The problem is that S may occupy only a tiny fraction of all possible assignments U. Thus the number of samples t would need to be huge in order to get a good estimate of q. We give a (rather extreme) concrete example to make this precise. Consider the very simple formula $\phi = x_1 \wedge x_2 \wedge \cdots \wedge x_n$. Clearly $|S| = 1$ (the only satisfying assignment is when all n variables are TRUE). The given algorithm will output zero unless it happens to choose this assignment in one of its t samples, i.e., it outputs zero with probability $(1 - 2^{-n})^t \geq 1 - t2^{-n} \sim 1$ for any t that is only polynomial in n. Thus the relative error of the algorithm will be arbitrarily large with probability arbitrarily close to 1. For a less extreme example, one could consider the formula consisting of a single clause $\phi = x_1 \wedge x_2 \wedge \cdots \wedge x_{n/2}$; this has $2^{n/2}$ satisfying assignments, which is still an exponentially small fraction of the universe U so the algorithm will output zero with probability $(1 - 2^{-n/2})^t \geq 1 - t2^{-n/2}$, which is again ~ 1 for any t that is polynomial in n. *3pts*

Note: It is not enough here to quote the bound from class $t = O(q/\epsilon^2 \ln(1/\delta))$, which tells us how large a sample size is sufficient to estimate the proportion q. The reason is that this is an upper *bound on t, whereas here we need a* lower *bound. The lower bound can be derived by the very simple argument given above.*

(c) Note that the first two lines of the algorithm select each pair $(x, S_i), x \in S_i$ with probability $\frac{|S_i|}{\sum_j |S_j|} \cdot \frac{1}{|S_i|} = \frac{1}{\sum_j |S_j|}$. In other words, the first two lines pick an element u.a.r. from the *disjoint union* of the sets S_i. (Note that the goal is really to pick an element u.a.r. from the *union* $\cup_i S_i$.) Let $\Gamma = \{(x, S_i) \mid \text{lowest}(x) = i\}$. Clearly, the algorithm outputs 1 with probability $\sum_{(x,S_i)\in\Gamma} \frac{1}{\sum_j |S_j|} = \frac{|\Gamma|}{\sum_j |S_j|}$. To see that $|\Gamma| = |S|$, simply observe that every element $x \in S$ corresponds to exactly one lowest S_i, or *3pts* equivalently $\Gamma = \{(x, S_{\text{lowest}(x)}) \mid x \in S\}$. It follows that the algorithm outputs 1 with probability $p = \frac{|S|}{\sum_j |S_j|}$.

(d) Clearly, for $i = 1, 2, \ldots, m$ we have $|S_i| \leq |S|$. Hence, $\sum_j |S_j| \leq m|S|$, and thus $p = \frac{|S|}{\sum_j |S_j|} \geq \frac{1}{m}$. *2pts*

(e) Note that $X_1, \ldots, X_t$ are independent 0-1 r.v.'s with mean p, so $\mathrm{E}[X] = pt$ and the Chernoff bound yields *3pts*

$$\Pr[|X - pt| \geq \epsilon pt] \leq 2e^{-\epsilon^2 pt/3}.$$

The quantity on the right is bounded above by δ provided we take $t = \lceil \frac{3}{\epsilon^2 p} \ln \frac{2}{\delta} \rceil$. Since $p \geq \frac{1}{m}$ from part (d), it suffices to take $t = \lceil \frac{3m}{\epsilon^2} \ln \frac{2}{\delta} \rceil = O(\frac{m}{\epsilon^2} \log \frac{1}{\delta})$.

(f) Each iteration of the algorithm in (c) requires $O(1)$ operations, so the final algorithm takes $O(t) = O(\frac{m}{\epsilon^2} \log \frac{1}{\delta})$ time. By definition, we have $|S| = \frac{\sum_j |S_j|}{t} \cdot tp$ and $Y = \frac{\sum_j |S_j|}{t} \cdot X$. This implies *2pts*

$$Y \in [(1-\epsilon)|S|, (1+\epsilon)|S|] \iff X \in [(1-\epsilon)tp, (1+\epsilon)tp]$$

and thus

$$\Pr\big[Y \in [(1-\epsilon)|S|, (1+\epsilon)|S|]\big] = \Pr\big[X \in [(1-\epsilon)tp, (1+\epsilon)tp]\big]$$

It follows by part (e) that $\Pr\big[Y \in [(1-\epsilon)|S|, (1+\epsilon)|S|]\big] \geq 1 - \delta$, as required.