ORIE 4520: Stochastics at Scale **Homework 3: Due Sep 28th, 12pm**

Fall 2015 Sid Banerjee (sbanerjee@cornell.edu)

Problem 1: (Practice with Chebyshev and Chernoff bounds)

When using concentration bounds to analyze randomized algorithms, one often has to approach the problem in different ways depending on the specific bound being used. Typically, Chebyshev is useful when dealing with more complicated random variables, and in particular, when they are pairwise independent; Chernoff bounds are usually used along with the union bound for events which are easier to analyze. We'll now go back and look at a few of our older examples using both these techniques.

Part (a)

(Number of collisions) Recall we showed that if we throw m balls in n bins, the average number of collisions $X_{m,n}$ to be $\mu_{m,n} = \binom{m}{2}\frac{1}{n}$. Use Chebyshev's inequality to show that:

$$\mathbf{P}[|X_{m,n} - \mu_{m,n}| \geq c\sqrt{\mu_{m,n}}] \leq 1/c^2.$$

Next suppose we choose $m = 2\sqrt{n}$, then $\mu_{m,n} \leq 1$. Use Chernoff bounds plus the union bound to bound the probability that no bin has more than 1 ball. Compare this to the more exact analysis you did in homework 1.

Part (b)

(Coupon collector) For n bins, recall that we defined T_i to be the first time when i unique bins were filled, and used these random variables to show that T_n, i.e., the number of balls we need to throw before every bin has at least one ball, satisfies $\mathbf{E}[T_n] = nH_n = \Theta(n \log n)$. Using the same random variables, show that $\mathbf{P}[|T_n - \mathbf{E}[T_n]| \geq \epsilon \mathbf{E}[T_n]] \leq \frac{\pi^2}{6c^2 H_n^2}$.

Next, suppose we throw in $m = n \log n + cn$ balls – using Chernoff bounds plus the union bound, choose c such that no bin is empty with probability greater than $1 - \delta$.

Hint: Use $\sum_{i=1}^{\infty} 1/i^2 = \pi^2/6$

Problem 2: (The Hoeffding Extension)

In class, we saw that for a r.v. $X_i \sim$Bernoulli(p_i), we have:

$$\mathbf{E}[e^{\theta X}] = 1 - p + pe^{\theta}$$

Plugging this into the Chernoff bound and optimizing over θ, we obtained a variety of bounds – in particular, for independent r.vs $\{X_i\}$ with $X = \sum_i X_i$ and $\mu = \mathbf{E}[X] = \sum_i p_i$, we showed for any $\epsilon > 0$:

$$\mathbf{P}[|X - \mu| \geq \epsilon\mu] \leq 2\exp\left(\frac{-\epsilon^2\mu}{2+\epsilon}\right)$$

We now extend this to more general bounded r.vs.

Part (a)

First, for any θ, argue that the function $f(x) = e^{\theta x}$ for $x \in [0,1]$ is bounded above by the line joining $(0,1)$ and $(1, e^\theta)$. Using this, find constants α, β such that $\forall\, x \in [0,1]$:

$$e^{\theta X} \leq \alpha x + \beta$$

Part (b)

Next, for any random variable X_i taking values in $[0,1]$ such that $\mathbf{E}[X_i] = \mu_i$, show that:

$$\mathbf{E}[e^{\theta X_i}] \leq 1 - \mu_i + \mu_i e^\theta.$$

Using this, for independent r.vs X_i taking values in $[0,1]$ with $\mathbf{E}[X_i] = \mu_i$, and defining $X = \sum_i X_i, \mu = \sum_i \mu_i$, show that:

$$\mathbf{P}[|X - \mu| \geq \epsilon\mu] \leq 2\exp\left(\frac{-\epsilon^2\mu}{2+\epsilon}\right)$$

(Note: You can directly use the inequality for Bernoulli r.v.s – no need to show the optimization.)

Part (c) (Optional)

Next, for any random variable X_i taking values in $[a_i, b_i]$ such that $\mathbf{E}[X_i] = \mu_i$, a similar bounding technique as above can be used to show:

$$\mathbf{E}\left[e^{\theta(X_i - \mu_i)}\right] \leq \exp\left(-\frac{1}{8}\theta^2(b-a)^2\right)$$

(This is sometimes referred to as Hoeffding's lemma – for the proof, see the wikipedia article)

Now consider independent r.vs X_i taking values in $[a_i, b_i]$ with $\mathbf{E}[X_i] = \mu_i$, and as before, let $X = \sum_i X_i, \mu = \sum_i \mu_i$. Using the above inequality, optimize over θ to show that:

$$\mathbf{P}[(X - \mu) \geq \epsilon\mu] \leq \exp\left(\frac{-2\epsilon^2\mu^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

Problem 3: (A Weaker Sampling Theorem: Adapted from MU Ex 4.9)

In class we saw the following 'sampling theorem' for estimating the mean of a $\{0,1\}$-valued random variable: In order to get an estimate within $\pm\epsilon$ with confidence $1-\delta$, we need $n \geq \frac{2+\epsilon}{\epsilon}\ln\frac{2}{\delta}$. A crucial component in this proof was using the Chernoff bound for Bernoulli(p) random variables. Suppose instead we want to estimate a more general random variable X (for example, the average number of hours of TV watched by a random person) – we may not be able to use a Chernoff bound if we do not know the moment generating function We now show how to to get a similar sampling theorem which only uses knowledge of the mean and variance of aX.

We want to estimate a r.v. X with mean $\mathbf{E}[X]$ and variance $Var[X]$, given i.i.d samples $X_1, X_2, \ldots$. Let $r = \sqrt{Var[X]}/\mathbf{E}[X]$ – we now show that we can estimate it up to accuracy $\pm\epsilon\mathbf{E}[X]$ and confidence $1-\delta$ using $O\left(\frac{\epsilon^2}{r^2}\ln\frac{1}{\delta}\right)$ samples.

Part (a)

Given n samples $X_1, \dots, X_n$, suppose we use the estimator $\widehat{X} = \left(\sum_{i=1}^n X_i\right)/n$. Show that $n = O(r^2/\epsilon^2\delta)$ is sufficient to ensure:

$$\mathbf{P}\left[|\widehat{X} - \mathbf{E}[X]| \geq \epsilon\mathbf{E}[X]\right] \leq \delta$$

Part (b)

We say an estimator is a *weak estimator* if it satisfies that $\mathbf{P}\left[|\widehat{X} - \mathbf{E}[X]| \geq \epsilon\mathbf{E}[X]\right] \leq 3/4$ – using part (a), show, that we need $O(r^2/\epsilon^2)$ samples to obtain a weak estimator. Now suppose we are given m weak estimates $\widehat{X}_1, \widehat{X}_2, \dots, \widehat{X}_m$, and we define a new estimator $\widetilde{X}$ to be the *median* of these weak estimates. Show that using $m = O(\ln(1/\delta)$ weak estimates gives us an estimate $\widetilde{X}$ that satisfies:

$$\mathbf{P}\left[|\widetilde{X} - \mathbf{E}[X]| \geq \epsilon\mathbf{E}[X]\right] \leq \delta$$

What could go wrong if we used the mean of $\widehat{X}_1, \widehat{X}_2, \dots, \widehat{X}_m$ instead of the median?

Problem 4: (Randomized Set-Cover)

In this problem, we'll look at *randomized rounding*, which is a very powerful technique for solving large-scale combinatorial optimization problems. The main idea is that given a problem which can be written as an optimization problem with integer constraints, we can sometimes solve the *relaxed problem* with non-integer constraints, and then *round* the solutions to get a good assignment. We will highlight this technique for the *Minimum Set-Cover* problem.

We are given a collection of m subsets $\{S_1, S_2, \dots, S_m\}$ which are subsets of some large set U of n elements, such that $\bigcup_i S_i = U$. The Minimum Set-Cover problem is that of selecting the smallest number of sets $\mathcal{C}$ from the collection $\{S_1, S_2, \dots, S_m\}$ such that they cover U, i.e., such that each element in U lies in at least one of the sets in $\mathcal{C}$.

Part (a)

Argue that the minimum-set cover problem is equivalent to the following integer program:

$$\begin{aligned}
\underset{x}{\text{Minimize}} \quad & \sum_i x_i \\
\text{subject to} \quad & \sum_{i|e\in S_i} x_i \geq 1,\ e \in U \\
& x_i \in \{0,1\},\ i \in \{1,2,\dots,m\}
\end{aligned}$$

Let the solution to this problem, i.e., the minimum set-cover, be denoted OPT.

Next, argue that if we solve the same problem, but now change the last constraint to $x_i \in [0,1]$ for all i, then the resulting solution OPT_{LP} of this *relaxed* problem obeys $OPT_{LP} \leq OPT$. Note that the relaxed problem is an LP and hence can be solved efficiently.

Part (b)

Given a solution z to the relaxed LP, we now round the values to obtain a feasible solution for the original minimum set-cover problem. For each set S_i, we generate $k = c\log n$ i.i.d Bernoulli(z_i) random variables $X_{i,1}, X_{i,2}, \ldots, X_{i,k}$ – if any of them is 1, then we set $x_i = 1$, i.e., we add S_i to our cover $\mathcal{C}$. Prove that the resulting set-cover obeys $\mathbf{E}[|\mathcal{C}|] \leq c\log n \cdot OPT$.

Part (c)

Finally, choose c to ensure that the probability that the resulting set-cover $\mathcal{C}$ does not cover any element $e \in \mathcal{U}$ is less than $1/n^2$.

Problem 5: (Papadimitrou's 2SAT Algorithm) (Optional)

The 2SAT problem(Wikipedia entry), and more generally, the boolean satisfiability (SAT) problem (Wikipedia entry) are one of the cornerstones of theoretical algorithms, and also a very useful modeling tool for a variety of optimization problems. In the general SAT problem, we want to find a satisfying assignment for a given a Boolean expression in n Boolean (i.e., $\{0,1\}$, or FALSE/TRUE) variables $\{X_1, X_2, \ldots, X_n\}$ typically involving *conjunctions* (i.e., logical AND, denoted as $\wedge$), *disjunctions* (i.e., logical OR, denoted as $\vee$) and *negations* (logical NOT; typically $\overline{X}$ denotes the negation of a variable X).

In 2SAT, the expression is restricted to being a conjunction (AND) of several *clauses*, where each clause is the disjunction (OR) of two *literals* (either a variable or its negation). For example, the expression $(X_1 \vee X_2) \wedge (\overline{X_1} \vee X_3) \wedge (X_2 \vee \overline{X_3})$ is a 2SAT formula, which has several satisfying assignments including $(X_1 = 1, X_2 = 1, X_3 = 1)$. Note that in order to find a satisfying assignment, we need to set the variables such that each clause in the formula has at least one literal which is TRUE. Although the general SAT problem is known to be NP-complete, 2SAT can be solved in polynomial time – we will now see a simple randomized algorithm that demonstrates this fact:
Papapdimitrou's 2SAT Algorithm): Given a 2CNF formula F involving n Boolean variables, and an arbitrary assignment τ, we check if τ satisfies F. If not, we pick an arbitrary unsatisfied clause, pick one of its literals *uniformly at random*, and flip it to get a new assignment τ'. We then repeat this until we find a satisfying assignment.

Part (a)

Assume that F has a unique satisfying assignment τ^*, and for any assignment τ, let $N(\tau)$ be the number of literals in τ which agree (i.e., have the same value) as the corresponding literal in τ. Argue that each time we execute an iteration of Papadimitrou's 2SAT algorithm with input assignment τ, the new assignment τ' satisfies:

$$N(\tau') = \begin{cases} N(\tau) + 1 & \text{with probability } \frac{1}{2} \\ N(\tau) - 1 & \text{with probability } \frac{1}{2} \end{cases}$$

Part (b)

Based on the above, argue that the running time T_n of the algorithm is upper bounded by the *first time* that a symmetric random walk starting from 0 hits n or $-n$ (equivalently, $T_n = \arg\min_{K>0}\{\sum_{i=1}^{K} X_i| = n\}$, where X_i are i.i.d Rademacher random variables). Next, show that $\mathbf{E}[T_n] = n^2$, and thus, prove that after $O(n^3)$ iterations, Papadimitrou's algorithm terminates with probability greater than $1 - 1/n$.