CS 6150: HW5 – Randomized algorithms

HINTS/SOLUTIONS

1. For the purposes of this exercise, $\log n$ denotes the natural logarithm.

   (a) (3 points) Suppose we have $m = 4n \log n$ balls, and we throw them into n bins (choosing a bin uniformly at random for each ball, as we saw in class). Prove that the probability that there exists an empty bin is $< 1/n$.

   Fix some bin. The probability that it's empty is

$$\left(1 - \frac{1}{n}\right)^{4n\log n} \leq \frac{1}{e^{4\log n}} = 1/n^4.$$

   Here, we used the inequality we saw in class, of $(1-1/n)^n \leq 1/e$, and the identity $e^{logx} = x$ for the natural logarithm. Thus the expected number of empty bins is $1/n^3$. Now, we can use Markov's ineq to conclude that the probability that the number of empty bins is ≥ 1 is at most $1/n^3 < 1/n$.

   (b) (3 points) What can you say about the probability above, when (a) $m = \frac{1}{2} \cdot n \log n$, and (b) $m = 100 n \log n$?

   Suppose $m = (1/2)n \log n$. For a given bin, the probability that it's empty is now roughly $(1/e)^{(1/2)\log n} = 1/\sqrt{n}$. (Here we are using the fact that $(1-1/n)^n \approx 1/e$ for large n.) Thus the expected number of empty bins is $\sqrt{n}$. Thus, if we denote by p the probability that there exists an empty bin, since the number of empty bins is never $> n$, we get $0(1-p) + pn \geq \sqrt{n}$, which gives $p \geq 1/\sqrt{n}$. (Indeed, this is a very weak bound; if we use ideas from part (d) below, we can show that the probability is really close to 1.)
   Suppose $m = 100 n \log n$. In this case, basically the same proof as in part (a) shows a bound of $1/n^{99}$.

   (c) (2 points) Let us now go back to the setting of $m = n$. We saw in class that the probability that bin j is empty is $\left(1 - \frac{1}{n}\right)^n \leq 1/e$ (and the quantity tends to $1/e$ as $n \to \infty$). Use Markov's inequality to bound the probability that 90% of the bins are empty.

   The expected number of empty bins is thus $\leq n/e$. So by Markov's inequality, the probability that the number of empty bins is $\geq 9n/10$ is at most $\frac{n/e}{9n/10} \leq 10/9e \approx 0.41$.

   (d) (7 points) Let us now obtain a much better bound. Let X_j denote the random variable that is 1 if bin j is empty, and 0 otherwise. We discussed in class why the X_j's are *not* independent. Thus, we cannot use the Chernoff bound to bound the probability that 90% of the bins are empty.

   However, the variables in this case are "anti-correlated". Formally, prove that for any distinct indices $j_1, j_2, \dots, j_k$, we have

$$\Pr[X_{j_1} = 1 | X_{j_2} = X_{j_3} = \dots = X_{j_k} = 1] \leq \Pr[X_{j_1} = 1].$$

   I.e., conditioning on other bins being empty only decreases the probability that a given bin is empty. Use this to prove that the probability that 90% of the bins are empty is at most $(0.9)^n$ (which is exponentially small in n).

   [*Hint:* first use the above inequality along with Bayes rule to prove that $\Pr[X_{j_1} = X_{j_2} = X_{j_3} = \dots = X_{j_k} = 1] \leq e^{-k}$.]

Conditioning on bins $j_2, j_3, \dots, j_k$ being empty is equivalent to the random process in which we have n balls and $n-k+1$ bins. Thus the probability that the bin j_1 is empty is $\left(1 - \frac{1}{n-k+1}\right)^n$, which is $\le \left(1 - \frac{1}{n}\right)^n$ which is the probability that bin j_1 is empty without any conditioning.

Now, using Bayes rule ($\Pr[A \wedge B] = \Pr[A|B]\Pr[B]$), we have:

$$\begin{aligned}
&\Pr[X_{j_1} = X_{j_2} = X_{j_3} = \dots = X_{j_k} = 1] \\
&\quad = \Pr[X_{j_1} = 1 | X_{j_2} = \dots = X_{j_k} = 1]\Pr[X_{j_2} = X_{j_3} = \dots = X_{j_k} = 1] \\
&\quad \le \Pr[X_{j_1} = 1]\Pr[X_{j_2} = X_{j_3} = \dots = X_{j_k} = 1] \\
&\quad \le \Pr[X_{j_1} = 1]\Pr[X_{j_2} = 1] \dots \Pr[X_{j_k} = 1].
\end{aligned}$$

(We used the observation above about negative correlation, and then induction.) This gives the upper bound $1/e^k$ on the above quantity.

Thus the probability that there exist $9n/10$ empty bins is at most

$$\binom{n}{9n/10} \frac{1}{e^{9n/10}} < \frac{2^n}{e^{9n/10}} < 0.9^n.$$

We used the simple fact that $\binom{n}{k} < 2^n$ for all k.

2. The point of this exercise is to learn to *apply* Chernoff bounds in simple cases.

(a) (3 points) Let $a_1, a_2, \dots, a_n$ be n real numbers in $[-1, 1]$. Say we would like to compute their mean by sampling indices j at random from $\{1, 2, \dots, n\}$ (with replacement) and computing the average of the sampled a_j's. How many indices must we sample in order to obtain an estimate $\widehat{\mu}$ of the mean μ such that $|\widehat{\mu} - \mu| \le \epsilon$ with probability $1 - \delta$?

[*Hint:* You may want to go through the survey on Concentration bounds by Chung and Lu, linked on the course page.]

Let X_i be the ith sample, and suppose we take m samples. Since sampling is done with replacement, the X_i's are all independent (and have an identical distribution), and X_i takes value a_j w.p. $1/n$ for all j. For each i, $\mathbf{E}[X_i] = \mu$, and the variance of X_i is $\mathbf{E}[(X_i - \mu)^2] = \mathbf{E}[X_i^2] - \mu^2 \le 1$ (because $|a_j| \le 1$ for all j).

We are interested in bounding the probability of the event $\left|\frac{1}{m}\left(\sum_{i=1}^m X_i\right) - \mu\right| > \epsilon$. This is the same as the event $|\sum_{i=1}^m (X_i - \mu)| > m\epsilon$.

Now, let us write $Y_i = X_i - \mu$, and use Theorem 3.1 from the Chung-Lu survey (the Y_i's have mean zero, and the same variance as X_i, so we can set $\sigma = 1$). We get, for all k, (as the number of X_i's we have is m)

$$\Pr[|\sum_i Y_i| \ge k] \le 2e^{-k^2/4m}.$$

From the discussion above, we wish to set $k = m\epsilon$, in which case the RHS becomes $2e^{-\epsilon^2 m/4}$. In order to make this probability $\le \delta$, we must have

$$\epsilon^2 m/4 \ge \log(2/\delta) \iff m \ge 4 \cdot \frac{\log(2/\delta)}{\epsilon^2}.$$

(b) (1 point) Suppose we sample without replacement. Does your proof above still work? (Answer yes/no, with a couple of lines of reasoning.)

No. We lose the fact that the different X_i are independent, which is crucial to using Chernoff.

(c) (2 points) Suppose we weaken the constraint on a_i to $a_i \in [-M, M]$, for some parameter M. How many samples must we now take in order to obtain an estimate $\widehat{\mu}$ with the same guarantee as in part (a)?

In this case, the variance bound is larger. If $a_i \in [-M, M]$, we can do the same calculation as before, to get $\mathbf{E}[(X_i - \mu)^2] \leq M^2$. Thus, following the calculation above, we get that $m \geq 4 \cdot \frac{M^2 \log(2/\delta)}{\epsilon^2}$ (additional M^2 term).

(d) (4 points) Suppose n is odd. Prove that it is not possible to obtain an estimate of the *median* via samples. Formally, show that if we have $a_1, a_2, \dots, a_n \in [0, 1]$, and given an $\epsilon < 1/8$, it is not possible to estimate the median of the a_i's up to a $\pm\epsilon$ error by sampling $o(n)$ of the a_i's (with or without replacement – it does not matter) and computing the sample median.

Let $n = 2k + 1$, and let $a_1, \dots, a_k$ be reals in the range $[0, 1/4]$ and $a_{k+2}, \dots, a_{2k+1}$ be in the range $[3/4, 1]$. Suppose $a_{k+1} = 1/2$. Now, unless the element a_{k+1} is sampled, the sample median will either be $\leq 1/4$, or $\geq 3/4$. And if we pick $o(n)$ of the a_i's there is only a $o(1)$ probability of picking a_{k+1}, hence only $o(1)$ probability of estimating the median up to an additive error of $1/4$.

3. We have seen in one of the lectures that *any* comparison-based sorting algorithm must make at least $n \log_2 n - O(n)$ comparisons in the worst case, when given an array of size n. Now we will give a randomized algorithm that achieves this bound asymptotically (in expectation).

Let $A[0, \dots, n-1]$ be the input array. Consider a variant of quick-sort, in which instead of picking a uniformly random pivot, we sample M random elements of A (without replacement), and pick the median of these entries as a pivot. The intuition is that this will lead to a near-perfect split with good probability, and thus the number of comparisons will follow the recurrence that "nearly" looks like $T(n) = 2T(n/2) + n$.

(a) (3 points) Suppose $M = 2m + 1$, for an integer $m \geq 1$. Let p_k be the probability that the k'th smallest element in A is chosen as the pivot by the above procedure ($1 \leq k \leq n$). Prove that

$$p_k = \frac{\binom{k-1}{m}\binom{n-k}{m}}{\binom{n}{2m+1}}.$$

In order to make the kth smallest element the median, we must choose m elements from the first $k - 1$, and m from the last $n - k$, and choose the kth element. The total number of ways of doing it is $\binom{k-1}{m}\binom{n-k}{m} \cdot 1$. Thus we have the expression as desired.

(b) (3 points) Give a recursive formula for the expected number of comparisons on an array of size n, in terms of the numbers p_k defined above.

We saw this in class. Let $T(n)$ denote the expected number of comparisons on an array of size n. If the k'th element is the median, then the number of comparisons is $T(k) + T(n - k) + n$. This leads to the recurrence:

$$T(n) = n + \sum_{k=1}^{n-1} p_k [T(k) + T(n-k)].$$

(c) (4 points) Now, we can use an approximation for p_k to solve the recursion. It turns out that we can approximate

$$p_k \approx \frac{(2m+1)!}{m!\, m!} \cdot \frac{1}{n} \left(\frac{k}{n}\right)^m \left(1 - \frac{k}{n}\right)^m,$$

and plug it into the recursion above. For $m = 1$ and $m = 5$, prove by induction that the expected number of comparisons is $\leq C_m \cdot n \log_2 n$, for appropriate constants. You get half credit if $C_5 < C_1$, and full credit if additionally, $C_5 \leq 1.6$. [*Hint:* approximate the summation with an integral,[1] and use Wolfram Alpha to obtain numeric approximations.]

[1] For those who haven't seen this trick, please talk to the TA's for explanation. (Or Google for good resources, and share!)

Since $p_k = p_{n-k}$, we can rewrite the recurrence above as

$$T(n) = n + \sum_{k=1}^{n-1} 2p_k T(k).$$

Let us prove by induction that $T(n) \leq C_m \cdot n \log_2 n$. Assume this is true for integers $< n$. (If you are worried about the base case $n = 1$, you can simply add a $2n$ term.) Then, in order to prove the inductive step, we need to show that

$$n + \sum_{k=1}^{n-1} 2p_k C_m k \log_2 k \leq C_m \cdot n \log_2 n.$$

The LHS can now be written, using the above approximation, as

$$n + \frac{2C_m(2m+1)!}{m!m!} \cdot \sum_{k=1}^{n-1} \frac{1}{n} \left(\frac{k}{n}\right)^m \left(1 - \frac{k}{n}\right)^m k \log_2 k.$$

Let us focus on the summation part. Writing $k \log_2 k = n \cdot (n/k) \log_2(k/n) + \log_2 n$, setting $(k/n) = t$, we can approximate the above summation by the integral

$$\begin{aligned} &n \int_0^1 t^m (1-t)^m t(\log_2 t + \log_2 n) dt \\ &\quad = n \log_2 n \int_0^1 t^{m+1} (1-t)^m dt + n \int_0^1 t^{m+1}(1-t)^m \log_2 t dt. \end{aligned}$$

Let us set $m = 1$. The first integral above evaluates to $1/12$, which precisely cancels out the $2(2m+1)!/m!^2$ term. [This isn't a coincidence, the integral in general evaluates to $m!^2/2(2m+1)!$.] Thus, the constant C_1 needs to satisfy

$$n + 12C_1 \cdot n \int_0^1 t^2 (1-t) \log_2 t dt \leq 0.$$

Evaluating this integral numerically, this simplifies to $(0.84157) * C_1 \geq 1$, or $C_1 \geq 1.19$, roughly. Thus, the inductive argument works for, say, $C_1 = 1.2$.
Let us now set $m = 5$. Even here, the first integral evaluates to $1/5544$, which is precisely $2 \cdot 11!/5!^2$. So now, the constant C_5 needs to satisfy

$$n + 5544C_5 \cdot n \int_0^1 t^6 (1-t)^5 \log_2 t dt \leq 0.$$

Again, evaluating numerically, this simplifies to $(0.94238) * C_5 \geq 1$, or $C_5 \geq 1.062$. Thus the inductive argument works for $C_5 = 1.07$, pretty close to 1!

4. The point of this exercise is to work out the details of the very simple, randomized algorithm for 'global min-cut', proposed by Karger. As this is now standard in randomized algorithms classes, you can easily find the solution online, but I encourage you to work out the calculations by yourself first.

   The global min cut problem is the following: given a connected undirected graph $G = (V, E)$ (no edge weights), the goal is to remove as few edges as possible so as to disconnect the graph.

   Karger's algorithm is extremely simple: at every step, pick a random edge in the graph (uniformly at random), and "collapse" its end points into a supernode (this gives a new graph with one vertex less, on which we recurse). In the process, we keep the parallel edges, and discard self loops.

   The algorithm ends when precisely two vertices remain, at which point we return the number of (parallel) edges between the vertices.

(a) (2 points) Let E' be one of the min cuts in the graph (it is the subset of edges that need to be cut). Prove that if we collapse an edge that is *not* in E', then the size of the min cut in the new graph is equal to that in G.

> No collapsing can *reduce* the size of the min cut (unless we reduce to one vertex), thus min cut in the new graph has value $\geq$ min cut in G. To see the other direction, if we do not collapse an edge in E' (which say corresponds to the cut across $(S, \overline{S})$), it means that we collapsed an edge between vertices in S, or an edge between two vertices in $\overline{S}$. In either case, the cut $(S, \overline{S})$ is preserved in the new graph.

(b) (2 points) If E' is one of the min cuts in the graph, prove that $|E'| \leq 2|E|/n$. [*Hint:* what is the average number of edges in a 'vertex cut'? (a vertex cut is a cut in which we remove all the edges incident on one vertex)]

> The vertex cut around vertex i has size equal to the degree d_i of i. The min cut is thus $\leq d_i$ for all i. Hence, it is also $\leq (1/n)\sum_i d_i$ (the average), which is equal to $2|E|/n$.

(c) (2 points) Use these observations to prove that the probability that the min cut value is maintained after one step is at least $\left(1 - \frac{2}{n}\right)$.

> Fix some min cut E'. By part (b), we have $|E'| \leq 2|E|/n$. Now by part (a), if we do not collapse any of the edges in E', then the min cut value stays the same. This happens with prob equal to $1 - \frac{|E'|}{|E|} \leq 1 - \frac{2}{n}$.

We can then recurse, and obtain that the overall probability of success is at least

$$\left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right) \dots \left(1 - \frac{2}{3}\right) \approx \frac{2}{n^2}.$$

Thus, if we run the algorithm $O(n^2)$ times, there is a very high probability that at least one of the times, the process *succeeds* in finding the cut E'.

(d) (4 points) One surprising consequence of the algorithm is that the *number* of min cuts in an unweighted graph is small. Prove, using the last observation above, that the number of min cuts is $\leq n^2/2$. (Significantly smaller than the total number of cuts, which is 2^n.)

[*Hint:* Suppose we have min cuts $E'_1, E'_2, ...E'_r$. We could have used the reasoning in parts (a)-(c) with any of the min cuts, and we have that the probability that we end up with E'_i is at least $2/n^2$ for each i.]

> The hint is basically the solution. Fix any min cut E', and suppose it corresponds to $(S, \overline{S})$. The probability that our sequence of collapses led to all vertices in S merging to one node and all vertices in S' merging to the other is at least $2/n^2$, by the reasoning after part (c). [This should strike you as surprising. It "looks" like the collapsing is random and can lead to many possible configurations, but in fact, the number of final outcomes is pretty small!] This is true for every min cut. For different cuts, we cannot be collapsing the same sets $S, \overline{S}$. Thus, these events are disjoint, and thus since the probabilities add up to ≤ 1, we have at most $n^2/2$ cuts.

5. (5 points) Let $a_1, a_2, \dots, a_m$ be random integers chosen independently from the interval $[1, N]$. Prove that the probability that the "argmin" of the a_i (i.e., the index j s.t. a_j is minimum) is unique with probability at least $\left(1 - \frac{1}{N}\right)^{m-1}$.

[*Remark:* This is a special case of a general and powerful result known as the *Isolation Lemma* (see the Wikipedia article on the same for more details).]

[*Hint:* Prove by induction on m.]

> Let p_m denote the probability of interest. Clearly, $p_1 = 1$ (the min is always unique). Thus, suppose $m \geq 2$, and assume that $p_{m-1} \geq \left(1 - \frac{1}{N}\right)^{m-2}$. Now, suppose we have already chosen $a_1, \dots, a_{m-1}$. The probability that the min is unique is p_{m-1}, by definition. Conditioned on

this, can we lower bound the probability that the min of $a_1, \ldots, a_{m-1}, a_m$ is unique? We can, by observing that this is true, as long as the value of a_m is not equal to the (unique) min of $a_1, \ldots, a_{m-1}$. If it's smaller, then a_m becomes the new (and unique) min, and if not, the same unique min remains! This gives:

$$\Pr[a_1, \ldots, a_m \text{ has unique min} \mid a_1, \ldots, a_{m-1} \text{ has unique min}] \geq 1 - \frac{1}{N}.$$

Thus, by Bayes theorem, $p_m \geq p_{m-1}\left(1 - \frac{1}{N}\right)$, which completes the proof by induction.

Alas, this proof does not generalize to a proof of the isolation lemma, and a different trick is necessary. See the wiki article.