CS 498ABD Spring 2019 — Homework 1 Solutions

1. Suppose you want to estimate the average of n numbers via sampling (for example the heights of students in the class). The average can be very skewed by outliers. However, we can obtain an accurate estimate if we assume that the numbers are within some limited range. Assume the input numbers $z_1, z_2, \ldots, z_n$ are from $[a, b]$ where $a, b \in \mathbb{R}$ with $a \leq b$. Suppose you sample k input numbers (with replacement) and output their average as the estimate for the true average $\alpha = (\sum_i z_i)/n$. Let X be the random variable denoting the output value.

   (a) Using Chebyshev's inequality, show that for $k \geq \frac{(b-a)^2}{\delta\varepsilon^2}$, we have

   $$\Pr[|X - \alpha| \geq \varepsilon] \leq \delta.$$

   Solution: Let X_i be the random variable for the i-th sample. Then $\mathrm{E}[X_i] = \alpha$, and since $X = \frac{1}{k}\sum_{i=1}^{k} X_i$ and the X_i's are independent, $\mathrm{E}[X] = \alpha$ and $\mathrm{Var}[X] = \frac{1}{k^2}\sum_i \mathrm{Var}[X_i]$.

   Since $X_i \in [a, b]$ so is $\mathrm{E}[X_i]$. Upper bounding $X_i \leq b$ and lower bounding $\mathrm{E}[X_i] \geq a$ gives us $\mathrm{Var}[X_i] = \mathrm{E}[(X_i - \mathrm{E}[X_i])^2] \leq \mathrm{E}[(b-a)^2] = (b-a)^2$.[1] Then $\mathrm{Var}[X] \leq (b-a)^2/k$, and so

   $$\Pr[|X - \alpha| \geq \varepsilon] \leq (b-a)^2/k\varepsilon^2 \leq \delta. \qquad \blacksquare$$

   (b) Using the Chernoff inequality, show that there exists a constant $c > 0$ such that for $k \geq \frac{c(b-a)^2 \log(2/\delta)}{\varepsilon^2}$, we have

   $$\Pr[|X - \alpha| \geq \varepsilon] \leq \delta.$$

   Solution: In order to apply the Chernoff bound, we apply a transformation so that the range is $[-1, 1]$. To do this, set define $f(x) = \frac{2x-b-a}{b-a}$ and set $Y_i = f(X_i)$. Let $Y = \sum_i Y_i$, then $Y = kf(X)$, and $\mathrm{E}[Y] = kf(\alpha)$. This implies that $Y - \mathrm{E}[Y] = \frac{2k}{b-a}(X - \alpha)$.

   Thus if $c = \frac{1}{2}$,

   $$\begin{aligned}\Pr[|X - \alpha| \geq \varepsilon] &= \Pr\left[\frac{b-a}{2k}|Y - \mathrm{E}[Y]| \geq \varepsilon\right] \\ &= \Pr\left[|Y - \mathrm{E}[Y]| \geq \frac{2k}{b-a}\varepsilon\right] \\ &\leq 2\exp(-2k\varepsilon^2/(b-a)^2) \\ &\leq \delta. \end{aligned} \qquad \blacksquare$$

[1] A tighter bound of $\mathrm{Var}[X_i] \leq (b-a)^2/4$ can be proved as follows. Let $Y_i = X_i - a$. Then $\mathrm{Var}[X_i] = \mathrm{Var}[Y_i]$. Setting $c = b - a$,

$$\mathrm{Var}[Y_i] = \mathrm{E}[Y_i^2] - \mathrm{E}[Y_i]^2 \leq \mathrm{E}[cY_i] - \mathrm{E}[Y_i]^2 = \mathrm{E}[Y_i](c - \mathrm{E}[Y_i]).$$

The function $f(x) = x(c-x)$ is maximized at $x = c/2$, so $\mathrm{Var}[X_i] \leq c^2/4 = (b-a)^2/4$.

2. Consider the following variant of Quick Sort. Given an array A of n numbers (which we assume are distinct for simplicity) the algorithm picks a pivot x uniformly at random from A and computes the rank of x. If the rank of x is between $n/4$ and $3n/4$ (call such a pivot a good pivot) it behaves like the normal Quick Sort in partitioning the array A and recursing on both sides. If the rank of x does not satisfy the desired property (the pivot picked is not good) the algorithm simply repeats the process of picking the pivot until it finds a good one. Note that in principle the algorithm may never terminate!

   (a) Write a formal description of the algorithm.

   Solution: The algorithm is as follows:

```
QuickSortVar($A[1..n]$)
  if $|A| \le 4$
      sort $A$ explicitly and return the sorted array
  else
      repeat
          Pick pivot $x$ uniformly at random from $A$
          Partition $A$ into $A_{\text{less}}$, $x$, and $A_{\text{greater}}$ using $x$ as pivot
      until $|A_{\text{less}}| \ge \frac{n}{4}$ and $|A_{\text{less}}| < \frac{3n}{4}$
      return the concatenation of QuickSortVar($A_{\text{less}}$), $x$, and QuickSortVar($A_{\text{greater}}$)
```

■

   (b) Prove that the expected run time of this algorithm is $O(n \log n)$ on an array of n numbers.

   Solution (Version 1): We modify the QuickSort recurrence from lecture. Let Z be the random variable for the number of times a pivot is selected before a good pivot is found, and X_i be an indicator random variable for the event that the rank i element is chosen as a *good* pivot. Then $Q(A)$ satisfies the recurrence

$$Q(A) \le Zn + \sum_{i=\frac{n}{4}+1}^{\frac{3n}{4}} X_i \left(Q(A^i_{\text{less}}) + Q(A^i_{\text{greater}})\right)$$

   Then $\text{E}[Z] = 2$, and for $\frac{n}{4} < i \le \frac{3n}{4}$, $\Pr[X_i = 1] = \frac{2}{n}$, so setting $T(n) = \text{E}[Q(A)]$,

$$T(n) \le 2n + \frac{2}{n} \sum_{i=\frac{n}{4}+1}^{\frac{3n}{4}} (T(i-1) + T(n-i)) = 2n + \frac{4}{n} \sum_{i=\frac{n}{4}}^{\frac{3n}{4}-1} T(i).$$

   We will show that $T(n) \le cn \lg n$ for some $c > 0$. The base case is trivial. Otherwise,

$$T(n) \le 2n + \frac{4}{n} \sum_{i=\frac{n}{4}}^{\frac{3n}{4}-1} T(i) \le 2n + \frac{4}{n} \sum_{i=\frac{n}{4}}^{\frac{3n}{4}-1} (ci \lg i) = 2n + \frac{4c}{n} \sum_{i=\frac{n}{4}}^{\frac{3n}{4}-1} i \lg i$$

   We can upper bound $\sum_{i=\frac{n}{4}}^{\frac{3n}{4}-1} i \lg i$ by $\int_{n/4}^{3n/4} x \log x \, dx$. Another possibility is as follows:

$$\begin{aligned}
\sum_{i=\frac{n}{4}}^{\frac{3n}{4}-1} i \lg i &\le \sum_{i=\frac{n}{4}}^{\frac{n}{2}} i \lg\left(\frac{n}{2}\right) + \sum_{i=\frac{n}{2}+1}^{\frac{3n}{4}-1} i \lg n \\
&= (\lg n) \sum_{i=\frac{n}{4}}^{\frac{3n}{4}-1} i - \sum_{i=\frac{n}{4}}^{\frac{n}{2}} i
\end{aligned}$$

$$
\begin{aligned}
&= (\lg n)\left(\frac{1}{2}\left(\frac{3n}{4}-1\right)\left(\frac{3n}{4}-2\right)-\frac{1}{2}\left(\frac{n}{4}-1\right)\left(\frac{n}{4}-2\right)\right) \\
&\quad -\left(\frac{1}{2}\left(\frac{n}{2}\right)\left(\frac{n}{2}-1\right)-\frac{1}{2}\left(\frac{n}{4}\right)\left(\frac{n}{4}-1\right)\right) \\
&\leq (\lg n)\frac{n^2}{4}-\frac{3n^2}{32}.
\end{aligned}
$$

For $c \geq 6$, the above bound gives us

$$T(n) \leq 2n + cn\lg n - 3cn/8 \leq cn\lg n. \qquad \blacksquare$$

Solution (Version 2): This is similar in spirit to the slick analysis of QuickSort from the lecture. Let $Q(A)$ be the number of comparisons done on the input A. Let X_{ij} be the indicator random variable for the event that the i-th element is compared to the pivot in the j-th recursive level. In each recursive call, the length of the array shrinks by a factor of at least 1/4. Thus the number of recursive calls is at most $\log_{4/3} n$. In other words,

$$Q(A) \leq \sum_{i=1}^{n} \sum_{j=1}^{\log_{4/3} n} X_{ij}.$$

To analyze X_{ij}, we observe that the number of times the i-th element is compared to the pivot is exactly the number of times a pivot is selected before a *good* pivot is found, so $\mathrm{E}[X_{ij}] = 2$. In other words,

$$\mathrm{E}[Q(A)] \leq \sum_{i=1}^{n} \sum_{j=1}^{\log_{4/3} n} \mathrm{E}[X_{ij}] = 2n\log_{4/3} n = O(n\log n). \qquad \blacksquare$$

(c) Prove that the algorithm terminates in $O(n\log n)$ time with high probability.

Solution: This is very similar to the analysis done in lecture. Instead of tracking recursive levels, we track the number of pivot choices.[2] If the split induced by the pivot is balanced, i.e., each partition has size between $\frac{1}{4}$ and $\frac{3}{4}$ of the input, we make a recursive call on the two partitions. Otherwise, we choose another pivot. Thus, each element is involved a sequence of pivot choices (until it finds itself in an array of length at most 4).

Fix an element x of the input, and let S_j be the partition containing x in the j-th pivot choice. If the pivot does not induce a balanced split of the input, $S_{j+1} = S_j$. Say that x is *lucky* for the j-th pivot choice if the pivot has rank between $|S_j|/4$ and $3|S_j|/4$, i.e., $|S_j|/4 \leq |S_{j+1}| \leq 3|S_j|/4$.

Let X_j be the indicator random variable for the event that x is lucky for the j-th choice. Then $\Pr[X_j = 1] = \frac{1}{2}$. If ρ is the number of lucky choices in the first k, $|S_k| \leq (3/4)^\rho n$. For $|S_k| \leq 4$, it suffices to set $\rho = 4\ln n$. Furthermore, the indicator random variables X_i are independent, as the pivots are picked independently. Finally, since $\rho = \sum_{i=1}^{k} X_i$, $\mathrm{E}[\rho] = \frac{k}{2}$.

Thus setting $k = 32\ln n$ and $\delta = \frac{3}{4}$, we can apply the Chernoff bound to show that the probability that x does not get $4\ln n$ lucky pivot choices out of $32\ln n$ is

$$\Pr[\rho \leq 4\ln n] = \Pr[\rho \leq k/8] = \Pr[\rho \leq (1-\delta)\mathrm{E}[\rho]] \leq \frac{1}{n^4}.$$

Taking a union bound over all x, we conclude that with probability $1 - \frac{1}{n^3}$, the running time of QUICKSORTVAR is $O(n\log n)$. $\blacksquare$

[2]For the version of QuickSort from lecture, counting recursive levels is exactly the same thing as counting pivot choices. With this change in point of view, the proof is actually *identical* to the one from lecture!

3. Suppose we want to generate N pairwise independent random variables in the range $\{0,1,2,\ldots,M-1\}$. We will assume that N and M are powers of 2 and let $N=\{0,1\}^n$ and $M=\{0,1\}^m$ (hence $n=\log N$ and $m=\log M$). We saw a scheme in the lecture using mn bits. Here we will revisit that scheme in a different way and then see how it can be made more randomness-efficient.

   Pick a uniformly random matrix $A\in\{0,1\}^{m\times n}$ and a random vector $b\in\{0,1\}^m$. Then for a vector $v\in\{0,1\}^n$, set $X_v=Av+b \mod 2$ (by this we mean component wise mod 2).

   (a) Suppose we pick A and b uniformly at random. Show that under this scheme, for all $w\in\{0,1\}^n$ where $w\neq\vec{0}$ and for all $\gamma\in\{0,1\}^m$,

   $$\Pr_A[Aw=\gamma \bmod 2]=\frac{1}{2^m}.$$

   Why does this guarantee that X_u and X_v are independent for $u\neq v$ and $u\neq\vec{0}, v\neq\vec{0}$?

   Solution: As we will see, the solutions to both this part and the next part mostly involve repeated application of the fact that if $x,y\in\{0,1\}$ are such that x is uniformly distributed and independent from y, then $x+y \bmod 2$ is uniformly distributed in $\{0,1\}$.

   Choosing A uniformly at random means independently choosing each entry uniformly at random. This means that the rows of A are independent from each other, and thus the entries of Aw are also independent from each other.

   Since $w\neq\vec{0}$, we may assume without loss of generality (by permuting the order of the entries of A and w if necessary) that the first entry of w is 1. With this assumption, we can write $(Aw)_i=A_{i,1}+\sum_{j=2}^n A_{i,j}w_j$. Then since $A_{i,1}$ is uniformly distributed and independent from $\sum_{j=2}^n A_{i,j}w_j$, $(Aw)_i=A_{i,1}+\sum_{j=2}^n A_{i,j}w_j$ is uniformly distributed, i.e., $\Pr[(Aw)_i=\gamma_i]=\frac{1}{2}$. We conclude

   $$\Pr[Aw=\gamma]=\Pr\left[\bigwedge_{i=1}^m (Aw)_i=\gamma_i\right]=\prod_{i=1}^m \Pr[(Aw)_i=\gamma_i]=\frac{1}{2^m}.$$

   To show that X_u and X_v are independent, we need to show that for fixed $\alpha,\beta\in\{0,1\}^m$,

   $$\Pr[X_u=\alpha\wedge X_v=\beta]=\Pr[X_u=\alpha]\Pr[X_v=\beta].$$

   Since the entries of b are independently chosen uniformly at random, we can apply the argument seen above to show that $\Pr[Au+b=\alpha]=\Pr[Av+b=\beta]=\frac{1}{2^m}$.

   The equations $Au+b=\alpha$ and $Av+b=\beta$ hold if and only if $A(u-v)=\alpha-\beta$ and $b=\beta-Av$ hold. Thus

   $$\begin{aligned}\Pr[X_u=\alpha\wedge X_v=\beta]&=\Pr[A(u-v)=\alpha-\beta\wedge b=\beta-Av]\\&=\Pr[A(u-v)=\alpha-\beta]\Pr[b=\beta-Av\mid A(u-v)=\alpha-\beta].\end{aligned}$$

   Since $u\neq v$, $u-v\neq 0$, so we can use the equation we proved to conclude that

   $$\Pr[A(u-v)=\alpha-\beta]=\frac{1}{2^m}.$$

   For any fixed A solving $A(u-v)=\alpha-\beta$, there is only one b that satisfies $b=\beta-Av$. Then since b is chosen uniformly,

   $$\Pr[b=\beta-Av\mid A(u-v)=\alpha-\beta]=\frac{1}{2^m}.$$

   Putting it all together, we obtain

   $$\Pr[X_u=\alpha\wedge X_v=\beta]=\frac{1}{2^{2m}}=\Pr[X_u=\alpha]\Pr[X_v=\beta].$$ ■

(b) We can make the following improvement. A matrix A is *Toeplitz* if the entries along each diagonal are constant, i.e., $A_{i,j} = A_{i+1,j+1}$. Suppose we choose A uniformly at random from the set of Toeplitz matrices whose entries are in $\{0,1\}$. Show that under this scheme, it is also the case that for all $w \in \{0,1\}^n$ where $w \neq \vec{0}$ and for all $\gamma \in \{0,1\}^m$,

$$\Pr_A[Aw = \gamma \bmod 2] = \frac{1}{2^m}.$$

Solution: Choosing a Toeplitz matrix uniformly at random is the same thing as independently choosing the entries in the first column and first row uniformly at random, since all the other entries are determined by these choices.

Since A is Toeplitz, the rows of A are not independent from each other, so we cannot conclude directly that the entries of Aw are independent. Instead, we have

$$\Pr[Aw = \gamma] = \Pr\left[\bigwedge_{i=1}^{m} (Aw)_i = \gamma_i\right] = \prod_{i=1}^{m} \Pr\left[(Aw)_i = \gamma_i \,\middle|\, \bigwedge_{j<i} (Aw)_j = \gamma_j\right].$$

If we show that $\Pr[(Aw)_i = \gamma_i \mid \bigwedge_{j<i}(Aw)_j = \gamma_j] = \frac{1}{2}$, then $\Pr[Aw = \gamma] = \frac{1}{2^m}$, as desired. This also shows that even though the rows of A are not independent, the entries of Aw are.

Again, assume without loss of generality that the first entry of w is 1. There are two cases.

For $i = 1$, $(Aw)_1 = \sum_{j=1}^{n} A_{1,j} w_j$. As noted above, the values $A_{1,j}$ are independently chosen uniformly at random, so $(Aw)_1$ is uniformly distributed by the same argument as in the previous part. Thus $\Pr[(Aw)_1 = \gamma_1] = \frac{1}{2}$.

For $i > 1$, fix any solution to the system $(Aw)_j = \gamma_j$ for all $j < i$. By definition of Toeplitz matrices, this also fixes the entries $A_{i,2}, \ldots, A_{i,n}$. We assumed that $w_1 = 1$, and $A_{i,1}$ is chosen uniformly at random independently from $A_{i,2}, \ldots, A_{i,n}$, so $(Aw)_i = A_{i,1} + \sum_{j=2}^{n} A_{i,j} w_j$ is uniformly distributed. Thus $\Pr[(Aw)_i = \gamma_i \mid \bigwedge_{j<i}(Aw)_j = \gamma_j] = \frac{1}{2}$. ■

(c) How many random bits do we need to generate a random Toeplitz matrix $A \in \{0,1\}^{m \times n}$ and how much storage (in bits) do you need to store it implicitly? Express this as a function of N and M and compare with the number of random bits needed for the case when A is picked as a random $\{0,1\}$ matrix.

Solution: Since a Toeplitz matrix T is determined by its first column and first row, the number of random bits required to generate T is $m + n - 1 = \log(M) + \log(N) - 1$. This is also the number of bits needed to store it implicitly. On the other hand, if A is a random $\{0,1\}$ matrix, then every entry needs to be chosen randomly, requiring $mn = \log(M)\log(N)$ bits. In the worst case, we also need to store that many bits. ■

4. In lecture we analyzed probabilistic counting: initialize a counter X to 1, and for every increment instruction, increment X with probability $1/2^X$. By averaging many such estimators, we obtained a $(1+\varepsilon)$-approximation to n with good probability and space usage was $O(\log\log n)$. In this problem you will investigate a minor modification. Imagine we still initialize X to 1, but we increment it with probability $1/(1+a)^X$ for some fixed $a > 0$. (Note that your estimator for n would have to change from $2^X - 1$ to something else.)

   How small must a be so that our estimate $\widetilde{n}$ of n satisfies $|\widetilde{n} - n| \le \varepsilon n$ with at least 9/10 probability when we return the output of a single estimator instead of averaging many estimators we did in the lecture? Also derive a bound $S = S(n)$ on the space (in bits) so that this algorithm uses at most S space with at least 9/10 probability by the end of the n increments.

   Solution: Let X_i be the counter after i events, and set $Y_i = (1+a)^{X_i}$. We claim that $\mathrm{E}[Y_n] = an+1$. The base case is when $n = 0$: $X_0 = 0$ and $Y_0 = 1$. Otherwise,

$$\begin{aligned}
\mathrm{E}[Y_n] &= \mathrm{E}[(1+a)^{X_n}] \\
&= \sum_j (1+a)^j \Pr[X_n = j] \\
&= \sum_j (1+a)^j \left(\Pr[X_{n-1} = j]\left(1 - \frac{1}{(1+a)^j}\right) + \Pr[X_{n-1} = j-1]\frac{1}{(1+a)^{j-1}}\right) \\
&= \sum_j (1+a)^j \Pr[X_{n-1} = j] + \sum_j ((1+a)\Pr[X_{n-1} = j-1] - \Pr[X_{n-1} = j]) \\
&= \mathrm{E}[Y_{n-1}] + a = a(n-1) + 1 + a = an + 1.
\end{aligned}$$

   Thus the correct estimator is $\boldsymbol{\widetilde{n} = ((1+a)^X - 1)/a}$.

   Next, we analyze $\mathrm{E}[Y_n^2]$. When $n = 0$, $Y_0 = 1$. Otherwise,

$$\begin{aligned}
\mathrm{E}[Y_n^2] &= \mathrm{E}[(1+a)^{2X_n}] \\
&= \sum_j (1+a)^{2j} \Pr[X_n = j] \\
&= \sum_j (1+a)^{2j} \left(\Pr[X_{n-1} = j]\left(1 - \frac{1}{(1+a)^j}\right) + \Pr[X_{n-1} = j-1]\frac{1}{(1+a)^{j-1}}\right) \\
&= \sum_j (1+a)^{2j} \Pr[X_{n-1} = j] \\
&\qquad + \sum_j ((1+a)^2 \cdot (1+a)^{j-1} \Pr[X_{n-1} = j-1] - (1+a)^j \Pr[X_{n-1} = j]) \\
&= \mathrm{E}[Y_{n-1}^2] + (a^2 + 2a)\,\mathrm{E}[Y_{n-1}].
\end{aligned}$$

   Standard techniques for solving linear recurrences (such as the characteristic method or Wolfram Alpha) can be used to solve the recurrence $b_n = b_{n-1} + (a^2+2a)(a(n-1)+1)$, $b_0 = 1$. The solution is $b_n = \frac{a^3+2a^2}{2}n^2 + \frac{4a-a^3}{2}n + 1$, so $\mathrm{Var}[Y_n] = a^3 n(n-1)/2$, and thus $\mathrm{Var}[\widetilde{n}] = \mathrm{Var}[(Y_n - 1)/a] = an(n-1)/2$.

   Chebyshev's inequality gives us $\Pr[|\widetilde{n} - n| \ge \varepsilon n] \le \frac{a}{2\varepsilon^2}$. If this is at most 1/10, then $\boldsymbol{a \le \varepsilon^2/5}$.

   By Jensen's inequality, $(1+a)^{\mathrm{E}[X_n]} \le \mathrm{E}[Y_n] = an+1$, so $\mathrm{E}[X_n] \le \log_{1+a}(an+1) = \frac{\ln(an+1)}{\ln(1+a)}$. For small a, we can use the approximation $\ln(1+a) \approx a$ to estimate the expected number of bits to store X_n by $O(\log \frac{\log an}{a}) = \boldsymbol{O(\log\log n + \log \frac{1}{\varepsilon})}$ if $a \approx \varepsilon^2$.

   More generally, if we want a $(1+\varepsilon)$-approximation with probability at least $(1-\delta)$, we need $a \le 2\delta\varepsilon^2$ so the number of bits needed is $O(\log \frac{\log n}{\delta\varepsilon^2}) = \boldsymbol{O(\log\log n + \log \frac{1}{\delta} + \log \frac{1}{\varepsilon})}$. ■