CS 255: Homework 1

Due: Feb. 16, at 9 pm

Instructions

1. You may work in groups of up to 3 students.
2. Each group makes one submission through Canvas.
3. Include the names and SJSU ids for each student in the group.

1 Problem 1

(a) Describe an algorithm that simulates a fair three-sided die, using independent fair coin flips as the only source of randomness. Your algorithm should return 1, 2, or 3, each with probability 1/3. What is the expected numer of coin flips used by your algorithm?

Solution. There are many possible solutions, any approach that produces a number 1, 2, or 3 with 1/3 probability in a constant number of expected coin flips was accepted. Below is one possible approach.

In one trial, we flip the coin twice. We create the assignment $\{HH\} = 3$, $\{HT\} = 2$, $\{TH\} = 1$, and $\{TT\}$ is a failure. For each $x \in \{1, 2, 3\}$, we have $P($ return $x|$ trial is not a failure $) = 1/3$. In the event of a failure, we perform another trial. Each trial succeeds with probability $p = 3/4$, so the expected number of *trials* until the first success is $1/p = 4/3$. Each trial is 2 flips, so on average we need 8/3 flips.

(b) Describe an algorithm that simulates a fair coin flip, using independent rolls of a fair three-sided did as the only source of randomness. What is the expected number of die rolls used by your algorithm?

Solution. Again there are many possible solutions; any approach with a constant expected number of rolls was accepted.

In one trial, we roll the die once. If the number one the die is 1 or 2 we return that number; otherwise the trial fails and we roll again. For $x \in \{1, 2\}$, we have $P($ return $x|$ trial is not a failure $) = 1/2$. Each trial succeeds with probability 2/3. The expected number of trials (rolls) until the first success is 3/2.

Problem 2

Recall the max-cut problem from lecture 2. Given an undirected graph $G = (V, E)$, we want to find a partition of the vertices (A, B) that maximizes the number of cut edges (edges with one end point in A and the other in B). In lecture, we analyzed a randomized algorithm: process the vertices in any order, and flip a fair coin (probability of heads 1/2) to assign the each vertex to the set A or B. We showed the randomized algorithm yields an average cut size of $m/2$, where $m = |E|$. This problem derives a simple greedy algorithm using a method called derandomization through conditional expectation.

(a) Let $v_1, \ldots, v_n$ be any ordering of the vertices, and assume that we process the vertices in that order. Since the coin is fair, conditioning on the outcome of the first coin flip gives:

$$\mathbb{E}[\text{size of cut}] = 0.5 \cdot \mathbb{E}[\text{size of cut}|v_1 \in A] + 0.5 \cdot \mathbb{E}[\text{size of cut}|v_1 \in B], \tag{1}$$

where the randomness is over the assignment of $v_2, \ldots, v_n$.

In lecture we showed that $\mathbb{E}[\text{size of cut}] = m/2$. Explain why at least one of the conditional expectations in the right hand side of (1) must be at least $m/2$.

Solution. If both conditional expectations are less that $m/2$, then

$$0.5 \cdot E[\text{size of cut}|v_1 \in A] + 0.5 \cdot E[\text{size of cut}|v_1 \in B] < 0.5(\frac{m}{2} + \frac{m}{2}) = \frac{m}{2} = E[\text{size of cut}],$$

which contradicts (1).

(b) We can generalize part (a) as follows. Given an assignment of the vertices $v_1, \ldots, v_{i-1}$, we have:

$$\begin{aligned} E[\text{size of cut}|\text{assignment of } v_i, \ldots, v_{i-1}] = 0.5 \cdot E[\text{size of cut}|v_i \in A,\ \text{assignment of } v_1, \ldots, v_{i-1}] \\ + 0.5 \cdot E[\text{size of cut}|v_i \in B,\ \text{assignment of } v_1, \ldots, v_{i-1}], \end{aligned} \tag{2}$$

where the randomness is over the assignment of $v_i, \ldots, v_n$.

The idea is to assign v_i to the set A or B which maximizes the conditional expectation on the right side above. Given an assignment of the vertices $v_1, \ldots, v_{i-1}$, explain how to compute

$$E[\text{size of cut}|v_i \in A,\ \text{assignment of } v_1, \ldots, v_{i-1}].$$

Solution. We use linearity of expectation, similar to part (b), and consider the probability that any edge (v_j, v_k) is cut given the partial assignment of $v_1, \ldots, v_{i-1}$. There are five cases:

i) Both endpoints are already assigned, that is $v_j, v_k \in \{v_1, \ldots, v_{i-1}\}$. Then, there is no randomness; the edge is already cut, or it isn't.

ii) Neither endpoint is in $\{v_1, \ldots, v_i\}$. This edge is cut with probability 1/2, by part (a).

iii) One endpoint, say v_j, is in $\{v_1, \ldots, v_i\}$, and the other, v_k, is not. Then, the edge is cut if v_k is assigned to the opposite side of the partition from v_j. This happens with probability 1/2.

iv) One endpoint, say v_j, is in $\{v_1, \ldots, v_{i-1}\}$, the other is v_i. Since we are conditioning on $v_i \in A$, the edge is cut if $v_j \in B$. (Similarly, if we conditioned on $v_i \in B$, the edge is cut if $v_j \in A$.)

From the cases above, we have

$$\begin{aligned} E[\text{size of cut}|v_i \in A,\ \text{assignment of } v_1, \ldots, v_{i-1}] &= (\#\text{ edges cut between } \{v_1 \ldots v_{i-1}\}) \\ &+ (\#\text{ edges } (v_i, v_j) \text{ with } v_j \text{ already assigned to } B) \\ &+ 0.5 \times (\#\text{ edges with at least one end point in } \{v_{i+1}, \ldots, v_n\}). \end{aligned}$$

(c) Given an assignment of the vertices $v_1, \ldots, v_{i-1}$, show that assigning v_i to maximize the conditional expectation in (2) does not depend on the vertices $v_{i+1}, \ldots, v_n$.

Solution. If we compare the conditional expectations in (2), then the only difference is in the terms:

$$(\#\text{ edges } (v_i, v_j) \text{ with } v_j \text{ already assigned to } B)$$

and

$$(\#\text{ edges } (v_i, v_j) \text{ with } v_j \text{ already assigned to } A).$$

These terms do not depend on the assignment of $v_{i+1}, \ldots, v_n$.

(f) Describe and analyze a deterministic algorithm that computes a cut of size at least $|E|/2$.

Solution. We process the vertices in an arbitrary order. Given the assignment of vertices $\{v_1, \ldots, v_{i-1}\}$, we assign v_i to *maximize* the number edges cut between v_i and the already assign vertices $\{v_1, \ldots, v_{i-1}\}$. That is, if there are more edges between v_i and the vertices of B, we assign v_i to A; otherwise we assign v_i to B. This algorithm runs in $O(V + E)$ time.

Problem 3

Consider a random walk on a path with nodes numbered $1, 2, \ldots, n$ from left to right. At each step, we flip a fair coin to decide which direction to walk, moving one step left or one step right with equal probability. The random walk ends when we fall off one end of the path, either by moving left from node 1 or by moving right from node n. (a) Prove that if we start at vertex 1, the probability that the walk ends by falling off the *right* end of the path is exactly $1/(n+1)$.

Solution: Let p_i = probability the walk ends by falling off the *right* end of the path, given that it starts from vertex $i \in [0 \ldots n+1]$. Here, we define the base cases, $p_0 = 0$ and $p_{n+1} = 1$ where walk has ended by falling of the left and right ends of the path respectively.

By conditioning on the outcome of the first flip, we see that for all $i \in [1 \ldots n]$:

$$p_i = 0.5p_{i-1} + 0.5p_{i+1}.$$

Using $p_i = 0.5p_i + 0.5p_i$ in the above, we find that

$$p_{i+1} - p_i = p_i - p_{i-1}.$$

Now, since $p_0 = 0$, we obtain:

$$\begin{aligned}
p_2 - p_1 &= p_1 \\
p_3 - p_2 &= p_2 - p_1 = p_1 \\
p_4 - p_3 &= p_3 - p_2 = p_1 \\
&\vdots \\
p_{i+1} - p_i &= p_i - p_{i-1} = p_1.
\end{aligned}$$

Adding the first $i - 1$ of these equations yields:

$$p_i - p_1 = \sum_{j=1}^{i-1} p_{j+1} - p_j = \sum_{j=1}^{i-1} p_1,$$

or

$$p_i = i \cdot p_1. \tag{3}$$

Now, since $p_{n+1} = 1$, we get

$$p_1 = \frac{1}{n+1}.$$

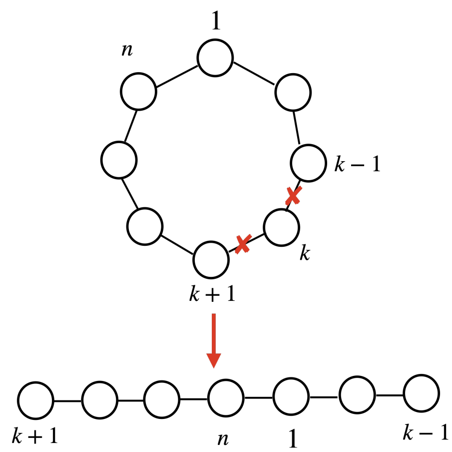
(b) Prove that if we start at vertex k, the probability that the walk ends by falling off the *right* end of the path is exactly $k/(n+1)$.

Solution: Using (3) and $p_1 = 1/(n+1)$, we find

$$p_k = \frac{k}{n+1}.$$

(c) Consider the same setting on a simple cycle. That is, connect the two ends of the path so that going right from node n leads to node 1. Fix any node $1 < k < n$. Starting from node 1 the walk will eventually reach node k either from the left (node $k-1$) or from the right (node $k+1$). Starting from node 1 what is the probability that the walk reaches node k from the left?

Solution. We can reduce this problem to part (c) by cutting the edges connected to node k, refer to the figure below. This makes node $k-1$ the right most node of graph and $k+1$ the left



most. Node 1 is now at position $j = n - (k+1) + 2$. Using this j in the formula from part b, we obtain:

$$P(\text{ reach node } k \text{ from node } k-1) = \frac{n-(k+1)+2}{n} = \frac{n-k+1}{n}.$$

Problem 4

Suppose we are given a coin that may or may not be biased, and we would like to compute an accurate *estimate* of the probability of heads. Specifically, if the actual unknown probability of heads is p, we would like to compute an estimate $\hat{p}$ such that

$$P(|\hat{p} - p| > \epsilon) \leq \delta,$$

where ϵ is a given accuracy parameter, and δ is a given confidence parameter.

```
MeanEstimate(ε):
    count ← 0
    for i ← 1 to N do
        if Flip() = Heads then
            count ← count + 1
    return count/N
```

The following algorithm is a natural first attempt; here Flip() returns the result of an independent flip of the coin.

(a) Let $\hat{p}$ denote the estimate returned by MeanEstimate(ϵ). Prove that $\mathbb{E}[\hat{p}] = p$.

Solution. Define the indicator random variables: $X_i = 1$ if the ith flip is heads, 0 otherwise. Then, $X = \sum_{i=1}^{N} X_i$ is the total number of heads. The value returned by MeanEstimate is: $\hat{p} = X/N$. Each X_i is a Bernoulli random variable with probability of success p and mean $E[X_i] = p$. By linearity of expectation:

$$E[\hat{p}] = E[X/N] = \frac{1}{N}\sum_{i=1}^{N} E[X_i] = \frac{1}{N}\sum_{i=1}^{N} p = p.$$

(b) Prove that if we set $N = \lceil \alpha/\epsilon^2 \rceil$ for some appropriate constant α, then we have

$$P(|\hat{p} - p| > \epsilon) < 1/4.$$

[Hint: use Chebyshev's Inequality.]

Solution. The variance of a Bernoulli random variable with parameter p is $Var[X_i] = p(1-p)$. Since the coin flips are independent, the X_i's are independent. It follows that:

$$Var[\hat{p}] = Var[X/N] = \frac{1}{N^2}\sum_{i=1}^{N} Var[X_i] = \frac{p(1-p)}{N}.$$

Using the result of part (a) and Chebyshev's inequality:

$$P(|\hat{p} - p| > \epsilon) \leq \frac{Var[\hat{p}]}{\epsilon^2} = \frac{p(1-p)}{N\epsilon^2}.$$

The simplest bound for the above is $p(1-p) \leq 1$, for any $p \in [0,1]$. In this case, choosing $N = \lceil 4/\epsilon^2 \rceil$, yields:

$$P(|\hat{p} - p| > \epsilon) \leq \frac{1}{4}.$$

We can get a slightly better bound by observing that function $f(x) = x(1-x)$ is maximized at $x^* = 1/2$ where $f(x^*) = 1/4$. This shows that we only need $N = \lceil 1/\epsilon^2 \rceil$.

(c) We can increase the previous estimator's confidence by running it multiple times, independently, and returning the *median* of the estimates. Let p^* denote the estimate returned by MedianOfMeansEstimate(δ, ϵ). Prove that if we set $N = \lceil \alpha/\epsilon^2 \rceil$ (inside MeanEstimate) and $K = \lceil \beta \ln(1/\delta) \rceil$, for some appropriate constants α and β, then $P(|p^* - p| > \epsilon) < \delta$. [Hint: use Chernoff bounds.]

MedianOfMeansEstimate(δ, ϵ)**:**

for $j \leftarrow 1$ **to** K **do**

| $\quad estimate[j] \leftarrow$ MeanEstimate(ϵ)

return Median($estimate[1 \ldots K]$)

Solution. We call each run of MeanEstimate a trial. For any $\epsilon > 0$, we say that a trial succeeds if $P(|\hat{p} - p| \leq \epsilon)$, otherwise it fails. Let $Y_j = 1$ if trial j fails, 0 otherwise. Then $Y = \sum_{j=1}^{K} Y_j$ is the number of failed trials. By part (b), if $N = \lceil 4/\epsilon^2 \rceil$ then $E[Y] \leq K/4$.

Observe that the MedianOfMeansEstimate exceeds ϵ only if at least $K/2$ trials fail. Recall that we can use an upper bound on the mean in a Chernoff bound. Using the bound $E[Y] \leq K/4$ from part (b) in the simplified Chernoff bound:

$$P(Y \geq K/2) = P(Y \geq (1+1) \cdot K/4) \leq e^{-\frac{1^2 K/4}{3}} = e^{-K/12},$$

as long as $N = \lceil 4/\epsilon^2 \rceil$. Choosing $K = \lceil 12 \ln(1/\delta) \rceil$, yields:

$$P(Y \geq K/2) \leq \delta.$$

Therefore, if $N = \lceil 4/\epsilon^2 \rceil$ and $K = \lceil 12 \ln(1/\delta) \rceil$, then the MedianOfMeansEstimate satisfies $|\hat{p} - p| \leq \epsilon$, with probability at least $1 - \delta$.