

Problem Set 2

Spring 2019

Issued: September 4, 2019

Due: 11:59 PM, Wednesday, September 11, 2019

1. Secret Hitler

In the game of Secret Hitler, there are 9 players. 4 of them are Fascist and 5 of them are Liberal. There is also a deck of 17 cards containing 11 Fascist “policies” and 6 Liberal “policies”. Fascists want to play Fascist policies, and Liberals want to play Liberal policies. Here’s how the play proceeds.

- A President and a Chancellor are chosen uniformly at random from the 9 players.
- The President draws 3 policies from the deck and gives 2 to the Chancellor.
- The Chancellor chooses one to play.

Now suppose you are the Chancellor, but the President gave you 2 Fascists. Being a Liberal, you wonder, did the President just happen to have 3 Fascist policies, or was the President a Fascist who secretly discarded a Liberal policy. In this scenario, what’s the probability that the President is Fascist? Let’s assume that Fascist presidents always try to discard Liberal policies.

2. Packet Routing

Packets arriving at a switch are routed to either destination A (with probability p) or destination B (with probability $1 - p$). The destination of each packet is chosen independently of each other. In the time interval $[0, 1]$, the number of arriving packets is $\text{Poisson}(\lambda)$.

- (a) Show that the number of packets routed to A is Poisson distributed. With what parameter?
- (b) Are the number of packets routed to A and to B independent?

3. Compact Arrays

Consider an array of n entries, where n is a positive integer. Each entry is chosen uniformly randomly from $\{0, \dots, 9\}$. We want to make the array more compact, by putting all of the non-zero entries together at the front of the array. As an example, suppose we have the array

$$[6, 4, 0, 0, 5, 3, 0, 5, 1, 3].$$

After making the array compact, it now looks like

$$[6, 4, 5, 3, 5, 1, 3, 0, 0, 0].$$

Let i be a fixed positive integer in $\{1, \dots, n\}$. Suppose that the i th entry of the array is non-zero (assume that the array is indexed starting from 1). Let X be a random variable which is equal to the index that the i th entry has been moved after making the array compact. Calculate $\mathbb{E}[X]$ and $\text{var}(X)$.

4. Message Segmentation

The number of bytes N in a message has a geometric distribution with parameter p . Suppose that the message is segmented into packets, with each packet containing m bytes if possible, and any remaining bytes being put in the last packet. Let Q denote the number of full packets in the message, and let R denote the number of bytes left over.

- Find the joint PMF of Q and R . Pay attention on the support of the joint PMF.
- Find the marginal PMFs of Q and R .
- Repeat part (b), given that we know that $N > m$.

Note: you can use the formulas

$$\sum_{k=0}^n a^k = \frac{1 - a^{n+1}}{1 - a}, \text{ for } a \neq 1$$

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1 - x}, \text{ for } |x| < 1$$

in order to simplify your answer.

5. Almost fixed points of a permutation

Let Ω be the set of all permutations of the numbers $1, 2, \dots, n$. Let an almost fixed point be defined as follows: If we put the numbers $i \in 1, 2, \dots, n$ around a circle in clockwise order (such that 1 and n are next to each other) and then assign another number $\omega(i) \in 1, 2, \dots, n$ to it, if the number $\omega(i)$ is next to i (or is equal to i), we will say that i is almost a fixed point. So, for the permutation $\omega(1) = 5, \omega(2) = 3, \omega(3) = 1, \omega(4) = 4, \omega(5) = 2$, we have that 1, 2, and 4 are almost fixed points.

Now, let $X(\omega)$ denote the number of almost fixed points in $\omega \in \Omega$. Find $\mathbb{E}[X]$ and $\text{var}(X)$.

6. Soliton Distribution

This question pertains to the **fountain codes** introduced in the lab.

Say that you wish to send n chunks of a message, X_1, \dots, X_n , across a channel, but alas the channel is a **packet erasure channel**: each of the packets you send is erased with probability $p_e > 0$. Instead of sending the n chunks directly through the channel, we will instead send n packets through the channel, call them Y_1, \dots, Y_n . How do we choose the packets Y_1, \dots, Y_n ? Let $p(\cdot)$ be a probability distribution on $\{1, \dots, n\}$; this represents the **degree distribution** of the packets.

- (i) For $i = 1, \dots, n$, pick D_i (a random variable) according to the distribution $p(\cdot)$. Then, choose D_i random chunks among X_1, \dots, X_n , and “assign” Y_i to the D_i chosen chunks.
- (ii) For $i = 1, \dots, n$, let Y_i be the XOR of all of the chunks assigned for Y_i (the number of chunks assigned for Y_i is called the **degree** of Y_i).
- (iii) Send each Y_i across the channel, along with metadata which describes which chunks were assigned to Y_i .

The receiver on the other side of the channel receives the packets Y_1, \dots, Y_n (for simplicity, assume that no packets are erased by the channel; in this problem, we are just trying to understand what we should do in the ideal situation of *no* channel noise), and decoding proceeds as follows:

- (i) If there is a received packet Y with only one assigned chunk X_j , then set $X_j = Y$. Then, “peel off” X_j : for all packets Y_i that X_j is assigned to, replace Y_i with $Y_i \text{ XOR } X_j$. Remove Y and X_j (notice that this may create new degree-one packets, which allows decoding to continue).
- (ii) Repeat the above step until all chunks have been decoded, or there are no remaining degree-one packets (in which case we declare failure).

In the lab, you will play around with the algorithm and watch it in action. Here, our goal is to work out what a good degree distribution $p(\cdot)$ is.

Intuitively, a good degree distribution needs to occasionally have prolific packets that have high degree; this is to ensure that all packets are connected to at least one chunk. However, we need “most” of the packets to have low degree to make decoding easier. Ideally, we would like to choose $p(\cdot)$ such that at each step of the algorithm, there is exactly one degree-one packet.

- (a) Suppose that when k chunks have been recovered ($k = 0, 1, \dots, N - 1$), then the expected number of packets of degree d (for $d > 1$) is $f_k(d)$. Assuming we are in the ideal situation where there is exactly one degree-one packet for any k : What is the probability that a given degree d packet is connected to the chunk we are about to peel off? Based on that, what is the expected number of packets of degree d whose degrees are reduced by one after the $(k + 1)$ st chunk is peeled off?
- (b) We want $f_k(1) = 1$ for all $k = 0, 1, \dots, n - 1$. Show that in order for this to hold, then for all $d = 2, \dots, n$ we have $f_k(d) = (n - k)/[d(d - 1)]$. From this, deduce what $p(d)$ must be, for $d = 1, \dots, n$. (This is called the **ideal soliton distribution**.)

[*Hint*: You should get two different recursion equations since the only degree 1 node at peeling $k + 1$ is going to come from the peeling of degree 2 nodes at peeling k , however, for other higher degree d nodes, there will be some probability that some degree d ones will remain from the previous iteration and some probability that they will come from $d + 1$ one that will be peeled off]

- (c) Find the expectation of the distribution $p(\cdot)$.

In practice, the ideal soliton distribution does not perform very well because it is not enough to design the distribution to work well in expectation.

7. [Bonus] Connected Random Graph

The bonus question is just for fun. You are not required to submit the bonus question, but do give it a try and write down your progress.

We start with the empty graph on n vertices, and iteratively we keep on adding undirected edges $\{u, v\}$ uniformly at random from the edges that are not so far present in the graph, until the graph is connected. Let X be a random variable which is equal to the total number of edges of the graph. Show that $\mathbb{E}[X] = O(n \log n)$.

Hint: consider the random variable X_k which is equal to the number of edges added while there are k connected components, until there are $k - 1$ connected components. Don't try to calculate $\mathbb{E}[X_k]$, an upper bound is enough.