Partner 1
Partner 2
Partner 3

Homework 2

CSCI 1550 / 2540
February 20, 2025

Due: March 6, 2025
Remember to show your work for each problem to receive full credit.

Problem 1 (40 Points)

In this exercise, we design a randomized algorithm for the following packet routing problem. We are given a network that is an directed connected graph G, where nodes represent processors and the edges between the nodes represent wires. We are also given a set of N packets to route. For each packet we are given a source node, a destination node, and the exact route (path in the graph) that the packet should take from the source to its destination. (We may assume that there are no loops in the path.) In each time step, at most one packet can traverse any single edge. A packet can wait at any node during any time step, and we assume unbounded queue sizes at each node.

A *schedule* for a set of packets specifies the timing for the movement of packets along their respective routes. That is, it specifies which packets should move and which should wait at each time step. Our goal is to produce a schedule for the packets that tries to minimize the total time and the maximum queue size needed to route all the packets to their destinations.

(a) The dilation d is the maximum distance traveled by any packet. The congestion c is the maximum number of packets that must traverse a single edge during the entire course of the routing. Argue that the time required for any schedule should be at least $\Omega(c+d)$.

Solution: If a packet must do d hops to reach its destination, then it is clear that the time required for any scheduling is $\geq d$. Also, if an edge must be traversed by at least c packets, any scheduling requires $\geq c$ time as there is the constraint that only one packet can traverse the same edge for any time step. By combining these two lower bounds, we obtain that the time required by any scheduling is at least $\geq \max\{c, d\}$. We conclude that any scheduling requires time $\Omega(c+d)$ by observing that $\max\{c, d\} \geq (c+d)/2$.

(b) Consider the following unconstrained schedule, where many packets may traverse an edge during a single time step. Assign each packet an integer delay chosen randomly, independently, and uniformly from the interval $[1, \lceil \frac{\alpha c}{\log(Nd)} \rceil]$, where $\alpha \leq 1$ is a constant. A packet that is assigned a delay of x waits in its source node for x time steps; then it moves on to its final destination through its specified route without ever stopping. Prove that with probability $1 - (Nd)^{-\frac{1}{3\alpha}}$ no more than $\frac{2\log(Nd)}{\alpha}$ packets use a particular edge E, at a particular step t.

Solution: Number the packets from 1 to N. Let $e \in E$ be an edge of the graph. Denote with $S(e) = \{\text{i} \ : \text{pckt i traverses edge e}\}$. Note that for any edge e, $|S(e)| \leq c$. Let $C_{i,e}(t)$ be 1 if packet i traverses edge e at time t, 0 otherwise. For any $i \in S(e)$, $C_{i,e}(t)$ is a Bernoulli random variable and $C_{i,e}(t) \leq 1/(\lceil \frac{c\alpha}{\log(Nd)} \rceil) \leq \frac{\log(Nd)}{c\alpha}$. This is due to the fact that the packet i waits a random time between 1 and $\lceil \frac{c\alpha}{\log(Nd)} \rceil$, so the probability

that it traverses at time t is at most the probability that it chooses the right value in this interval to start its routing (at most, as this may not be possible if t is too large). Note that $C_{i,e}(t)$ and $C_{j,e}(t)$ are independent for $i, j \in S(e)$, with $i \neq j$. Let $B_1, \ldots, B_N$ be N indepenent Bernoulli random variables with mean $\frac{\log(Nd)}{c\alpha}$.

We now have that for any edge e and time t:

$$\Pr\left(\sum_{i \in S(e)} C_{i,e}(t) \geq 2 \cdot \frac{\log(Nd)}{\alpha}\right) \leq \Pr\left(\sum_{i=1}^{c} B_i \geq 2 \cdot \frac{\log(Nd)}{\alpha}\right)$$

where we used the fact that $|S(e)| \leq c$, i.e. we are taking the sum over more elements. We can apply Chernoff's bound with $\delta = 1$ and obtain that:

$$\Pr\left(\sum_{i \in S(e)} C_{i,e}(t) \geq 2 \cdot \frac{\log(Nd)}{\alpha}\right) \leq e^{-\log(Nd)/(c\alpha)\cdot c \cdot 1/3} = (Nd)^{-1/3\alpha}$$

which gives the desired bound.

(c) Again using the unconstrained schedule of part (b), show that there exists a constant α such that the probability that more than $O(\log(Nd))$ packets pass through any edge at any time step is at most $\frac{1}{Nd}$. [Hint: argue that since there are N packets, and each packet traverses $\leq d$ edges, we need to apply union bound over no more than Nd events.]

Solution: We observe that any packet arrives to its destination in time $\leq d + \lceil \frac{c\alpha}{\log(Nd)} \rceil$, hence for any packet i and edge e, $C_{i,e}(t) = 0$ if $t > d + \lceil \frac{c\alpha}{\log(Nd)} \rceil$. Also, any packet can traverse at maximum d edges, hence the number of edges that are traversed by at least one packet in the graph is at most Nd. Thus, we want to bound the sum $\sum_{i \in S(e)} C_{i,e}(t)$ only for $\leq Nd$ edges and $\leq d + \lceil \frac{c\alpha}{\log(Nd)} \rceil$ possible time steps. By an union bound, we have that:

$$\begin{aligned}\gamma = \left(\text{at any time, an edge in the network is traversed by } \geq 2 \cdot \frac{\log(Nd)}{\alpha} \text{ packets}\right) \\ \leq Nd\left(d + \left\lceil \frac{c\alpha}{\log(Nd)} \right\rceil\right)(Nd)^{-1/3\alpha}\end{aligned}$$

Choose some sufficiently small α so that $\lceil \frac{c\alpha}{\log(Nd)} \rceil < 1$ and $\alpha < 1/9$. Then, γ is bounded by

$$\gamma < (Nd) \cdot (d+1)(Nd)^{-3}$$

Finally, because d is a positive integer, we know that $d \geq 1/(N-1)$ (assuming $N > 1$). This rearranges to $Nd \geq d + 1$. Therefore,

$$\gamma < (Nd)(Nd)(Nd)^{-3} = (Nd)^{-1}$$

as desired.

(d) Use the unconstrained schedule to devise a simple randomized algorithm that, with high probability (in N), produces a schedule of length $O(c + d\log(Nd))$ using queues of size $O(\log(Nd))$ and following the constraint that at most one packet crosses any single edge per time step.

Solution: Consider scaling the unconstrained schedule by $2\log(Nd)/\alpha$; that is, for each time step, make it of size $2\log(Nd)/\alpha$ so that if there are $c > 1$ packets at one node about to cross a certain edge, they cross the edge one at a time. As seen in part c, with probability at least $1 - 1/Nd$ there is no edge that will be crossed by more than $2\log(Nd)/\alpha$ packets at one point in time in the unconstrained schedule, so it is guaranteed that in this new process that only one packet will cross one edge at a time, and there will be queues of size $O(\log(Nd))$ at the nodes.

Notice that the unconstrained schedule is of length at most

$$d + \left\lceil \frac{\alpha c}{\log(Nd)} \right\rceil.$$

Scaling this up by $O(\log(Nd))$, this new process has complexity $O(c + d\log(Nd))$ as desired.

Homework 2

Problem 2 (20 points)

In many wireless communication systems, each receiver listens on a specific frequency. The bit $b(t)$ sent at time t is represented by a 1 or -1. Unfortunately, noise from other nearby communications can affect the receiver's signal. A simplified model of this noise is as follows. There are n other senders, and the ith has strength $p_i \leq 1$. At any time t, the ith sender is also trying to send a bit $b_i(t)$ that is represented by 1 or -1. The receiver obtains the signal $s(t)$ given by

$$s(t) = b(t) + \sum_{i=1}^{n} p_i b_i(t)$$

If $s(t)$ is closer to 1 than -1, the receiver assumes that the bit sent at time t was a 1 otherwise, the receiver assumes that it was a -1.

Assume that all the bits $b_i(t)$ can be considered independent, uniform random variables. Give a Chernoff bound to prove the probability that the receiver makes an error in determining $b(t)$ is less than or equal to following quantity

$$\exp(\frac{-1}{2\sum_{i=1}^{n} p_i^2}).$$

Solution: We know that an error occurs if the original bit $b(t) = 1$ and the received bit $s(t) = -1$ or vice versa. By symmetry of $\sum_{-=1}^{n} p_i b_i(t)$ around 0, it is sufficient to calculate the probability for the case where $b(t) = 1$ and $s(t) = -1$. Then taking a Chernoff bound, we get that for some parameter θ:

$$\mathbb{P}\left(\sum_{i=1}^{n} p_i b_i(t) > 1\right) = \mathbb{P}\left(\prod_{i=1}^{n} \left(e^{\theta b_i(t)}\right)^{p_i} > e^{\theta}\right) \leq \frac{\prod_{i=1}^{n} \mathbb{E}[e^{\theta b_i(t) p_i}]}{e^{\theta}} \leq \frac{\prod_{i=1}^{n} \left(\frac{1}{2}e^{\theta p_i} + \frac{1}{2}e^{-\theta p_i}\right)}{e^{\theta}}$$

We can bound the last term with the Taylor series, giving

$$\frac{1}{2}e^{\theta} + \frac{1}{2}e^{-\theta} = \sum_{k=0}^{\infty} \frac{\theta^{2k}}{(2k)!} \leq \sum_{k=0}^{\infty} \frac{\theta^{2k}}{(2^k k)!} = e^{\theta^2/2}$$

After a quick substitution, we conclude that our error rate is bounded by $e^{\frac{1}{2}\theta^2(\sum_{i=1}^{n} p_i^2) - \theta}$.

We can minimize our error by taking the minimum of $\frac{1}{2}\theta^2(\sum_{i=1}^{n} p_i^2) - \theta$. Setting the first derivative to 0, we see that $\theta = \frac{1}{\sum_{i=1}^{n} p_i^2}$. Then we conclude that the error rate is at most $e^{\frac{-1}{2\sum_{i=1}^{n} p_i^2}}$.

Homework 2

Problem 3 (35 points)

Bob is facing a very challenging math question in **CSCI1550/2540**. Even if this math question is very hard, it has a simple binary answer $Y \in \{0,1\}$ (both answers are equally likely). Bob asks for help from n fellow math-loving friends (numbered from 1 to n), and each of them provides an answer to this math question. However, as this math question is very hard, there is no guarantee that these answers are the same. In particular, friend i provides an answer $X_i \in \{0,1\}$, for $i = 1, \ldots, n$. Bob knows the expertise of each friend; in particular, he knows that for each $i = 1, \ldots, n$, we have that:

$$X_i = \begin{cases} Y & \text{with probability } p_i > 1/2 \\ 1 - Y & \text{with probability } 1 - p_i \end{cases}$$

Formally, $X_1, \ldots, X_n$ are random variables function of Y. Bob also assumes that these friends won't collaborate with each other; that is, given Y, the random variables $X_1, \ldots, X_n$ are independent.

Bob wants to use a function $f(X_1, \ldots, X_n) : \{0,1\}^n \to \{0,1\}$ to obtain the final answer to the hard math problem. He would like to minimize the error that the function f makes a mistake, i.e., he wants to minimize:

$$\Pr(f(X_1, \ldots, X_n) \neq Y) \tag{1}$$

If a function f minimizes (1), we say that f is optimal. Let $\vec{X} = (X_1, \ldots, X_n)$.

(a) For $y \in \{0,1\}$, let

$$g(\vec{x}, y) = \Pr(\vec{X} = \vec{x} | Y = y) = \prod_{i:x_i=y} p_i \prod_{i:x_i=1-y} (1 - p_i) = \exp\left(\sum_{i:x_i=y} \log p_i + \sum_{i:x_i=1-y} \log(1 - p_i) \right)$$

Show that a function f is optimal if and only if for any $\vec{x} \in \{0,1\}^n$, it holds that

$$f(\vec{x}) = \arg \max_{y \in \{0,1\}} g(\vec{x}, y)$$

Solution: Let f be a function $f : \{0,1\}^n \to \{0,1\}$. The error of this function is:

$$\Pr(f(X_1, \ldots, X_n) \neq Y) = \sum_{\vec{x} \in \{0,1\}^n} \Pr(f(\vec{x}) \neq Y | \vec{X} = \vec{x}) \Pr(\vec{X} = \vec{x}) =$$
$$= \sum_{\vec{x} \in \{0,1\}^n} \Pr(\vec{X} = \vec{x} | f(\vec{x}) \neq Y) \Pr(f(\vec{x}) \neq Y)$$

Note that since the function f always returns the same value given $\vec{x}$, we have that $\Pr(f(\vec{x}) \neq Y) = \frac{1}{2}$. Also, $Y \neq f(\vec{x})$ if and only if $Y = 1 - f(\vec{x})$ Therefore, we have that:

$$\begin{aligned}\Pr(f(X_1, \ldots, X_n) \neq Y) &= \frac{1}{2} \sum_{\vec{x} \in \{0,1\}^n} \Pr(\vec{X} = \vec{x} | Y = 1 - f(\vec{x})) \\ &= \frac{1}{2} \sum_{\vec{x} \in \{0,1\}^n} g(\vec{x}; 1 - f(\vec{x}))\end{aligned}$$

Observe that the error is minimized if and only if for any $\vec{x}$, we choose $f(\vec{x}) = \text{argmax}_{y \in \{0,1\}} g(\vec{x}, y)$.

(b) Bob considers a family of functions that is called weighted majority vote. That is, he wants to assign a different weight to the answer of the different friends, based on their competence. Let $\vec{w} = (w_1, \ldots, w_n) \in \mathbb{R}^n$. Given $\vec{w}$, we define:

$$f(\vec{x}; \vec{w}) = \begin{cases} 1 & \text{if } \sum_{i=1}^n X_i w_i \geq \sum_{i=1}^n (1 - X_i) w_i \\ 0 & \text{otherwise} \end{cases}$$

Let $\vec{w}^* = (w_1^*, \ldots, w_n^*)$, where $w_i^* = \ln\left(\frac{p_i}{1-p_i}\right)$. Use the answer to question a. to show that the function $f(\bullet; \vec{w}^*)$ is optimal.

Solution: We have that $f(\vec{x}, \vec{w}^*)$ is equal to 1 if and only if

$$\begin{aligned} f(\vec{x}, \vec{w}^*) = 1 &\iff \sum_{i=1}^n X_i \ln\left(\frac{p_i}{1-p_i}\right) \geq \sum_{i=1}^n (1 - X_i) \ln\left(\frac{p_i}{1-p_i}\right) \\ &\iff \prod_{i:X_i=1} \frac{p_i}{1-p_i} \geq \prod_{i:X_i=0} \frac{p_i}{1-p_i} \\ &\iff \prod_{i:X_i=1} p_i \prod_{i:X_i=0} (1-p_i) \geq \prod_{i:X_i=0} p_i \prod_{i:X_i=1} (1-p_i) \\ &\iff g(\vec{x}, 1) \geq g(\vec{x}, 0) \end{aligned}$$

The first equivalence is due to the definition of f, and the second equivalence is obtained by applying the exponential function to both sides of the inequality. In the last equivalence, we used the definition of g. Optimality follows by Part (a).

(c) Let $f^*(\bullet) = f(\bullet; \vec{w}^*)$. Use Hoeffding's bound to show an upper bound on the error probability

$$\Pr(f^*(X_1, \ldots, X_n) \neq Y)$$

Hint: Show that if $f^*(X_1, ..., X_n) \neq Y$, then the sum of weights w_i, whose corresponding answer X_i is correct, is less than or equal to $\frac{1}{2} \sum_{i=1}^n w_i$.

Solution: For $i = 1, \ldots, n$, let Z_i be a binary random variable that denotes the event $\{X_i = Y\}$, i.e. $Z_i = 1$ if and only if $X_i = Y$, else $Z_i = 0$. We have that Z_i is distributed as a Bernoulli of parameter p_i, i.e. $Z_i \sim B(p_i)$, and for $j \neq i$, Z_j is independent with Z_i by problem assumptions.

We have that f^* makes an error if and only if $\sum_{i=1}^n Z_i w_i \leq \frac{W}{2}$, where $W = \sum_{i=1}^n w_i$ (the sum of the weights of the correct votes is less than the sum of the incorrect votes; in case of a tie we make the worst case assumption that we are wrong).

Let $\tilde{Z}_i = Z_i w_i$. Observe that $\tilde{Z}_i = w_i$ with probability p_i, and $\tilde{Z}_i = 0$ with probability $1 - p_i$. It is clear that $\tilde{Z}_i \in [0, w_i]$ if $p_i > 1/2$, and $\tilde{Z}_i \in [w_i, 0]$ if $p_i < 1/2$, and $\tilde{Z}_i = p_i w_i$.

We have that:

$$\begin{aligned}\Pr(f^*(X_1, \ldots, X_n) \neq Y) &\leq \Pr\left(\sum_{i=1}^n \tilde{Z}_i \leq W/2\right) \\ &= \Pr\left(\sum_{i=1}^n \tilde{Z}_i - \sum_{i=1}^n p_i w_i \leq \sum_{i=1}^n w_i \left(\frac{1}{2} - p_i\right)\right)\end{aligned}$$

Note that $w_i\,(1/2 - p_i) \leq 0$ for each $i = 1, \ldots, n$. Hence, we can apply Hoeffding's bound and obtain that:

$$\Pr(f^*(X_1, \ldots, X_n) \neq Y) \leq \exp\left(-\frac{2\left[\sum_{i=1}^n w_i(p_i - 1/2)\right]^2}{\sum_{i=1}^n w_i^2}\right)$$

(d) Suppose that for each $i = 2, \ldots, n$, we have that $p_i = 0.9$, and let $p_1 \to 1$. What happens to the upper bound computed in question *c.*? Is this upper bound useful or not in this scenario?

Solution: If we have that $p_1 \to 1$ and the other $p_i's$ are all fixed, then $w_1 \to \infty$, and by taking the limit we have that:

$$Pr(f^*(X_1, \ldots, X_n) \neq Y) \leq \exp(-1/2) \simeq 0.6$$

Hence we have the bound becomes vacuous. This point shows a limitation of the applicability of Hoeffding's bound in the case of very heterogenous sums.

Homework 2

Problem 4 (25 points)

This problem demonstrate the difference between additive and multiplicative error deviation bounds.

Let $G = (V, E)$ be an undirected graph, $V = \{1, \ldots, n\}$ and $E \subseteq \{\{i, j\} : i, j \in V \text{ and } i \neq j\}$. We know the number of vertices $|V| = n$. We want to estimate the fraction of pairs $\{i, j\}$ of connected by an edge, $\rho = m/\binom{n}{2}$, where $m = |E|$. We can query an oracle, that given a pair $\{i, j\}$, tells us if i and j are connected by an edge in the graph G, i.e. whether $\{i, j\} \in E$ or not.

(a) **Additive error bound:** Use the Hoeffding's bound to bound the number of queries of pairs, chosen uniformly at random, needed to estimate ρ within an ϵ additive error, i.e. output $\tilde{\rho}$ such that $|\tilde{\rho} - \rho| \leq \epsilon$ with probability at least $1 - \delta$.

Solution: Let $\tilde{\rho}$ be the fraction of edges found out of s random queries. It holds that $\tilde{\rho} = \rho$. By Hoeffding's bound, we have that:

$$\Pr\left(|\tilde{\rho} - \rho| > \epsilon\right) \leq 2\exp\left(-2s\epsilon^2\right) \leq \delta$$

Therefore, by taking $s \geq \frac{1}{2\epsilon^2} \ln \frac{2}{\delta}$ samples, we have that with probability at least $1 - \delta$, it holds that $|\tilde{\rho} - \rho| \leq \epsilon$.

(b) **Multiplicative error bound:**

1. Assume that you given a lower bound d on the fraction ρ. If this lower bound is true, how many random queries are needed to find an estimate $\tilde{\rho}$ that satisfies an ϵ multiplicative error, i.e. $|\tilde{\rho} - \rho| \leq \epsilon\rho$, with probability at least $1 - \delta$?
2. Assume now that you don't have a lower bound of ρ. Design and analyze an algorithm that estimates ρ with a number of sample adjusted to the unknown ρ. [**Hint:** Assume first that $\rho > 1/4$, if the condition doesn't hold assume $\rho > 1/8$, etc. Remember to bound the total error probability.]
3. For which values of ρ is it better to just check all pairs?

Solution: Using the same strategy above, we have that:

$$\Pr\left(|\tilde{\rho} - \rho| > \rho\epsilon\right) \leq 2\exp\left(-2s\epsilon^2\rho^2\right) \leq 2\exp\left(-2s\epsilon^2 d^2\right)$$

In the second inequality, we use the fact that $\rho \geq d$. Again, we set

$$2\exp\left(-2s\epsilon^2 d^2\right) \leq \delta \ .$$

Therefore, by taking $s \geq \frac{1}{2\epsilon^2 d^2} \ln \frac{2}{\delta}$ samples, with probability at least $1 - \delta$, we have that $|\tilde{\rho} - \rho| \leq \rho\epsilon$.

Observe that if $m = O(n)$, then $\rho = o(1/n)$, and we need at least $\Omega(\frac{n^2}{\epsilon^2} \ln \frac{1}{\delta})$ samples to obtain an estimate $|\tilde{\rho} - \rho| \leq \rho\epsilon$ with probability at least $1 - \delta$. In comparison, we only need $\binom{n}{2}$ queries to compute exactly the value ρ. Therefore, we observe that a sampling strategy is not effective to obtain a multiplicative bound if $m = O(n)$, i.e. the graph is sparse. A sampling strategy is asymptotically convenient only if $m = \Omega(n^{1+c})$ for $c > 0$

Assume now that we do not know a lower bound d to ρ. We design an algorithm that is able to compute multiplicative bound by iteratively guessing the value of this lower bound.

Let $d_0 = \frac{1}{4}$, and let $d_i = \frac{1}{2}d_{i-1}$ for $i = 1, 2, \ldots$.

At iteration i, we do s_i random queries and compute the fraction of edges found $\tilde{\rho}_i$. Fixed ϵ and δ_i (see next equation), the number of samples s_i is chosen to satisfy:

$$\Pr(|\tilde{\rho}_i - \rho| \geq \epsilon d_i) \leq 2\exp(-2s_i\epsilon^2 d_i^2) \leq \delta_i \tag{2}$$

Suppose that the event $|\tilde{\rho}_i - \rho| \leq \epsilon d_i$ is true for every iteration (we will further discuss this later). There are two situations: (i) $\tilde{\rho}_i - \epsilon d_i \geq d_i$ and (ii) $\tilde{\rho}_i - \epsilon d_i \leq d_i$.

In case (i), we have that $\rho \geq \tilde{\rho}_i - \epsilon d_i \geq d_i$ (first inequality due to event, second inequality due to the fact that we are in case (i)), therefore we are guaranteed that our lower bound is correct and we can return $\tilde{\rho}_i$. In case (ii), we cannot have this guarantee, therefore we iterate again.

Observe that if $d_i \leq \rho/(1 + 2\epsilon)$, then we have the guarantee that situation (i) will occur (if event $|\tilde{\rho}_i - \rho| \leq \epsilon d_i$ is true). In fact, we have that:

$$|\tilde{\rho}_i - \rho| \leq \epsilon d_i \implies \tilde{\rho}_i - \epsilon d_i \geq \rho - 2\epsilon d_i \geq d_i$$

In the last inequality, we used the fact that $d_i \leq \rho/(1 + 2\epsilon)$. Therefore, if d_i is small enough, and $|\tilde{\rho}_i - \rho| \leq \epsilon d_i$ holds, then we have that situation (i) will occur and the algorithm terminates.

Therefore, the total number of iteration is

$$\leq -\log_2(\rho) + \log(1 + 2\epsilon) = O\left(\log_2 \frac{1}{\rho}\right) \quad .$$

Observe that after $\log_2 n$ iteration, we can say that $\rho \in O(1/n)$ (as the algorithm did not terminate earlier) and a sampling strategy is not more effective (see discussion above). Therefore, we run at maximum $\log_2 n$ iterations of the algorithm, and if the algorithm

did not terminate, we query all the edges in the graph.

This algorithm always returns a correct estimate if for $i = 1, \ldots, \log_2 n$, the events $|\tilde{\rho}_i - \rho| \leq \epsilon d_i$ hold (we are doing a worst case analysis, if one of this single event is not true we say that the algorithm fails).

We set $\delta_i = \delta / \log_2 n$. By union bound, the algorithm returns a correct estimate with probability $\geq 1 - \delta$. The number of random queries required at iteration i are (solving (2))

$$s_i \geq \frac{1}{2\epsilon^2 d_i^2} \ln \left(\frac{\log_2 n}{\delta} \right) .$$