Warm-up

Problem 1. Suppose E_1 and E_2 are two *independent* events, each happening with probability p. What is the probability that at least one of them happens? Compare to what the union bound gives.

Generalise to k independent events $E_1, \dots, E_k$ each happening with probability p.

Problem 2. Prove Chebyshev's inequality using Markov's inequality.

Problem 3. Compute the expectation and variance of a Poisson(λ) random variable. (Recall that if $X \sim \text{Poisson}(\lambda)$, then $\Pr[\, X = k \,] = e^{-\lambda} \frac{\lambda^k}{k!}$ for any integer $k \geq 0$.)

Problem 4. Let X be a Binomial random variable with parameters n and p. Compute (or recall) the expectation and variance of X.

a) Bound the probability that X deviates from its expectation by more than $2\sqrt{np}$.

b) Suppose that $p = \frac{1}{4}$.
   - Use Markov's inequality to bound $\Pr[\, X \geq n/2 \,]$.
   - Use Chebyshev's inequality to bound $\Pr[\, X \geq n/2 \,]$.
   - Use the Chernoff bound to bound $\Pr[\, X \geq n/2 \,]$.
   - Use Hoeffding's bound to bound $\Pr[\, X \geq n/2 \,]$.
   - Compare the 4 bounds.

c) Suppose now that $p = \frac{1}{2n}$.
   - Use Markov's inequality to bound $\Pr[\, X \geq 1 \,]$.
   - Use Chebyshev's inequality to bound $\Pr[\, X \geq 1 \,]$. Comment.
   - Use the Chernoff bound to bound $\Pr[\, X \geq 1 \,]$.
   - Use Hoeffding's bound to bound $\Pr[\, X \geq 1 \,]$.
   - Compute $\Pr[\, X \geq 1 \,]$ exactly, and compare the bounds obtained.

Problem solving

Problem 5. Prove Theorem 8 of the lecture notes:

> Let A be a Monte Carlo algorithm with worst-case running time $T(n)$ and constant failure probability $p \in (0, 1)$, with the following extra guarantee: one can detect whether the output of A is incorrect in time $O(1)$.

Then there exists a *Las Vegas* algorithm A' for the same task with expected running time $O(T(n))$ (where the hidden constant in the $O(\cdot)$ depends on p).

Problem 6. Suppose that we have two Monte Carlo algorithms A and B for a decision problem P, with the following behaviour: on any input x,

- if the true answer $P(x)$ is yes, then A outputs yes with probability at least $1/2$, while B outputs yes with probability one.
- if the true answer $P(x)$ is no, then A outputs no with probability one, while B outputs no with probability at least $1/2$.

Both A and B run in worst-case time $T(|x|)$. Using A and B, design a Las Vegas algorithm C for P. Analyse its expected running time.

Problem 7. Let A be a randomised algorithm which, on input x, consumes (at most) T "resources" and uses (at most) r random bits, outputs good or bad, such that

- If x is good, then $\Pr[\,A(x) = \text{good}\,] \geq 9/10$;
- If x is bad, then $\Pr[\,A(x) = \text{good}\,] \leq 1/10$.

For any $\delta \in (0, 1]$, give a randomised algorithm A' such that, on input x,

- If x is good, then $\Pr[\,A(x) = \text{good}\,] \geq 1 - \delta$;
- If x is bad, then $\Pr[\,A(x) = \text{good}\,] \leq \delta$.

Bound the amount of resources T' and random bits r' this algorithm A' uses.

Problem 8. Similar, but a little different: Let A be a randomised algorithm which, on input x, consumes (at most) T "resources" and uses (at most) r random bits, outputs good or bad, such that

- If x is good, then $\Pr[\,A(x) = \text{good}\,] \geq 1/10$;
- If x is bad, then $\Pr[\,A(x) = \text{good}\,] = 0$.

For any $\delta \in (0, 1]$, give a randomised algorithm A' such that, on input x,

- If x is good, then $\Pr[\,A(x) = \text{good}\,] \geq 1 - \delta$;
- If x is bad, then $\Pr[\,A(x) = \text{good}\,] = 0$.

Bound the amount of resources T' and random bits r' this algorithm A' uses.

Problem 9. We will prove (a simplified version of) the Chernoff bound. Namely, given $X_1, \dots, X_n$ i.i.d. random variables taking values in $\{0, 1\}$, each with expectation p, set $X = \sum_{i=1}^{n} X_i$. We will show that

$$\Pr[\,X > (1+\gamma)\mathbb{E}[X]\,] \leq e^{-\gamma^2 \mathbb{E}[X]/3}, \qquad \gamma \in (0, 1]$$

In what follows, fix any $\gamma \in (0, 1]$.

a) Show that, for every $t > 0$,

$$\Pr[\, X > (1+\gamma)\mathbb{E}[X] \,] = \Pr\left[e^{tX} > e^{t(1+\gamma)\mathbb{E}[X]} \right].$$

b) Deduce that, for every $t > 0$,

$$\Pr[\, X > (1+\gamma)\mathbb{E}[X] \,] \leq \frac{\mathbb{E}\left[e^{tX_1}\right]^n}{e^{t(1+\gamma)\mathbb{E}[X]}}.$$

c) Compute $\mathbb{E}\left[e^{tX_1}\right]$, and deduce that, for every $t > 0$,

$$\Pr[\, X > (1+\gamma)\mathbb{E}[X] \,] \leq \frac{(1+p(e^t-1))^n}{e^{t(1+\gamma)np}}.$$

d) Use the inequality $\ln(1+x) \leq x$ to show that, for every $t > 0$,

$$\Pr[\, X > (1+\gamma)\mathbb{E}[X] \,] \leq e^{-pn\cdot f(t)}.$$

where $f(t) = (1+\gamma)t - (e^t - 1)$.

e) Choose the best value of $t > 0$ (which is a free parameter) to show that

$$\Pr[\, X > (1+\gamma)\mathbb{E}[X] \,] \leq e^{-pn((1+\gamma)\ln(1+\gamma)-\gamma)}.$$

Show (or take for granted, and verify by plotting the two functions) that $(1+\gamma)\ln(1+\gamma) - \gamma \geq \gamma^2/3$ for all $\gamma \in (0,1]$. Conclude.

Advanced

Problem 10. Use the same approach to show the "other side" of the Chernoff bound:

$$\Pr[\, X < (1+\gamma)\mathbb{E}[X] \,] \leq e^{-\gamma^2\mathbb{E}[X]/2}$$

for $\gamma \in (0,1]$. Do you see how to generalise the above argument to $X_1, \ldots, X_n \in [0,1]$? To independent (but non-identically distributed) $X_i's$?

Problem 11. We will prove the *median trick*. Suppose that any given input x is associated with an interval $[a_x, b_x] \subseteq \mathbb{R}$ of "good values." We don't know this interval: our goal is, given any input x to find a good value for x with very high probability, say $1-\delta$ for arbitarily small δ.

All we are given is an algorithm A which, on any input x, is guaranteed to output a good value with reasonably good probability. Specifically,

$$\Pr[\, A(x) < a_x \,] \leq \alpha, \qquad \Pr[\, A(x) > b_x \,] \leq \alpha$$

for some known constant $\alpha < 1/2$. Consider the following algorithm B: on input x, run A on x independently k times, and output the median of all k values obtained.

a) Analyse the probability that the output of B is a good value, as a function of α and k.

b) Set the integer k to achieve our original goal: output a good value with probability at least $1 - \delta$.