Overview

Submit your writeup including all code and plots as a PDF via Gradescope. We recommend reading through the entire homework beforehand and carefully using functions for testing procedures, plotting, and running experiments. Taking the time to test, maintain, and reuse code will help in the long run!

Data science is a collaborative activity. While you may talk with others about the homework, please write up your solutions individually. If you discuss the homework with your peers, please include their names on your submission. Please make sure any handwritten answers are legible, as we may deduct points otherwise.

1 Robust Mean Estimation via Concentration Inequalities

Suppose we observe a sequence of i.i.d. random variables $X_1, \ldots, X_n$. Their distribution is unknown, and has unknown mean μ and known variance σ^2. In this question, we will investigate two different estimators for the mean μ: the sample mean, and the so-called "median of means" estimator. In particular, we will analyze them in terms of how many samples n are required to estimate μ to a given precision ϵ and for a confidence threshold δ.

We'll start with the sample mean for parts (a) - (c): in other words, we'll use $X_1, \ldots, X_n$ to compute an estimate $S_n = \frac{1}{n}\sum_i X_i$ for the mean μ. We want to see what sample size n guarantees that $\mathbb{P}(|\hat{\mu} - \mu| \geq \epsilon) \leq \delta$.

Hint: for this problem, you might find it useful to watch the walk-through of Discussion 9, question 2.

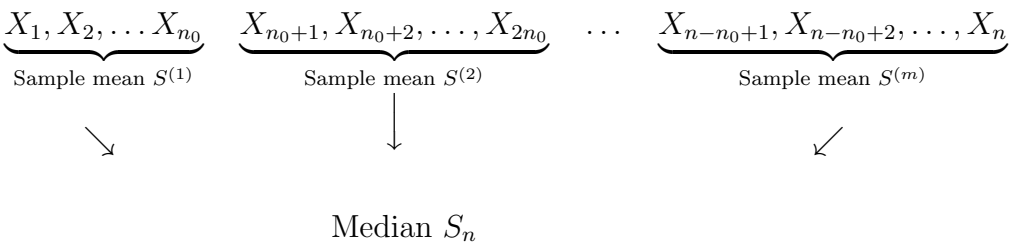
(a) (2 points) Let $S_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Use Chebyshev's inequality to show that $n = \frac{\sigma^2}{\delta\epsilon^2}$ samples are sufficient for $|S_n - \mu| \leq \epsilon$ with probability at least $1 - \delta$.

(b) (3 points) Now assume that each X_i is bounded between a and b. Use Hoeffding's inequality to compute the number of samples n sufficient for $|S_n - \mu| \leq \epsilon$ with probability at least $1 - \delta$. In particular, show that the dependence of n on δ is $O(\log(1/\delta))$.

(c) (2 points) Suppose we can't assume that each X_i is bounded. In that case, we can no longer apply Hoeffding's inequality, and can only use Chebyshev's inequality. This is a problem if we require a very high confidence level: in this part, you'll show why.

For this part only, assume that $\sigma^2 = 1$, $a = -1, b = 1$, and $\epsilon = 0.1$. Make a plot that shows, for particular values of δ, the number of samples n required based on your answers from parts (a) and (b). Your plot should show a range of ten δ values between 1/2 and 1/1000, using `np.geomspace(1/2,1/1000,10)`, and should be shown on a log-log scale. What do you observe?

To overcome this problem, we'll replace the sample mean with another estimator, and construct bounded random variables that will help us reason about the new estimator. To construct this estimator, we'll start by considering m groups of X_i, each with fixed size n_0. We'll compute the sample mean for each group, and call these sample means $S^{(1)}, \ldots, S^{(m)}$. Then, we'll use the median of all these group means as our estimate for the mean. The diagram below summarizes our approach.



We do this because even though one such sample mean $S^{(i)}$ might be far from the true mean μ, we hope (and will show) that the median of all of them is more likely to be close to the true mean μ.

(d) (2 points) Fix a sample size $n_0 = \lceil \frac{4\sigma^2}{\epsilon^2} \rceil$. For each of the group means i, we define a binary random variable Z_i:

$$Z_i = 1(|S^{(i)} - \mu| \geq \epsilon).$$

In other words, Z_i is 1 if the corresponding group mean is close to the true mean μ (within ϵ), and 0 otherwise.

Show that $\mathbb{E}[Z_i] \leq 1/4$.

Hint: Z_i is a Bernoulli random variable.

(e) (2 points) We set $S_{Med} := \text{Median}(\{S^{(1)}, \ldots, S^{(m)}\})$. This is called the *median-of-means estimator*. Explain in words why having $|S_{Med} - \mu| \geq \epsilon$ implies that $\sum_{i=1}^{m} Z_i \geq \frac{m}{2}$.

Hint: If y is the median of m numbers, it means that $\lceil m/2 \rceil$ of the numbers are greater than or equal to y, and similarly $\lceil m/2 \rceil$ of the numbers are less than or equal to y.

(f) (2 points) By taking probabilities, part (e) implies

$$\mathbb{P}(|S_{Med} - \mu| \geq \epsilon) \leq \mathbb{P}\left(\frac{1}{m}\sum_{i=1}^{m} Z_i \geq \frac{1}{2}\right).$$

If we combine this fact with the result of (d), we can show that

$$\mathbb{P}(|S_{Med} - \mu| \geq \epsilon) \leq \mathbb{P}\left(\frac{1}{m}\sum_{i=1}^{m} \left(Z_i - \mathbb{E}[Z_i]\right) \geq \frac{1}{4}\right).$$

Now use Hoeffding's inequality to compute what number m is sufficient to ensure that $|S_{Med} - \mu| \leq \epsilon$ with probability at least $1 - \delta$. What is the final number of samples of X required?

2 Simulation Study of Bandit Algorithms

In this problem, we evaluate the performance of three algorithms for the multi-armed bandit problem. The general protocol for the multi-armed bandit problem with K arms and n rounds is as follows: in each round $t = 1, \ldots, n$ the algorithm chooses an arm $A_t \in \{1, \ldots, K\}$ and then observes reward r_t for the chosen arm. The bandit algorithm specifies how to choose the arm A_t based on what rewards have been observed so far. In this problem, we consider a multi-armed bandit $K = 2$ arms, $n = 50$ rounds, and where the reward at time t is $r_t \sim \mathcal{N}(A_t - 1, 1)$, i.e. $\mathcal{N}(0, 1)$ for arm 1 and $\mathcal{N}(1, 1)$ for arm 2.

(a) (4 points) Consider the multi-armed bandit where the arm $A_t \in \{1, 2\}$ is chosen according to the explore-then-commit algorithm (below) with $c = 4$. Let $G_n = \sum_{t=1}^{n} r_t$ denote the total reward after $n = 50$ iterations. Simulate the random variable G_n a total of $B = 200$ times and save the values $G_n^{(b)}$, $b = 1, \ldots, B$ in a list. Report the expected value $\frac{1}{B} \sum_{b=1}^{B} G_n^{(b)}$ of the total reward and plot a normalized histogram of the rewards.

Algorithm 1 Explore-then-Commit Algorithm

input: Number of initial pulls c per arm
for $t = 1, \ldots, cK$: **do**
  Choose arm $A_t = (t \bmod K) + 1$
end
Let $\hat{A} \in \{1, \ldots, K\}$ denote the arm with the highest average reward so far.
for $t = cK + 1, cK + 2, \ldots, n$: **do**
  Choose arm $A_t = \hat{A}$
end

(b) (4 points) Consider the multi-armed bandit where the arm $A_t \in \{1, 2\}$ is chosen according to the UCB algorithm (below) with $c = 4$. Repeat the simulation in Part (a) using the UCB algorithm, again reporting the expected total reward and the histogram of $G_n^{(b)}$ after $n = 50$ rounds. How does the reward G_n compare to your results from part (a)? Compare both the average and the overall distribution.
Note: if $T_A(t)$ denote the number of times arm A has been chosen (before time t) and $\hat{\mu}_{A,t}$ is the average reward from choosing arm A (up to time t), then use the upper confidence bound $\hat{\mu}_{A,T_A(t-1)} + \sqrt{\frac{2\log(20)}{T_A(t-1)}}$. Note also that this algorithm is slightly different than the one used in lab and lecture as we are using an initial exploration phase.

Algorithm 2 UCB Algorithm

input: Number of initial pulls c per arm
for $t = 1, \ldots, cK$: **do**
| Choose arm $A_t = (t \bmod K) + 1$
end
for $t = cK + 1, cK + 2 \ldots$: **do**
| Choose arm A_t with the highest upper confidence bound so far.
end

QUESTION 1

1a) Chebyshev's inequality: for any random variable Z with mean μ and variance σ^2. Then, for any $t > 0$

$$P(|Z-\mu| \geq t) \leq \frac{\text{Var}(Z)}{t^2}. \text{ To fin}$$

To find first n which satisfies this, let $\delta(|Z-\mu| \geq t) = \frac{\text{Var}(Z)}{t^2}$.

Therefore, by Chebyshev:

If $S_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, then $E(S_n) = \mu$ and $SD(S_n) = \frac{\sigma}{\sqrt{n}}$

$$P(|S_n-\mu| \geq \varepsilon) = \frac{\sigma^2/n}{\varepsilon^2}$$

$$P(|S_n-\mu| \leq \varepsilon) = 1 - \frac{\sigma^2}{n\varepsilon^2}$$

$$P(|S_n-\mu| \leq \varepsilon) = 1 - \frac{\sigma^2}{n\varepsilon^2}$$

$$1 - \delta = 1 - \frac{\sigma^2}{n\varepsilon^2}$$

$$-\delta = -\frac{\sigma^2}{n\varepsilon^2}$$

$$\delta = \frac{\sigma^2}{n\varepsilon^2}$$

$$n = \frac{\sigma^2}{\delta\varepsilon^2}$$

Therefore, $\boxed{n = \frac{\sigma^2}{\delta\varepsilon^2}}$ samples are sufficient for $|S_n-\mu| \leq \varepsilon$ w/ probability at least $1-\delta$.

1b) Hoeffding's inequality: Let $X_1, \ldots, X_n$ be i.i.d random variables bound between a and b. Then:

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}\left(X_i - E(X_i)\right)\right| \geq t\right) \leq 2 \cdot \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

To find the first n which satisfies this, let:

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}\left(X_i - E(X_i)\right)\right| \geq t\right) = 2 \cdot \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

Here,

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{1}{n}\sum_{i=1}^{n} E(X_i)\right| \geq \varepsilon\right) = 2 \cdot \exp\left(\frac{-2n\varepsilon^2}{(b-a)^2}\right)$$

$$P\left(\left|S_n - \frac{\mu}{n}\right| \geq \varepsilon\right) = 2 \cdot \exp\left(\frac{-2n\varepsilon^2}{(b-a)^2}\right)$$

$$P\left(\left|S_n - \frac{\mu}{n}\right| \leq \varepsilon\right) = 1 - 2 \cdot \exp\left(\frac{-2n\varepsilon^2}{(b-a)^2}\right)$$

$$1 - \delta = -2 \cdot \exp\left(\frac{-2n\varepsilon^2}{(b-a)^2}\right)$$

$$-\delta = -2 \cdot \exp\left(\frac{-2n\varepsilon^2}{(b-a)^2}\right)$$

$$\delta = 2 \cdot \exp\left(\frac{-2n\varepsilon^2}{(b-a)^2}\right)$$

$$\log\left(\frac{\delta}{2}\right) = \frac{-2n\varepsilon^2}{(b-a)^2}$$

$$\frac{(b-a)^2 \cdot \log(\delta/2)}{-2\varepsilon^2} = n$$

$$n = \frac{(b-a)^2 \log(2/\delta)}{2\varepsilon^2} \quad \therefore \; n \text{ depends on } \delta \text{ by } O\left(\log\left(\tfrac{1}{\delta}\right)\right)$$

1c)

Part c

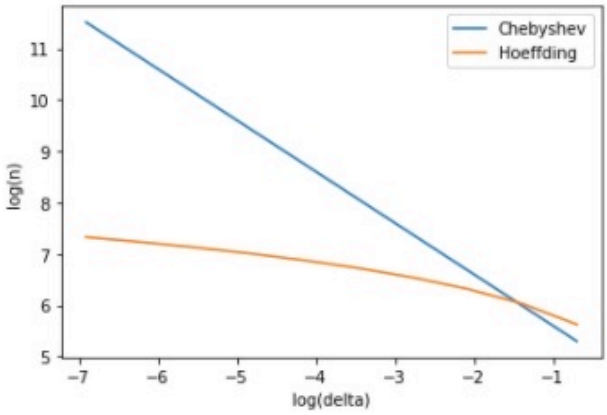
```
sigma = 1
a = -1
b = 1
epsilon = 0.1
```

```
deltas = np.geomspace(1/2,1/1000,10)
deltas
```

```
array([0.5       , 0.25065966, 0.12566053, 0.06299605, 0.03158114,
       0.01583223, 0.00793701, 0.00397897, 0.00199474, 0.001     ])
```

```
n_a = (sigma**2)/(deltas * epsilon**2)
n_b = ((b-a)**2)*(np.log(2/deltas))/(2*epsilon**2)
```

```
plt.plot(np.log(deltas), np.log(n_a), label='Chebyshev');
plt.plot(np.log(deltas), np.log(n_b), label='Hoeffding');
plt.xlabel('log(delta)');
plt.ylabel('log(n)');
plt.legend();
```



d) If Z_i is a Bernoulli random variable, $E(Z_i) = P(Z_i)$

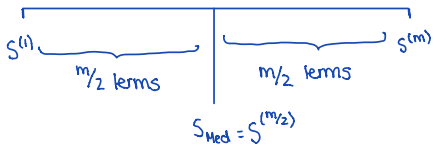
$$E[Z_i] = E[\mathbf{1}(|S^{(i)} - \mu| \geq \varepsilon)]$$

$$= P(|S^{(i)} - \mu| \geq \varepsilon)$$ using Chebyshev

$$\leq \frac{\sigma^2/n_0}{\varepsilon^2} = \frac{\sigma^2}{\frac{4\sigma^2}{\varepsilon^2}} \cdot \frac{1}{\varepsilon^2} = \frac{1}{4}$$

$$\therefore \boxed{E[Z_i] \leq \frac{1}{4}}$$

e) S_{med} can be visualized as follows:



If S_{med} is NOT within ε of the true mean μ...

① All $S^{(m/2+1)}, \ldots, S^{(m)}$ are also not within ε of the true mean μ

② $S^{(1)}, S^{(2)}, \ldots, S^{(m/2)}$ may or may not be within ε of the true mean μ

Similarly, if S_{med} is within ε of the true mean μ...

① All $S^{(1)}, S^{(2)}, \ldots, S^{(m/2)}$ are within ε of the true mean μ

② $S^{(m/2+1)}, \ldots, S^{(m)}$ may or may not be within ε of the true mean μ

Therefore, this implies that the minimum number of means $S^{(i)}$ that are close to the true mean is $m/2$.

We know that the sum of the indicator variable Z_i denotes the number of $S^{(i)}$ that are close to the true mean, hence $\boxed{\sum_{i=1}^{m} Z_i \geq \frac{m}{2}}$.

f) Hoeffding's inequality: Let $X_1, \ldots, X_n$ be i.i.d random variables bound between a and b. Then:

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - E(X_i))\right| \geq t\right) \leq 2 \cdot \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

(no need for 2 sided bound here,

We know: $P(|S_{Med} - \mu| \geq \varepsilon) \leq P\left(\frac{1}{m}\sum_{i=1}^{m}(Z_i - E(Z_i)) \geq \frac{1}{4}\right) \leq \exp\left(\frac{-2m(1/4)^2}{(1-0)^2}\right)$ since Z_i is bounded)

To find the first n which satisfies this, let:

$$P(|S_{Med} - \mu| \geq \varepsilon) = \cdot \exp\left(-2m(1/4)^2\right)$$

$$1 - P(|S_{Med} - \mu| \leq \varepsilon) = \cdot \exp\left(-2m(1/4)^2\right)$$

$$1 - (1 - \delta) = \cdot \exp(-m \cdot 1/8)$$

$$\delta = \exp(-m \cdot 1/8)$$

$$\log(\delta) = -m \cdot \frac{1}{8}$$

$$8 \log\left(\frac{1}{\delta}\right) = m$$

∴ Therefore, $\boxed{m = 8 \log\left(\frac{1}{\delta}\right)}$ samples are sufficient for $|S_n - \mu| \leq \varepsilon$ w/ probability at least $1 - \delta$.

QUESTION 2

2a)

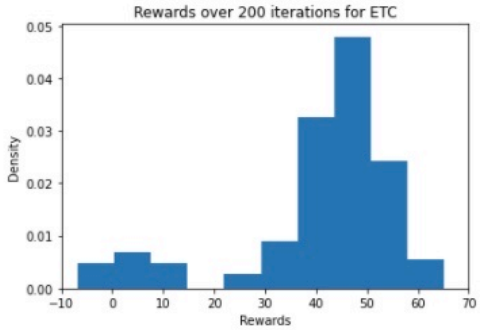
Part a

```
K = 2
n = 50
```

```
def part_a(c):
    rewards = []
    reward_arm1 = []
    reward_arm2 = []
    for t in range(1, c*K+1):
        # choosing arm
        arm = (t%K)+1

        # adding rewards
        if (arm == 1):
            r = np.random.normal(0, 1)
            reward_arm1.append(r)
            rewards.append(r)
        else:
            r = np.random.normal(1, 1)
            reward_arm2.append(r)
            rewards.append(r)

    if (np.mean(reward_arm1) > np.mean(reward_arm2)):
        best_arm = 1
    else:
        best_arm = 2

    for t in range(c*K+1, n+1):
        # choosing arm
        arm = best_arm

        # adding rewards
        if (arm == 1):
            r = np.random.normal(0, 1)
            reward_arm1.append(r)
            rewards.append(r)
        else:
            r = np.random.normal(1, 1)
            reward_arm2.append(r)
            rewards.append(r)

    return rewards
```

```
all_g = []

for i in np.arange(200):
    rewards = part_a(4)
    all_g.append(sum(rewards))
```

```
plt.hist(all_g, density=True);
plt.xlim(-10, 70);
plt.xlabel('Rewards');
plt.ylabel('Density');
plt.title('Rewards over 200 iterations for ETC');
```




```
np.mean(all_g)
```

```
40.76798597939278
```

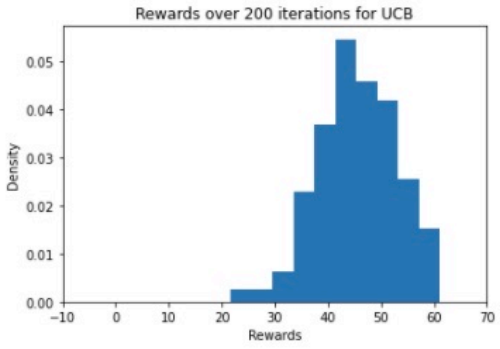
Avg reward ≈ 40.77

26) **Part b**

```
def part_b(c):

    rewards = []
    reward_arm1 = []
    reward_arm2 = []

    for t in range(1, c*K+1):
        # choosing arm
        arm = (t%K)+1

        # adding rewards
        if (arm == 1):
            r = np.random.normal(0, 1)
            reward_arm1.append(r)
            rewards.append(r)
        else:
            r = np.random.normal(1, 1)
            reward_arm2.append(r)
            rewards.append(r)

    for t in range(c*K+1, n+1):
        # choosing arm
        arm_1_cb = np.mean(reward_arm1) + np.sqrt(2*np.log(20)/len(reward_arm1))
        arm_2_cb = np.mean(reward_arm2) + np.sqrt(2*np.log(20)/len(reward_arm2))

        if (arm_1_cb > arm_2_cb):
            arm = 1
        else:
            arm = 2

        # adding rewards
        if (arm == 1):
            r = np.random.normal(0, 1)
            reward_arm1.append(r)
            rewards.append(r)
        else:
            r = np.random.normal(1, 1)
            reward_arm2.append(r)
            rewards.append(r)

    return rewards
```

```
all_g = []

for i in np.arange(200):
    rewards = part_b(4)
    all_g.append(sum(rewards))
```

```
plt.hist(all_g, density=True);
plt.xlim(-10, 70);
plt.xlabel('Rewards');
plt.ylabel('Density');
plt.title('Rewards over 200 iterations for UCB');
```




```
np.mean(all_g)
```

```
45.70216461420108
```

Avg reward ≈ 45.70

The average reward for ETC is ~40.77 (40-43 range), while the average reward for UCB is ~45.70 (43-46 range). Hence, the average reward for UCB is higher than the average reward for ETC.

In terms of the distribution, the UCB distribution seems to have a smaller range of X-axis values i.e. it does not have a left tail. On the other hand, the ETC distribution has a left tail and a larger range of X-axis values, with occassional gaps in the distribution. Hence, the ETC distribution seems to be more left-skewed while the UCB distribution looks more normal.