Statistical Inference

HW Author: Mohammad Arabzadeh , Nikoo Paknia
Instructor: Mohammadreza A. Dehaqani

Fall 2024

Homework 1

- If you have any questions about the homework, don't hesitate to drop an email to the HW Authors.
- Feel free to use the class group to ask questions — our TA team will do their best to help out!
- Please consult the course page for important information on submission guidelines and delay policies to ensure your homework is turned in correctly and on time.
- Please note that for computing questions, a major part of your grade is based on analyzing your results, so be sure to include explanations along with your code.
- This course aims to equip you with the skills to tackle all problems in this domain and encourages you to engage in independent research. Utilize your learnings to extend beyond the classroom teachings where necessary.
- As we mentioned in class, you'll have a quick (5 minute) in-person (or virtual!) hand-in session to help us check your understanding of the work you've submitted. For each assignment, about 25 students will be randomly chosen by an algorithm designed to ensure fairness. This algorithm will make sure you only present around 2 times during the term to keep things stress-free. However, if we notice inconsistencies between your work and what you present, the algorithm will adjust, increasing the chances you'll be selected again. Think of it as a dynamic process that adapts based on your performance—ensuring everyone gets a fair shot!

Problem 1: Visa

Consider a PHD or Master's student finishing their degree and going in OPT (a period which the student is allowed to work in the US). The student can be in this status for a year, and if their field is in the STEM subset (Science, Tech, Engineering, Mathematics) they can extend this period by another 2 years (3 in general).

While on OPT, students tend to get the H1B working visa, which is a stepping stone for a green card. The H1B visa has a lottery system which is held annually, hence, a STEM student can apply for it 3 times. The lottery system works like this: there are 85,000 visas available, and the number of applicants is usually twice or thrice this number. 65,000 visas are given to all applicants and the remaining 20,000 are reserved for applicants with a Master's degree or higher. So a Master's or PHD student gets 2 shots at the visa lottery, once in the 65,000 pool an another in the 20,000 pool.

(a) Suppose that the student of our interest has finished their degree in 2021 and is now in the beginning of their OPT period. They are planning to apply for the H1B visa in 2021 and 2022 and 2023, potentially if they are not granted. If applicants are 308,613 in 2021 (121,000 graduate students), 483,927 in 2022 (127,600 graduate students), 780,884 in 2023 (314,000 graduate students) and everybody gets an equal chance to win, What is the probability of getting the visa in 3 years?

(b) Let's say that the student is in a **non-STEM** field. What is the probability of winning? Do you think that a non-STEM student can rely only on getting the H1B visa (to get to green-card), given that they have only 1 shot and if they lose they have to leave the country?

(c) Back to first scenario (a), now assume along the way the student can get married to a US citizen. If they do, they can apply for a green card. The probability of getting married in each year is 15 percent. What is the probability of getting a green card in 3 years?

Problem 2: Independence (Optional)

(a) Let N_1 and N_2 be independent random variables following Poisson distributions with parameters λ_1 and λ_2. Show that the distribution of $N = N_1 + N_2$ is Poisson with parameter $\lambda_1 + \lambda_2$.
(b) Let T_1 and T_2 be independent exponential random variables with parameters λ_1 and λ_2. Find the density function of $T_1 + T_2$.

Problem 3: Mean, Median, and Mode

(a) Show that in unimodel symmetric distributions, the mean, median, and mode are equal.
(b) See if the statement in (a) holds when the distribution is symmetric but multimodal; if it doesn't, give a graphical example.

Problem 4: A geometric point of view

A point is chosen randomly in the interior of an ellipse:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

(a) Find the marginal densities of the x- and y-coordinates of the point.
(b) What is the joint density $f_{X,Y}(x, y)$ of the point within the ellipse?
(c) Compute the probability that the x-coordinate is within half of its maximum possible value, i.e., find:

$$P\left(|x| \leq \frac{a}{2}\right)$$

Problem 5: Inequalities

(a) Let X be a non-negative random variable with $\mathbb{E}[X] = 5$.

1. Use the Markov inequality to find an upper bound for $P(X \geq 10)$.

2. Explain why the Markov inequality can only provide an upper bound.

3. Discuss a situation where this bound might be loose or not very informative.

(b) Suppose X is a random variable with $\mathbb{E}[X] = 20$ and $\text{Var}(X) = 36$.

1. Use the Chebyshev inequality to find an upper bound for $P(|X - 20| \geq 12)$.

2. Explain the significance of the choice of k in your calculation.

3. If you were to draw a distribution of X, how might this bound compare to the actual probability of $|X - 20| \geq 12$?

(c) Let X and Y be random variables with $\mathbb{E}[X] = 2$, $\mathbb{E}[Y] = 3$, $\mathbb{E}[X^2] = 8$, and $\mathbb{E}[Y^2] = 12$. Use the Cauchy-Schwarz inequality to find an upper bound for $\mathbb{E}[XY]$.

1. Calculate the upper bound using the values provided.

2. Discuss a potential scenario where the actual value of $\mathbb{E}[XY]$ may be significantly lower than this upper bound.

Problem 6: Sort, sort, sort

Simulate this problem with python, preferebly in an ipynb file.

Consider the 4 sotring algorithms: bubble sort, insertion sort, merge sort, and quick sort.

randomly generate a list of 1000 integers (make sure it's shuffled) and sort them using the above algorithms, each 100 times (reshuffle each time). each time record the time it took to sort the list.

(a) Plot the duration distribution for each method. Include the mean, median, standard deviation in the plot.

Sort the algorithms based on the mean duration.

Looking at the standard deviation, which algorithm is the most stable? Which one is the most volatile? Why do you think so?

(b) Are any of the distributions skewed?

Calculate the Kurtosis of the distributions. Which algorithm has the least kurtosis? What does this say about the sorting-algorithm's consistency in terms of duration?

(c) Draw the boxplots for the durations of each algorithm.

Which algorithm has the most outliers?

Note: Feel free to use built-in sorting functions—no need to reinvent the wheel! Let's focus on the fun part: analyzing the results!

Problem 7: Cookie Monster

Simulate this problem with python, preferebly in a ipynb file

Salt and pepper is a type of noise sometimes seen on digital images, similar to the one in **Figure 1**: each noise pixel is either complete white or complete black.

Figure 1: Example of a Salt-and-pepper noisy image

now an image (**Figure 2**) consider 3 types noise: uniform, gaussian and exponential.

You're going to apply these noises on the image.

(a) For a noise ratio of %10, %15, %20, %30, %40, %50 of all pixels apply the uniform noise across the whole image, apply the gaussian noise with a mean and standard deviation that are randomly sampled (uniformly), and the exponential noise with $\lambda = 1$.

Plot the results of the above in three rows of 6 columns (for each noise ratio).

Note: remember, the noise color may only be black, or white.

(b) Alright, let's say that the face of the cookie monster is the region of interest. if at least the 20% of the pixels of his face are corrupted by noise, consider it as corrupted.

For each of the noise ratios, %20, %30, %40, %50, and the three types of noise, apply the noise to the image 100 times and count the times that the face is corrupted.

Report the average count of the corrupted faces for each noise ratio and type of noise.

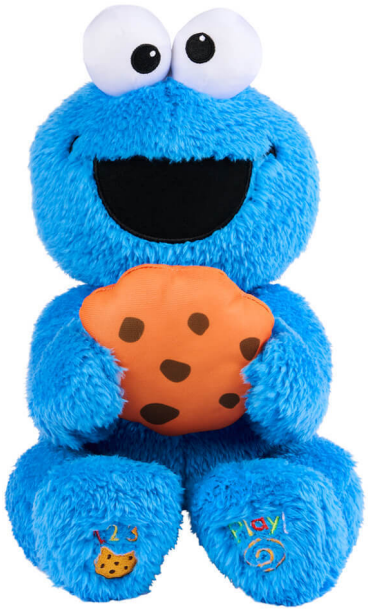

Figure 2: Subject image (Download)

Are the power of the noise types the same? If not, which one is the most detrimental? Why do you think so?

(c) One of the effective ways of salt-and-pepper noise removal is the median filter(don't freak out, it's just a filter that replaces the pixel value with the median of the neighboring pixels, you can use a library for this). apply the noise ratios of the section (b) with the 3 noise types (just once, not 100 times), and then apply the median filter to the image.

Plot the results of the above in three rows of 4 columns (for each noise ratio). Do the images look good now? depends? why ,if so?

Problem 8: Conditional Independence

In the throw of two independent dice, the events that the first die shows a 6 and the second die shows a 5 are independent.

(a) Prove the above statement.

(b) Examine whether these two events remain independent given that the sum of the two dice is greater than 10.

(c) Consider three different events involving a single die, such that event 1 is a subset of event 2, and event 2 is a subset of event 3. Analyze which of these events can act as a condition for the other two events to become independent.

(d) Each of the two responses above represents a basic form of conditional independence analysis in graphical models. What other fundamental case do we have to discuss the independence or dependence of two events conditioned on another event?

Problem 9: Limit Distributions

Using the Poisson distribution, first prove that it satisfies the properties of a valid probability function. Then, show that the Poisson distribution can be derived as a special case of the Binomial distribution. (**Hint:** Consider the limit as $n \to \infty$, $p \to 0$, and $\lambda = np$.)

Problem 10: Memorylessness

If X is a geometric random variable, show that:

$$P(X > n + k - 1 \mid X > n - 1) = P(X > k)$$

Explain this equation based on the interpretation of the geometric distribution.

Problem 11: Transformation of Random Variables

Let $Y = g(X)$ and the PDF of X be $f_X(x)$. First, derive $f_Y(y)$ using the concept of the CDF, and then prove that:

$$E_X[y] = E_Y[y]$$

Problem 12: What are AutoEncoders doing here?

Autoencoders are a group of neural networks used for feature extraction. The autoencoder consists of an encoder and a decoder. The encoder is a function of input images. Let X represent input images drawn from $N(0, 1)$, and let the encoder function be $\text{En}(X)$. It is known that the latent space of this autoencoder, which compacts all image information, is a sample from $\exp(\text{En}(X))$. What is the average of the latent vector of these images?

Problem 13: Plants vs. Enemies

In a computer game, two agents are competing. The first agent has a static strategy and is called the "enemy". The second agent is called the "farmer". The game's ground is a triangular farm, where the farmer must choose a fraction of the land to maximize farming while evading the enemy. The farm is represented as the area under the line $X + Y = 1$.

The enemy's movement along the x-axis follows $U[0, 1]$, and its movement along the y-axis follows $U[0, 1-X]$. The farmer must divide its farming area using either a horizontal or vertical line. What would be the optimal placement of this line to ensure the probability of encountering the enemy is minimized, while maximizing the farming area?

Problem 14: Random Genes (Optional)

A genetic engineer is investigating the effect of a chemical on a sequence of 10 genes responsible for a specific disease. It is observed that the genes change in this substance with frequencies drawn from a uniform distribution, and the robustness to change increases from gene 1 to gene 10.

Using Python or R, simulate this scenario by sampling data from the uniform distribution 100 times for 10 genes. First, plot the distribution of change frequency for gene 5, analyze the resulting distribution, and determine its type. Then, compare this distribution with the simulation results.