Indian Institute of Technology Mandi

भारतीय प्रौद्योगिकी संस्थान मण्डी

IC252-Data Science 2

Assignment- 05

General instructions:

- Utilize Python programming language for implementation.
- Ensure the program is well-documented to enhance comprehension.
- Employ functions and loops for efficient code organization.
- Implement error handling to manage invalid inputs or unexpected scenarios.
- Optimize the code for performance and readability where applicable.

1. For the given file “timeMachine.txt”, do the following:

   (a) Find the frequency (the number of occurrences of all the words) and plot a histogram of the **probability distribution** of it.

   (b) Find the probability of occurrence of all ordered pairs of letters. Print the probabilities of the top 10 ordered pairs (Ignore punctuation and white spaces).

   (c) Repeat *b* part while ignoring the punctuation but considering the white spaces.

2. The Community Mobility Reports provided by Google show movement trends by region across different categories of places. These reports are created with aggregated, anonymized sets of data from users who have turned on the Location History setting, which is off by default. The data shows how visitors to (or time spent in) categorized places change compared to our baseline days. A baseline day represents a normal value for that day of the week. The baseline day is the median value from the 5-week period Jan 3 – Feb 6, 2020. The baseline isn’t a single value for each region-category—it’s 7 individual values. The same number of visitors on 2 different days of the week result in different percentage changes.

   You are given a dataset containing India’s mobility data in 2021. You are expected to do the following for Mumbai City:

   (a) Consider the data given for mobility to be probabilistic in nature. Let’s assume that the data given by Google is probabilistic in nature, and each mobility has a certain probability associated with it. So, on a particular day for a region, we compute the expected mobility using the formula:

   $$E[Mobility] = \sum P(Mobility) \times Mobility$$

   (b) Consider the following distribution during the lockdown period (1/04/2021 to 20/05/2021):

Grocery/Pharma	Retail	Transport	Parks	Residential	Workplace
$p = 0.2$	$p = 0.2$	$p = 0.05$	$p = 0.02$	$p = 0.5$	$p = 0.03$

   Plot the Expected Mobility along with all the other mobilities on the same graph to compare the mobilities.

   (c) Find the Error between the expected mobility and other mobilities using the following error measures,

      i. Root Mean Squared Error.

      ii. Mean Absolute Error.

      iii. KL Divergence.

3. A random walk, sometimes known as a drunkard’s walk, is a random process that describes a path that consists of a succession of random steps on some mathematical space. An elementary example of a random walk is the random walk on the integer number line $\mathbb{Z}$ starts at 0 and moves +1 or −1 at each step with equal probability.

   (a) Simulate an elementary(equiprobable) $1D$ random walk experiment 10000 times. Let the number of steps $n = (100, 1000, 10000)$. Plot the probability distribution of the final locations on the number line.

   (b) Let the probability of going right = 0.6 and the probability of going left = 0.4. Repeat part *a* for the same.

4. Create a Python class **BivariateGaussianDistribution** for a continuous random variable (X, Y). Include methods for:

   - $init(self, mean\_x, mean\_y, var\_x, var\_y, cov)$: Initialize with a function that calculates the joint PDF for given values of X and Y.
   - $calculate\_pdf(self, x, y)$: This method should return the joint PDF value for input x and y.
   - $marginal\_pdf\_x(self, x)$: This method should calculate and return the marginal PDF of X for a given value of x (integrate the joint PDF over all possible Y values). (Bonus: Implement $marginal\_pdf\_y$ for Y)
   - $plot\_pdf\_contour(self)$: This method should create a contour plot visualizing the joint PDF using libraries like matplotlib.

   A snippet of the code for above problem is:

```
class BivariateGaussian(BivariateDistribution):
def __init__(self, mean_x, mean_y, var_x, var_y, cov):
# ... (initialize parameters for a bivariate Gaussian distribution)

# Example usage
distribution = BivariateGaussian(...)
pdf_value = distribution.calculate_pdf(1.5, 2.0)
marginal_pdf_x = distribution.marginal_pdf_x(1.0)
distribution.plot_pdf_contour()
```

5. Write a python program to estimate the value of π using Monte Carlo simulation. Generate animation of the simulation where estimates of π converges with an increase in the number of samples. Plot the Monte Carlo estimate of π where x-axis represent number of sample (1 to 3000) and y-axis represent estimate of π.

6. Evaluate the integral

$$f(x) = \int_0^{\pi} \sin^4(3x)dx$$

   using a Monte Carlo approach. Generate animation of the simulation where estimates of the integral converges with an increase in the number of samples. Plot the Monte Carlo estimate of integral where $x-$axis represent number of sample (1 to 2000) and $y-$axis represent estimate of $\int_0^1 f(x)dx$. Also, calculate the exact value of the integration and compare it with the simulated one.

7. (a) Calculate the entropy of a fair coin. Now suppose that the coin is biased, i.e., probability of head is not equal to 0.5. Plot the entropy curve where $x-$axis represents the probability of head and $y-$axis is the corresponding entropy.

   (b) Generate and plot two Gaussian distributions with different mean and variance. Calculate the KL divergence and cross-entropy between these two distributions. Repeat this experiment for different mean and variance, and observe the value of KL divergence and cross-entropy when these two distributions:

      i. overlap each other,
      ii. partially overlap,
      iii. do not overlap.

8. (a) Simulate a random number generator for the following distributions:

      i. Uniform distribution
      ii. Normal distribution
      iii. Truncated exponential distribution

   Generate a sample dataset of 1000 points for each case. Plot the histogram of the samples and the density function of the given distributions in a single subplot.

   (b) Let

$$f_X = \frac{1}{40}(2x + 3), \quad 0 < x < 5$$

   be a density function. Generate a random number simulator for this density function and sample 1000 random draws. Plot the graph of given density function and histogram plot of the drawn samples in a single figure.