(Fall 2020) 1175 Probability **(Due: 2020/12/23 in class)**

Homework 4: Multivariate Normal & Sum of Independent Random Variables

Submission Guidelines: For Problems 1 and 2, please turn in your write-up after the class on 12/23. For Problems 3 and 4, please briefly summarize your observation in a technical report (no more than 2 pages), compress your report and source code into one .zip file, and submit the compressed file via E3.

Problem 1 (Strong Law of Large Numbers) (15 points)

Consider two sequences of random variables $X_1, X_2, \cdots$ and $Y_1, Y_2, \cdots$ defined on the same sample space. Suppose that X_n converges to a and Y_n converges to b, almost surely. Show that $X_n + Y_n$ converges to $a + b$, almost surely. (Hint: Consider two events A, B defined as $A = \{\omega : X_n(\omega) \text{ does not converge to } a\}$ and $B = \{\omega : Y_n(\omega) \text{ does not converge to } b\}$)

Problem 2 (Concentration Inequalities) (10+15=25 points)

(a) Suppose that X is a random variable with $E[X] = \mathrm{Var}[X] = \mu$. What does Chebyshev's inequality say about $P(X > \pi\mu)$?

(b) Imagine we have an algorithm for solving some decision problem (e.g., is a given number a prime?). Suppose that the algorithm makes a decision at random and returns the correct answer with probability $\frac{1}{2} + \delta$, for some $\delta > 0$ (which is just a bit better than a random guess). To improve the performance, we run the algorithm N times and take the majority vote. Show that, for any $\epsilon \in (0, 1)$, the answer is correct with probability at least $1 - \varepsilon$, as long as $N \geq (1/2)\delta^{-2} \ln(\varepsilon^{-1})$. (Hint: For each i, define X_i to be a Bernoulli random variable for which $X_i = 1$ when the algorithm return the correct answer at the i-th trial. Under majority vote, we know that if the final answer is incorrect, then $X_1 + X_2 + \cdots + X_N \leq N/2$. Use the negative part of Hoeffding's inequality. This scheme is usually called "boosting randomized algorithms.")

Problem 3 (Multivariate Normal for Regression) (15+25=40 points)

One interesting application of multivariate normal (MVN) random variables is to solve regression tasks. In this problem, you will implement a simple MVN-based predictor that predicts the outputs of the testing queries based on the training data. Specifically, let $D_{train} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ be the training dataset and let $D_{test} = \{x_{N+1}, \cdots, x_{N+M}\}$ be the testing queries. The goal is to predict $\{y_{N+1}, \cdots, y_{N+M}\}$ that correspond to $\{x_{N+1}, \cdots, x_{N+M}\}$.

(a) As a prep work, please show the following property that we discussed in class: Let Z_1, Z_2 be a pair of bivariate normal random variable with mean μ_1, μ_2, variance σ_1^2, σ_2^2, and correlation coefficient ρ. Show that conditioned on that $Z_1 = z_1$, the conditional distribution of Z_2 is normal with mean $\mu_2 + \frac{\rho\sigma_2(z_1 - \mu_1)}{\sigma_1}$ and variance $(1 - \rho^2)\sigma_2^2$.

(b) The property in (a) can be extended to the multivariate normal case. Suppose that for every $k \in \{N + 1, \cdots, N + M\}$, $\{Y_1, \cdots, Y_N, Y_k\}$ is multivariate normal with mean vector $\mu = [\mu_1, \cdots, \mu_N, \mu_k]^\intercal$ and a $(N + 1) \times (N + 1)$ covariance matrix Σ, where the covariance between Y_i and Y_j (denoted by $\Sigma_{i,j}$) has the following form:

$$\Sigma_{i,j} = \sigma_f^2 \exp\big(-\frac{(x_i - x_j)^2}{2\ell^2}\big) + \sigma^2\delta_{i,j}, \forall i, j,$$

where σ_f is a scale factor, ℓ is called the lengthscale, σ^2 is some positive constant (usually called the noise parameter), and $\delta_{i,j}$ is the delta function (i.e. $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ if $i \neq j$). Given that $Y_1 = y_1, \cdots, Y_N = y_N$, it can be shown that the conditional distribution of Y_k is normal with mean $K(x_k, x_{1:N})[K(x_{1:N}, x_{1:N}) + \sigma^2 I]^{-1} y_{1:N}$ and variance $K(x_k, x_k) - K(x_k for, x_{1:N})[K(x_{1:N}, x_{1:N}) + \sigma^2 I]^{-1} K(x_{1:N}, x_k)$, where

- $K(x_k, x_{1:N}) = [\Sigma_{k,1}, \cdots, \Sigma_{k,N}]$ is a $1 \times N$ vector.
- $K(x_{1:N}, x_{1:N})$ is an $N \times N$ matrix with the (i, j)-th entry equal to $\Sigma_{i,j}$.
- I is an identity matrix of size $N \times N$.

- $K(x_{1:N}, x_k)$ is the transpose of $K(x_k, x_{1:N})$.
- $y_{1:N} = [y_1, \cdots, y_N]^\intercal$ is an $N \times 1$ vector.

Based on the above conditional distribution, please write a program (e.g. in Python or MATLAB) to find the predictive distributions of the outputs of the test query points $\{x_{N+1}, \cdots, x_{N+M}\}$. What is the prediction result of the testing dataset under $\sigma_f = 1, \sigma = 0.1, \ell = 0.06$? What is the prediction result of the testing dataset if ℓ is set to be 0.2 instead?

Problem 4 (Monte Carlo Method for Integration) (15+15=30 points)

In probability research, one common task is to find the expected value of a function that depends on a random variable. If the random variable is continuous, then calculating such an expected value may involve a complex integral that has no simple closed-form expression. In this scenario, one work-around is to leverage Monte Carlo method to find an approximate answer.

(a) In this subproblem, you will be asked to tackle the following integral: Let $Z \sim \mathcal{N}(0, 1)$. Define another random variable $Y = \cos(Z) + \sin(2Z)$. Our goal is to find out the expected value of Y using Monte Carlo method. Specifically, let $z_1, z_2, \cdots, z_n$ be n independent numbers drawn from a standard normal distribution. Then,

$$E[Y] = \int \big(\cos(z) + \sin(2z)\big)\frac{1}{\sqrt{2\pi}}e^{-\frac{-z^2}{2}}dz \approx \frac{1}{n}\sum_{i=1}^{n}\big(\cos(z_i) + \sin(2z_i)\big).$$

Please write a short program (either in Python or MATLAB) to implement the above procedure. Suppose we set $n = 10^3$ and repeat the same estimation procedure for 20 times. What are the estimation results? What if we reconfigure $n = 10^5$ and again repeat the same estimation procedure for 20 times. Do you observe any differences in the estimation results?

(b) Define a closed region $A = \{(x, y) : x^2 + 2y^2 \le 1\}$. Our goal is to approximately compute the area of the region A (denoted by Δ_A) using Monte Carlo integration. To begin with, we may write Δ_A as

$$\Delta_A = \int_A dxdy.$$

To apply Monte Carlo method, we consider two independent random variables $X \sim \text{Unif}(-1, 1)$ and $Y \sim \text{Unif}(-1, 1)$. Then, the joint PDF of X and Y (denoted by $f_{XY}(x, y)$) is 1/4 for any $x \in [-1, 1], y \in [-1, 1]$ and is 0 elsewhere. Accordingly, we have

$$\Delta_A = 4 \cdot P((X, Y) \in A) = 4\int_{-1}^{1}\int_{-1}^{1}\mathbb{I}_{\{(x,y)\in A\}}f_{XY}(x, y)dxdy \approx 4 \cdot \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_{\{(x_i,y_i)\in A\}},$$

where each (x_i, y_i) is an independent sample from the joint distribution f_{XY} and $\mathbb{I}_{\{\cdot\}}$ is the indicator function. Therefore, Δ_A can be approximated by counting the number of random points that fall in the region A. Please write a short program (either in Python or MATLAB) to implement the above procedure. What are your estimates of Δ_A under $n = 10^3, 10^5$, and 10^7?