Lecture Summary: Continuous Random Variables - Colab Simulation

Source: Lecture 4.9.docx

Key Points

- **Purpose:**
  - Simulate continuous random variables in Python using Google Colab.
  - Explore Monte Carlo techniques for estimating probabilities and comparing theoretical and empirical results.
- **Packages Used:**
  - `NumPy`: For generating numerical data.
  - `scipy.stats`: For generating random samples from continuous distributions.
  - `matplotlib.pyplot`: For plotting histograms and densities.
- **Key Steps:**
  1. Import necessary libraries:

     ```
     import numpy as np
     from scipy.stats import uniform, expon, norm
     import matplotlib.pyplot as plt
     ```

  2. Generate samples from continuous distributions using `scipy.stats`:
     - Uniform distribution: `st.uniform.rvs(loc, scale, size)`.
     - Exponential distribution: `st.expon.rvs(scale, size)`.
     - Normal distribution: `st.norm.rvs(loc, scale, size)`.
  3. Plot histograms and overlay theoretical PDFs for comparison.
- **Examples and Results:**
  - **Uniform Distribution:**
    - Samples generated in range $[0, 3]$ with 10,000 values.
    - Histogram with bins overlaid by theoretical density $f_X(x) = \frac{1}{3}$ for $0 \leq x \leq 3$.
    - Visual comparison confirms empirical data approximates the theoretical PDF.
  - **Exponential Distribution:**
    - Parameter $\lambda = 2$, 10,000 samples generated.
    - Histogram plotted with theoretical density $f_X(x) = 2e^{-2x}$.
    - Density decays rapidly, matching the histogram.
  - **Normal Distribution:**

        - Mean $\mu = 0$, standard deviation $\sigma = 1$.
        - Histogram of samples overlaid with theoretical PDF.
        - Matches closely, showing the bell-shaped curve.

- **Practical Insights:**
    - Histograms provide a visual representation of data distribution and are useful for verifying the fit of a theoretical model.
    - Comparing histograms with theoretical PDFs helps identify suitable distributions for data modeling.
    - Adjusting bin sizes in histograms can improve visualization and interpretation.
- **Monte Carlo Simulations:**
    - Generate random samples to estimate probabilities and distribution properties.
    - Verify theoretical results through empirical data.
    - Useful for exploring large datasets where analytical solutions are complex.

Simplified Explanation

Objective: Simulate and analyze continuous random variables using Python.

Examples: 1. Uniform: Constant density across an interval. 2. Exponential: Rapidly decaying density. 3. Normal: Symmetric bell curve.

Tools: - Histograms to visualize data. - PDFs to verify theoretical distributions.

Key Takeaway: Simulation bridges theory and practice, offering insights into real-world data distributions.

Conclusion

In this lecture, we:

- Explored continuous random variable simulation in Colab.
- Visualized data using histograms and theoretical densities.
- Discussed the use of Monte Carlo techniques for empirical validation.

Simulation is a powerful tool for understanding and modeling continuous random variables, providing practical verification of theoretical concepts.