Project i: Basic Data Generation and Manipulation

Due Feb 12 by 11:59pm **Points** 10 **Available** until Feb 12 at 11:59pm

Note: This assignment will be updated regarding grading and programming language. The problems within it will not change, simply the submission format and grading details will be revised/included. Also note that Matlab may be accepted (dependent on the outcome of the programming language preference survey (https://goo.gl/forms/isbG7Ur7v6rsKHeh2)). These details will be added no later than Monday Feb 4th. If you'd like to get a head-start on it this weekend though please feel free-- again, the problems themselves will not change.

1 - Visualizing the Multivariate Gaussian (Normal) Probability Density Function

In this problem you will visualize the 2-dimensional Normal PDF by plotting the PDF itself as well as its contour map. The deliverables for this problem are 2 plots (a contour plot and a plot of the PDF itself) and your code.

- Generate a 2D Gaussian PDF with mean vector of $\boldsymbol{\mu} = [1, 1]^T$ and covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$
- Show a contour plot of this PDF over the range [-5,5] in both dimensions
- Show a surface plot of the PDF over the range [-5,5] in both dimensions
- Use this as an opportunity to gain intuition on how the mean vector and covariance matrix affect the normal distribution
  - Note that the covariance matrix must be symmetric and positive semi-definite
  - Explore various values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$
- Helpful functions (packages)
  - multivariate_normal (numpy/random)
  - contour and plot_surface (matplotlib/pyplot)

2 - Generate Synthetic Data for Classification

In this problem you will generate synthetic data from 2 different normal distributions, each representing one class of a binary classification problem. The deliverables for this problem are a plot of the synthetic data and your code.

- Generate 100 observations from a zero-mean 2-d normal distribution with covariance matrix $\Sigma = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$
- Generate 100 observations from a 2-d normal distribution with mean vector $\boldsymbol{\mu} = [1, 1]^T$ and covariance matrix $\Sigma = \begin{bmatrix} 1 & -0.1 \\ -0.1 & 1 \end{bmatrix}$
- Plot all 200 observations on one scatter plot, giving each group its own color (e.g., red and blue)
- Helpful functions (packages)
  - multivariate_normal (numpy/random)

- scatter (matplotlib/pyplot)

3 - Generate Synthetic Data for Regression

This problem is similar to Problem 2, however, now you will generate data to use for regression, i.e., instead of class labels (or colors) you now will have continuous target values. The deliverables for this problem are a plot of the synthetic data and your code.

- Generate 100 observations from a uniform distribution over the interval -5 to 5. These are your "inputs".
- Generate 100 ground-truth (no noise) target values using the function $y_i = \frac{1}{10}x_i^3 + 3$
- Add independently sampled zero-mean unit-variance Gaussian noise to each target value
- Plot all 100 noisy observations and ground-truth on the same scatter plot, color coding each with a unique color
- Helpful functions
  - uniform (numpy/random)
  - multivariate_normal (numpy/random)
  - normal (numpy/random)
  - randn (numpy/random)
  - scatter (matplotlib/pyplot)

4 - Standardizing Data

In this problem you will "standardize" data, meaning you will zero-center it and re-scale such that all dimensions have unit variance. The deliverables for this problem are plots of the data and your code.

- Generate 10 observations from a 2-d normal distribution with mean vector $\boldsymbol{\mu} = [2 \quad 5]^T$ and covariance matrix $\Sigma = \begin{bmatrix} 1 & -0.1 \\ -0.1 & 1 \end{bmatrix}$
- Stack all 10 observations in 10x2 matrix $\boldsymbol{X} = \begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \\ \vdots & \vdots \\ X_{10,1} & X_{10,2} \end{bmatrix}$
- Standardize the data by subtracting the sample mean from each dimension and dividing by the standard deviation. This will give a centered and scaled dataset of the following form $\hat{X} = \begin{bmatrix} \frac{X_{1,1}-\mu_1}{\sigma_1} & \frac{X_{1,2}-\mu_2}{\sigma_2} \\ \frac{X_{2,1}-\mu_1}{\sigma_1} & \frac{X_{2,2}-\mu_2}{\sigma_2} \\ \vdots & \vdots \\ \frac{X_{10,1}-\mu_1}{\sigma_1} & \frac{X_{10,2}-\mu_2}{\sigma_2} \end{bmatrix}$, where $\mu_i = \frac{1}{10}\sum_{j=1}^{10} X_{j,i}$ and $\sigma_i = \sqrt{\frac{1}{10}\sum_{j=1}^{10}(X_{j,i}-\mu_i)^2}$. Note the mean and standard deviations are computed for the *columns* of X. Further, use built in functions to calculate the mean and standard deviation (don't implement the sums shown above as-is).
- Create 2 scatter plots showing the original data X and the standardized data $\hat{X}$.

Submissions

Plots

Submit a single document (pdf) with screenshots of your plots. Make sure it is clear in the document what each image is. Indicate the question number for each image and a short (1 sentence) explanation of what it is.

Code

Submit a .py file for each question in the assignment. The code should be ready to run and it is expected that when it is run the plots in the aforementioned document are produced. Comment your code well.

For this project, you will submitting 5 files: 4 .py script files and 1 .pdf file.