CS 215: Data Analysis and Interpretation: Assignment

Instructor: Suyash P. Awate

Submission Instructions:

- IITB and CSE have zero tolerance to plagiarism.

- If you submit the solution to the assignment, you agree that every line of code and every line in the report is your (or your group's) own, and hasn't been copied from any other source offline or online.

- If you submit the solution to the assignment as a group and any member of the group is found to have committed plagiarism, then the full penalty will be applicable to every member of the group.

- For the sake of effective learning, if you submit the solution to the assignment as a group, then each member of the group agrees to have participated fully (100%) in performing every part of every question in the assignment.

- Submit your solution, i.e., (i) the code, (ii) the results, e.g., graphs or other data, and (iii) the report (in Adobe PDF format), for each question, through moodle. Put the code within the folder "code", all results in the folder "results", and the report in the folder "report". You may make subfolders for each question in the assignment.

- Submit a single zip file that contains the solution to each problem below in a separate folder.

- To get any possible partial credit for the code, ensure that the code is very well documented. The documentation should specify the reason / meaning for which every line of code is written.

- To get partial credit for the derivations, include all derivation steps in their full details.

- To avoid non-deterministic results in each program run, and to make your results reproducible while your assignment is being graded, use rng(seed) where seed is a fixed hard-coded integer in your code.

- While submitting the results of a question, use may use the "publish" feature in MATLAB.

- **5 points** for submission in the proper format.

- If you feel there is a typo in the question, please make suitable assumptions, consistent with those in the question, and proceed to solve the problem. Also, please let the TAs or the instructor know.

1. (20 points) Estimating π Using Simulation.

Read all instructions before you start.

Consider a bivariate random variable $X := (X_1, X_2)$, where X_1, X_2 are both independent and have a uniform distribution over $(-1, 1)$.

(a) (5 points) What is the probability that the random variable X takes values within a circle of radius 1 ? Derive the expression.

(b) (5 points) Use the previous result to estimate the value of π purely relying on simulation of X. Justify your approach in words.

(c) (5 points) Report the estimates of π using sample sizes $N = 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8$. Use the "single" datatype for the simulation. How will you write code to handle the situation when you desire to increase the accuracy of the estimate with a sample size as large as $N = 10^9$ or larger ? Does your code handle this case ?

(d) (5 points) How will you estimate the sample size M required to have the estimate of π within $[\pi - 0.01, \pi + 0.01]$ with 0.95 probability ? Assume the true value of π to be known, so that you know the interval exactly. Describe an algorithm and justify it. Compute this estimate for the sample size M and report it.

2. (20 points) Multivariate Gaussian.

Read all instructions before you start.

Generate N points (with N taking the values $10, 10^2, 10^3, 10^4, 10^5$) from a multivariate $2D$ Gaussian probability density function with mean $\mu = [1, 2]'$ and a covariance matrix C with the first row as $[1.6250, -1.9486]$ and the second row as $[-1.9486, 3.8750]$.

For this generation, you are only allowed to use the randn() and eig() functions in Matlab.

For each data sample of size N, compute the maximum likelihood (ML) estimates of the mean and the covariance matrix.

For this estimation, you are only allowed to use the sum() function in Matlab.

(a) (5 points) Describe and justify your method for generating sample points from the 2D Gaussian.

(b) (5 points) For each value of N, repeat the experiment 100 times, and plot a box plot of the error between the true mean μ and the ML estimate $\widehat{\mu}_N$ (which depends on N), where the error measure is $\| \mu - \widehat{\mu}_N \| \, / \, \| \mu \|$. Use a logarithmic scale on the horizontal axis, i.e., $\log_{10} N$.

(c) (5 points) For each value of N, repeat the experiment 100 times, and plot a box plot of the error between the true covariance C and the ML estimate $\widehat{C}_N$ (which depends on N), where the error measure is $\| C - \widehat{C}_N \|_{\text{Fro}} \, / \, \| C \|_{\text{Fro}}$. Use a logarithmic scale on the horizontal axis, i.e., $\log_{10} N$.

(d) (5 points) For each value of N, for a single data sample, within a single figure, plot the $2D$ scatter plot of the generated data and show the principal modes of variation of the data by plotting a line starting at the empirical mean and going a distance equal to the empirical eigen-value along a direction given by the empirical eigen-vector.

3. (15 points) PCA and Hyperplane Fitting.

   Read all instructions before you start.

   Consider the observed set of points of the form $(x, y) \in \mathbb{R}^2$ in the file "points2D_Set1.mat". Assume each observation (x, y) is drawn independently from the joint probability density function $P(X, Y)$ of random variables X and Y.

   For this question, you cannot use the functions mean(), cov(), and pca() in Matlab.

   • (5 points) How can principal component analysis (PCA) be used to best approximate a linear relationship between random variables X and Y. Describe the method clearly, using appropriate mathematical descriptions for clarity. Your description should be clear enough to lead to a programmable implementation.

   • (5 points) Show a scatter plot of the points. Overlay on the scatter plot, the graph of a line showing the linear relationship between Y and X.

   • (5 points) Repeat the same analysis for the set of points in "points2D_Set2.mat". Show a scatter plot of the points. Overlay on the scatter plot, the graph of a line showing the linear relationship between Y and X. Compared to the result on the other set of points, justify the quality of the approximation resulting in this question using logical arguments.

4. (25 points) Principal Component Analysis (PCA).

   Read all instructions before you start.

   Download the dataset comprising images of handwritten digits in `http://yann.lecun.com/exdb/mnist`; this has been downloaded in the folder "data" and stored as "mnist.mat".

   Each image is stored as a matrix (28×28) of numbers. You can visualize these images (or matrices) in Matlab using the functions imagesc() or imshow(). Use the Matlab command "axis equal" to use the same units on each axis of the image.

   For the following computations, make sure to convert (cast) the integer data type to a floating-point type. For this question, you cannot use the functions mean(), cov(), and pca() in Matlab.

   For every digit, from 0 to 9, compute:

   (i) the mean μ (3 points),

   (ii) the covariance matrix C (5 points), and

   (ii) the first mode of variation determined by the eigenvector v_1 and the corresponding eigenvalue λ_1 (where λ_1 is the largest of all eigenvalues) of the covariance matrix C (7 points).

   Note: Before computing the mean and covariance matrix, convert each 28×28 pixel image matrix to a $28^2 \times 1$ vector by concatenating its columns. To visualize the $28^2 \times 1$ mean vector, convert it back to a matrix and then visualize it using imagesc(). Use the reshape() function to change matrices to vectors and vice versa. The covariance matrix will be of size $28^2 \times 28^2$.

   • (5 points) For each digit, sort the 28^2 eigenvalues of the covariance matrix and plot them as a graph. Comment and justify what you observe. How many "principal" / significant modes of variation (i.e., number of "large" eigenvalues) do you find, for each digit ? Are the significant modes of variation equal to 28^2 or far less ? Why ?

   • (5 points) For each digit, show the 3 images side by side: (i) $\mu - \sqrt{\lambda_1} v_1$, (ii) μ, and (iii) $\mu + \sqrt{\lambda_1} v_1$, to show the principal mode of variation of the digits around their mean. Comment and justify what you observe. For a certain digit, say 1, what does the principal mode of variation tell you about how people write that digit ?

5. (10 points) Principal Component Analysis (PCA) for Dimensionality Reduction.

   Read all instructions before you start.

   Download the dataset comprising images of handwritten digits in `http://yann.lecun.com/exdb/mnist`; this has been downloaded in the folder "data" and stored as "mnist.mat".

   As of now, for each digit, each 28×28 pixel image is represented using 28^2 coordinate values in the Euclidean space of dimension 28^2. Suppose you decide to re-represent the images using only 84 coordinates (instead of $28^2 = 784$) in a 84-dimensional basis for some 84-dimensional hyperplane within the original Euclidean space, such that the chosen 84-dimensional hyperplane maximizes the total dispersion of the original data (for the chosen digit) within the hyperplane.

   • (5 points) Write a function to compute those 84 coordinates, for each of the ten digits (0–9).

   • (5 points) Give an algorithm for regenerating / reconstructing the image using those 84 coordinates (and the knowledge of the designed 84-dimensional basis). For each of the ten digits (0–9), pick an image, and show the original and the reconstructed images side by side.

6. (25 points) Principal Component Analysis (PCA) for Another Image Dataset

   Read all instructions before you start.

   Consider the dataset provided within the folder "data_fruit"

   For this question, you cannot use the functions mean(), cov(), and pca() in Matlab.

   Each datum is an image of size 80×80 pixels with 3 color channels red (R), green (G), and blue (B), i.e., a $80 \times 80 \times 3$ array. For PCA, each image should be resized to a vector of length 19200. For visualization, reshape each vector back to a RGB image of size 80×80 pixels using the function reshape(), followed by a shift and rescaling of the values into the range $[0, 1]$, followed by displaying the matrix using the function image().

   • (9 points: 1 + 4 + 4) Similar to the analysis done in previous question, find the mean μ, the covariance matrix C, and the the first 4 **eigenvectors** of C. Display the **mean** and the **eigenvectors** as images (side by side, in the same figure); you can use the function subplot(). Find the first 10 **eigenvalues**, sort them, and plot their values on a graph. Use the function eig**s**() for efficient computation.

   • (8 points: 4 + 4) For each fruit image in the dataset, finds its **closest** representation as a linear combination of the first 4 eigenvectors added to the mean. Use the measure of closeness as the Frobenius norm of the difference. Describe the algorithm used to produce this closest representation in mathematical terms and describe the logic behind your algorithm. Display the original fruit image and its closest representation, as images (side by side, in the same figure).

   • (9 points: 3 + 3 + 3) Using all of the top 4 eigenvectors and the mean image, **sample** random images to generate new images of "fruit". Describe the underlying algorithm clearly in words and including suitable mathematical notation. Display three such images that are distinct from any image in the given the dataset, but are representative of the dataset and can be considered as that of a new / generated fruit.