CS215, IIT Bombay
Assignment #4

Prof. Suyash P. Awate

Tulip Pandey (190050125)
Vibhav Aggarwal (190050128)

Problem 1

Solution

(a) X_1 and X_2 are I.I.D random variables following the uniform distribution (U(-1,1)).
Therefore,

$$PDF_{X_i}(x) = \frac{1}{2} \quad for \quad i \in \{1, 2\}$$

Let Y be a random variable defined as follows:-

$$Y := X_1^2 + X_2^2$$

Therefore, $P(Y \leq 1) = P(-1 \leq X_1 \leq 1; -\sqrt{1 - X_1^2} \leq X_2 \leq \sqrt{1 - X_1^2})$
This can be calculated as follows:-

$$\int_{-1}^{1} \frac{1}{2} \left(\int_{-\sqrt{1-x_1^2}}^{\sqrt{1-x_1^2}} \frac{1}{2} dx_2 \right) dx_1$$

This evaluates to :-

$$P(Y \leq 1) = \frac{\pi}{4}$$

(b) We consider a random variable Z which is defined as follows :-

$$Z = 1; \quad if \quad Y \leq 1$$

$$Z = 0 \ ; \ \text{otherwise}$$

This is a bernoulli random variable with parameter $p = \frac{\pi}{4}$. We obtain N data points of the form X_{1i} and X_{2i} from U(-1,1) and convert these data points into corresponding Z_i for i $\in \{1, 2, \ldots N\}$. We now carry out maximum likelihood estimation of the parameter p. For a bernoulli random variable, the ML estimate of it's parameter is as follows (denoted by $\hat{p}$) :-

$$\hat{p} = \frac{\sum_i^N Z_i}{N}$$

Therefore our estimate for π would be as follows:-

$$\pi = 4\frac{\sum_i^N Z_i}{N}$$

for Z defined as above.

(c) The estimates for different values of N can be found in `results/q1.pdf`
When N is as large as 10^9, then we cannot generate and store all the points simultaneously in a MATLAB array due to memory constraints. However, we can make use of the fact that we don't need all the points at once. Therefore, we can generate the points in multiple batches of size say 10^5 and keep adding the sum of Z for all batches to obtain the final sum. Our code handles this case too.

(d) Using the central limit theorem, we can say that the empirical mean of Z (as defined in previous part(s)), denoted by $\hat{\mu}$, follows a Gaussian distribution with mean $\frac{\pi}{4}$ and standard deviation $\frac{\sqrt{\frac{\pi}{4}\left(1 - \frac{\pi}{4}\right)}}{\sqrt{N}}$.

Problem 1 continued on next page...

Therefore, the following holds :-

$$P\left(-\frac{0.01}{4} \leq \hat{\mu} - \frac{\pi}{4} \leq \frac{0.01}{4}\right) = \text{erf}\left(\frac{0.0025}{\sqrt{2}\sigma}\right)$$

where $\sigma = \dfrac{\sqrt{\frac{\pi}{4}(1-\frac{\pi}{4})}}{\sqrt{N}}$

$\text{erf}\left(\dfrac{n}{\sqrt{2}}\right) = 0.95$ for n = 2. Therefore,

$$\frac{0.0025}{\sigma} = 2$$

$$\frac{\sqrt{N}}{\sqrt{\frac{\pi}{4}\left(1-\frac{\pi}{4}\right)}} = 800$$

$$N = 64 \times 10^4 \times \frac{\pi}{4}\left(1-\frac{\pi}{4}\right)$$

Therefore,

$$N = 107871 \simeq 10^5$$

Location of MATLAB script: `code/q1.m`
Instructions to run the code: Simply run the script and it will output the estimate for pi denoted as pi-estimate in the code for the different values of N :- [10,10^2,10^3,10^4,10^5,10^6,10^7,10^8] in that order.
Location of results: `results/q1.pdf`

Problem 2

Solution

(a) If X is a 2D multivariate Gaussian random variable, then it can be expressed as: $X = AW + \mu$, where A is a 2×2 matrix, μ is a 2D vector and $W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$ where W_1 and W_2 are independent random variables following the standard normal distribution. The covariance matrix C of X is given as: $C = AA^T$. If we know A and μ, then we can easily sample data from X by sampling values of W_1 and W_2 from the standard normal distribution and then applying the linear transformation $g(W) = AW + \mu$.

We are given μ and C. Now to find A, we can proceed as follows:
We can assume that A is a combination of scaling and rotation (proper or improper). Thus, $A = RS$ where S is a diagonal matrix and R is an orthogonal matrix. Now, $C = AA^T = RSS^TR^T = RS^2R^T$. Hence, we can simply diagonalize the matrix C using `eig` function to obtain R and S^2. Taking square root of each diagonal entry of S^2 gives us S which can then be multiplied with R to get A, as required.

(b), (c), (d)
Location of MATLAB script: `code/q2.m`
Instructions to run the code: Simply run the script and it will produce 7 figures in separate windows (5 for part (d), 1 for part (b) and (c) each).
Location of results: `results/q2.pdf`

Problem 3

Solution

Principal Component analysis can be used to approximate a linear relationship between the different components (here x and y) of the sample data . The method followed is as follows :-

If we have N samples of some d-dimensional measurement (here d = 2), then we can represent this data in the form of a $d \times N$ matrix $A = [v_1\ v_2\ \dots\ v_N]$. First step is to shift the origin to the mean of these N sample points to facilitate the computations. So we deduct the mean from A. Now we need to compute the $d \times d$ covariance matrix, C, of the data. The element at i-th row and j-th column of C is given by $\frac{1}{N-1}\sum_{k=1}^{N} v_{ki}v_{kj}$. Thus, C can be conveniently written as $AA^T/(n-1)$. The eigenvectors of this symmetric matrix give the directions of deviation of this data. The eigenvalue corresponding to the eigenvectors determines the measure of deviation along that direction. Therefore, the eigenvector having the largest eigenvalue is the principle mode of deviation and is the best approximation for the linear relationship between x and y. The eigenvalues and eigenvectors of C are obtained using the `eig` function of MATLAB. The coefficients of the eigenvector (with the max eigenvalue) are the coefficients in the linear expression and the mean alongwith the eigenvector coefficients determines the constant term (as mentioned in the code).

The line obtained by PCA does not represent the data in the second case efficiently. PCA carries out linear dimensional reduction and in cases where the data is not linearly related, it returns the line on which projection of the data gives the maximum variance which does not replicate the precise relationship between the data components. PCA carries out analysis between data components solely on the basis of their linear relationship, and as seen while dealing with correlation co-efficients, widely varying data may correspond to the same correlation coefficient. The measure of relation between data components is based only on the nature of change in one on linearly incrementing/decrementing the other. In non linear graphs like the one given here, the linear changes in one lead to changes in the same and opposite direction in the other which effectively cancel out. Thus, the data is represented by a straight, almost horizontal line passing through the mean representative of lack of any linear relationship overall. In the first case however, the data, actually (almost) follows a linear relationship. Here, the analysis of the eigen decomposition shows that the eigenvalue corresponding to the principal eigenvector is sufficiently larger than the other which means that the fluctuations in the data perpendicular to the principle mode of variation are minimal and as a result, the line gives a good fit for the data.

Location of MATLAB script: `code/q3.m`
Instructions to run the code: Simply run the script. It will produce 2 figures in separate windows. First figure shows the the scatter plot of the points in the first dataset and the line representing the linear relationship between x and y in the first dataset. The second figure shows the the scatter plot of the points in the second dataset and the line representing the linear relationship between x2 and y2 in this dataset.
Location of results: `results/q3.pdf`

Problem 4

Solution

Location of MATLAB script: `code/q4.m`
Instructions to run the code: Simply run the script. It will produce 6 figures in separate windows. First figure shows the the first 50 eigenvalues for each digit (shown in different colors). The next 5 figures contain 2 subplots each which show the images $\mu - \sqrt{\lambda_1}v_1$, μ and $\mu + \sqrt{\lambda_1}v_1$ side by side in that order.
Location of results: `results/q4.pdf`

After running the script, the mean, covariance matrix, largest eigenvalue and corresponding eigenvector for each digit gets stored in the matrices `Means` (784x10), `Covs` (784x784x10), `Eigvals` (1x10) and `Eigvecs` (784x10) respectively.

The plot of eigenvalues shows that initially, there is a rapid decrease in eigenvalues for all digits. The significant modes of variation can be safely assumed to be around 30 to 35 (or maybe less) which is far less than $28^2 = 784$. This is because, we don't actually need 784 dimensions to represent an image as simple as a single digit. We can approximate it very well in about 35 dimensions only.

The general trend in the triplet of images is that the left and right images are rotated by some angle with respect to the middle image in opposite directions. This implies that the major difference between the digits written by different people lies in the orientation of digits, while the shapes of digits almost remain similar in general. In images consisting of straight lines, this is almost always true, while for the images with some curved portions, there are slight differences in the extent of the bulge of the curves as well. This is because of the extra possibility of deviation in this aspect but the principal mode of variation continues to be the orientation in most cases. The large number of sample points for each digit also guarantee that most possibilities corresponding to the principal mode of variation lie within the two extremes of $+\lambda_1$ and $-\lambda_1$ with a high probability of around $\text{erf}(\sqrt{1000/2}) \simeq \text{erf}(\sqrt{22})$ (according to CLT).

Problem 5

Solution

Location of MATLAB script: `code/q5.m`
Instructions to run the code: Simply run the script. It will produce 5 figures in separate windows. They contain 2 subplots each which show the original and reconstructed images for each digit side by side in that order.
Location of results: `results/q5.pdf`

Let U represent the 784x84 matrix containing the first 84 eigenvectors. Any sample image (after subtracting mean) V which is a 748x1 vector, can be projected on each of the eigenvectors to get the 84 coordinates. The 84x1 vector X containing these 84 coordinates can be computed as: $X = U^T V$. Once we have this, the original image can be (approximately) reconstructed by simply taking the linear combination of the eigenvectors according to the vector X and adding back the mean. Thus the reconstructed image W is given as: $W = UX + \mu$ (where μ is the mean).

Problem 6

Solution

For producing the best approximation of the given images as linear representations of the first 4 eigenvectors and the mean, we took the sum of the projection of the mean shifted images on each of the unit eigenvectors and added the mean to this sum. The idea behind it is that the projections give the representation of the data in the 4 dimensional eigenspace and the mean shifts back the image into the original co-ordinate frame . This representation also minimises the Frobenius norm of the difference between the actual image and its 4 dimensional approximation. Following is the proof for the same :-

Let $v = a_1 v_1 + a_2 v_2 + a_3 v_3 + a_4 v_4$ be a linear combination of unit eigenvectors v_1, v_2, v_3, v_4 where a_1, a_2, a_3, a_4 are some arbitrary scalars. If X represents our image vector (19200 dimensional) in the mean shifted

co-ordinate frame, then we need to minimize the Frobenius norm of $X - v$. Therefore, the objective function to be minimised is as follows :-

$$\begin{aligned}
J &= |X - (a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4)|^2 \\
&= |X|^2 + |(a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4)|^2 - 2X \cdot (a_1v_1 + a_2v_2 + a_3v_3 + a_4v_4) \\
&= |X|^2 + \sum_{i=1}^{4} a_i^2 - 2\sum_{i=1}^{4}(a_i X \cdot v_i) \quad \text{(pairwise dot products of perpendicular eigenvectors is 0)} \\
&= |X|^2 + \sum_{i=1}^{4} |a_i - X \cdot v_i|^2 - \sum_{i=1}^{4} (X \cdot v_i)^2
\end{aligned}$$

This is minimised when each $a_i = X \cdot v_i$ which means that the taking the projections of the image on the eigenvectors, for obtaining the coefficients for linear representation indeed minimises the Frobenius norm.
For sampling new images using the 4 eigenvectors and the mean, the following process is followed :-
As the eigenvalues give the variance of projections of the data on the eigenvectors, the net variance from the mean in the hyper plane determined by these vectors would be a linear combination of the variances along these directions. This can be modelled using random draws from a standard normal distribution and multiplying them with the square root of the eigenvalues of the corresponding eigenvectors to finally get the co-efficients by which each eigenvector is multiplied. Finally, this representation is shifted back to the original co-ordinate frame by adding the mean. This produces a random sample representative of the data set.

Location of MATLAB script: `code/q6.m`
Instructions to run the code: Simply run the script. It will produce 19 figures in separate windows. First figure shows the the image corresponding to the first 4 eigenvalues (in descending order of eigenvalues from left to right followed by the mean image on the extreme right). The next figure shows the plot of the 10 largest eigenvalues. The next 16 figures show the image sample from the dataset and the approximate estimation of the image as a linear representation of the eigenvectors and the mean in one image. The last figure shows 3 images of new fruits -that are linear combinations of the eigenvectors and the mean and can be considered to be representative of the dataset.
Location of results: `results/q6.pdf`