EEE598: Generative AI: Theory to Practice

Instructor: Dr. Lalitha Sankar
Homework Assignment #1
Total points: 100

Due Date and Time: Tuesday 02/11/2025 11:30 PM

Please follow carefully the instructions on how to submit homework assignments found in the HW module on Canvas (right after Module 0).
For all plotting exercises, make sure to label your axes and legends as appropriate. Python libraries can be used for all analyses and plots.

Principal Component Analysis (PCA)

Total: *[58 points]* In this problem we will explore the USPS digits dataset (https://www.kaggle.com/datasets/bistaumanga/usps-dataset) with principal component analysis. The Kaggle page where you can find the dataset also has the necessary instructions to load it into memory. This dataset consists of handwritten digits scanned by the USPS from mail. Each image is of a number between 0 and 9 and has size 16×16. Images are in black and white. Note that this is not the MNIST digits dataset.
You have been assigned a group (you can find this on Canvas and the grouping is entitled HW1 groups on canvas) – there are 10 groups for the 10 digits. Group $i, i = 1, \ldots, 10$ will work on digit $i - 1$. In short, you will only use **ONE DIGIT**, the digit assigned to you is random and depends on the group you are in. For example, if you are in group 4, you will work on digit 3.
Remember to properly scale and normalize your data.

1. *[7 points]* Let $\mathbf{X}$ be the **centered** (mean-subtracted) data matrix. Perform SVD decomposition on this centered data matrix which should yield the left singular unitary matrix $\mathbf{U}$, right singular $\mathbf{V}$, and diagonal singular value matrix $\mathbf{\Sigma}$ as:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

where the diagonal entries of $\mathbf{\Sigma}$ are $(\sigma_1, \sigma_2, \ldots, \sigma_r, \ldots, 0)$ such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$.

Let q denote the the size of the low-dimensional representation; clearly $q \in \{1, \ldots, r\}$ where $r = \text{rank}(X) \leq \min(d, n)$ is the rank of the $\mathbf{X}$ matrix.

Create the following visualizations:

(a) *[2 points]* Plot singular values in descending order.

(b) *[4 points]* Let $\hat{\mathbf{X}}_q$ denote the data matrix reconstructed after projecting to q principal vectors, $q \in \{1, 2, \ldots\}$. Then,

$$\hat{\mathbf{X}}_q = \mathbf{U}_q\mathbf{\Sigma}_q\mathbf{V}_q^T$$

where as defined in class, the matrices with a subscript q use only their first q singular vectors (U and V) or singular values ($\mathbf{\Sigma}$). The mean squared error (MSE) is given by

$$\text{MSE}(q) = \frac{1}{nd}\sum_{i=1}^{n}\sum_{j=1}^{d}(\mathbf{X}[i,j] - \hat{\mathbf{X}}_q[i,j])^2, \tag{1}$$

where $\mathbf{A}[i,j]$ denotes the $(i,j)^{\text{th}}$ component of the matrix $\mathbf{A}$, n is the number of samples, and d is the dimension of the data. Plot MSE as a function of the number of components used for reconstruction.

(c) *[1 points]* The explained variance is defined as $\sum_{i=1}^{q}\sigma_i^2$. Plot explained variance as a function of q.

2. *[3 points]* What is an appropriate value or range of values for q, the dimension of low dimensional space, that you'd choose based on explained variance.

3. *[12 points]* For the remaining problems, let $q = 5$. Visualize the columns of $\mathbf{U}$ in an array of 1×5 where the $i^{\text{th}}, i = 1, 2, \ldots, 5$ panel of the image array includes the following plots; make sure to use clear legends to identify the 4 plots in each panel.

(a) *[4 points]* Histograms of the i^{th} column of $\mathbf{U}$. (that is, use the data in each column to fit a histogram by cleverly choosing the input parameters to the histogram function).

(b) *[4 points]* Fit probability distributions to each column using `scipy.stats`. Try the Gaussian and Laplace distributions. Explain what metric is being optimized to fit by this library.

(c) *[4 points]* Overlay the fitted distributions on the histograms.

4. *[10 points]* In the fitting process described above, one possible output is the CDF of the distribution. We now address the question of which fitted distribution above, i.e., Gaussian or Laplace, is a better fit to the histogram for each of the five columns of $\mathbf{U}$. To this end, we will use two different metrics:
(i) the Kolmogorov-Smirnoff (KS) one-sample test, and
(ii) the Anderson-Darling (AD) one-sample test.
While it is ideal to find formal textbook-based references, the following simpler expositions may be helpful:
(i) https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm
(ii) A simple overview of fitting, evaluating, and testing the process of fitting a probability distribution to a dataset.https://www.linkedin.com/advice/0/how-do-you-determine-fit-probability-distribution-kdoje#:~:text=Data%20Scientist%20Associate-,To%20determine%20the%20fit%20of%20a%20probability%20distribution%20to%20your,data%20with%20the%20expected%20distribution.
(iii) This site lists many possible light and heavy tailed distributions that may be of broader interest for fitting https://rpubs.com/ChrisSchmidt/777597
Finally, a PDF of the classic textbook in statistics, All of Statistics by Larry Wasserman may be helpful to some of you.

Now create a table showing:

(a) *[5 points]* The best fitting distribution based on each of the two tests (for each column). You are also welcome to try some of the other tests listed in the links and add it to your table (explain the desired range/values for each test).

(b) *[5 points]* The parameters of the distribution learned from the fitting. For example, when fitting a Gaussian distribution, the two parameters of interest are mean and variance. Identify the parameters for the two distributions used to fit ane list the name and values of the parameters clearly.

5. *[8 points]* Implement the generative model:

$$\hat{\mathbf{X}}_{\text{synth}} = \mathbf{U}_{\text{sampled},q}\boldsymbol{\Sigma}_q\mathbf{V}_q^T + \mathbf{1}\bar{\mathbf{x}}^T + \mathbf{W} \tag{2}$$

where $\mathbf{U}_{\text{new}}$ is generated by sampling from your fitted distributions, $\mathbf{1}$ is the all ones vector (dimensions?), and $\mathbf{W}$ is random noise with $\mathbf{W}_{ij} \sim \mathcal{N}(0, \sigma^2)$, i.e., each entry of $\mathbf{W}$ is an independent and identically distributed Gaussian random variable (RV). Since the data was centered, the mean of the data must be added in in the generative model, hence the addition of $\bar{\mathbf{x}}$. Let $\sigma = \text{MSE}(q)$ computed in equation (1).

(a) *[2 points]* Display 5 synthetic images for the digit assigned to you in a 1×5 panel.

(b) *[2 points]* Generate the mean image for the true dataset (for the digit assigned to you) and the mean synthetic data (you'd need a decent number of generated images to do this); display them side by side.

(c) *[4 points]* Comment on the quality of the synthetic images generated. Does increasing q from 5 (try 1-2 other values) improve performance? Provide some visual or other arguments to justify your answer.

6. *[18 points]* *Validating your synthetic data by computing cross-correlations between features*: It is always a challenge to compare the quality of the true and synthetic data. A measure of closeness of the generated and true images is the correlation coefficient matrix.

(a) *[1 points]* For the data matrix $\mathbf{X}$, write down the matrix form of (empirical or sample) covariance matrix. [Assume here that the mean matrix written as $\mathbf{1}\bar{\mathbf{x}}^T$ has been subtracted from $\mathbf{X}$].

(b) *[1 points]* What are the dimensions of this matrix: $d \times d$ or $n \times n$? [Hint: try to recall how sample covariance is defined for two scalar RVs.]

(c) *[2 points]* If you generated the same number n of synthetic samples as your real data and denoted the resulting synthetic data matrix as $\hat{\mathbf{X}}$, how is the empirical or sample covariance matrix of $\hat{\mathbf{X}}$ computed? [Make sure you keep track of the mean!]

(d) *[10 points]* Generate the same number of images as the real data. Compute the correlation matrix using Numpy's `corrcoef`. Please note the difference between how covariance of a data matrix $\mathbf{X}$ is defined in the classroom (in the lecture on PCA) and in Numpy and compute the covariance matrix appropriately. Plot heat maps of the correlation matrices (normalized covariance matrix) of real and synthetic data. Use Matplotlib's `imshow` for these images. How similar are the heat maps by visual inspection?

(e) *[4 points]* Report the L_2 difference between the two correlation coefficient matrices as a measure of similarity. Let k be the dimensions of your correlation coefficient matrix. The L_2 difference is simply:

$$\frac{1}{nk}\sum_{i=1}^{n}\sum_{j=1}^{k}(\mathbf{C}[i,j]-\hat{\mathbf{C}}_q[i,j])^2, \tag{3}$$

where $\mathbf{C}$ and $\hat{\mathbf{C}}_q$ are the correlation coefficient matrices of the real and generated data, respectively.

Probabilistic Principal Component Analysis (PPCA)

Total points: *[32 points]* In this problem we will explore the same dataset as previously, but with probabilistic PCA. For this problem, use the same SVD and choice of q code from the previous problem.

7. *[6 points]* (since there isn't any writing for this problem, these points are for the code) Choose the same q as in the previous problem. Implement the PPCA generative process as follows:

   (a) Sample latent variables $\mathbf{z}_{\text{new}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_q)$ where $\mathbf{I}_q$ is the q-dimensional identity matrix.

   (b) Compute the same mean $\bar{\mathbf{x}}$.

   (c) Generate an n-length vector $\boldsymbol{\epsilon}$ whose entries are zero mean independent and identically distributed Gaussian noise RVs with variance σ^2, i.e., $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$ where

$$\sigma^2 = \frac{1}{d-q} \sum_{j=q+1}^{d} \lambda_j$$

   (d) Generate observations

$$\mathbf{x}_{\text{new}} = \mathbf{V}_q \mathbf{z}_{\text{new}} + \bar{\mathbf{x}} + \boldsymbol{\epsilon} \tag{4}$$

8. *[4 points]* How is σ^2 related to the explained variance for this choice of q.

9. *[8 points]* Generate and display new synthetic digits:

   (a) *[2 points]* Display 5 synthetic images for the digit assigned to you in a 1×5 panel.

   (b) *[2 points]* Generate the mean image for the true dataset (for the digit assigned to you) and the mean synthetic data (you'd need a decent number of generated images to do this); display them side by side.

   (c) *[4 points]* Comment on the quality of the synthetic images generated. Does increasing q from 5 (try 1-2 other values) improve performance? Provide some visual or other arguments to justify your answer.

10. *[14 points]* Validating synthetic data via cross-correlations:

   (a) *[10 points]* Generate the same number of images as the real data. As in problem (6.d), compute the correlation matrix using Numpy's `corrcoef`. [Please note the difference between how covariance of a data matrix $\mathbf{X}$ is defined in class and in Numpy and compute the covariance matrix appropriately.] Plot heat maps of the correlation matrices (normalized covariance matrix) of real and synthetic data. Use Matplotlib's `imshow` for these images. How similar are the heat maps by visual inspection?

   (b) *[4 points]* Report the L_2 difference between the two correlation coefficient matrices as a measure of similarity.

Comparing Synthetic PCA with PPCA

11. *[10 points]* Let's compare PPCA with the approach of using PCA to generate synthetic data.

   (a) *[5 points]* How does PPCA outlined via the three steps in problem (7.) which lead to the generative model given by equation (4) differ from the method outlined earlier in equation (2) to generate synthetic samples from a basic PCA method? Hint: how and why do the slightly different mathematical formulations change what we obtain?

   (b) *[5 points]* Are there any advantages of the probabilistic formulation?