Homework 3 Fall 2024

04650 Mathematical Foundations of Machine Learning (CMU-Africa)

Release Date: October 7th, 2024
DUE: **October 28th, 2024, 11:59 PM CAT**

- **Collaboration policy:** You can discuss HW problems with other students, but the work you submit must be written in your own words and not copied from anywhere else. However, do write down (at the top of the first page of your HW solutions) the names of all the people with whom you discussed this HW assignment. We strongly encourage you to type your solution using LaTeX, but this is not required

- **Submitting your work:** Assignment will be submitted using Gradescope. You will submit a single pdf file containing your solutions and any python code you wrote for the questions that require programming.

- **Getting Help**: Please use office hours and Piazza to ask any questions related to this assigments.

- **Refrain from using ChatGPT**: These problems are easy and can be easiliy solved with the use of generative AI's like ChatGPT. However, you will not learn by using these tools. Not learning will impact your performance in other courses which require this knowledge like introduction to deep learning and introduction to machine learning for engineers. Ask questions on Piazza about anything you don't understand, the TA's and instructors will respond to you as fast as possible. Moreover, you will not have access to internet during the exam.

1. Eigenvalues and Eigenvectors (10 Points)

Consider the matrix,

$$\mathbf{A} = \begin{bmatrix} 9 & 1 & -1 \\ -1 & 11 & 1 \\ -2 & 2 & 10 \end{bmatrix}$$

1. Obtain the characteristic polynomial of $\mathbf{A}$. (Hint. Find $\det(\mathbf{A} - \lambda\mathbf{I})$)
2. From the characteristic polynomial, obtain the 3 eigenvalues, λ_1, λ_2 and λ_3, such that $\lambda_1 < \lambda_2 < \lambda_3$
3. Using Gaussian elimination, obtain a corresponding eigenvector for each eigenvalue obtained in (2) above.
4. Using np.linalg.eig, compute the eigenvalues and eigenvectors of $\mathbf{A}$
5. Compare the eigenvectors computed using np.linalg.eig and that obtained using Gaussian elimination in (3) above. Are they the same? If they are the same, explain why this is the case. Otherwise, explain why they are different.
6. Now, given a diagonal matrix, $\mathbf{D}$, whose diagonal entries are the eigenvalues of $\mathbf{A}$ and a matrix, $\mathbf{P}$ whose columns are the eigenvectors of $\mathbf{A}$, use the eigenvalues you obtained in (2) and the eigenvectors you obtained in (3) to show that
$$\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$$
7. Calculate the determinant of $\mathbf{A}$, $\det(\mathbf{A})$
8. What is the trace of $\mathbf{A}$, $tr(\mathbf{A})$

9. Confirm that the following holds

$$\det(\mathbf{A}) = \lambda_1 \lambda_2 \lambda_3$$
$$tr(\mathbf{A}) = \lambda_1 + \lambda_2 + \lambda_3$$

2. Image Compression (10 points)

In class we explored the concept of low rank approximation where we approximated a matrix as the product of two low rank matrices. With Singular Value Decomposition (SVD), you can approximate a matrix as the sum of several rank-1 matrices, and this was also demonstrated.

An image has been provided for you in the handout.

1. What are the dimensions of the provided image?
2. Obtain the SVD decomposition of the image and visualize the rank-1 matrix corresponding to the largest singular value.
3. Add up the rank-1 matrices for the top 3 singular values and visualize your result. What do you observe?
4. Add up the rank-1 matrices for the top 10 singular values and visualize your result. What do you observe?
5. Assuming 1 byte of memory is required for one element of the image matrix, determine the top k rank-1 matrices required to represent the image if a compression ratio of 13:1 is desired. What is the value of k ?. Add up the rank-1 matrices for the top k singular values and visualize your result. Remember to show your workings

3. Linear Regression (10 points)

In linear regression, SVD can help by decomposing the design matrix, identifying dependencies between predictor variables, and enabling regularization techniques like ridge regression. By reducing the rank or eliminating singular values, SVD assists in stabilizing regression models and improving their predictive accuracy. In this question you will solve a linear regression problem using SVD.

Complete the cells in the provided starter code and answer the following questions.

1. What is the size of the dataset?
2. How many singular values were obtained during the calculation of the coefficients? Why is this number exact — neither more nor less?
3. Given the SVD decomposition of a matrix, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, show that $\mathbf{A}^{-1} = \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{V}^\top$
4. After training the model, what coefficients did you obtain? Interpret the significance of this coefficients.
5. What is the mean squared error of the train split and test split?
6. Visualization: complete the visualization tasks in the starter notebook and report your observation.

4. Principal Component Analysis (PCA) and eigenfaces (70 points)

PCA is used to project high dimensional vectors onto a lower dimensional subspace whose basis vectors capture the maximum variance in the data of high dimensional vectors.

An image can be seen as a high dimensional vector. For example, a 100x100 image is a 10000 dimensional vector. PCA can be used to reduce this dimensionality by projecting images onto a lower dimensional subspace, given a dataset of images.

Eigenfaces is an application of PCA to obtain a lower dimensional subspace onto which face images can be projected, such that face images can be reconstructed by linearly combining the basis vectors of the lower dimensional subspace. The basis vectors of this lower dimensional subspace are called the eigenfaces. Eigenfaces can be applied in face recognition.

[

You can read more about PCA in chapter 10 of the textbook and study this Wikipedia article on Eigenface

]

In this exercise, you will go through the process of implementing a face recognition system using eigenfaces. You will use faces from the olivetti faces dataset. A starter notebook has been provided for you in the handout.

1. In the starter code, the olivetti faces dataset has been loaded, using a scikit-learn function. A database of faces, a list of test faces, and a list of faces of people not in the database have been created.
   (a) How many face images are in the database?
   (b) How many images are in the list of test faces?
   (c) How many images of people not in the database are available?
   (d) What are the dimensions of each face image?
2. Create a matrix $\mathbf{T}$ containing all faces in the database, such that each row corresponds to the pixels of a person's face. Note that the height of this matrix should be equal to the number of faces in the database. What are the dimensions of $\mathbf{T}$?
3. Obtain and visualize the mean face
4. Obtain the covariance matrix, $\mathbf{S}$, of $\mathbf{T}$.
5. Obtain the eigenfaces of the covariance matrix $\mathbf{S}$. Report the time (in seconds) it took to compute the eigenfaces.
6. Why is the process of computing the eigenfaces as done above computationally expensive? Please explain.
7. Given the mean face $\bar{\mathbf{x}}$, and the matrix $\mathbf{V}$ whose columns are the eigenvectors of $\mathbf{S}$, if $\mathbf{M}$ is a matrix whose rows are equal to the result of subtracting the mean face from the rows of $\mathbf{T}$, show that

$$\mathbf{V} = \mathbf{M}^\top \mathbf{U}$$

   where $\mathbf{U}$ is a matrix whose columns are the eigenvectors of $\mathbf{M}\mathbf{M}^\top$
8. Compute the eigenfaces using $\mathbf{M}^\top \mathbf{U}$
9. Why is obtaining the eigenfaces using $\mathbf{M}^\top \mathbf{U}$ more computationally efficient than the approach used in (5)? Please explain.
10. Observe the numerical values of the eigenfaces obtain using both approaches. Are they the same? If they are the same, explain why? otherwise explain why they are different.
11. Now visualize the first eigenface (the eigenface that captures the most variance in faces in the database) obtained using the approach in (5) and (8). Compare the images, they should be similar, why are they similar?
12. Using the eigenfaces obtain in (5), select the top eigenfaces that capture 90% of the variance in the database. How many eigenfaces were selected?
13. Visualize the top 10 eigenfaces

14. Given a matrix, $\mathbf{A}$, whose columns are the top eigenfaces selected in (12), show that the projection matrix unto the column space of $\mathbf{A}$ is the identity matrix.
15. Given a new face, $\mathbf{x}$, and the mean image $\bar{\mathbf{x}}$, show that the coordinate of $\mathbf{x}$ in the eigenfaces subspace is given by $\mathbf{A}^T(\mathbf{x} - \bar{\mathbf{x}})$
16. Given the coordinate of a new face, we can calculate the distance between this coordinate and the coordinates of all faces in the database. The new face is recognized if the minimum distance is less than a certain threshold. **With the information you now have about eigenfaces, design a face recognition system that recognizes faces similar to those in the face database. Identify a threshold such that the accuracy of your system is greater than 80%**
17. For the obtained threshold, Evaluate the face recognition system's performance using the following metrics:
    - Accuracy
    - Precision
    - Recall
    - F1-score
18. For your face recognition system, since you already know the coordinates of all faces in the database with respect to the eigenfaces, you can decide to discard the database of faces and only work with this coordinates. This is also done when you decide to include faces of new people to the system, so that they can be recognized. Using this information, calculate the compression ratio of your face recognition system if you decide to store only the coordinates of faces. Show your workings.
19. Conduct experiments to assess the impact of different parameters and settings on your face recognition system. Consider factors like the number of eigenfaces.
20. Analyze and interpret the results, discussing the strengths and limitations of the Eigenfaces method in your context.