Assignment

Homework 04

Andrew Park

Applied Machine Learning Homework Assignment

CORNELL TECH | HOME OF THE JACOBS TECHNION-CORNELL INSTITUTE

November 3, 2023

1. General Summary

The overall assignment was about understanding eigendecompostion, eigenfaces, applying logistic regression to these and SVD. The assignement helped me to understand and experiment practical application of eigendecomposition in face recognition and understand mathematical concept of SVD.

In the coding assignment, I was amazed to use image data by the label in the text by the code script as follows: plt.imread. The overall process was about performing face recognition which was very amazing to see the results on each code blocks. First, beginning with part (b), I prepared the Face Dataset by reshaping 50x50 pixel images into 2500-dimensional vectors to form the matrices $\mathbf{X}$ and $\mathbf{X}_{\text{test}}$ for the training and test sets. In part (c), I computed the average face μ by averaging all image vectors in $\mathbf{X}$. In part (d), I performed mean normalization by subtracting μ from every image vector in both $\mathbf{X}$ and $\mathbf{X}_{\text{test}}$. Part (e) involved eigendecomposition on $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ to extract eigenfaces, with the first ten displayed as images. For part (f), these eigenfaces were utilized to transform the high-dimensional face images into a lower-dimensional 'face space'. Lastly, in part (g), logistic regression was used on these features for face classification, and the model's accuracy was plotted for different dimensions of the feature space.

Next, the first written exercise focused on understanding Singular Value Decomposition (SVD), especially the process of obtaining the eigendecomposition of X^TX from the SVD. By proving the equation, I was able to understand that X^TX can be used to derive the eigenvalues and eigenvectors for X in SVD. The next exercise was about finding the eigenvalues, eigenvectors, SVD, and the one-dimensional approximation of M. Starting from question (b), I was able to apply the equation learned from exercise 1, benefiting from the concept of the relationship between SVD and eigendecomposition by using the matrices M^TM and MM^T. It was also a great opportunity to observe that the columns of eigenvectors correspond to the matrices U and V in SVD. Lastly, by computing the one-dimensional approximation of M, I learned that capturing the most dominant pattern with the greater value of Σ (which has a higher variance) can represent the overall scheme of the data (matrix) effectively. This approach may cause some difference between M and M_{approx}, due to the loss of information, but doing so will be highly efficient when understanding high-dimensional data.

WRITTEN EXERCISES

Problem 1

SVD and eigendecomposition. Show that we can obtain the eigendecomposition of X^TX from the SVD of a matrix X.

Solution.

According to the definition of SVD (Singular Value Decomposition), SVD of an $m \times n$ matrix X is the factorization of X into three matrices as follows: $X = UDV^T$

a) U is a $m \times m$ orthonormal matrix, then $U^TU = I$ (identity matrix)

b) D is a $m \times n$ diagonal matrix with non-negative real numbers on the diagonal

c) V is a $n \times n$ orthonormal matrix, then $V^TV = I$

Then,

$$X^T = (UDV^T)^T = VD^TU^T$$

$$X^TX = (VD^TU^T)(UDV^T) = VD^TDV^T = VD^2V^T \quad \because D \text{ is a diagonal matrix}$$

Since D is diagonal, D^TD is a diagonal matrix where each diagonal element is the square of the according element in D.

Considering the form of eigendecomposition,

$$Av = \lambda v \ , AQ = Q\Lambda \ , A = Q\Lambda Q^{-1}$$

A as a square n $\times$ n matrix, Q as a square n $\times$ n matrix whose ith column is the eigenvector q_i, Λ as the diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\Lambda_{ii} = \lambda_i$

$X^TX = VD^TDV^T$ is the form of eigendecomposition, V being the matrix of eigenvectors, D^TD $(= D^2)$ being the diagonal matrix of eigenvalues of X^TX, the D matrix position in $X = UDV^T$. Also, the eigenvectors of X^TX are the columns of V, and the eigenvalues of X^TX are the squared singular values, which is the elements of D^TD $(= D^2)$.

Problem 2

SVD of Rank Deficient Matrix.
(a) Compute the matrices M^TM and MM^T .
(b) Find the eigenvalues for your matrices of part (a).
(c) Find the eigenvectors for the matrices of part (a).
(d) Find the SVD for the original matrix M from parts (b) and (c). Note that there are only two nonzero eigenvalues, so your matrix Σ should have only two singular values, while U and V have only two columns.

(e) There are 2 non-zero singular values, if we only keep one by setting the smaller singular value to 0, then the data will be represented in 1D only. Compute such one-dimensional approximation to M.

Solution.

(a) By computing matrix M^TM and matrix MM^T, we can see the result is a square matrix of size equal to the number of columns and rows in M (3×3 matrix for M^TM and 5×5 matrix for MM^T).

$$M^TM = \begin{bmatrix} 39 & 57 & 60 \\ 57 & 118 & 53 \\ 60 & 53 & 127 \end{bmatrix}$$

$$MM^T = \begin{bmatrix} 10 & 9 & 26 & 3 & 26 \\ 9 & 62 & 8 & -5 & 85 \\ 26 & 8 & 72 & 10 & 50 \\ 3 & -5 & 10 & 2 & -1 \\ 26 & 85 & 50 & -1 & 138 \end{bmatrix}$$

(b) By using NumPy's 'np.linalg.eig' function, we can find the eigenvalues.

Eigenvalues of M^TM =
$\begin{bmatrix} 2.14670489e+02 & -1.33871866e-14 & 6.93295108e+01 \end{bmatrix}$
Eigenvalues of MM^T =
$\begin{bmatrix} 2.14670489e+02 & -4.13378832e-15 & 6.93295108e+01 & -3.28163351e-15 & -4.10937465e-16 \end{bmatrix}$

(c) Similarly, by using NumPy's 'np.linalg.eig' function, we can also find the eigenvectors.

$$\text{Eigenvectors of } M^TM : \begin{bmatrix} 0.42615127 & 0.90453403 & -0.01460404 \\ 0.61500884 & -0.30151134 & -0.72859799 \\ 0.66344497 & -0.30151134 & 0.68478587 \end{bmatrix}$$

$$\text{Eigenvectors of } MM^T : \begin{bmatrix} -0.16492942 & -0.95539856 & 0.24497323 & -0.83645189 & 0.14590955 \\ -0.47164732 & -0.03481209 & -0.45330644 & 0.32249423 & -0.1726319 \\ -0.33647055 & 0.27076072 & 0.82943965 & 0.40049076 & 0.06871429 \\ -0.00330585 & 0.04409532 & 0.16974659 & -0.03474164 & -0.97055978 \\ -0.79820031 & 0.10366268 & -0.13310656 & -0.18640247 & 0.04691152 \end{bmatrix}$$

(d) To get the singular values of M, I extracted the square roots of the eigenvalues (M^TM, MM^T) which are the squares of the singular values of M. Also, since there are only two singular values according to the question, Σ will be a diagonal matrix with these two singular values. Finally, U correspond to eigenvectors of MM^T, and V correspond to eigenvectors of M^TM. I applied a threshold of '1e-10' for numerical stability, calculated singular value Σ, and constructed U and V using the first two eigenvectors that corresponds to the nonzero eigenvalues.

$$\text{Matrix } U = \begin{bmatrix} -0.16492942 & -0.95539856 \\ -0.47164732 & -0.03481209 \\ -0.33647055 & 0.27076072 \\ -0.00330585 & 0.04409532 \\ -0.79820031 & 0.10366268 \end{bmatrix}$$

$$\text{Matrix } \Sigma = \begin{bmatrix} 14.65163776 & 0 \\ 0 & 8.32643446 \end{bmatrix}$$

$$\text{Matrix } V = \begin{bmatrix} 0.42615127 & 0.90453403 \\ 0.61500884 & -0.30151134 \\ 0.66344497 & -0.30151134 \end{bmatrix}$$

(e) According to the question, by setting the smaller singular value to 0, I only kept the largest singular value. By these conditions, I computed the one-dimensional approximation to M.

$$\text{One-dimensional approximation of M} = \begin{bmatrix} -1.02978864 & -1.48616035 & -1.60320558 \\ -2.94487812 & -4.24996055 & -4.58467382 \\ -2.10085952 & -3.031898 & -3.27068057 \\ -0.02064112 & -0.02978864 & -0.0321347 \\ -4.9838143 & -7.19249261 & -7.75895028 \end{bmatrix}$$

By calculating the SVD of M, the 1D representation is expected to capture the most dominant pattern in the dataset. The reduction results in different values for M and M_{approx}, reflecting the loss of information. However, this 1D representation is expected to perform better in visualizing and understanding high-dimensional data.