Linear Algebra and its Applications
Assignment 5
Due: 18 April 2024

In this assignment you will explore the application of SVD to an image classification problem.

MNIST set is a large collection of handwritten digits. It is a very popular dataset in the field of image processing. It is often used for benchmarking machine learning algorithms. It has a collection of $70,000, 28 \times 28$ images of handwritten digits from 0 to 9. The dataset is already divided into training and testing sets. You can use import the dataset using the Keras module. Each image is stored as vector (by unwrapping the image matrix); the first coordinate is the label (i.e., a number between 0 to 9 indicating the digit whose image the vector represents). You can print/display the image using pyplot, by wraping it using " reshape(28, 28)". Spend some time to get familiar with this dataset.

1. For each digit i form a matrix whose columns are images of that digit. Make sure that the label is not part your column; i.e., there are exactly 784 rows in each column. The dataset contains $4,000$ images of each of the digits. You may decide to take them all in your matrices or you may choose a fixed number N (sufficiently big) columns for each digit.

2. The 10 matrices formed above are in general not of full rank. For instance, consider the matrix $D5$ containing images labeled 5. The columns span a proper subspace $D(5) \subset \mathbb{R}^{784}$ that would contain all 28×28 images of 5. So, given a test image e if we find an algorithm that detects whether e is in $D(5)$ or not would lead us to a model that would solve the classification problem.

3. For each of this matrix $D(n)$ write a routine to extract the first 4 left singular vectors (i.e., the principal components). You may call function provided by numpy. Recall that these are orthonormal basis vectors of the column space of $D(n)$. Now for each digit print 5 images - the first image should be from the training data sets and for the remaining use the left singular vectors (by reshaping them into 28×28 image).

4. Finally, the classification stage. Develop your model using the following idea. Given a test image, as a vector z, one woould like to compute

$$\rho_n(z) := \min ||D(n)x - z||,$$

   where n is a digit between 0 to 9. The value of n for which $\rho_n(z)$ is the least should be the label of z. It should be clear to you that using the full matrix $D(n)$ to solve the LSP is ineffecient. This is where SVD should enter, i.e., instead of $D(n)$ you should use a low rank approximation of the factorization. Further hint, try using only first 4 or so columns of U. Moreover, we are not really interested in finding an x that minimizes the expression, we are interested in finding the minimum norm.

5. Try using between $2500 - -3000$ images of each digit in the training set to form the matrices and use the remaining as (labeled) test images.

6. The final output should contain 50 images (from Point 3 above) - one original image and 4 images corresponding to left singular vector. As markdown include the exact expression used in clssification and some justification as to why you are using that particular expression. Also, the confusion matrix and the efficiancy results.

Note: Your submission should be a PDF output of your Juypter Notebook and the python files. Make sure that unnecessary things like, error messages, trial runs etc. are not included in the submission. Do upload a compressed archive. Make sure that a PDF file is uploaded.