Math 156: Lab assignment #2

Due: Tuesday October 17th

The goal of this assignment is to learn how to use principal component analysis (PCA) to reduce the dimensionality of data. Many real datasets, such as collections of natural images, are embedded in very high dimensional spaces, however the data is usually clustered near a much lower dimensional space. One way to exploit this phenomena is to project the data onto a linear subspace of lower dimension. We will do this using PCA. For this assignment you will not need to turn in code.

1 Theory

Let $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^d$ be a dataset of N points. Recall that the sample mean is given by

$$\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \tag{1}$$

and the sample covariance matrix is given by

$$\boldsymbol{S} = \frac{1}{N-1} \sum_{n=1}^{N} (\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T. \tag{2}$$

Let $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d\}$ be the eigenvectors of the sample covariance matrix, arranged in the order so that their associated eigenvalues $\{\lambda_1, \ldots, \lambda_d\}$ are decreasing, i.e. $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$. These eigenvectors are known as the principal components of the dataset.

Task 1: Show that every data point $\boldsymbol{x}_n \in \mathcal{D}$ can be written in the form

$$\boldsymbol{x}_n = \bar{\boldsymbol{x}} + \sum_{i=1}^{d} \langle \boldsymbol{x}_n - \bar{\boldsymbol{x}}, \boldsymbol{u}_i \rangle \, \boldsymbol{u}_i.$$

Task 2: Let $V \subset \mathbb{R}^d$ be a k-dimensional subspace spanned by the orthonormal vectors $\{v_1, \ldots, v_k\}$. Show that the orthogonal projection p_V of some $\boldsymbol{x} \in \mathbb{R}^d$ onto V is given by

$$p_V(\boldsymbol{x}) = \sum_{i=1}^{k} \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \, \boldsymbol{v}_i.$$

Principal component analysis reduces the dimensionality of the data by projecting the data onto the subspace spanned by the first m principal components for some $m < d$. The PCA projection onto the first m components is given by

$$\hat{\boldsymbol{x}}_n = \bar{\boldsymbol{x}} + \sum_{i=1}^{m} \langle \boldsymbol{x_n} - \bar{\boldsymbol{x}}, \boldsymbol{u}_i \rangle \, \boldsymbol{u}_i.$$

The mean squared error of the PCA projection onto an m dimensional subspace is

$$J_m = \frac{1}{N} \sum_{n=1}^{N} ||\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n||^2.$$

Task 3: The mean squared error can also be expressed in terms of the eigenvalues of the sample covariance matrix. Show that

$$J_m = \frac{N-1}{N} \sum_{i=m+1}^{d} \lambda_i.$$

By looking at the eigenvalues of the sample covariance matrix we can decide how to choose the dimension cutoff value m.

2 Experiments

For this assignment we will work with two different datasets, the three moons dataset, and a dataset we will create from the digit_image.mat file.

Begin by downloading the three moons dataset from CCLE. Next load the data into MATLAB from the .mat file using the load command. The three moons dataset is a set of 1500 points in $\mathbb{R}^{100}$.

Task 4: Compute the sample mean and covariance matrix of the three moons dataset. Plot the eigenvalues of the covariance matrix. Is there a sudden drop off in the size of the eigenvalues? What does this say about the dimensionality of the data?

Task 5: Project the data onto the first two principal components. Create and print out a 2-D scatter plot of the *coefficients* of the data points in the two dimensional subspace given by the first two principal components. Does the name three moons make sense?

Now we will create a data set by making various rotated copies of the image in the digit_image.mat file. Download the digit_image.mat file and load the data into MATLAB. To visualize the data, use the function imshow(*your image*). The image is a 64×64 pixel image, and is thus a vector in $\mathbb{R}^{4096}$.

Important note: When you are visualizing the principal components you will need to use imshow(*your image*,[]) in order to scale the grayscale values into the correct range (otherwise your image will not look right). You will also need to convert between working with the images as a 64×64 matrix and a 4096×1 vector. You will want to use the 64×64 representation when using imshow, and you will want to use the vector representation when computing the sample mean and covariance matrix. You can convert between the representations using the reshape() function.

Task 6: Generate $N = 200$ new images, by rotating the image by an angle $\theta \in [0, 180]$, in degrees (use evenly spaced angles between 0 and 180). Note that the rotation of an image is not a trivial task, because of the discretization of the plane, necessary for digital images (you are not asked to re-code this, and can use built-in functions. Useful functions: linspace and imrotate, with the option 'crop', so as to keep a constant size, use the 64×64 matrix representation here). Print out and turn in an image of the digit rotated by 90 degrees.

Task 7: What is the inherent dimensionality of this dataset? (Hint: consider how many parameters are needed to distinguish one element of the dataset from another).

Task 8: Compute the sample mean and sample covariance matrix of the dataset you just created, then compute the principal components of the dataset. Explain why it is not necessary to compute all of the eigenvectors if $N < d$, where $d = 4096$ is the dimension of the image space.

Task 9: Display the first few principal components of the dataset. What do these components look like? Do they make sense in the context of the dataset you generated?

Task 10: Reconstruct the original image using only the first 3 principal components. In other words compute the vector

$$\hat{\boldsymbol{x}}_1 = \bar{\boldsymbol{x}} + \sum_{i=1}^{3} \langle \boldsymbol{x}_1 - \bar{\boldsymbol{x}}, \boldsymbol{u}_i \rangle \, \boldsymbol{u}_i$$

where $\boldsymbol{x}_1$ is the original image. Does $\hat{\boldsymbol{x}}_1$ look like $\boldsymbol{x}_1$ when you use imshow? Is it reasonable to project the data onto a 3-dimensional subspace?

Task 11: Approximately how many principal components do you need to make the projection $\hat{\boldsymbol{x}}_1$ look like the original $\boldsymbol{x}_1$? Is this surprising given your answer to task 6?