Homework 9

Due Apr 9 at 11 pm

Unless stated otherwise, justify any answers you give. You can work in groups, but each student must write their own solution based on their own understanding of the problem.

When uploading your homework to Gradescope you will have to select the relevant pages for each question. Please submit each problem on a separate page (i.e., 1a and 1b can be on the same page but 1 and 2 must be on different pages). We understand that this may be cumbersome but this is the best way for the grading team to grade your homework assignments and provide feedback in a timely manner. Failure to adhere to these guidelines may result in a loss of points. Note that it may take some time to select the pages for your submission. Please plan accordingly. We suggest uploading your assignment at least 30 minutes before the deadline so you will have ample time to select the correct pages for your submission. If you are using LaTeX, consider using the minted or listings packages for typesetting code.

1. (Normalization) In this problem we study the effect of normalizing by the standard deviation before performing principal component analysis. Let $\tilde{x}$ be a zero-mean 3-dimensional random vector with covariance matrix

$$\Sigma_{\tilde{x}} := \begin{bmatrix} 100 & 25 & 0 \\ 25 & 400 & 0 \\ 0 & 0 & 0.16 \end{bmatrix}. \tag{1}$$

We define the normalized vector $\tilde{y}$ as

$$\tilde{y}[i] := \frac{\tilde{x}[i]}{\sqrt{\mathrm{Var}(\tilde{x}[i])}} \quad 1 \leq i \leq 3. \tag{2}$$

(a) Compute the covariance matrix of $\tilde{y}$.

- observe that the random vector $\tilde{y} \in \mathbb{R}^3$ can be written as $\tilde{y} = v * \tilde{x}$ that is as the element-wise product of the original random vector $\tilde{x} \in \mathbb{R}^3$ and the vector $v = \begin{pmatrix} \frac{1}{\sqrt{\tilde{x}[1]}} \\ \frac{1}{\sqrt{\tilde{x}[2]}} \\ \frac{1}{\sqrt{\tilde{x}[3]}} \end{pmatrix}$
- then we can consider each of the elements of the new covariance matrix $\Sigma_{\tilde{y}}$
- note that $\Sigma_{\tilde{y}}[i,j] = cov(\tilde{y}[i], \tilde{y}[j]) = cov(\frac{1}{\sqrt{\tilde{x}[i]}}\tilde{x}[i], \frac{1}{\sqrt{\tilde{x}[j]}}\tilde{x}[j]) = E[\frac{1}{\sqrt{\tilde{x}[i]}}\tilde{x}[i] * \frac{1}{\sqrt{\tilde{x}[j]}}\tilde{x}[j]] - E[\frac{1}{\sqrt{\tilde{x}[i]}}]\tilde{x}[i]E[\frac{1}{\sqrt{\tilde{x}[j]}}\tilde{x}[j]] = \frac{1}{\sqrt{\tilde{x}[i]}}\frac{1}{\sqrt{\tilde{x}[j]}}(E[\tilde{x}[i]\tilde{x}[j]] - E[\tilde{x}[i]]E[\tilde{x}[j]])$ $= \frac{1}{\sqrt{\tilde{x}[i]}}\frac{1}{\sqrt{\tilde{x}[j]}}cov(\tilde{x}[i], \tilde{x}[j])$
- computing this out we get $\Sigma_{\tilde{y}} = \begin{pmatrix} 1 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

(b) Is the directional variance of $\tilde{y}$ equal to one in every direction?

- this is false
- consider the vector $a = \begin{pmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{pmatrix}$
- we know the variance of $\tilde{y}$ in the direction of a is given by the variance of the projection of $\tilde{y}$ onto a
- the projection $P_a(\tilde{y}) = \frac{a^t \tilde{y}}{||a||} = a^t \tilde{y}$
- then we know $var(P_a(\tilde{y})) = var(a^t \tilde{y}) = a^t \Sigma_{\tilde{y}} a$
- then computing this out we can see $var(P_a(\tilde{y})) = 1.08333333 \neq 1$

(c) We decide to reduce the dimensionality of $\tilde{x}$ and $\tilde{y}$ to two dimensions using PCA. Report what directions are selected for each of the random vectors. (Feel free to use a computer for your calculations, but explain what you are doing.)

- given we would like to reduce the dimensionality of our random vector $\tilde{x} \in \mathbb{R}^d$ to $k \leq d$
- our steps are first to find the covariance $\Sigma_{\tilde{x}}$
- then we compute an eigendecomposition of $\Sigma_{\tilde{x}}$ and take it's top k eigenvectors $u_1 \dots u_k$, as by the spectral theorem these are the directions of maximal variance for the random vector $\tilde{y}$
- then we project $\tilde{y}$ back onto those directions to get the k principal components of $\tilde{y}$ $p_1 ... p_k = P_{u_1}(\tilde{x}) \dots P_{u_k}(\tilde{x}) = \frac{u_1^t \tilde{x}}{||u_1||_2} \dots \frac{u_k^t \tilde{x}}{||u_k||_2} = u_1^t \tilde{x} \dots u_k^t \tilde{x}$
- we can apply this methodology to our problem by looking at the random vector $\tilde{y}$ with k=2
- we can compute the eigenvalues of our covariance matrix as $\begin{pmatrix} 1.125 \\ 0.875 \\ 1. \end{pmatrix}$
- and see that $1.125, 1$ are our largest two eigenvalues
- then we can find the principal directions of our data as there associated eigenvectors which are $\lambda_1 = 1.125, u_1 = \begin{pmatrix} 0.70710678 \\ 0.70710678 \\ 0 \end{pmatrix}$ and $\lambda_2 = 1, u_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

(d) Explain which of the two options for dimensionality reduction to 2D would make more sense in each of the following situations and why that is the case: (1) The entries of $\tilde{x}$ represent the weight (in kilograms), heart rate (in beats per minute), and height (in meters) of a set of hospital patients. (2) The entries of $\tilde{x}$ represent the length, width and height (all in centimeters) of a set of cars.

- i think dimensionality reduction makes more sense in case a.
- in case a we are dealing with health data. each of our quantities is potentially related that is for example someones weight and height could have notable information about there heart rate. Further, in case a we are trying to understand

health outcomes so reducing this data to principal directions that may be a mixture of features could tell us something integrable about health care outcomes

- option b on the other hand is dealing with the dimensions of objects (in this case cars) and there may not be a strong corelation between the features. For instance knowing that a car is long may not tell you anything about its height or width for instance both a traditional limo and jeep limo are abnormally long, but they have very different heights and widths. This further implies that in the case of objects taking principal directions that may be a combination of multiple features may not be useable. Further, the data in this set already exists in a form that can be understood in 3 dimensions so dimensionality reduction is kind of pointless.

2. (Faces) The following questions refer to the code in the folder `faces` The Olivetti faces dataset used in `faces` contains images of faces of people associated with a unique numeric id to identify the person.

   (a) Complete the `compute_nearest_neighbors()` function in `nearest_neighbors.py` that finds the image in the training data that is closest to a given test image. Include the generated images in your submitted homework.

   Create a new file in which you must write code to complete the following tasks:

   (b) Generate a plot of k vs. σ_k^2, where σ_k^2 is the variance of the kth principal component of the data (e.g., σ_1^2 is the largest variance). Include the plot in your submitted homework document. You can limit the x axis to a reasonable number.

   (c) Plot (using `plot_image_grid()` in `plot_tools.py`) the vectors corresponding to the top 10 principal directions of the data. Your principal direction vectors should be in $\mathbb{R}^{4096}$ (they represent images). Include the plot in your submitted homework document.

   (d) Use the variance of principal directions plot to determine a relatively small number k of principal components that explains the training data reasonably well. Project the training data and the test data onto the first k principal components, and run nearest neighbors for each test image in this lower dimensional space. Include your choice for k, and the plots of your nearest neighbor results in your submitted homework document. You should use the code from `nearest_neighbors.py` to generate your image plots.

   (e) Give a reason why applying nearest-neighbor approach after performing dimensionality reduction could potentially produce better results.

Some notes to keep in mind:

(a) The function `np.linalg.eig` might return complex eigenvectors.

(b) The data points in the training and test data are given as rows.

(c) Include all new code (or functions) you have filled in your final PDF.