CS6923 Machine Learning, Fall 2019
Prof. Linda Sellie, NYU School of Engineering

Homework 7

Submit on NYU Classes by Friday. Nov. 1 at 8:00 p.m. You may work together with one other person on this homework. If you do that, hand in JUST ONE homework for the two of you, with both of your names on it. You may *discuss* this homework with other students but YOU MAY NOT SHARE WRITTEN ANSWERS OR CODE WITH ANYONE BUT YOUR PARTNER.

IMPORTANT SUBMISSION INSTRUCTIONS: Please submit your solutions in 3 separate files: one file for your written answers to Part I, one file for your written answers/output for the questions in Part II, and one file with your code (a zip file if your code requires more than one file).

Part I: Written Exercises

The written questions below will use the following matrices:
diagonal matrix D, orthogonal matrix V, and zero centered data matrix X

$$D = \begin{bmatrix} 157 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad V = \begin{bmatrix} -0.477 & 0.522 & 0.480 & 0.520 \\ 0.476 & -0.521 & 0.521 & 0.480 \\ 0.561 & 0.475 & -0.479 & 0.480 \\ -0.480 & -0.479 & -0.519 & 0.520 \end{bmatrix} \quad X = \begin{bmatrix} 4.5 & -3.5 & -3. & 1.5 \\ 1.5 & -2.5 & -4. & 4.5 \\ -3.5 & 4.5 & 2. & -2.5 \\ -2.5 & 1.5 & 5. & -3.5 \end{bmatrix}$$

1. Using D from above, which $\mathbf{v} \in \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.71 \\ 0.50 \\ 0.50 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.50 \\ 0.50 \\ 0.50 \\ 0.50 \end{bmatrix} \right\}$ will maximize $\mathbf{v}^T D \mathbf{v}$?

   Could you find a better $u$nit vector $\mathbf{v} \in \mathbf{R}^4$ than the one you selected as the answer for this question?

   (FYI $(0.71 * 0.71) \approx 0.50$.)

2. Using the orthogonal matrix V defined above, what is the value of $V^T\mathbf{v}$

   for every $\mathbf{v} \in \left\{ \begin{bmatrix} -0.477 \\ 0.476 \\ 0.561 \\ -0.480 \end{bmatrix}, \begin{bmatrix} 0.522 \\ -0.521 \\ 0.475 \\ -0.479 \end{bmatrix}, \begin{bmatrix} 0.480 \\ 0.521 \\ -0.479 \\ -0.519 \end{bmatrix}, \begin{bmatrix} 0.520 \\ 0.480 \\ 0.480 \\ 0.520 \end{bmatrix} \right\}$.

   Round your answer to 3 decimal places[1]

[1]Hint: Think of what the answer should be. Verify you are right by performing the calculations using a calculator. You will encounter round-off error.

3. Given $A = X^T X = VDV^T$ where X, V and D are defined above. Which unit vector $\mathbf{v} \in \mathbb{R}^4$ maximizes $\mathbf{v}^T A\mathbf{v}$.

   Remember: $\mathbf{v}$ is a *unit* vector if $\|\mathbf{v}\|_2 = 1$.

4. Given $\frac{1}{N}A = \frac{1}{N}X^T X = \frac{1}{N}VDV^T$ where X, D and V are defined above.

   Determine $\mathbf{v}^* = \arg\max_{\mathbf{v}, \|\mathbf{v}\|_2 = 1} \mathbf{v}^T \frac{1}{N} A\mathbf{v}$.[2]

5. If you wanted to project all the points in X onto a single line $\mathbf{v} \in \mathbb{R}^4$ so they had maximum variance, which line should you project the points onto?

   Project all the points in X onto the first column of V and compute the variance of the points after you projected them on to the line.

   Can you find another line to project the points onto such that the points would have larger variance? If so, find this line. If you cannot find a better line, choose any other line $\mathbf{v}' \in \mathbb{R}^4$, and project all the points onto this line, and compute the variance of the points after you projected onto this line.

   Remember, the formula for projecting a point $\mathbf{x}$ onto a line is $\frac{(\mathbf{x}^T\mathbf{v})}{\|\mathbf{v}\|}\mathbf{v}$ (Notice that if $\mathbf{v}$ is a unit vector then the formula to project a point $\mathbf{x}$ onto a line is $\frac{(\mathbf{x}^T\mathbf{v})}{1}\mathbf{v} = (\mathbf{x}^T\mathbf{v})\mathbf{v}$.). Therefore the distance from the origin to the point is $|\mathbf{x}^T\mathbf{v}|$.

   If we were going to classify the points in X, and we only wanted to use one feature. Which feature should we use if we thought the best feature would be the one where the examples had the highest variance?

6. The *total variance* of a data set X is the sum of the variances of all the principle components. If we project onto the top k principle components when performing PCA, the amount of variance remaining is the amount of variance of the top k principle components. [3]

   The proportion of variance maintained after projecting onto the top k principle components is $\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^d \lambda_i}$ when $\frac{1}{N}X^T X = VDV^T$ such that V is an orthogonal matrix, and $D = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{bmatrix}$

   where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$.

   For the previous problem. How much of the original variance do we retain have after projecting onto the top two principle components?

[2] i.e. Which unit vector $\mathbf{v} \in \mathbb{R}^4$ maximizes $\mathbf{v}^T \frac{1}{N} A\mathbf{v}$.

[3] Learn more at: `https://ro-che.info/articles/2017-12-11-pca-explained-variance`

Part II: Programming Exercises

7. Load the python notebook `homework 7 PCA.ipynb`. The first part of the notebook is a tutorial on PCA. At the bottom of the tutorial you will find a section called **Homework**. Complete the following:

   (a) Display the fourth face in the dataset using the appropriate command above. Print the resulting display.

   (b) Compute the mean of all the examples in the dataset fea. (That is, compute an image such that each pixel i of the image is the mean of pixel i in all the images in fea.) Display it using a modification of the above command. Give the Python commands you used and show the resulting display.

   (c) Let's do dimensionality reduction with pca. Using Python, compute the 5 top principal components of the data matrix fea. Give the Python commands you used. What are the values of the associated 5 attributes of the fourth image in the dataset?

   (d) Project the fourth face in the dataset onto the first 5 principle components.

   (e) Project the fourth face back into the original space.

   Give the Python commands you used. Display the resulting approximate image and print the resulting display. Repeat with the first 50 principle components (instead of 5).

   (These are representations of the fourth image, based on 5 or 50 features instead of the original 2914 pixel features.)