

CARNEGIE MELLON UNIVERSITY  
DEPARTMENT OF COMPUTER SCIENCE  
15-826 MULTIMEDIA DATABASES AND DATA MINING  
C. FALOUTSOS, FALL 2019

Due: hard copy, in class, at 1:30pm, on 11/13/2019  
No tarball.

**VERY IMPORTANT:**

- For each question, we expect only the **hard copy** of answers and code.
  1. **Separate** your answers, on different page(s) for each question
  2. **Type** the full info on **each** page: your **name**, **Andrew ID**, **course#**, **Home-work#**, **Question#** on each of the pages.

**Reminders:**

- *Plagiarism*: Homework is to be completed *individually*.
- *Typeset* your answers. Illegible handwriting may get zero points.
- *Late homeworks*: follow usual procedure: please email it
  - to all TAs and graders
  - with the subject line exactly 15-826 Homework Submission (HW 4)
  - and the count of slip-days you are using.

**For your information:**

- Graded out of **100** points; **3** questions total
- Rough time estimate: *12-18 hours ( $\approx$  4-6 hours per question)*

*Revision : 2019/11/05 23:03*

Question	Points	Score
Eigen values power method	35	
SVD - Visualization	30	
Fourier and wavelets	35	
Total:	100	

**Code packaging info:**

As before, for your convenience, we provide a *tar-file package*, at <http://www.cs.cmu.edu/~christos/courses/826.F19/HOMEWORKS/HW4/hw4.tar.gz>. We will refer to it as the *tar-file package* from now on. It has 3 directories /Q1, /Q2, /Q3.

**Question 1: Eigen values power method ..... [35 points]**

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

**Motivation:** As we have seen in class SVD has many practical applications. A common method of computing SVD for a given matrix  $A$  is by computing the eigen values of  $A^T A$  and  $AA^T$ .

If we want only the first (= dominant) eigenvalue and eigenvector, then we can use the *power iteration* method: we can multiply a random vector  $\vec{r}$  with the matrix, multiple, consecutive times, and the resulting vector will be very close to the corresponding eigenvector  $\vec{u}_1$  (times a large scalar).

**Problem Description:** Implement the power iteration method to compute the dominant eigen value and vector for a given matrix  $A$ .

- (a) [25 points] Give the code for the `power_iteration` in `./Q1/power_iteration.py`

.....  
 .....

- (b) [10 points] Compute the dominant eigen value  $\lambda_1$  and the corresponding eigen-vector  $\vec{u}_1$  for the following matrices.

$$A = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

.....  
 .....

**What to turn in:**

- **Answers:** Hard copy of the code `./Q1/power_iteration.py` and the answers for part (b).

## Question 2: SVD - Visualization ..... [30 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

**Motivation:** Very often in our career as a data analyst, we are given a cloud of  $N$  points in  $M$  dimensions, and we have to find patterns, clusters, anomalies. If the dimensionality  $M$  is high, it is hard to plot and visualize the dataset. Here we see how to reduce the dimensionality, and how to find patterns and anomalies.

**Problem Description:** In this problem, we will use the Singular Value Decomposition (SVD) to explore such a cloud of points.

Consider the 6-dimensional *mystery* dataset `./Q2/mystery.dat` in *tar-file package*. The  $N$  data points lie in a lower dimensionality hyper-plane of dimensionality  $k$  - you have to guess  $k$  and project the points into a  $k$ -dimensional (hyper-)plane, using SVD. Specifically, we are told that the  $i$ -th mystery data point  $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,6})$  was generated by the equations:

$$\begin{aligned} x_{i,1} &= a_1 * y_{i,1} + a_2 * y_{i,2} + \dots + a_k * y_{i,k} + \epsilon_{i,1}, \\ x_{i,2} &= b_1 * y_{i,1} + b_2 * y_{i,2} + \dots + b_k * y_{i,k} + \epsilon_{i,2}, \\ &\dots \\ x_{i,6} &= f_1 * y_{i,1} + f_2 * y_{i,2} + \dots + f_k * y_{i,k} + \epsilon_{i,6}, \end{aligned}$$

where  $\mathbf{y}_i = (y_{i,1}, \dots, y_{i,k})$  is the  $i$ -th point on the  $k$ -dimensional hyper-plane ( $k \leq 6$ ). The coefficients  $a_1, \dots, a_k, b_1, \dots, f_k$  are constant for all the  $N$  points in the dataset, and  $\epsilon_{i,j}$  indicates a small amount of noise.

Answer the following questions using SVD. We recommend MatLab.

- (a) **[6 points]** Guess: What is the dimensionality  $k$  of the mystery dataset? (Do NOT use the fractal dimension - it is not the right tool to guess  $k$ .)
- (b) **[2 points]** Give the singular values  $(\lambda_1, \dots)$  of the matrix  $X = (x_{i,j})$
- (c) **[2 points]** Briefly justify your answer for your guess for  $k$ .
- (d) **[10 points]** If  $k \leq 2$ , give the scatter-plot (of first, vs second, principal components). If  $k > 2$ , give all the pair-plots, that is, the scatter-plots of all the  $k$ -choose-2 possibilities.
- (e) There are two outliers in the dataset. Find those two outliers by manually looking at your scatter-plot(s).
  - i. **[4 points]** Mark them and hand in the resulting plots
  - ii. **[6 points]** report the (6-dimensional) coordinates of the two outliers.

**What to turn in:**

- **Answers:** Submit hard copy for the answers.

**Question 3: Fourier and wavelets ..... [35 points]**

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

**Motivation:** Digital Signal Processing (DSP) and specifically the Discrete Fourier (DFT) and Discrete Wavelet (DWT) transforms, are powerful tools for de-noising, anomaly detection and feature extraction in time sequences. Here we demonstrate

- how they help us spot outliers, by extracting valuable features (frequencies, amplitudes), from periodic time sequences like natural sounds (flying insects), and
- how they can help us discover signals buried inside noise, like a phone conversation in a noisy street.

**Problem Description:** You will analyze the signal `./Q3/signal_with_noise.mat` using DFT (also called FFT) and wavelets to detect a high frequency injection which occurs for a short duration. The signal is a 1-d time series of 2500 samples that were sampled at a frequency of 4000 Hz, and it is a mixture of sine/cosine functions, plus the short-lived injection.

(a) DFT/FFT analysis:

- [10 points] Plot the spectrum, i.e., the frequency and amplitude plot using `./Q3/wave_analysis.py`.
  - [5 points] Report the main frequencies and their amplitudes you see in the plot. Can you spot the frequency of the injection? (It's OK if not).
- .....  
.....

(b) Wavelet analysis: The DFT/FFT spectrum can not indicate the start-end of the high-frequency injection. It can only provide information about the overall frequencies the signal wavelet transforms also localize the frequencies in the signal in time.

- [10 points] Plot the Wavelet scaleogram using `wavelet_scaleogram.m`.
  - [10 points] Report the approximate start-time, end-time, and approximate frequency of the injection.
- .....  
.....

**What to turn in:**

- **Answers:** Submit hard copy for the answers and the plots.