Linear Algebra Assignment - II

TAs: Anirban, Soham, Tony

October 15, 2018

Instructions

- This is a coding assignment. The code has to be written in Python. You can use Python version 2 or 3 to solve all the problems. Please mention the version you have used in the report.
- The assignment deliverable is your python code, results, plots and your report of observations.
- Zip and send your solution to **E0226@outlook.com** with the following format for both zip file name and subject name of email: `LA_2_<last 5 digits of serial number>.zip`.
- You are encouraged to discuss among yourselves, but **DO NOT COPY** solutions or code. The consequences will be severe.
- Read the complete assignment carefully, before attempting to solve it.
- Follow the instructions provided for how your code is to be run, the formats for input and output files and the naming conventions. — Detailed Instructions
- The submission deadline is **November 15, 2018**.

Best of luck for the assignment. HAPPY CODING!

Problem - I
A Network of Friends and Foes in GoT

Suppose in the final season of the popular TV show "Game of Thrones", the directors decided to introduce one more House – the House of Algebrician (House Words: "*Keep calm and do algebra linearly!*")! The inhabitants of this House are quite fond of mathematics, especially the subdomain of algebra, namely, Linear Algebra. They want to analyze the friendship (or the closeness of interaction) between the houses and how strongly they are connected to each other. But, unlike most of the other houses, they do not have any "birds". Hence, not much

secret information about the other houses is known a priori. The only resource they have is the interaction network among different houses in the form of a graph. Being passionate about linear algebra, they decided to analyze certain aspects, such as who plays the central role in between the Houses or inside one House, and whether any small groups are formed. They decide to use the concept of eigenvalues and eigenvectors to detect the formation of communities. They also want to know which connections (edges) are more important than the rest, so that they can try to break that bond if they want to break two groups apart. Additionally, they want to find, among the communities, which characters are central!

You are given a part of the entire friendship network in the form of an unweighted and undirected graph. The graph is in the `gml` format. Use the `networkx` package to read it.

Task - I (0.5 Point)

Write code to obtain the degree distribution of the nodes. Plot it. What can you infer from the distribution? Does it follow any familiar distribution?

Task - II (0.5 Point)

Calculate the node centrality using any one from the different standard node centrality measures. Plot a histogram, taking uniform interval from the entire range of centrality values obtained. Name the top two central nodes of the given graph.

Task - III (0.5 Point)

Find the edge centrality using any centrality measure for all edges. Which edge is most central?

Task - IV (2 Points)

Create the **adjacency matrix** of the given network. Find out the **Laplacian matrix** and normalized Laplacian matrix. Calculate the eigenvalues and eigenvectors of the Laplcian. You are **not allowed** to use any library function to find the eigenvalues and eigenvectors. You can only use libraries to find the roots of a polynomial equation (one option is to use `numpy.roots`). Check if the eigenvalues are real or not.

Task - V (0.5 Point)

Note down the the smallest two eigenvalues and their corresponding eigenvectors. Ideally, you should get the smallest eigenvalue to be (approximately) zero and components of eigenvector to be all (approximately) one. Tally these values and report how much your answer differs from the ideal values.

Task - VI (0.5 Point)

Take the eigenvector corresponding to the second smallest eigenvalue obtained earlier. Find two clusters in the graph based on positive and negative entries of the eigenvector. Try out a different technique other than this which you think more appropriate to divide these entries into two groups. Give two different colors to the nodes and draw the graph. Save it. What do you observe? Can you infer something from it? Use `networkx.draw` for visualization.

Task - VII (0.5 Point)

The House of Algebrician wants to connect with that house which contains more information. If the information is measured in terms of the variability in the data, which house would they want to make friends with?

Bonus - I (0.25 Point)

Use `numpy` to find the eigenvectors and cluster them into two groups. Check to see if you find any difference.

Bonus - II (0.25 Point)

What would you do if you are asked to find three or more clusters? How can you extend this idea to find more than two clusters in a graph?

Bonus - III (0.25 Point)

What do you observe if you take the eigenvector corresponding to the second largest eigenvalue instead of the second smallest for the clustering process? As before, color and draw the network.

Bonus - IV (0.25 Point)

How do you relate the central edge in the graph to the clustering obtained?

Important : You can use `networkx` to read and draw the graph. You may have to use `matplotlib` along with `networkx` for plotting!

Input Data: The input for this problem will be a friendship network. The network will be given in the `gml` format. All the necessary information about the graph is stored into the `gml` file.

Output Data: Save all your plots in a directory. Name it `output_plots_1`. Store all the data asked for in a text file in an understandable manner. Naming convention is mentioned later. Write everything in the report file, both the results and graphs for the given network. We'll test your code on some different network.

Problem-II
Identify the Assassin of Tywin!

The House of Lannisters was ruling the King's Landing, the capital of Westeros, with the extremely powerful Tywin Lannister serving as the king. Everything was going smoothly for the Lannisters, but by a sudden turn of events, one fine morning, Tywin Lannister was found dead in his room in Red Keep, with an arrow stuck deep in his chest! The time of the death is suspected to be midnight, when everybody was probably asleep. Not a single clue was left by the murderer, except for the arrow! Now, the only way to find the killer was to identify whose arrow it was. Interestingly, the arrow has a particular number inscribed on it! And there are exactly ten people who use this kind of specialized arrows in all of King's Landing! Each person's arrow has a unique number (from 1 to 10) inscribed on it. The only problem is that the number on the arrow used by the killer is very ambiguous-looking and is extremely difficult to identify, even for an expert!

By this time, the House of Lannisters has developed a friendship with the House of Algebricians, and thus they sought their help to solve this mystery! You, being an expert in image recognition and mathematics (read: Linear Algebra ☺), are given a set of images of all possible numbers from the manufacturer of these arrows and the murderer's arrow. Identify the assassin!

Task - I (0.5 Point)

Find the mean and covariance matrix of the given data. Don't use any library function to calculate these.

Task - II (0.5 Point)

Obtain the eigenspaces of the given data. How many eigenvalues are repeating? What can you say about the eigenvectors obtained for different eigenspaces? Do they turn out to be orthogonal?

Task - III (1 Point)

Use the process of **Gram-Scmidt orthogonalization** to get the orthogonal eigenvectors of different eigenspaces (eigenvalues with multiplicity more than one). You will be given a set of vectors as input and the output should be a set of orthogonal vectors.

Task - IV (0.5 Point)

Pick M eigenvectors corresponding to the top-M eigenvalues obtained from the covariance matrix. Use these vectors to convert the original data into M-dimensional space (the original data is D-dimensional and $M \leq D$).

Task - V (0.5 Point)

Calculate and plot reconstruction error for different values of M. Set the rest of the $(D - M)$ dimensions to be zero.

Task - VI (1 Point)

Find the mean of each column of the given data and subtract it from original data. This is called *Data Centering* in statistics. Divide your dataset into training and validation sets using cross validation technique. Use K-Nearest Neighbor to classify the validation points. Use euclidean distance or cosine similarity for calculating the nearest neighbour. Report the accuracy for different values of K and M. Plot the results. What do you observe as you vary K and M? Finalize your model fixing K and M which give best accuracy.

Bonus - I (0.5 Point)

Project the vectors into 2D space and plot them. Do you see any pattern in the plot?

Bonus - II (0.5 Point)

Use K-NN classifier available in `sklearn` and compare your results.

Optional: Try out CNN (you can use `keras` library) for classification and see the results! Sample code can be found here.

Input Data: In this assignment we use a subset of the MNIST dataset of cardinality $11,000$, with each sample being 784-dimensional. For training, we use $10,000$ samples, and for testing, $1,000$. (Note: In the provided files, the first column has digit labels. Ignore this column when you reconstruct the matrix.)

Output Data: Save all your plots in a directory. Name it `output_plots_2`. Store all the data asked for in a file in a well-documented manner. Naming conventions are stated in the next section. Report your findings in a report file, including both the results and graphs for the given data. We'll test your code on some different test images.

How to run

You need to write your code in two separate Python files for two problems - `first_problem.py` and `second_problem.py`. You should print your output to **the standard output, as well as to text files** for the two problems. The naming convention for the output files is mentioned below:

Command to run the first problem:
`python first_problem.py <input_file_path>`

Command to run the second problem:
`python second_problem.py <test_data_file_path>`

Command to run the Gram-Schimdt process for the second problem:
`python second_problem.py -type=gram-schimdt <test_data_file_path>`

File naming convention

- The name of output file for the first problem needs to be `output_problem1.txt`. Put it into the `output_data/` directory.
- The name of output file for the second problem needs to be `output_problem2.txt`. Put it into the `output_data/` directory.
- The name of the plots asked for should follow the following naming convention: `problem_i_task_j.png`. For example, the filename for task-I in the first problem should be `problem_1_task_1.png`. Put these plots into the `output_plots/` directory.
- Prepare a report and name it `LA_Report_2.pdf`. It should contain the plots and output data for the given network and datasets.

Sample Input and Output

Please look at the input and output files provided along with for the precise format.

Second Problem

Gram-Schimdt Orthogonalization

Input

1 0 0 0 0
0.71 0.71 0 0 0
0.58 0.58 0.58 0 0
0.50 0.50 0.50 0.50 0
0.45 0.45 0.45 0.45 0.45

Explanation of input

The input data given above is basically the coefficients of five-dimensional vector. There are a total five of these vectors. Your task is to use gram-schimdt orthogonalization procedure to convert it to set of orthogonal vectors. These

vectors may not be unique. A possible set of coefficients of orthogonal vectors is mentioned below.

Output

1.00 0.00 0.00 0.00 0.00
0.00 0.71 0.00 0.00 0.00
0.00 0.00 0.58 0.00 0.00
0.00 0.00 0.00 0.50 0.00
0.00 0.00 0.00 0.00 0.45