BLG202E Numerical Methods in Comp. Eng.

Spring 2025 - Term Project

Due: May 20$^{\text{th}}$, 2025

By turning in this assignment, I agree by the ITU honor code and declare that all of this is my own work.

Important Notes

- Upload your solutions through **Ninova**. Homeworks sent via e-mail and late submissions **will not be accepted**.
- Please make sure that you write your **full name** and student identification **number** to **every file** you submit.
- Cheating is highly discouraged. It will be punished by a negative grade. Also disciplinary actions will be taken. Please do your homework on your own. Team work is not allowed. Pattern of your **solutions must belong only to you**.
- All codes and reports will be run through **plagiarism check**s. Please **do not copy any text or code** from other sources.
- If you have any questions, please contact with **Ali Esad Uğur** (ugura20@itu.edu.tr).
- Remember, there are only 10 types of people in the world – those who understand binary, and those who don't.

Unnormalized Spectral Clustering

Please read the following prerequisites.

- Install a Conda environment if you do not have it already.
- Install Jupyter Notebook (Optional).
- The project should be done in a Python notebook. You may use Jupyter Notebooks or Google Colab.
- Do not forget to format your code and leave comments for non-trivial sections.
- You may use matplotlib, numpy / SciPy, json and pandas libraries.

Problem Description

In this project, you will implement the Unnormalized Spectral Clustering algorithm described in the fifth page of [1]. It is an algorithm to cluster a graph into partitions that are highly related inside. You will use your knowledge on eigenvalues that you learned in the course.

To apply your implementation, you will use the roadmap graph of our campus. You will get it in your code as shown:

```
import osmnx as ox

# There are spaces between words
place_name = "Istanbul Teknik Universitesi, Sariyer, Istanbul, Turkey"
G_roadmap = ox.graph_from_place(place_name, network_type='walk')

# You can visualize the graph (Optional step)
ox.plot_graph(G_roadmap)
```

You will notice that in the final step of the algorithm, KMeans clustering is necessary. You can do this using the KMeans class from the sklearn.clustering library.

You are expected to perform experiments with different number of clusters. Two types of results will be delivered: The first one is Modularity. It is a metric used to measure the effectiveness of a clustering. It is calculated with the provided formula:

$$Q = \frac{1}{2*m} * \sum_{i,j}(A_{ij} - \frac{k_i k_j}{2*m})\delta(c_i, c_j)$$

Where:

- A_{ij} is the ij'th element of the adjacency matrix.
- k_i is the degree of node i.
- m is the total number of edges.
- c_i is the cluster label of node i.
- $\delta(c_i, c_j)$ is 1 if nodes i and j belong to the same cluster, 0 otherwise.

And the second result will be the visualization of the clustered graph. You may use this code block for it:

```
import matplotlib.pyplot as plt
import networkx as nx

# Visualization
plt.figure(figsize=(10, 8))
# Get 'x' and 'y' attributes for each node
x_pos = nx.get_node_attributes(G_roadmap, 'x')
y_pos = nx.get_node_attributes(G_roadmap, 'y')

# Combine 'x' and 'y' into a single 'pos' dictionary
pos = {node: (x_pos[node], y_pos[node]) for node in G_roadmap.nodes if
                                            node in x_pos and node in y_pos}

# Now you can draw the graph with the updated 'pos' dictionary
nx.draw(G_roadmap, pos, node_color=y_pred, cmap=plt.cm.tab10,
        node_size=50, edge_color='gray', alpha=0.5)

plt.title(f"Spectral Clustering on ITU RoadMap Network")
plt.show()
```

And finally, you will cluster the graph with regular KMeans only to show the effect of eigenvalue operations in the spectral clustering algorithm on the cluster performance. As instructed above, you may use the KMeans class from sklearn.clustering library.

Instructions

- Re-implement Unnormalized Spectral Clustering algorithm from scratch.
  - Read the first four sections of [1] carefully to understand the algorithm. You do not have to read the full paper.
  - Implement the algorithm based on the one given at the fifth page of the paper.
  - Read the documentations of osmnx and networkx libraries of python to better understand the graph objects that you will work on.
  - Do not use any builtin functions to get the eigenvectors. You may implement the power method that you learned during the lectures.
- Report:
  - Write a maximum of 3 pages report using IEEE Latex Template. State the problem, implementation details, available data and the experiments.
  - Try to add visual and numeric results to your report.
- Submission
  - Submit a python notebook that includes your implementation, necessary visualizations and results through Ninova.
  - Upload your project report until the deadline through Ninova.
- Assesment Criteria:
  - Implementation Quality (Correctness of implementation, comparative analysis, visualization clarity) - **80%**
  - Report - **20%**

References

[1] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.