Introduction

Assignment 2: Social and Large Scale Networks

This project provides implementation and analysis of class-learned concepts to generate graphical representations using NetworkX, Matplotlib, and Numpy packages in Python. The program computes cluster coefficients and neighborhood overlaps within an input graph. The program also partitions graphs in order to emphasize primary groups and store them in a new graph.

Pre-Requisites

I recommend running the program in the terminal on Windows.

You must have the latest version of Python installed. For installation, visit:
https://www.python.org/downloads/

Additionally, ensure that the NetworkX, Matplotlib, and Numpy packages are installed.
To do so, you can install using pip:

```
pip install networkx matplotlib numpy
```

To Run the Program:

1. On windows, start by opening your Powershell or Command Prompt window.
2. Use the 'cd' command to navigate to the directory where the python file is located.
   ```
   cd path/to/directory
   ```
   You should now be located in the directory where the python file is saved.
3. Program name:
   ```
   graph_analysis.py
   ```
4. Command inputs follow this format:
   ```
   python ./graph_analysis.py graph_file.gml
   --components n --plot
   [C|N|P] --verify_homophily
   --verify_balanced_graph --output
   out_graph_file.gml
   ```

   DESCRIPTION OF COMMANDS:

   ```
   graph_file.gml        : Input GML file representing the
   graph to analyze.
   --components n        : Partition the graph into n
   components (subgraphs/clusters).
   ```

```
--plot [C|N|P]           : Plot the graph in different
styles:
C                        : Clustering coefficient.
N                        : Neighborhood overlap.
P                        : Color nodes by attribute or
default.
--verify_homophily       : Test for homophily in the graph
(nodes with the same color are more likely to be
connected).
--verify_balanced_graph  : Check if the graph is balanced
based on edge signs.
--output out_graph_file.gml  : Save the updated graph with
results to the specified output GML file.
```

5. Gml files needed:
   i. graph_file.gml
   ii. homophily.gml
   iii. balanced_graph.gml
   iv. imbalanced_graph.gml

   Note: gml file must reside in the same location as .py

6. Example inputs:
   - For using components and plot:
     - `python ./graph_analysis.py graph_file.gml --components 3 --plot C --output out_graph_file.gml`
     - `python ./graph_analysis.py graph_file.gml --components 3 --plot N --output out_graph_file.gml`
     - `python ./graph_analysis.py graph_file.gml --components 3 --plot P --output out_graph_file.gml`
   - For homophily:
     - `python ./graph_analysis.py homophily.gml --plot N --verify_homophily`
   - For balance and imbalanced graph:
     - `python ./graph_analysis.py balanced_graph.gml --plot P --verify_balanced_graph`
     - `python ./graph_analysis.py imbalanced_graph.gml --plot P --verify_balanced_graph`

     Note: When the graph is displayed, the nodes may appear clustered together. To view them more clearly, try zooming in.

Author

Karla Chuprinski