PROJECT: ATTRIBUTED GRAPH COMMUNITY DETECTION

Overview: Existing community detection methods are usually based on the structural features and do not take into account the attributes of nodes. However in the real world graphs such as social network, we can have interesting amount of information derived from its social aspect, such as profile information, content sharing and annotations, among others. By incorporating such node attributes we aim to find more meaningful communities in these networks.

Goal: To implement the given community detection algorithm for the real-world attributed graphs.

To do that, you will need:
1. To implement the algorithm assigned to you with an "obsession" with the highest performance possible.
2. To provide a detailed analysis of the performance of your implementation.

Input: You will be provided with the following materials in advance:

- Graphs with Node attributes:
    - **Facebook graph:** The dataset contains a facebook network of a US university where each node is the user profile having attributes as student/faculty status, gender, major, second major, dorm, and year information. For the similarity convenience, these attribute values are converted into asymmetric binary variables. You will have edgelist and attribute list datasets representing the graph connectivity and attributes.
        - fb_caltech_small_edgelist.txt
        - fb_caltech_small_attrlist.csv
    - **Political blog graph:** A directed network of hyperlinks between weblogs on US politics. Each node has an attribute "value" describing its political learning as either liberal or conservative. Other attributes are the blog sources of the hyperlinks. You will have edgelist and attribute list datasets representing the graph connectivity and attributes.
        - polblogs_small_edgelist.txt
        - polblogs_small_attrlist.csv
- Scientific publication that describes the algorithm to be implemented.

Output: The implementation of the algorithm and a report (see details below).

Project Details:

1. Read and understand your scientific publication.
2. Implement the method described in the publication assigned to you.
3. Implementation can be only in programming language (R, Python, Matlab)
4. To evaluate the extracted communities, you need to compare it against certain common metrics. Report the following metrics:

(i) Number of communities
(ii) Size of the communities (Include Plot)
(iii) Modularity
(iv) Similarity (As per scientific publication)
(v) Density

5. Measure performance of the algorithm. The algorithm "runtime" is an additional performance criterion that will be taken into account.

Bonus Section: The methods and the algorithms described in the scientific publications for this project, can be further optimized by avoiding re-computing certain metrics such as modularity and similarity in every iteration, which can scale up the performance.
You can choose to implement such optimized version of the implementation and provide the results for the original version of the graphs.

- Facebook Graph:
  - fb_caltech_org_edgelist.txt
  - fb_caltech_org_attrlist.csv
- Political Blog Graph:
  - polblogs_org_edgelist.txt
  - polblogs_org_attrlist.csv

Submission Details:

1. Algorithm code with detailed comments.
2. README file with detailed instructions. It should include the following information:
3. Software that needs to be installed (if any) with URL's to download and instructions to install them.
4. Environment variable settings (if any) and OS it should/could run on.
5. Instructions on how to run the program.
6. Instructions on how to interpret the results.
7. Sample input and output files.
8. Citations to any software you may have used or any dataset you may have tested your code on.

In short, the TA and any other student that chooses your implementation should be able to install any required software, set up the environment, execute your program, and obtain results without any prior knowledge about your project.

3. Project Report

This report should describe, in your own words, the algorithm discussed in the scientific publication assigned to you. This section should also discuss the performance of this algorithm. If the algorithm is parameterized, then include some discussion and empirical results on the effect of the parameters on the identified communities. Provide a detailed analysis of your algorithm and report the goodness metrics and performance metrics as described in the paper.

Grading Rubric

Criteria	Percentage
Implementation	40%
Code executes and produces required results	30%
Project Report	30%
Bonus Section	25%

Papers:

1. Dang, T. A., and E. Viennet. "Community detection based on structural and attribute similarities." International Conference on Digital Society (ICDS). 2012.

2. Cruz, Juan David, Cécile Bothorel, and François Poulet. "Entropy based community detection in augmented social networks." Computational aspects of social networks (cason), 2011 international conference on. IEEE, 2011.

Other Readings:

1. Blondel, Vincent D., et al. "Fast unfolding of communities in large networks."Journal of Statistical Mechanics: Theory and Experiment 2008.10 (2008): P10008.