Sandhya
Data Science 2
HW03

1. Is the above organization (given also from the nets.py template) the best way to organize the library? Would you change any methods to functions or functions to methods?

The given structure might not be the best way to organize the graph library. We know that a graph is set of nodes. It would be better to make a separate Node class that would contain methods that are related to Nodes without involving graph per say. In the assignment, I have implemented a separate Node Class having node specific methods. My implementation of graph is a dictionary of Node objects where each node has an id and set of neighbor nodes. I found it easier to work with this structure.

2. What is missing? What are the most important calculations you think should be added to the library to make the library more useful or flexible?

This implementation is covering basic functionality needed for making graphs (networks) but it lacks some of the aspects. The library does not involve weighted graphs and things that come with it. It is possible to include functionality like finding dijikstra shortest path, minimum spanning tree, bipartite graph etc. It provides a solid foundation to get started on graphs but many things can be included for advanced functionality.

3. Is your library fast or slow? (Demonstrate this.) Can it be made faster? If so, how? Would optimizing the code lead to overly complicated code that would be hard to maintain?

My library seems to be pretty fast. I timed the code using python time library and calculated the difference between start and end time which is around 0.01470 seconds. I feel this might not be the best way to test this tool but it seems to convey the message. I believe that my library is not awfully slow even when checked with huge file but there is always a room for improvement. Though the class design and method implementation works fine but there is some repetition of code. The library can be made more efficient by getting rid of the iterative function calls. This may or may not require implementation of more efficient data structure which is more efficient but harder to comprehend. So, there is a tradeoff involved.

4. How well tested is your code? What are you not testing and why?

My test coverage is 100%. Checked this using nosetests coverage tool. I am testing almost all the methods in Node class, Graph class and functions in the file.

5. What were particularly easy or challenging parts of this assignment?

It took me a bit of time to design the class structure i-e Node and Graph. Reading and writing took some time in the beginning but it was not particularly hard. The network functions like network backbone, modularity, degree distribution involved understanding of concepts. So, these methods were time consuming. Overall, the assignment was helpful for me to understand the python object oriented side and building test suite.

references:
https://gist.github.com/bagrow/11181518