Homework Assignment 3

Fall 2022

IE 532 Analysis of Network Data

Due on Tuesday 12/06 (at 11:59 pm)

There are 4 questions in this assignment.

Please answer every question to the best of your knowledge and make sure to show your work for partial credit.

You are encouraged/expected to use software (networkx on Python and/or Gurobi) for part of the questions.

Make sure to cite your sources, if you decide to use more material to help you solve the exercises.

Remember that collaboration between students is allowed and encouraged, but please give your collaborators proper reference by letting me know who you worked with and what they contributed.

Only electronic submissions are accepted for this assignment.

You may either type your answers using LaTeX or other word processing software, or scan your handwritten answers and submit them. If the latter, please make sure the scanned copy is legible.

If submitting multiple files, create a folder and compress it (in either .zip, .rar, or .7z format, among others) before submitting.

Question 1: Centrality metrics

In class, we saw the definitions of three of the most standard centrality metrics as in degree, closeness, betweenness, PageRank (eigenvector). In the last few years, people have observed that such *nodal* metrics (that is, applying to a node) are often limited. They fail to address questions such as "what is the most central group on Twitter?" or "are female researchers central in bioinformatics?" or even "what are the most influential songs of all time?", among others.

(a) Propose three definitions of degree, closeness, and betweenness for a set of nodes S (of given size $|S| = k$) rather than a single node. Properly define how they would be calculated.
(b) Formulate mathematically the problem of identifying the group of nodes S of highest degree/closeness/betweenness of given size $|S| = k$. Make sure to carefully define your parameters, variables, constraints, and the objective function.
(c) Identify the groups of highest degree/closeness/betweenness for the network of the Les Misérables graph. Use $k = 2, 3, 5, 10$.

Question 2: Modularity clustering (and a spectral property)

In class, we discussed modularity clustering (also known as Louvain community detection). As a reminder, we are aiming to maximize

$$Q = \frac{1}{2|E|} \sum_{i,j \in V} \left[a_{ij} - \frac{d_i d_j}{2|E|} \right] \delta_{ij}$$

using a greedy approach, where we iteratively combine communities until the modularity value starts dropping. Answer the following questions.

(a) The modularity of a graph can be negative! Construct an example where the modularity of the set of communities in the graph is indeed negative. What is the minimum value the modularity can get?
(b) Implement the modularity clustering approach discussed in class (using modularity). Then, identify the average number of communities identified in randomly generated networks. Specifically, report the average number of communities identified after generating $k = 20$ networks with the following parameters:

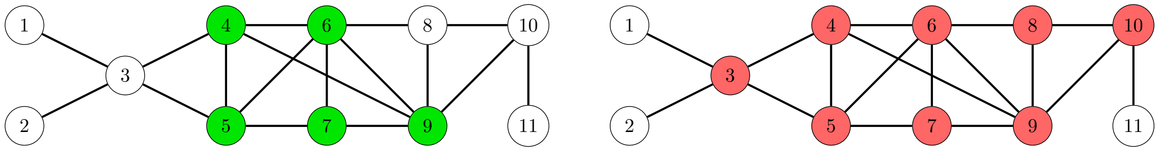
- Erdös-Rényi graphs with $n = 30$ and $p \in \{0.2, 0.5, 0.8\}$.
- Barabási-Albert graphs with $n = 30$ and $k \in \{1, 2, 3\}$.

Question 3: *k*-core decomposition

For any graph, we define a k-core as the induced subgraph where every node has degree at least k. Answer the following questions.

(a) Devise an algorithm to detect the largest k-core for a given value of k. For example, for $k = 3$ it should produce the structure in green in Figure 1 (left). For $k = 2$, it should produce the structure in red in Figure 1 (right).
(b) Let a node have coreness equal to c if it belongs to a c-core, but not to a $c+1$-core. Use the **karate club** and the **les miserables** networks from networkx and paint every node according to its coreness.

Figure 1: The largest 3-core (left) and 2-core (right) in the network.



Question 4: Algorithms for cliques

(a) Consider a greedy algorithm for finding the maximum clique. The greedy algorithm works as follows.

1. Find the node with the maximum degree. Add it to your clique C, that is $C = \{i\}$, where i is the node of maximum degree. Remove every node from the network that is not adjacent to it.
2. Among the remaining nodes, and excluding the nodes that are in C already, consider the one with maximum degree in the new network. Again, add it to C and remove every node that is not adjacent to it.
3. Stop when you have a clique.

Does that algorithm produce a maximum clique? Why/why not? What is the worst-case runtime complexity of the algorithm?

(b) Consider another greedy algorithm which works slightly differently.

1. Find the node with the minimum degree. Remove it from the network.
2. Among the remaining nodes, consider the one with the minimum degree in the new network. Again, remove it.
3. Stop when you have a clique.

Does that algorithm produce a maximum clique? Why/why not? What is the worst-case runtime complexity of the algorithm?

(c) Finally, consider one last algorithm for the maximum clique problem. First, calculate all cliques of size $k = 2$ (those would be the edges of the graph). Then, compare them pairwise: if they have $k - 1$ nodes in common and the graph also contains the missing edge between the pair, we can merge them into a clique of size $k + 1 = 3$. This algorithm can be generalized for any k. Does that algorithm produce a maximum clique? Why/Why not? What is the worst-case runtime complexity of the algorithm?

Good luck!