Wenqin Chen, Georgina Xu

MTH 255: Graph Theory

Initial Project Proposal

Due: 04/13/2018

Instructor: Professor Gwen Spencer

**Modeling Smith Houses Social Structures**

**Overview**

Our project is to build graph models to represent the social structure of each residential houses at Smith College. After building the models, we would like to find patterns (similarities and differences) between different houses’ social structure.

Each student in the house will be represented by one node; an edge exists between two nodes if and only if the two students are immediate friends. We are not going to ask for the data of the real social structure of each Smith houses. Instead, for each house, we will get the total number of students and the number of students of each class year; and we will construct a graph using the rules below.

**Model Building Rules and Techniques**

1. Simulate the social structure of a house, denote as graph G
   1. Start with S isolated nodes, representing S seniors entering house G who don’t know each other at first
   2. These S people start socializing with each other, making friends in the following way (research clustering coefficient):
      1. For a pair of nodes (u,v), if there is no path between them, they have a distance of infinity, which means there is a 5% (will update this after further research) chance for them to become friends (for u, v to have an edge)
      2. The shorter the u-v path is, the higher chance there will be an edge between u and v
      3. If edge uv already exists, the probability becomes 0
   3. After T time steps, another J isolated nodes join G, representing that a year passes and J juniors enter the house. The same rule for making friends applies to all (S + J) nodes
   4. After another T time steps, P isolated nodes, representing P sophomores, join G. Same rule for making friends applies to all (S + J + P) nodes
   5. After another T time steps, F isolated nodes, representing F first-years, join G. Same rule for making friends applies to all (S + J + P + F) nodes
   6. Stop adding more edges after T time steps and return the current graph
2. Next Step:
   1. Collect data (class year) of all Smith houses --> build model G for each house --> visualize each graph G
   2. Try if we can find any interesting patterns, similarities, differences between the social structure of each house (for example, Is it a connected graph? Are there any cliques?)
3. Further Explorations:
   1. Explore cut-nodes and cut-edges: removing which person (i.e. juniors study abroad) or which connection in a house’s social structure will make the most impact on the house?
   2. Which house has the weakest or the strongest (most stable) social structure and why?

**Computation Techniques**

1. computeDistance(G): For each node x, we are going to use MatLab to run Dijkstra algorithm and build a distance matrix of x, representing the length of the shortest path from x to all other nodes
2. computeProbability(G): For each node x, given its distance matrix, we are going to build a probability matrix of x, such that entry (x, v) represents the probability of adding an edge between x and v --- this method should call computeDistance(G)
3. socialize(G,T): simulates the process of socialization within T time steps. It takes T and G, as inputs, and return the resulting G. -- This method should call computeProbability(G)
4. simulateHouse(S,J,P,F): Start with a graph G with no edges and S nodes. Add new nodes three times (J, P, F) and call socialize(G,T) after adding new nodes
5. After building the models, we are going to use Java to implement graph visualization of each graph
6. Further Explorations (if we have time):
   1. We are going to build an user-interactive Java GUI interface so that user can remove nodes or edges and see what impacts it brings to the graph