Girvan-Newman Model

CSCI 4964 Final Project Report

Yifan(Elio) Li, liy57@rpi.edu

1.0 Introduction

The paper I've chosen for this project is 'Finding and evaluating community structure in networks' by Newman, Mark EJ, and Michelle Girvan.

This paper briefly talks about how to effectively finds the communities of network. Traditionally, the algorithm for finding a community is as follows:

1. For every edge in a graph, calculate the edge betweenness centrality.
2. Remove the edge with the highest betweenness centrality.
3. Repeat steps 2 until there are no more edges left.

However, this algorithm creates a problem where each edge does not recalculate the betweenness for each edge removal, this would result an issue resembled in figure 1 and figure 2. Figure 1 & 2 contains 2 clusters, where those clusters are connected by 2 edges, one of the edge have high betweenness while the other one have low betweenness. After performing the above algorithm, the edge that have high betweenness will get removed where the remaining edge will receive the highest betweenness instead. Since the algorithm does not recalculate betweenness this edge will get removed at the very end of the algorithm.

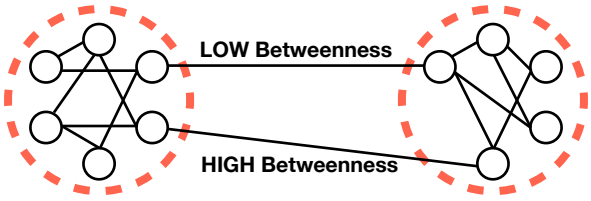

Figure 1. Example

Figure 2. Example After Edge Removal

The author of the paper proposed a new approach that simply add a new step for the previous algorithm:

1. For every edge in a graph, calculate the edge betweenness centrality.
2. Remove the edge with the highest betweenness centrality.
3. Calculate the betweenness centrality for every remaining edge.
4. Repeat steps 2-3 until there are no more edges left.

We will simply recalculate the betweenness for every iterations, this would be really costly but the author believed that it's worth the time and effort.

2.0 Betweenness

Betweenness is a measure that favors edges that lie between communities and disfavors those that lie inside communities. There are 3 ways to calculate betweenness.

I. Shortest-Path Betweenness: The easiest and most common type of measure, which is to find the shortest path between all pairs of vertices and count how many run along each edge

II. Random-Walk Betweenness: Calculate the expected net number of times that a random walk between a particular pair of vertices will pass down a particular edge and sum over all vertex pairs.

III. Current-Flow Betweenness: This is similar for random-walk betweenness but especially for the resistor network.

3.0 Modularity

A way to measure the community is modularity (Q). e_{ii} is the probability or the fraction of edge is in the same module i, a_i^2 is the probability of random edge belongs in module i, $TR(e)$ is the same as e_{ii} where $\sum_i e_{ii}$ which represents the trace of e, and lastly, $\|e^2\|$ is the sum of the matrix elements.

$$Q = \sum_i (e_{ii} - a_i^2) = TR(e) - \|e^2\|$$

4.0 Visualization

The result from the paper was a modularity of 0.54 with 11 communities. In my implementation I have also achieved 11 communities with a modularity of 0.538 which is pretty close to the original paper.

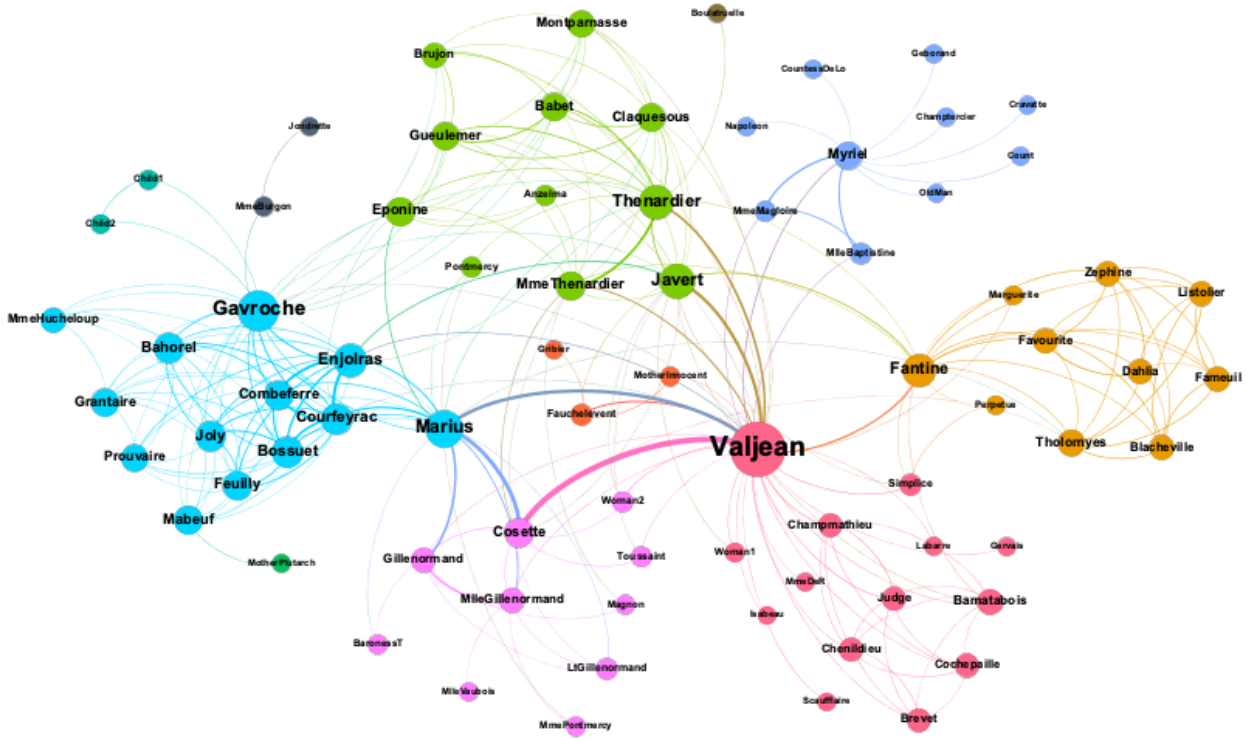

Figure 3. Network Visualization