1 Lancichinetti, Fortunato, and Radicchi: Benchmark graphs for testing community detection algorithms

1.1 Introduction

Testing an algorithm for community detection is just as important as deriving a new one. According to this paper, testing an algorithm for community detection should entail analyzing a well defined community structure and trying to recover its communities. Instead of using real-world examples, where true community membership is often not precisely known, a select number of computer-generated networks with built-in community structure can serve as a benchmark for checking community detection algorithms.

The Girvin and Newman (GN) class of networks is the most famous computer-generated network to serve as a benchmark for community detection algorithms. Each network has 128 nodes divided into 4 groups, with an average degree of 16 for each network and all nodes with approximately the same degree. If the majority of edges connected to a node are connected with members of a certain group, that node is a member of that group.

While they have been used to test algorithms for decades, GN models are not perfect. There are three key characteristics of GN models that make them unhelpful proxies of real network structures:

1. They have too few nodes
2. Each group is of roughly the same size
3. Each node connects to roughly the same amount of other nodes

Meanwhile, real networks have varying node degrees. This allows them to have things such as:

- resilience to random failures/attacks
- absence of a threshold for percolation
- epidemic spreading

A better benchmark then should have more degree variation than the GN model. More specifically, the degree distribution should be skewed such that it matches real networks who have often have distributions that taper in size according to power laws. Likewise, a better benchmark should also have differing community sizes. The size of network data has increased proportionally with the computing innovations needed to handle such data, so it is time that a model with many more nodes than 128 be constructed.

1.2 The Benchmark

Assume both degree and community size distributions are power laws, with exponents γ and β. The number of nodes is N and the average degree is (k).

Thus, construct a network using the following procedure:

1. Each node is given a degree from the power law distributions as defined above, so that the average degree is (k).

2. Each node shares fraction μ of its nodes with other members of its network, where μ is termed the **mixing parameter**.

3. The sizes of the community are taken from the power law distribution defined above, with the sum of all community sizes equal to N.

4. Nodes are assigned to any community at random. If the size of the community exceeds the internal degree of the node, the node enters the community. If not, it does not get assigned to that community. If a community is complete, any attempts to add another node will result in another node being displaced from the community and replaced by the node being assigned. Once all nodes have been assigned, stop assigning nodes.

5. Changing μ does not mean changing the degree of each node, but only the ratio between internal and external degrees(i.e. how many edges are going out of the community versus staying in).

The above procedure works generally, but fails to set μ exactly for smaller networks. The distribution of μ for a given benchmark graphs generally follows a bell-shaped curve.

Generating this benchmark is also not horribly computationally intensive.

1.3 Testing

For two of the more popular community detection algorithms, the algorithms perform relatively well on this new benchmark. To assess performance, the built-in modular structure of the network is compared with the communities detected by the two algorithms using **normalized mutual information**, a measure of similarity. This performance is affected by a number of observed factors:

- For smaller μ, the algorithms begin to fail

- Detection is better with larger average degree (k), but worse with larger community size β

- For larger N, detection begins to fail

Thus, a newer, computationally efficient way to assess community detection has been created that better matches the structure of modern real-world network data..