Revisiting the behavior of a network with preferential deletion

Steven Perdomo
COT 5405, Design and Analysis of Algorithms, University of Central Florida

Date: 05/06/20

Abstract

This project revisits the work done by Narsingh and Aurel [1] in helping analyze the behavior of web-like networks. Their paper developed underlying rules, specifically preferential deletion, for simulating the development of a network and created numerical solutions. The goal was to be able to study the degree distribution of the simulation and determine if it was in agreement with previous works. Here, their simulations and solutions are replicated and their results verified.

1. Introduction

Analyzing the behavior of networks has been a continual effort for research scientists. It has been found that many systems, virtual or real life, draw upon similar behavior. Consequently, the ability to understand the underling properties of a generalized network could provide insight into many areas of interest. Different efforts have created different principles upon which to simulate and analyze these networks.

The article of interest for this paper [1] took into consideration previous works that used dynamic random graph models. A dynamic random graph model can be described as "a discrete-time process which starts out with a small fixed graph and in each subsequent time step a new node is added to the graph or an existing node is deleted from the graph [1]." At each step, the model is updated depending on predefined rules. Even though previous dynamic model efforts have used different rules, the degree distributions have been shown to follow a power-law distribution. Using a combination of preferential attachment and deletion rules [1], also demonstrated this behavior. This project recreates the methodology constructed in the main article to similar results.

2. Methodology

Python was used to run both the model and numerical calculations and classes were used to represent both simulation and numerical operations. A full test cycle runs for a predefined amount of time steps t using a probability p for birth and q for death to dictate the result at each time step. Both classes record the number of nodes and edges throughout their life cycles and also compile their respective degree distributions as lists. These figures of data were then used for visual processing and validation. Numerical calculations for node count, edge count, and degree distribution were based off of the equations of the focal article.

Network Model

The network was modeled using an adjacency list representation with a python dictionary storing each list for the corresponding node. To begin with, a graph of one node with a self loop is created. For each time step, the probability values determine whether a node will be added or deleted from the graph.

In the event of a birth, a new node is added to the graph dictionary and an edge is created to an existing node based on the birth probability distribution[1]. Using the *NumPy* random choice function, the target node is randomly selected after calculating the probabilities for each node at the particular time step. In the event of death, a node is chosen to be deleted from the graph and its adjacency list is used to remove any edges from adjacent nodes. The deletion node is chosen based off of the death probability distribution[1] and similar to birth, *NumPy* is used to process the corresponding probabilities for each node.

There were certain situations that the main article didn't provide many specifics for handling. One such event involved nodes that would be left disconnected from each other after a deletion. This project's implementation simply leaves those nodes without any edges until the birth process connects a new node to it. Another case was the result of an empty graph after a deletion. This implementation continues the graph once a birth occurs and attaches the new node to itself, similar to the initial graph.

3. Results and Discussion

In conjunction with the main paper, the simulation was run with t = 50000 time steps and with three different p values: 0.6, 0.75, and 0.9. Figures 1 and 2 show the comparison between the simulation and numerical analysis for the quantity of nodes and edges. Dotted data points correspond to the simulation and the solid lines correspond to the analytical values.

The number of nodes and edges seemed to follow very similar trends in growth. The simulation data of the two graph properties look identical when compared with their corresponding analytical data. Simulation data matches well with the analytical data which helps verify the validity of this implementation of the research. Programming the model to fo-

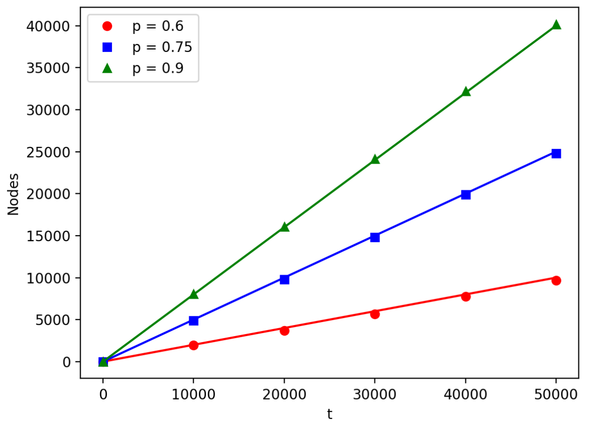

Fig. 1: Comparison of node quantities between modeling and analytical data

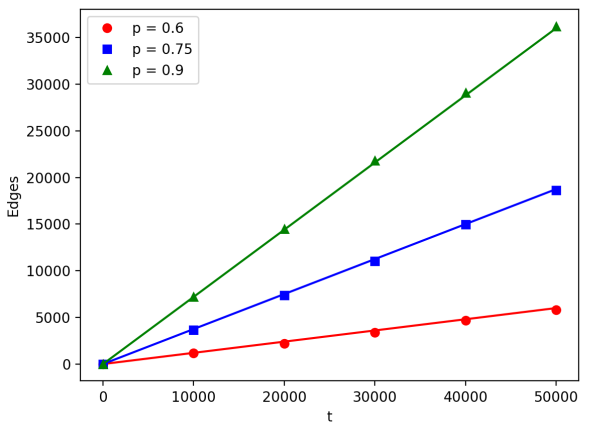

Fig. 2: Comparison of edge quantities between modeling and analytical data

llow the rules created by Narsingh and Aurel [1] created consistent results where the simulation almost exactly followed the analytical calculations. Any variance in the data could be attributed to some slight irregularity from the random choice generator.

The agreement of the data helps verify that the network simulation is running as expected and that the growth of the network in terms of node and edge counts is following the predicted values based on general graph development even with the integration of the preferential deletion rule.

With this in mind, the last desired analysis involves the actual degree distribution of the network model. Figure 3 shows the results of monitoring the degree distribution through an entire test interval plotted on a logarithmic scale. Again with $t = 50000$ time steps and with a p value of 0.8, the simulation was executed and its degree distribution was updated accordingly. The desire here is to be able to show that this preferential deletion model still follows a power law distribution in terms of the degrees of its nodes.

The model degree distribution doesn't seem to be an exact power law distribution. An exact power law distribution is demonstrated by the analytical solution. A power law distribution should be linear in the logarithmic scale. Despite the slightly curved nature of the model distribution, it is still promising that it follows a similar trend to the analytical solution. It's possible that model wasn't simulated to full correctness due to misinterpretations in recreating the methodology of the original paper. Another oddity of the data is that there was at times a slight increase in the distribution at the end of the scale. This relapse was created because the nodes with the highest degrees were likely to have another node added to them based on the initial birth probability distribution.

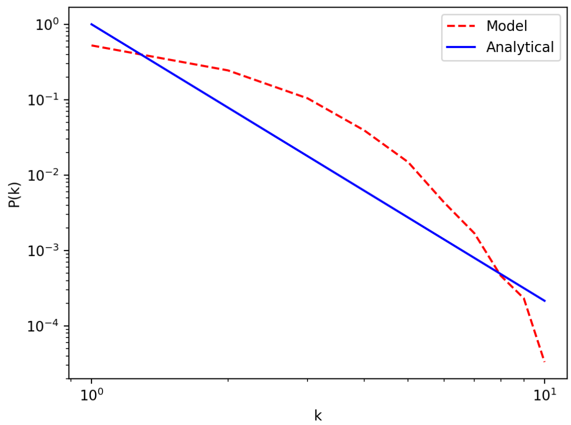

Fig. 3: Comparison of degree distribution data between modeling and analytical data on a logarithmic scale

4. Conclusion

This project revisited work done in implementing a new set of rules to define network behavior. The inclusion of preferential deletion agreed well with analytical methods and allowed a network to be accurately simulated. Using these rules created a network with a degree distribution curve that is close to the power law distribution that has been attributed to many real world networks and former research. The initial observation that network nodes with smaller degrees are more likely to be deleted in real world applications is shown to be a viable assumption when summarizing the behavior of networks.

Moreover, the data shows that it's possible that there are even more properties that can exist in networks that should be investigated to fully predict network behavior. Although this set of rules created a degree distribution curve that was desirable, different combinations of uniform and preferential node adjustment could lead to even more accurate modeling capabilities and consequently impact different areas of interest. This simulation did not meet expected values exactly which could mean that these rules might need probability adjustment. It will also be valuable to understand what rules might govern different networks and cause variations and behaviors.

References

[1] Narsingh Deo and Aurel Cami. “Preferential deletion in dynamic models of web-like networks”. Information Processing Letters, vol. 102, no. 4 (2007), pp. 156–162.