Simulating Dynamic Random Networks

Gary Hoppenworth

Introduction:

Random dynamic networks are of interest because they can be used to model complex, web-like systems. A perfect example is the World-Wide Web, a connection of websites all connected through hyperlinks. The World-Wide Web is *dynamic* -- websites are created and destroyed each day. However, these insertions and deletions happen in a very specific way -- sites with small degree are more likely to be deleted than large websites. This makes sense, since a website like www.google.com would cause massive shockways on the internet if it were to be deleted, whereas small websites with zero or only one link are less valuable to the internet community. In this paper we implement the dynamic random graph model with preferential attachment and preferential deletion studied by Deo and Cami in [1].

Definitions:

- Our initial graph G1 contains a single vertex v0 with a single edge to itself.
- At each time step t of our simulation, we either insert a vertex and edge with probability p, or delete a vertex and all incident edges with probability (1-p).
- Consider current graph G = (V, E). The insertion sequence and deletion sequence are described below.
- Insertion Sequence:

    - Pick a vertex v in G with probability d(v) / (2 |E|) where d(v) is the degree of v.
    - Attach a new vertex u to vertex v with a single edge.
- Deletion Sequence:
    - Pick a vertex v in G with probability (|V| - d(v)) / (|V|^2 - 2 |E|) where d(v) is the degree of v.
    - Remove vertex v from G as well as all incident edges.

Implementation:

I implemented this preferential dynamic random graph model in python using the Numpy and Random packages. All data points presented in the experimental section are the average of five simulations of the graph model.

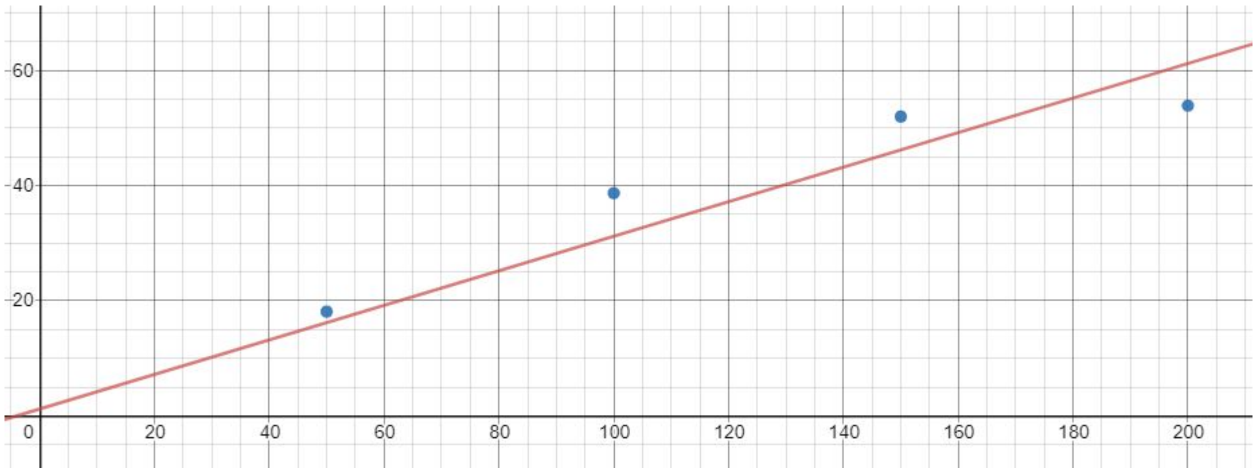
Experimental Results:

Number of Nodes:

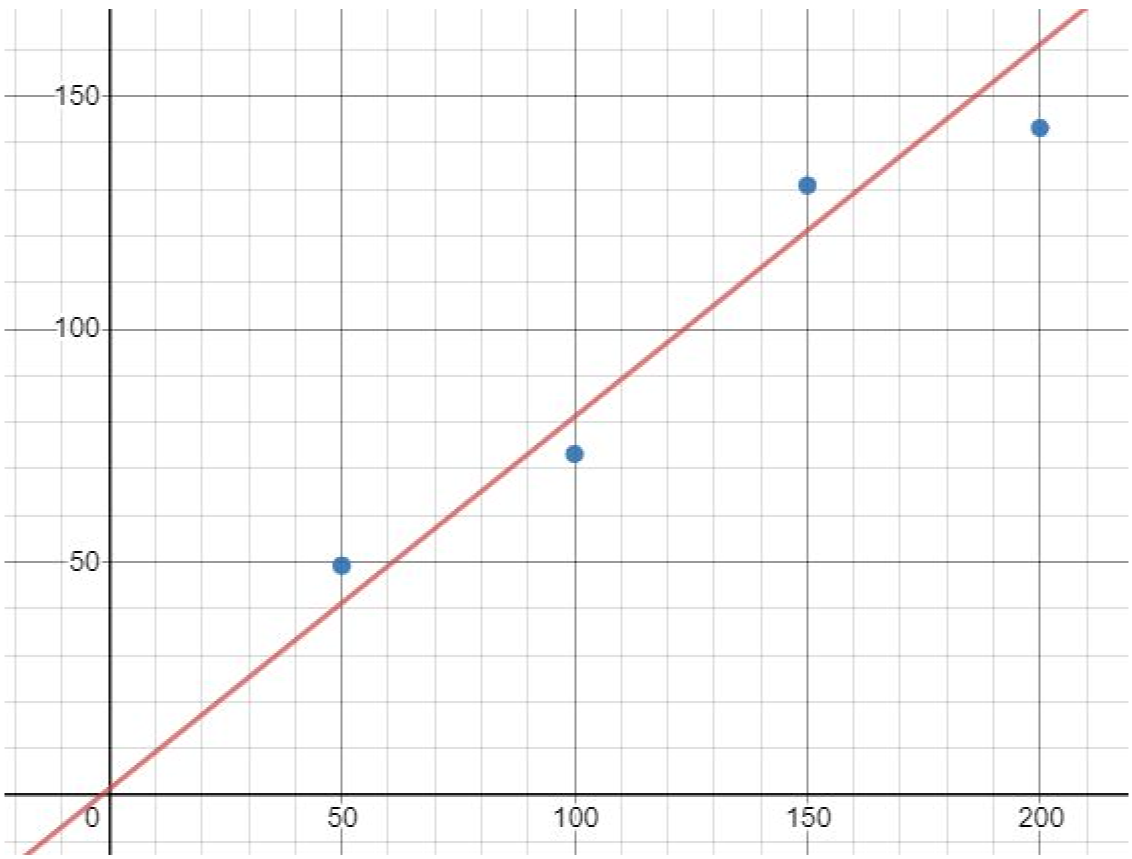
We experimentally measure the number of nodes as the graph model progresses. We compare this to the predicted value described in [1].

We evaluate this graph model for values p = 0.6, 0.9 at time steps t = 50, 100, 150, 200.

Number of vertices over time when p = 0.6



Number of vertices over time when p = 0.9



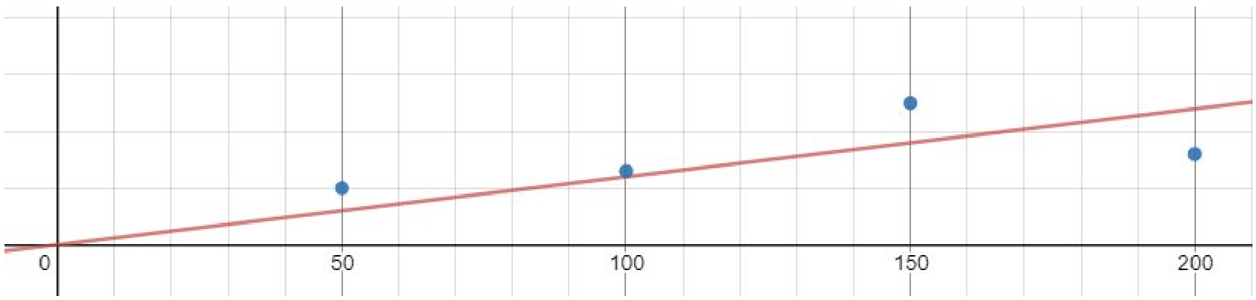
The expected number of vertices calculated in [1] is shown in red and our experimental results are shown as blue dots in the above graphs.

Number of Edges:

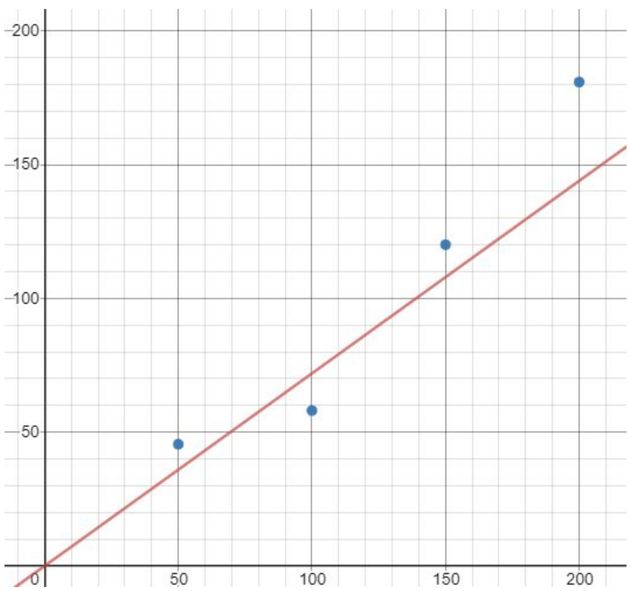
We experimentally measure the number of edges as the graph model develops. We compare this to the predicted value described in [1].

We evaluate this graph model for values p = 0.6, 0.9 at time steps t = 50, 100, 150, 200.

Number of edges over time when p = 0.6



Number of vertices over time when p = 0.9



The expected number of vertices calculated in [1] is shown in red and our experimental results are shown as blue dots in the above graphs.

Analysis:

Our experimental values seem to validate the mathematical analysis of the dynamic random graph evolution presented in [1]. Specifically, we observe that as p increases, the linear factor at which the number of edges and vertices increases is larger, which intuitively makes sense, since the value of p corresponds to the probability that a new vertex (and hence a new edge) is created.

Conclusion:

The mathematical analysis of dynamic random graphs that is given in [1] has been experimentally validated by our experimental data. Future work should explore the behavior of the preferential dynamic graph model on very large time scales. Unfortunately, due to my limited computing resources, I could only simulate these dynamic graph models on relatively small time scales (i.e. $t \leq 200$).

References:

[1] Deo, Narsingh, and Aurel Cami. "Preferential deletion in dynamic models of web-like networks." Information processing letters 102.4 (2007): 156-162.