STAT 428 Statistical Computing Final Project

Monte Carlo Simulation of Scale-free Networks

Tinghao Guo

May 13, 2014

Abstract

The scale-free network is a class of network whose degree follows a power law distribution. It has been widely studied in many fields such as WWW, food web, systems biology and etc. The most well-known model for generating the scale-free network is preferential attachment. It is a mechanism that incorporates the idea of "rich get richer" in the network growing process. In the project, I simulated the preferential attachment process and generated the scale-free network using Barabäsi-Albert (BA) model [5]. The numerical results have shown that the resulting scale-free network follows the power-law distribution with exponent parameter $\gamma \approx 2.89$, which agrees with the theoretical value as well as the simulation results in Ref. [5].

1 Introduction

Network theory is a research area focusing on the study of the graph. A large number of real networks such as genetic regulatory networks, citation networks, World Wide Web or social networks can be described as scale-free networks, where the degree distribution $P(k)$ follows the power-law or has a heavy tail:

$$P(k) \sim k^{\gamma} \tag{1}$$

where k is degree and γ is an exponent parameter [6]. For instance, in a movie collaboration network, the probability of an actor with k links to others has the power-law distribution, i.e. $P(k) \sim k^{-\gamma_{\text{actor}}}$, where $\gamma = 2.3 \pm 0.1$ [5]. Albert et al. showed that the World-Wide Web degree distribution is also scale-free based on a network model with 325,729 nodes and 1,469,680 edges. Its in-degree and out-degree distributions have the form of $P(k) \sim k^{-\gamma}$, where $\gamma^{\text{out}} = 2.45$ and $\gamma^{\text{in}} = 2.1$ [3]. In the metabolic network of H. pylori, it has been estimated that the power-law distribution has the exponent $\gamma = 2.32$ [11, 2]. Figure 1 shows an example of an undirected scale-free network with 50 nodes and 49 edges.

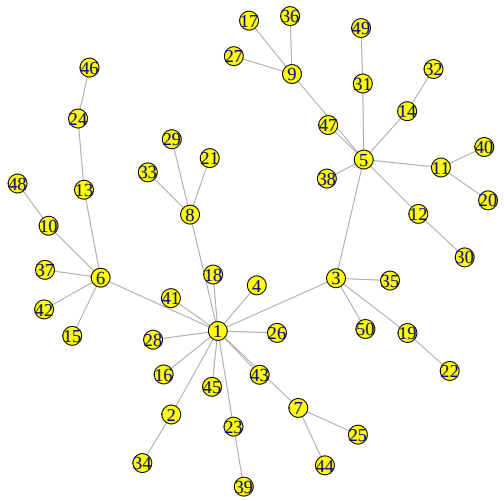

Figure 1: An example of scale-free network

Price was the first to propose a simple model that can produce a scale-free network with the power-law distribution [9]. The model was primarily focused on the citation network, where each vertex represents a

paper and the edge indicate the citation relation between the papers. Inspired by the Simon model that primarily studied the occurrence of power-law distribution in sociological, biological and economic empirical data [10], Price developed the mathematical model for the growth of the citation networks. In the citation network, the papers are published continually. The newly appearing paper can only cite the existing ones and not the other way around. As a result, the citation relation forms a directed network pointing from the new paper to the existing ones. The key assumption in Price's model is that the newly published paper cite the existing ones with the probability proportional to citation number (in-degree) those existing papers already have. Figure 2 shows an example of the citation network. This model has been shown to follow the power-law distribution:$P(k) \sim k^{\gamma}$ with the exponent $\gamma = 2 + a/c$, where c is the average number of papers cited by the new one (i.e. the average in-degree of each vertex) and a is a constant.

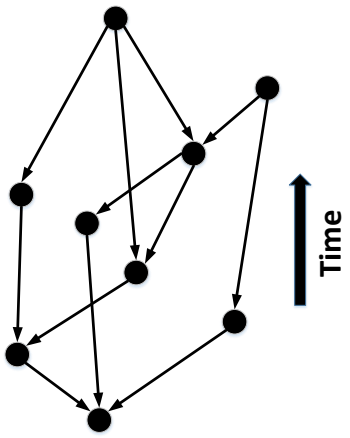

Figure 2: An example of the citation network [7].

While the Price model was initially called *cumulative advantage*, it was not until in 1999 that a well-known name *preferential attachment* was given by Barabäsi and Albert [4]. The Barabäsi-Albert (BA) model is perhaps the most commonly used generative growth model nowadays. In the BA model, one new node is added to the current graph at each time step. The connections between the new node and the existing ones are proportional to degree that the existing nodes already have. While the growing mechanism is similar, the BA model is mainly focused on the undirected graph, rather than the directed graph in the Price's model. It has been shown that the degree distribution $P(k)$ in the BA model exhibits the power law distribution [5, 8]:

$$P(k) \sim k^{-3} \text{ as } k \to \infty \tag{2}$$

The objective of the project is to simulate the BA model and generate a scale-free network using Monte Carlo method. In next section, I will review the BA model in detail and develop Monte Carlo the algorithm for the model. The numerical results are presented in Section 3.

2 Methods

Consider an undirected graph with m_0 nodes, the BA model can be stated as follows [5]:

- Growth: at each step, one adds a new node with $m(\leq m_0)$ connected to m different nodes.
- Preferential attachment: the probability p that the new node connects to existing node i (with degree k_i, for $i = 1, 2, \ldots, m_0$) is proportional to the degree that node i already has, i.e.

$$p = \frac{k_i}{\sum_{j}^{m_0} k_j}, \text{ for } i, j = 1, 2, \ldots, m_0. \tag{3}$$

To simulate the BA model, one can use adjacency matrix $A_{m_0 \times m_0}$ to represent the network. For any node i and j in the network, entry A_{ij} is defined as:

$$A_{ij} = A_{ji} = \begin{cases} 1 & \text{if node } i \text{ and node } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Note that the adjacency matrix is symmetric for the undirected graph. Since one new node is added to the network each time, the adjacency matrix is changed to $A_{m_0 \times m_0+1}$ at each time. For the additional m edges, one needs to choose m target nodes with the probability proportional to the degrees. A node label list is used to simulate the preferential attachment process [8]. Each entry in the label list represents the node ID that has edges. Therefore, one can choose m nodes uniformly at random from the list and the new node is then connected to the m edges, which is equivalent to choosing m target nodes proportional to the degrees in the network. Figure 3 illustrates this mechanism in detail. Suppose one adds a new node (Node 5) with $m = 2$ new edges. The current graph has degree 1, 2, 3 and 2 (solid lines) for Node 1-4 receptively. The node label list is written as $[1, 2, 3, 3, 2, 3, 4, 4]$. For example, because Node 4 has degree of 2, then node ID "4" are shown in the label list twice and the same idea applies to the other nodes. Node 5 (the new node) can choose two targets (because of $m = 2$) from Nodes 1-4 with probability proportional to the degree. In other words, this process is equivalent to choosing two targets uniformly at random from the list as the probability of

choosing Node 1-4 turns out to be $1/8, 2/8, 3/8$ and $2/8$ after the simple calculation. After adding two edges to the graph (dashed lines), one can update the node label list to $[1, 2, 3, 3, 2, 3, 4, 4, 3, 2, 5, 5]$. Therefore the algorithm for simulating the BA model can be described as follows:

1. Add one node to the network.
2. Choose m target nodes uniformly at random from the set of node currently in the list.
3. Create the edges from the new node to m target nodes.
4. Update the label list by adding the new node and targets to the end of the list. Return to Step 1.

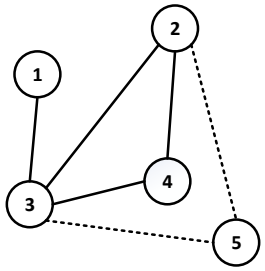

Figure 3: An example of node label list

3 Results

I used R to implement the algorithm and igraph to analyze the scale-free network. Igraph is an open source and free network analysis package that can be programmed in GNU R [1]. Table 1 reports the numerical results for the algorithm described in the previous section. Here the total network size is set to be 10,000. The power law exponent, KS statistics and p-value were estimated by igraph. It can be seen that the exponent values γ agree with simulation values $\gamma = 2.9$ and the theoretical value $\gamma = 3$ as $k \to \infty$ in Ref. [5]. Note that the initial graph with $m_0 = 1, 3, 5, 7$ respectively does not affect the exponent parameter value γ and the network performance significantly. The KS statistics and p-value also indicate the resulting network follows the power law distribution. Figure 4 shows the degree distribution for different initial graphs in a log-log plot and the slops are approximately 2.9. The R code can be found in Appendix.

In summary, I have simulated the BA model and generated the network using Monte Carlo method. The simulation results have shown that the resulting network is a scale-free network with degree distribution $P(k) \sim 2.89$. These numerical results are consistent with the data and the theoretical value given by Ref. [5].

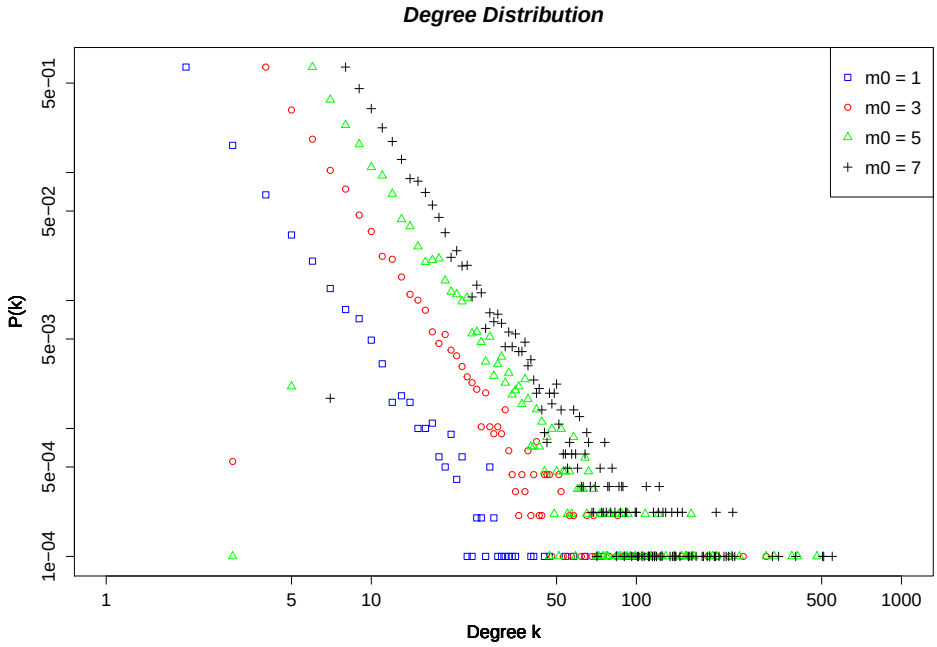

Figure 4: The power-law distribution with network size 10,000

Table 1: Power law distribution with network size 10,000

	γ	**KS-Stat.**	**p-value**
$m_0 = m = 1$	2.88	0.0285	0.999
$m_0 = m = 3$	2.87	0.0237	0.5650
$m_0 = m = 5$	2.89	0.0219	0.1406
$m_0 = m = 7$	2.89	0.0112	0.5442

References

[1] Official Site of igraph, 2014. http://igraph.org/redirect.html.

[2] Albert, R. Scale-free Networks in Cell Biology. *Journal of Cell Science 118*, 21 (2005), 4947–4957.

[3] Albert, R., Jeong, H., and Barabási, A.-L. Internet: Diameter of the World-Wide Web. *Nature 401*, 6749 (1999), 130–131.

[4] Barabási, A., Albert, R., and Jeong, H. Mean-field Theory for Scale-free Random Networks. *Physica A: Statistical Mechanics and its Applications 272*, 1 (1999), 173–187.

[5] Barabási, A.-L., and Albert, R. Emergence of Scaling in Random Networks. *Science 286*, 5439 (1999), 509–512.

[6] Jeong, H., Tombor, B., Albert, R., Oltvai, Z., and Barabási, A.-L. The Large-scale Organization of Metabolic Networks. *Nature 407*, 6804 (2000), 651–654.

[7] Leicht, E., and Clarkson, G. Large-scale Structure of Time Evolving Citation Networks. *The European Physical Journal B 59*, 1 (2007), 75–83.

[8] Newman, M. *Networks: An Introduction*. Oxford University Press, Oxford, UK, 2010.

[9] Price, D. Networks of Scientific Papers. *Science*, 3683 (1965), 510–515.

[10] Simon, H. On a Class of Sknew Distribution Functions. *Biometrika 42*, 3/4 (1955), 425–440.

[11] Tanaka, R. Scale-rich Metabolic Networks. *Physical Review Letters 94*, 16 (2005), 168101(4).

Appendix: R Code for the BA Model

```
## STAT 428 Final Project
## Simulate preferential attachment
## Tinghao Guo
rm(list=ls())
library(igraph)
n_total = 10000          # total number of nodes
m0= c(1,3,5,7)           # initial starting graph m0 = 1,3,5,7 repectively
alpha = list()
KS = list()
p = list()
```

```
degree = list()
degree_dist = list()


for (l in 1:length(m0)){
  label_list = NULL
  num_add = n_total - m0[l]
  # edges to be added each time
  m = m0[l]
  # pre-allocate the adjacency matrix
  adj = matrix(0, m0[l] + num_add, m0[l] + num_add)
  for (i in 1:num_add){
    if (length(label_list) != 0){
      # uniformly at random choose two target nodes from the label list
      r = sample(1:length(label_list), m, replace = FALSE)
      for (j in 1:m){
        # connect the edge between the targets and the new node
        adj[i+m0[l], label_list[r[j]]] = 1
        adj[label_list[r[j]], i+m0[l]] = 1
        # update the label list
        label_list = c(label_list,label_list[r[j]])
        label_list = c(label_list,i+m0[l])
      }
    }
    else{
      for (k in 1:m){
        # base case: the initial graph starts with m0 nodes
        # connect the new node to the initial m0 nodes
        adj[m0[l]+1,k] = 1;
        adj[k,m0[l]+1] = 1;
        label_list = c(label_list,m0[l]+1,k)
      }
```

```
    }
  }
  ## create a graph object using igraph
  g = graph.adjacency(adj, mode = "undirected", weighted = NULL, diag = FALSE)
  fit = power.law.fit(degree(g),10)
  degree[[l]] = degree(g)
  alpha[[l]] = fit$alpha
  KS[[l]] = fit$KS.stat
  p[[l]] = fit$KS.p
  degree_dist[[l]] = degree.distribution(g)
}

## Figure
plot(degree_dist[[1]], col = "blue", xlim = c(1,1000),
pch = 0, log = "xy", xlab = "Degree k", ylab = "P(k)")
par(new = T)
plot(degree_dist[[2]], col = "red", xlim = c(1,1000), pch = 1,
     log = "xy", xlab = "Degree k", ylab = "P(k)", axes = FALSE)
par(new = T)
plot(degree_dist[[3]], col = "green", xlim = c(1,1000), pch = 2,
     log = "xy", xlab = "Degree k", ylab = "P(k)", axes = FALSE)
par(new = T)
plot(degree_dist[[4]], col = "black", xlim = c(1,1000), pch = 3,
     log = "xy", xlab = "Degree k", ylab = "P(k)", axes = FALSE)
title(main = "Degree Distribution", xlab = "Degree k",
ylab = "P(k)", font.main = 4)
legend('topright', pch = c(0,1,2,3), col = c("blue","red","green","black"),
       legend = c("m0 = 1","m0 = 3","m0 = 5","m0 = 7"))
```