Complex Network Simulation

Morgan Grant
The University of Queensland
morgan.grant@uqconnect.com.au

Contents

Contents

Complex Network Examples

There is no solid definition for what constitutes a complex network.

The following are widely considered to be complex networks:

- The Internet
- The World Wide Web
- Social networks (real-world and online)
- Various biological networks

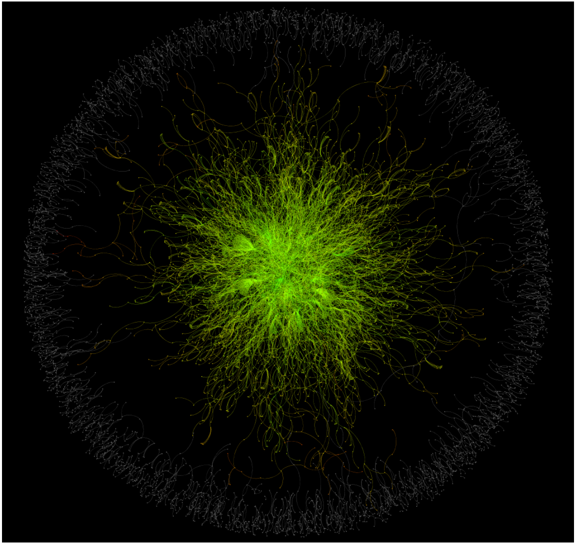
Simulation Motivation

Real-world complex networks can not be studied in their entirety because:

- The network is too large to store in computer memory.
- The network frequently changes.
- The resources to record the network are not available.

This motivates the simulation of complex networks using random graphs.

Random Graph Example



Sparse Graph Processes

Let G_n be a random graph of order n.
Consider the graph process $(G_n)_{n\geq 1}$.

Let $P_k^{(n)}$ be the proportion of vertices with degree k in G_n.

Definition

A graph sequence $(G_n)_{n\geq 1}$ is called *sparse* when

$$\lim_{n\to\infty} P_k^{(n)} = p_k, \tag{1}$$

for some deterministic limiting probability distribution $(p_k)_{k\geq 0}$.

Scale-Free Graph Processes

Definition

A sparse graph process $(G_n)_{n\geq 1}$ is *scale-free with exponent* α if

$$\lim_{k\to\infty} \frac{\log(p_k)}{-\log k} = \alpha \tag{2}$$

where $\alpha > 1$ and $(p_k)_{k\geq 0}$ is the limiting probability distribution for the degree of each vertex.

This means that for large n, each vertex has degree $\propto k^{-\alpha}$.

Contents

The Barabási-Albert Model

Denote the *Barabási-Albert* model with n vertices and parameter m by $\mathrm{BA}_n(m)$.

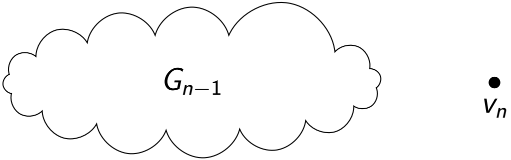
Can think of $\mathrm{BA}_n(m)$ as an algorithm which takes m and n as input and returns a random graph as output.

Alternatively $\mathrm{BA}_n(m)$ denotes the distribution of a random graph constructed by the *Barabási-Albert* model.

Informal Description (1)

We can construct $G_n \sim \mathrm{BA}_n(m)$ recursively.
Say we have a graph $G_{n-1} \sim \mathrm{BA}_{n-1}(m)$. To make G_n we:

1. Add a vertex v_n to G_{n-1}.

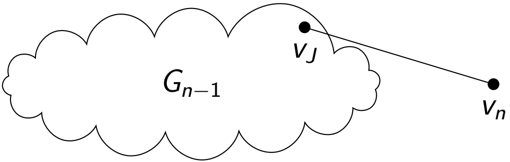


Informal Description (2)

2 Add edge $\{v_n, v_J\}$ where $v_J \in V(G_{n-1}) \cup \{v_n\}$ with probability

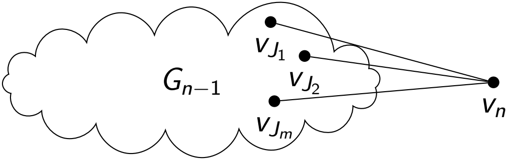
$$\mathbb{P}(J = j) = \frac{\delta_{jn} + \deg(v_j)}{1 + \sum_{v_i} \deg(v_i)}.$$

where δ_{ij} is the Kronecker delta.



Informal Description (3)

3. Repeat step 2 until m edges have been added to G_{n-1}. Call the resulting graph G_n.



Properties

For $m \geq 1$ we have:

- Degrees follow a power-law with $\alpha = 3$.
- Graph is not simple.
- Graph is not necessarily connected.

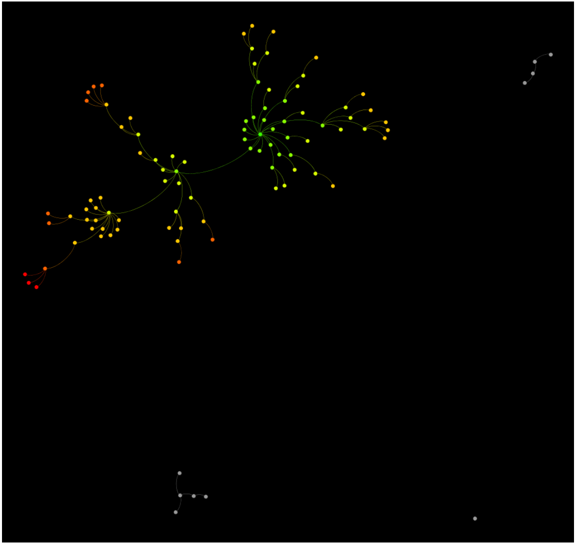
A simple and intuitive rule was used to simulate a complex network.
This "preferential attachment" scheme could explain how complex networks arise.

Efficient Barabási-Albert

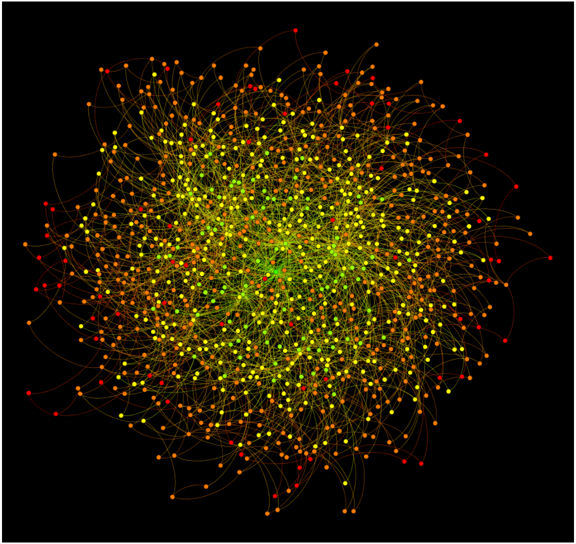
Algorithm 1: $\mathrm{BA}_n(m)$

Data: Number of vertices n, parameter m
Result: Barabási-Albert graph G
Initialize graph G to a single vertex with m self-loops;
Initialize degree stack $D = (1, 1, ..., 1)$ which has length m;
for $i \leftarrow 2$ **to** n **do**
    Add vertex v_i to G;
    for $j \leftarrow 1$ **to** m **do**
        Append i onto D;
        Generate $J \sim$ Discrete Uniform$(1, |D|)$;
        Add edge $\{v_{D(J)}, v_i\}$ to G;
        Append $(D(J))$ to D;
    end
end

$BA_{100}(1)$ Example



$BA_{1000}(2)$ Example



Contents

The Configuration Model

Denote $\text{Config}_n(\boldsymbol{x})$ as the *configuration* model with n vertices and degree sequence $\boldsymbol{x}$.

The output of the configuration model is be a random graph with degree sequence $\boldsymbol{x}$.

To make the random graph scale-free, we make the degree sequence a random vector $\boldsymbol{X}$.

Random Degree Sequences

The goal is for $\boldsymbol{X} = (X_1, \cdots, X_n)$ to exhibit power-law behaviour. Take $X_i \sim \text{Zeta}(\alpha)$, with $2 \leq \alpha \leq 3$.

Recall that the density function for the Zeta(α) distribution is

$$f_\alpha(k) = k^{-\alpha}/\zeta(\alpha), \; k = 1, 2, \cdots \tag{3}$$

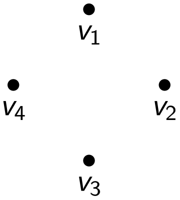
where $\zeta(s)$ is the Riemann-Zeta function.

Informal Description (1)

Assume that $\boldsymbol{x} = (x_1, x_2, \cdots x_n)$ has been simulated from $\boldsymbol{X}$ and that $\sum_i x_i$ is even. For example take $\boldsymbol{x} = (3, 1, 2, 2)$.

1 Add n vertices, $v_1, v_2, \cdots, v_n$
2 For each vertex v_i, add x_i copies of index i to stack H

$$H = (1, 1, 1, 2, 3, 3, 4, 4)$$



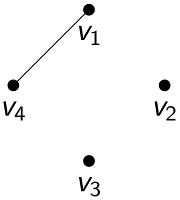
Informal Description (2)

3 Pop k off the top of stack H

4 Remove l uniformly from the rest of stack H

5 Add edge $\{v_k, v_l\}$

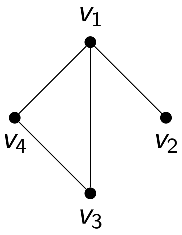
$$H = (\cancel{1}, 1, 1, 2, 3, 3, \cancel{4}, 4) = (1, 1, 2, 3, 3, 4)$$



Informal Description (3)

6 Repeat from step 3 until H is empty

$$H = ()$$



Properties

For $X_i \sim \text{Zeta}(\alpha)$ we have:

- Degrees follow a power-law with parameter α
- Graph is not necessarily simple
- Graph is not necessarily connected

The configuration model has greater flexibility than BA. However the configuration model does not explain how/why complex networks exist.

Formal Configuration I

Algorithm 2: $\text{Config}_n(\boldsymbol{x})$

Data: Number of vertices n, degree sequence x
Result: Multigraph G
Initialize the graph G to have n vertices;
Initialize empty stack of half-edges H;
for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** x_i **do**
        Push i onto H;
    end
end

Formal Configuration II

while $|H| \geq 2$ **do**

$i \leftarrow H[1]$;
Remove $H[1]$ from H;
Generate $U \sim \mathrm{DU}(1, |H|)$;
$j \leftarrow H[U]$;
Remove $H[U]$ from H;
Add edge $\{v_i, v_j\}$ to G;

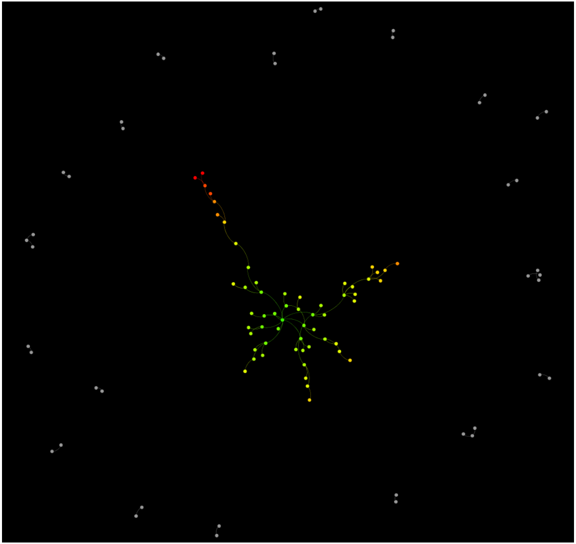
end

Forcing Simplicity

There are two main options to force the graph to be simple:

- Erased Configuration
  - If degree sum is odd, discard the left-over half-edge
  - Search and destroy parallel edges and self-loops
  - Advantage: fast and reliable
  - Disadvantage: each configuration does not occur with equal probability
- Repeated Configuration
  - Keep generating $\boldsymbol{X}$ and the random graph until the resulting graph is simple
  - Advantage: each configuration occurs with equal probability
  - Disadvantage: Slow and inefficient

$\text{Config}_{100}(\boldsymbol{X})$ Example ($\alpha = 2.5$)



$\mathrm{Config}_{1000}(\boldsymbol{X})$ Example ($\alpha = 3$)