Random Graphs

François Théberge
theberge@ieee.org

August 2019

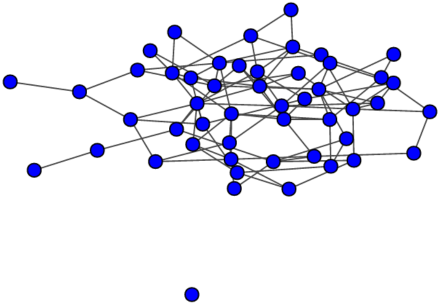
Outline

1. Erdos-Renyi models (ER)
2. Chung-Lu model (CL)
3. Configuration model

ER Models

The $G(n, m)$ model:

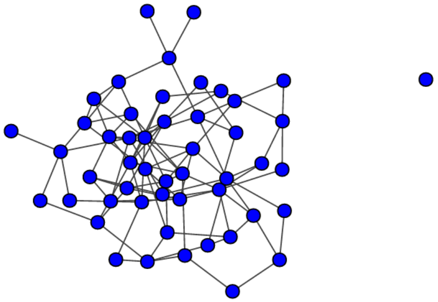
- n nodes
- m edges chosen at random from $N = \binom{n}{2}$ pairs
- average degree $k = 2m/n$



ER Models

The $G(n, p)$ model:

- n nodes
- each possible $N = \binom{n}{2}$ pair of nodes is connected with probability p
- expected number of edges is Np
- expected average degree $k = p(n-1)$



ER Models

The $G(n, p)$ model:

- with $p = m/N$, the expected average degree is $k = 2m/n$
- this is the model typically used in practice
- allows for easy calculation of graph statistics

ER Models

Number of edges:

Let P_m the probability of getting m edges with the $G(n,p)$ model, and let $N = \binom{n}{2}$.

$$P_m = \binom{N}{m} p^m (1-p)^{N-m}$$

Given that N is large and p is (typically) small, we can use the Poisson approximation to the binomial with $\lambda = Np$:

$$P_m \approx \frac{e^{-\lambda}\lambda^m}{m!}$$

ER Models

Degree distribution:

Let p_k, the probability that some node has degree k in $G(n, p)$.

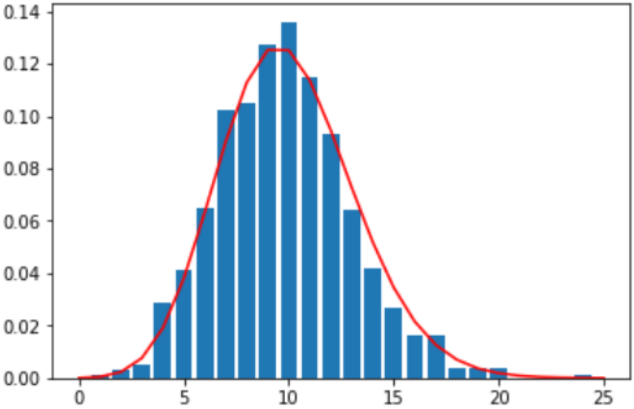
$$p_k = \binom{n-1}{k} p^k (1-p)^{n-1-k} \approx \frac{e^{-\lambda}\lambda^k}{k!}$$

with $\lambda = (n-1)p$, the expected average degree.

In view of the Poisson distribution, such models do not generate *hubs*, the high-degree nodes typically seen in real networks.

ER Models

Degree distribution ($n = 1000,\ p = .01$):



ER Models

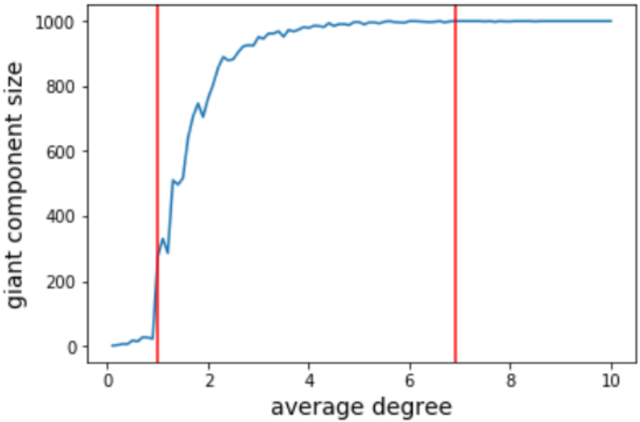
Giant connected component

For $G(n, p)$ with expected average degree $k = p(n - 1)$, we distinguish 3 regimes:

1. $k < 1$: subcritical regime; no giant component, clusters are mostly trees.
2. $k > 1$: supercritical regime; single giant component, small clusters are mostly trees.
3. $k >> \log(n)$: connected regime; no isolated nodes or small clusters.

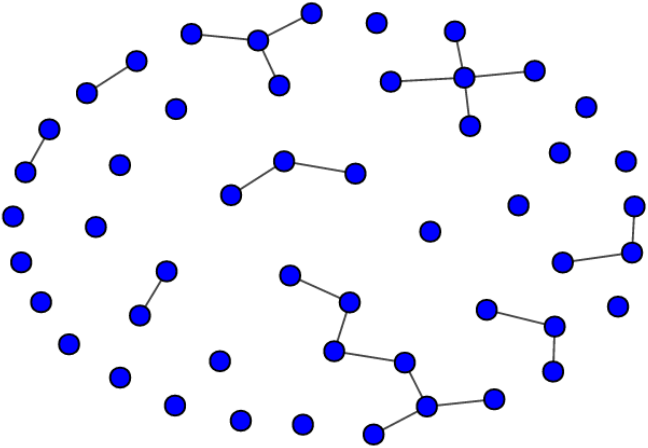
ER Models

Giant connected component with $n = 1000$:



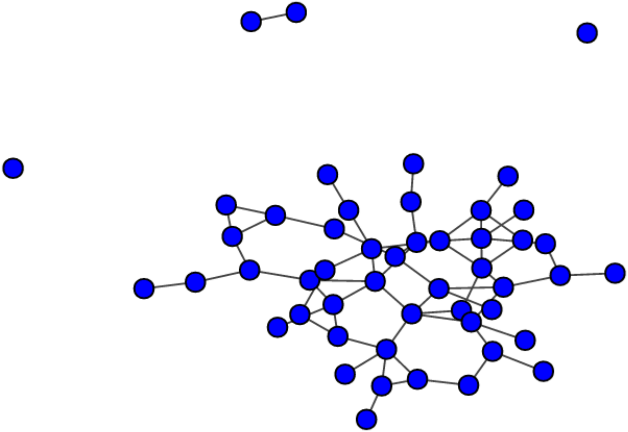
ER Models

Subcritical regime:



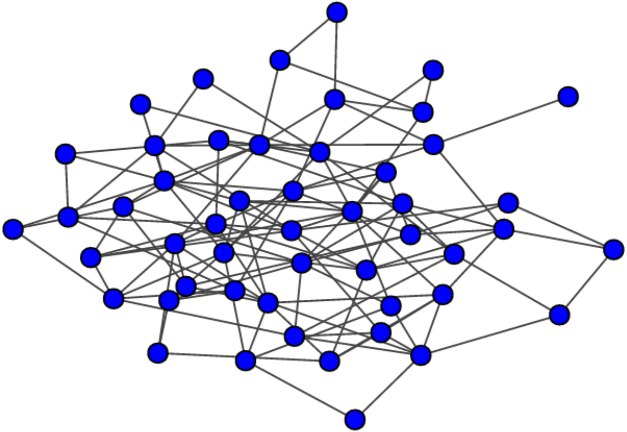
ER Models

Supercritical regime:



ER Models

Connected regime:



Chung-Lu Models

Next algorithms taken from [WDS]:

LAWRENCE LIVERMORE NATIONAL LABORATORY

LLNL-TR-678729

An In-Depth Analysis of the Chung-Lu Model

M. Winlaw, H. DeSterck, G. Sanders

October 28, 2015

Chung-Lu Models

Let G be a graph with vertices $V = \{v_1, \ldots, v_n\}$.

Assume also the degree sequence: $k_i = deg_G(v_i)$.

Model I: probability of an edge between vertices v_i and v_j is given by:

$$p_{ij} = \frac{deg_G(v_i)deg_G(v_j)}{vol(V)}, \ i \neq j \text{ and } p_{ii} = \frac{(deg_G(v_i))^2}{2vol(V)}.$$

This is also known as the Bernoulli Chung-Lu model.

Chung-Lu Models

Model I:

Algorithm 1: Bernoulli Chung-Lu Algorithm

```
for i = 1 to n do
    for j = i to n do
        Draw a random number from the uniform distribution on [0,1];
        if The random number is less than or equal to p_ij then
            /* Add edge (i, j) to the graph */
            if i ≠ j then
                a_ij = a_ji = 1;
            else
                a_ii = 2;
            end
        end
    end
end
```

Chung-Lu Models

If we define $\mathcal{CL}_1(G)$ to be the distribution of graphs obtained with Model I, then $\mathbb{E}_{G' \sim \mathcal{CL}_1(G)}(deg_{G'}(v_i)) = deg_G(v_i), 1 \leq i \leq n$.

For $G' = (V, E') \sim \mathcal{CL}_1(G)$:

- $\mathbb{E}(|E'|) = |E|$, but we may have $|E'| \neq |E|$,
- there are no multi-edges, and
- there can be self-edges.

This algorithm is not useful in practice, as it requires $O(n^2)$ multinomial experiments.

Chung-Lu Models

Model II is more practical algorithm with $O(|E|)$ steps only.

Generate a graph over vertices V by selecting $|E|$ edges $e = (u_1, u_2)$ where each u_i is independently sampled from V according to the multinomial distribution where $p(v_i) = deg_G(v_i)/vol(V)$.

Edges can be repeated, so what we have are expected number of edges instead of probabilities.

Chung-Lu Models

Model II:

Algorithm 3: $\mathcal{O}(|\text{Edges}|)$ Chung-Lu Algorithm

for $k = 1$ *to* m **do**
  Draw node i with probability $\frac{k_i}{2m}$;
  Draw node j with probability $\frac{k_j}{2m}$;
  `/* Add edge (i, j) to the graph */`
  if $i \neq j$ **then**
    $a_{ij} = a_{ji} = 1$;
  else
    $a_{ii} = 2$;
  end
end

Chung-Lu Models

If we define $\mathcal{CL}_2(G)$ to be the distribution of graphs obtained with Model II allowing for multi-edges, then $\mathbb{E}_{G' \sim \mathcal{CL}_2(G)}(deg_{G'}(v_i)) = deg_G(v_i), 1 \leq i \leq n$.

For $G' = (V, E') \sim \mathcal{CL}_2(G)$:

- we always have $|E'| = |E|$,
- there can be multi-edges, and
- there can be self-edges.

Chung-Lu Models

For Model II, we can ignore multi-edges, which reduces the overall expected degree (volume) of the graph.

For both models, we can ignore self-edges, again reducing the overall volume.

Chung-Lu Models

For model II:

Algorithm 4: $\mathcal{O}(|\text{Edges}|)$ Chung-Lu Algorithm without Self-Edges.

```
for k = 1 to m do
    Draw node i with probability k_i/2m;
    Draw node j with probability k_j/2m;
    /* Add edge (i, j) to the graph */
    if i ≠ j then
        a_ij = a_ji = 1;
    end
end
```

Chung-Lu Models

In summary:

		Probability of Edge (i, j)	Probability of Edge (i, i)	Expected Degree $E(\mathbf{D}_i)$
Bernoulli Chung-Lu	Self-Edges: Model I	$\frac{k_i k_j}{2m}$	$\frac{k_i^2}{4m}$	k_i
	No Self-Edges: Model III	$\frac{k_i k_j}{2m}$	0	$k_i - \frac{k_i^2}{2m}$
$\mathcal{O}(m)$ Chung-Lu	Self-Edges: Model II	$1-(1-2\frac{k_i k_j}{4m^2})^m$ $< \frac{k_i k_j}{2m}$	$1-(1-\frac{k_i^2}{4m^2})^m$ $< \frac{k_i^2}{4m}$	$\sum_{j\neq i} 1-(1-2\frac{k_i k_j}{4m^2})^m$ $+1-(1-\frac{k_i^2}{4m^2})^m < k_i$
	No Self-Edges: Model IV	$1-(1-2\frac{k_i k_j}{4m^2})^m$ $< \frac{k_i k_j}{2m}$	0	$\sum_{j\neq i} 1-(1-2\frac{k_i k_j}{4m^2})^m$ $< k_i - \frac{k_i^2}{2m}$

Table 1.2: Model Summary

Configuration Model

The Chung-Lu models are probabilistic models for edge generation, with *expected* degree sequence.

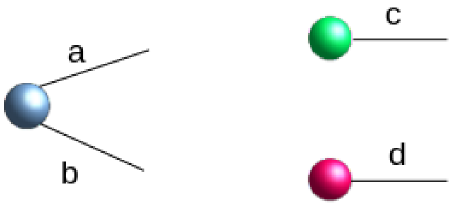
For the *configuration model*, we specify a precise degree sequence for the nodes $d_1, \ldots, d_n$.

Each graph with n nodes and this exact degree sequence is assigned the same probability.

Configuration Model

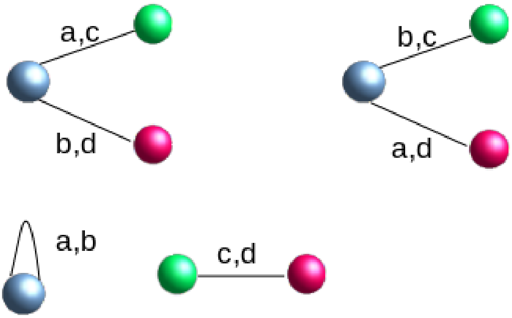
For each node i, we assign d_i stubs (half-edges);

Stubs are connected at random.



Configuration Model

This can generate self-edges and multi-edges



Configuration Model

For this model, the expected number of edges between nodes $i \neq j$ is:

$$e_{ij} = \frac{d_i d_j}{2m - 1}$$

and:

$$e_{ii} = \frac{d_i(d_i - 1)}{2(2m - 1)}$$

The overall expected number of edges is $m = \frac{1}{2} \sum_i d_i$.

Random Graphs

Notebook #2