Networks Structure and Dynamics
4. Random graph models

Maximilien Danisch, Marwan Ghanem, Lionel Tabourier

LIP6 – CNRS and Sorbonne Université

first_name.last_name@lip6.fr

October 9^{th} 2018

Outline

Random graphs – Motivation

1: understand the structure
Are the observed properties normal?
Answer: compare to a synthetic random graph

Draw randomly (**uniform probability**) in the set of graphs (of a given size)
→ common properties to the large majority of graphs
→ expected properties

2: simulate processes
Note: also possible with a non-random generative model

Motivation
Graphs with given density
Non-uniform random models
Graphs with fixed degree distribution

Random graphs – Motivation

1: understand the structure
Are the observed properties normal?
Answer: compare to a synthetic random graph

Draw randomly (**uniform probability**) in the set of graphs (of a given size)
→ common properties to the large majority of graphs
→ expected properties

2: simulate processes
Note: also possible with a non-random generative model

3/26

Random graphs – Motivation

1: understand the structure

Are the observed properties normal?
Answer: compare to a synthetic random graph

Draw randomly (**uniform probability**) in the set of graphs (of a given size)
→ common properties to the large majority of graphs
→ expected properties

2: simulate processes

Note: also possible with a non-random generative model

Outline

Erdős-Rényi model

$$G_{n,p}$$

- n nodes
- Any edge exists with a given probability p

Exercise: write a pseudocode to generate $G_{n,p}$

Complexity: $\mathcal{O}(n^2)$

Erdős-Rényi model

$$G_{n,p}$$

- n nodes
- Any edge exists with a given probability p

Exercise: write a pseudocode to generate $G_{n,p}$

Complexity: $\mathcal{O}(n^2)$

Erdös-Rényi model

$$G_{n,m}$$

- n nodes
- m edges chosen uniformly at random

Exercise: write a pseudocode to generate $G_{n,p}$. How about multiple edges?

Complexity: $\mathcal{O}(m)$

Erdös-Rényi model

$$G_{n,m}$$

- n nodes
- m edges chosen uniformly at random

Exercise: write a pseudocode to generate $G_{n,p}$. How about multiple edges?

Complexity: $\mathcal{O}(m)$

Equivalence between $G_{n,p}$ and $G_{n,m}$

In a $G_{n,p}$ ER graph, probability p is the density (δ)

$$p = \frac{2m}{n(n-1)}$$

$G_{n,m}$ et $G_{n,p}$ are “equivalent” if p and m verify this relationship

Notion of expected property

Example: Erdős-Rényi random graph $G_{n,m}$, $n = m = 4950$

Result: clique of 100 nodes and other 4850 nodes degree 0

Surprising?

Probability to have degree 0: $\mathcal{P}(k=0) = (1-p)^m \sim 0.135$.
each of the m link which could connect to the node does not exist

$\Rightarrow$ Expected number of degree 0 nodes:
$\mathcal{P}(k=0) \cdot n \sim 670$ to be compared with 4850...

$\rightarrow$ seems very unlikely

Notion of expected property

Example: Erdős-Rényi random graph $G_{n,m}$, $n = m = 4950$

Result: clique of 100 nodes and other 4850 nodes degree 0

Surprising?
Probability to have degree 0: $\mathcal{P}(k = 0) = (1 - p)^m \sim 0.135$.
each of the m link which could connect to the node does not exist

$\Rightarrow$ Expected number of degree 0 nodes:
$\mathcal{P}(k = 0) \cdot n \sim 670$ to be compared with 4850...

$\rightarrow$ seems very unlikely

Notion of expected property

Example: Erdős-Rényi random graph $G_{n,m}$, $n = m = 4950$

Result: clique of 100 nodes and other 4850 nodes degree 0

Surprising?
Probability to have degree 0: $\mathcal{P}(k = 0) = (1 - p)^m \sim 0.135$.
each of the m link which could connect to the node does not exist

$\Rightarrow$ Expected number of degree 0 nodes:
$\mathcal{P}(k = 0) \cdot n \sim 670$ to be compared with 4850...

$\rightarrow$ seems very unlikely

Notion of expected property

Example: Erdős-Rényi random graph $G_{n,m}$, $n = m = 4950$

Result: clique of 100 nodes and other 4850 nodes degree 0

Surprising?
Probability to have degree 0: $\mathcal{P}(k = 0) = (1 - p)^m \sim 0.135$.
each of the m link which could connect to the node does not exist

$\Rightarrow$ Expected number of degree 0 nodes:
$\mathcal{P}(k = 0) \cdot n \sim 670$ to be compared with 4850...

$\rightarrow$ seems very unlikely

Properties of Erdős-Rényi graphs

- Density
- Connectedness
- Average distance, diameter

Properties of Erdős-Rényi graphs

- Density set by operator
- Connectedness giant component, size $\mathcal{O}(n)$

  (if $m \geq \mathcal{O}(n)$)
- Average distance, diameter $\sim \log(n)$

  (for $m \geq \mathcal{O}(n)$)

Properties of Erdős-Rényi graphs

- Degree distribution
- Clustering coefficient
- Communities

Properties of Erdős-Rényi graphs

- Degree distribution homogeneous
- Clustering coefficient $\simeq$ density
- Communities no

Properties of Erdős-Rényi graphs

	real	ER
density	low	low
connectedness	giant comp.	giant comp.
distances	low	low
degree distrib.	heterogeneous	homogeneous
clustering	high	low
communities	yes	no

Conclusion on Erdős-Rényi graphs

Real-world complex networks are very different from random Erdős-Rényi graphs

Consequences

- Resemblances (connectedness, distances) can be explained with this simple model
- In general, not a good model for simulations, proofs . . .

→ Other models?

Outline

Small-World networks

Watts et Strogatz - *Nature, 1998*

Small-world: small average distance, high clustering

From a regular network, random reconnections of edges with probability p:

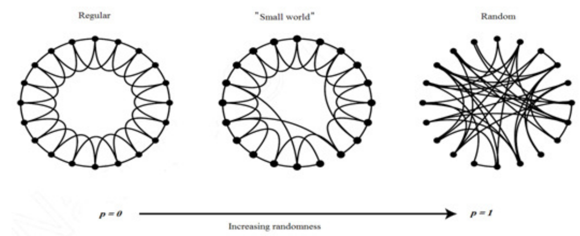
Produce high clustering but distribution non heterogeneous...

Small-World networks

Watts et Strogatz - *Nature, 1998*

Small-world: small average distance, high clustering

From a regular network, random reconnections of edges with probability p:



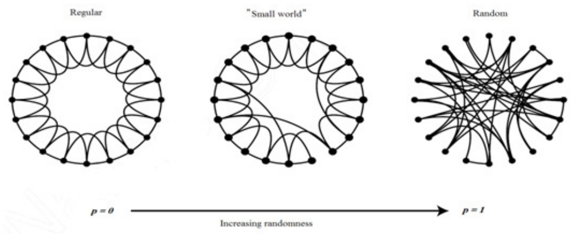
Produce high clustering but distribution non heterogeneous...

Small-World networks

Watts et Strogatz - *Nature, 1998*

Small-world: small average distance, high clustering

From a regular network, random reconnections of edges with probability p:



Produce high clustering but distribution non heterogeneous...

Properties of the *Small-World* model

	real	small-world
density	low	low
connectedness	giant comp.	giant comp.
distances	low	low
degree distrib.	heterogeneous	homogeneous
clustering	high	high
communities	yes	no

Scale-Free model

fr: modèle sans échelle

Barabási et Albert - *Science, 1999*

Graph built according to the **preferential attachment** law:

probability for a node i to be connected to a new comer proportional to degree $d^\circ(i)$

Motivation: this process leads to a power-law degree distribution (not proved here)

Ad hoc justification: generative process in agreement with the *"rich gets richer"* rule (or Merton's *"Matthew's effect"*)

Scale-Free model

fr: modèle sans échelle

Barabási et Albert - *Science, 1999*

Graph built according to the **preferential attachment** law:

probability for a node i to be connected to a new comer proportional to degree $d^\circ(i)$

Motivation: this process leads to a power-law degree distribution (not proved here)

Ad hoc justification: generative process in agreement with the *"rich gets richer"* rule (or Merton's *"Matthew's effect"*)

Scale-Free model

fr: modèle sans échelle

Barabási et Albert - *Science, 1999*

Graph built according to the **preferential attachment** law:

probability for a node i to be connected to a new comer proportional to degree $d^\circ(i)$

Motivation: this process leads to a power-law degree distribution (not proved here)

Ad hoc justification: generative process in agreement with the *"rich gets richer"* rule (or Merton's *"Matthew's effect"*)

Barabási-Albert scale-free model

Algorithm 1: BA generation (n nodes and $(n-n_0)\alpha+m_0$ edges)

Parameters: n , G, α (degree arriving node)
with G connected graph with n_0 nodes and m_0 edges,
for i *from* (n_0+1) *to* n **do**
  add node i to G
  $k=0$
  while $k<\alpha$ **do**
    draw $j\in[\![0;i-1]\!]$ with probability $\mathcal{P}(j)=\frac{d^\circ(j)}{\sum_{q=0}^{i-1} d^\circ(q)}$
    if $(i,j)\notin G$ **then**
      add edge (i,j) in G
      $k{+}{+}$
    end
  end
end

Properties of the *Scale-Free* model

	real	scale-free
density	low	low
connectedness	giant comp.	giant comp.
distances	low	low
degree distrib.	heterogeneous	heterogeneous
clustering	high	low
communities	yes	no

Outline

Generative method: configuration model

Degree distribution

$p_1, p_2, p_3, \ldots$

Draw nodes degree according to the distribution

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)

Draw random pairs of stubs and connect them

Take care of possible loops or multi-edges

Generative method: configuration model

Degree distribution

$p_1, p_2, p_3, \ldots$

Draw nodes degree according to the distribution

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)

Draw random pairs of stubs and connect them

Take care of possible loops or multi-edges

Generative method: configuration model

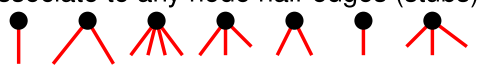
Degree distribution

$p_1, p_2, p_3, \ldots$

Draw nodes degree according to the distribution

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)



Draw random pairs of stubs and connect them

Take care of possible loops or multi-edges

Generative method: configuration model

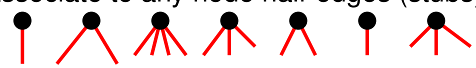
Degree distribution

$p_1, p_2, p_3, \ldots$

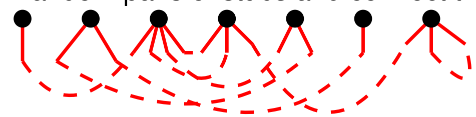
Draw nodes degree according to the distribution

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)



Draw random pairs of stubs and connect them



Take care of possible loops or multi-edges

Generative method: configuration model

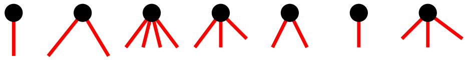
Degree distribution

$p_1, p_2, p_3, \ldots$

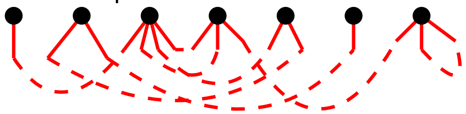
Draw nodes degree according to the distribution

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)



Draw random pairs of stubs and connect them



Take care of possible loops or multi-edges

Implementing the configuration model

Table : node i occurs exactly $d^\circ(i)$ times

0	1	1	2	2	2	2	3	3	3	4	4	5	6	6	6

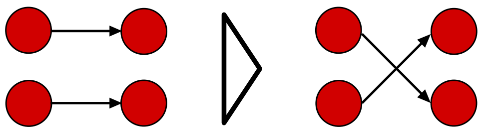
Algorithm 2: Generating a graph with fixed degree distribution

$i = 2m$
while $i > 0$ **do**
  u = random (0 , $i-1$)
  swap boxes u and $i-1$
  v = random (0 , $i-2$)
  swap boxes v and $i-2$
  i = i-2
  // edge (u, v) created
end

Other implementation: switching method

Principle

- we must start from a graph having the input degree distribution
- iterate switching of edges ends
- after a *sufficient amount* of switches, the graph produced is a random element of the set of graphs



Other implementation: switching method

Why does it work?

- The degree of any node remains unchanged

  so we keep the distribution unchanged
- The process is a Markov chain

  can be seen as a **random walk** in the set of graphs (defined by this degree distribution)

  after a while, we visit all elements with the same probability (not proved here)

  if we make enough switches, we obtain a random graph with this degree distribution

Other implementation: switching method

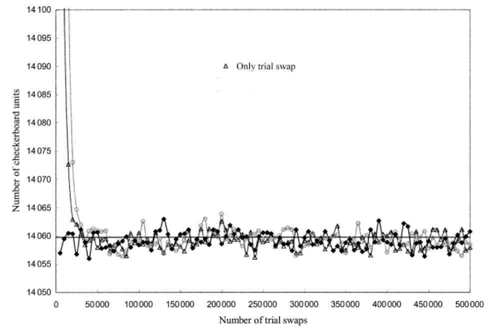
Why does it work?

- The degree of any node remains unchanged
  so we keep the distribution unchanged
- The process is a Markov chain
  can be seen as a **random walk** in the set of graphs (defined by this degree distribution)
  after a while, we visit all elements with the same probability (not proved here)
  if we make enough switches, we obtain a random graph with this degree distribution

Other implementation: switching method

When to stop switchings?

Measuring some features (ex: clustering) during the process until these features do not evolve any more...



Properties – Comparison

	real	Erdős-Rényi	fixed d.d.
density	low	low	low
connectedness	giant comp.	giant comp.	giant comp.
distances	low	low	low
degree	heterogene	homogene	heterogene
clustering	high	low	low
communities	yes	no	no

→ clustering is not a consequence of heterogeneous degree

Graph with fixed degree distribution: the bipartite case

Newman, Watts et Strogatz - *PNAS, 2002*

Example of the *Internet Movie Data Base*: what means a link between two actors?
Richer representation: network actor/movie

Vocabulary

- bipartite graph: *fr: graphe biparti*
  2 subsets of nodes A and B,
  links only connect nodes in A to nodes in B
- the actor network is a projection of this network, note that it has less information

Graph with fixed degree distribution: the bipartite case

The direct generative method (as well as the switching method) can be applied:

- using two degree distributions (for nodes A and B)
- connecting only nodes of A to nodes of B

Results

- explains clustering and degree in projections
  for some graphs
  in Newman *et al.*: *coboarding* ok, not in collaboration networks
- no large-scale structure (communities)

More models

Exercise: suggest other relevant models of networks.

Motivation
Graphs with given density
Non-uniform random models
Graphs with fixed degree distribution

More models

Exercise: suggest other relevant models of networks.

- Stochastic Block Model (SBM)
- Spatial models