MU5IN075
Network Analysis and Mining
5. Random Graph Models II

Esteban Bautista-Ruiz, Lionel Tabourier

LIP6 – CNRS and Sorbonne Université

first_name.last_name@lip6.fr

October 12, 2021

Notion of uniform generation

Until now 3 random models, 2 different families of models:

1. Erdős-Rényi
2. Watts-Strogatz, Barabási-Albert

Why are they fundamentally different?

1. ER: there is a target set (graphs with fixed density)
   all graphs have the same probability to be produced
2. BA, WS: no explicit target set...

$\Rightarrow$ ER model is uniform (or homogeneous)

Notion of uniform generation

Until now 3 random models, 2 different families of models:

1. Erdős-Rényi
2. Watts-Strogatz, Barabási-Albert

Why are they fundamentally different?

1. ER: there is a target set (graphs with fixed density)
   all graphs have the same probability to be produced
2. BA, WS: no explicit target set...

$\Rightarrow$ ER model is uniform (or homogeneous)

Notion of uniform generation

Until now 3 random models, 2 different families of models:

1. Erdős-Rényi
2. Watts-Strogatz, Barabási-Albert

Why is it important?

Because we cannot say that a BA is a *standard* SF graph
or that a WS is a *standard* graph with small-world properties

$\Rightarrow$ more relevant to have uniform graph generation

Notion of uniform generation

Until now 3 random models, 2 different families of models:

1. Erdős-Rényi
2. Watts-Strogatz, Barabási-Albert

Why is it important?

Because we cannot say that a BA is a *standard* SF graph or that a WS is a *standard* graph with small-world properties

$\Rightarrow$ more relevant to have uniform graph generation

Outline

1. Uniform graph generation with fixed degree distribution
   - Configuration model and variants
   - A few words on switching methods
   - The bipartite case

The configuration model

Degree distribution

$p_1, p_2, p_3, \ldots$

Draw nodes degree according to the distribution

$\rightarrow$ degree sequence

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)

Draw random pairs of stubs and connect them

Deal with possible loops or multi-edges

The configuration model

Degree distribution

$p_1, p_2, p_3, \ldots$

Draw nodes degree according to the distribution

$\rightarrow$ degree sequence

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)

Draw random pairs of stubs and connect them

Deal with possible loops or multi-edges

The configuration model

Degree distribution

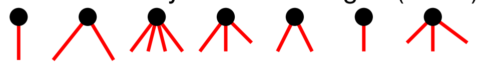
$p_1, p_2, p_3, \ldots$

Draw nodes degree according to the distribution

$\rightarrow$ degree sequence

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)



Draw random pairs of stubs and connect them

Deal with possible loops or multi-edges

The configuration model

Degree distribution

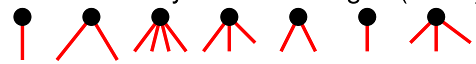
$p_1, p_2, p_3, \ldots$

Draw nodes degree according to the distribution

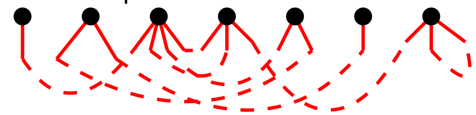
$\rightarrow$ degree sequence

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)



Draw random pairs of stubs and connect them



Deal with possible loops or multi-edges

The configuration model

Degree distribution

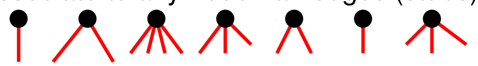
$p_1, p_2, p_3, \ldots$

Draw nodes degree according to the distribution

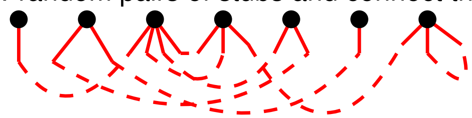
$\rightarrow$ degree sequence

1 2 4 3 2 1 3

Associate to any node half-edges (stubs)



Draw random pairs of stubs and connect them



Deal with 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 1: Generating a graph with fixed degree sequence

```
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
```

$*$ to be discussed...

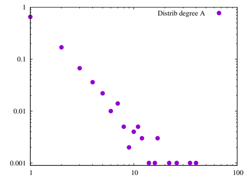
Deal with possible loops or multi-edges

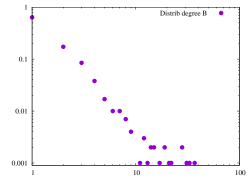
Answer 1: generation with rejection

Loop or multi-edge generated, restart the generation process

Deal with possible loops or multi-edges

- advantage: uniform generation
- drawback: can be long. . .





average number of trials (1000 nodes):

2180 17300

average generation time (1000 nodes):

1.2s 8.1s

Quiz: for what kind of distribution can it be long?

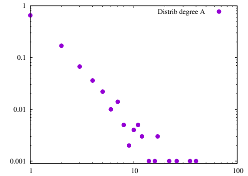
Deal with possible loops or multi-edges

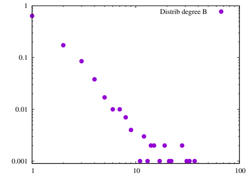
Answer 2: suppress loops or multiple edges

When a loop or a multiedge is generated, exclude it

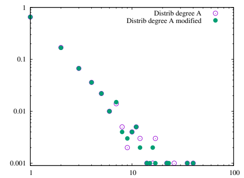
Deal with possible loops or multi-edges

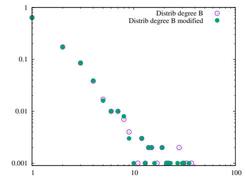
- advantage: fast
- drawback: does not have the exact degree sequence





after loops and multi-edges deletion, become:

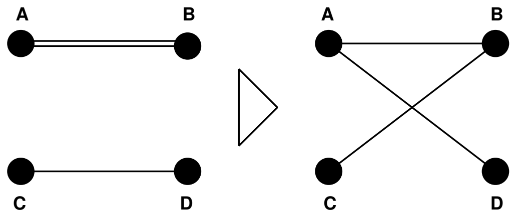




Deal with possible loops or multi-edges

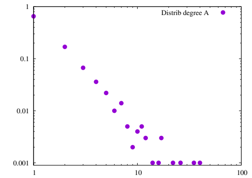
Answer 3: reconnect

When a loop or a multiedge is generated, switch to destroy it

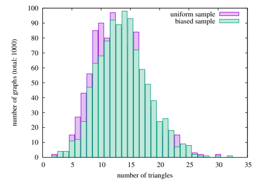


Deal with possible loops or multi-edges

- advantage: relatively fast, have the exact sequence
- drawback: not uniform = biased

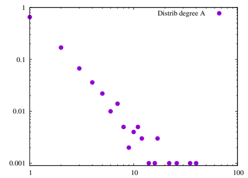


number of triangles for 1000 graphs

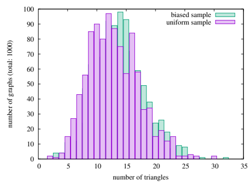


Deal with possible loops or multi-edges

- advantage: relatively fast, have the exact sequence
- drawback: not uniform = biased



number of triangles for 1000 graphs



Properties – Comparison

	real	fixed d.d.
density	low	?
connectedness	giant comp.	?
distances	low	?
degree	heterogeneous	?
clustering	high	?
communities	yes	?

Properties – Comparison

	real	fixed d.d.
density	low	low
connectedness	giant comp.	giant comp.
distances	low	low
degree	heterogeneous	heterogeneous
clustering	high	lower
communities	yes	no

→ heterogeneous degree only partly accounts for the c.c.
→ see practical work

Other implementation: switching method

Principle

- start from a graph with the given degree sequence
- iterate switching of edge 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 degree sequence unchanged
- The process is a Markov chain
  - can be seen as a **random walk** in the set of graphs (defined by this degree sequence)
  - 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 sequence

Uniform graph generation with fixed degree distribution
Configuration model and variants
A few words on switching methods
The bipartite case

Other implementation: switching method

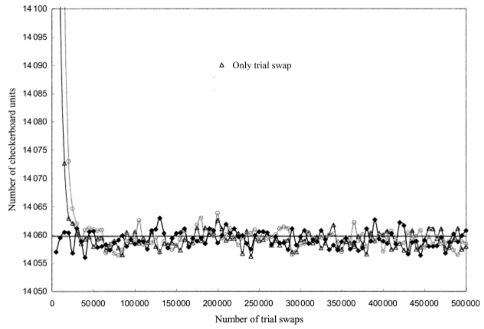
Why does it work?

- The degree of any node remains unchanged
  so we keep the degree sequence unchanged
- The process is a Markov chain
  - can be seen as a **random walk** in the set of graphs (defined by this degree sequence)
  - 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 sequence

Other implementation: switching method

When to stop switchings?

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

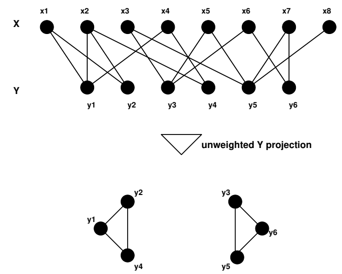


credits image: I.Miklós and J.Podani

Graph with fixed degree sequence: the bipartite case

Newman, Watts, Strogatz - *PNAS, 2002*

- Bipartite graph: two distinct types of nodes U and V
  $\rightarrow$ links between U and V
- Projection: if u_1 and u_2 connected to v in bipartite
  $\rightarrow$ u_1 and u_2 are connected in the U-projection



bipartite data richer, but not always available

Graph with fixed degree sequence: the bipartite case

Newman, Watts, Strogatz - *PNAS, 2002*

Underlying bipartite structure $\Rightarrow$ cliques in the projection

Bipartite configuration model

- fixed degree sequence for nodes X: $d_1^x, d_2^x, \ldots, d_{n_X}^x$
- fixed degree sequence for nodes Y: $d_1^y, d_2^y, \ldots, d_{n_Y}^y$
- random connections

$\rightarrow$ no possible self-loops, but multiedges still a problem

Graph with fixed degree sequence: the bipartite case

Newman, Watts, Strogatz - *PNAS, 2002*

Experimental results - comparison of the projections:

	average degree		clustering coef.	
projected network	Model	Real	Model	Real
Company directors	14.53	14.44	0.590	0.588
Movie actors	125.6	113.4	0.084	0.199
Physics collaboration	16.74	9.27	0.192	0.452

Conclusions:

- more realistic clustering
  than ER, or usual configuration model on unipartite networks
- still no large-scale structure
  visible on scientific collaboration networks

Graph with fixed degree sequence: the bipartite case

Newman, Watts, Strogatz - *PNAS, 2002*

Experimental results - comparison of the projections:

	average degree		clustering coef.	
projected network	Model	Real	Model	Real
Company directors	14.53	14.44	0.590	0.588
Movie actors	125.6	113.4	0.084	0.199
Physics collaboration	16.74	9.27	0.192	0.452

Conclusions:

- more realistic clustering
  than ER, or usual configuration model on unipartite networks
- still no large-scale structure
  visible on scientific collaboration networks

Perspective: more models

- Fix other constraints beyond degree distribution? but how?
- Exponential Random Graphs
- Stochastic Block Model
- Spatial models
- ...

$\rightarrow$ still many open research questions