COMPLEX NETWORKS:
STRUCTURE AND FUNCTIONALITY
III. EXPLORATION

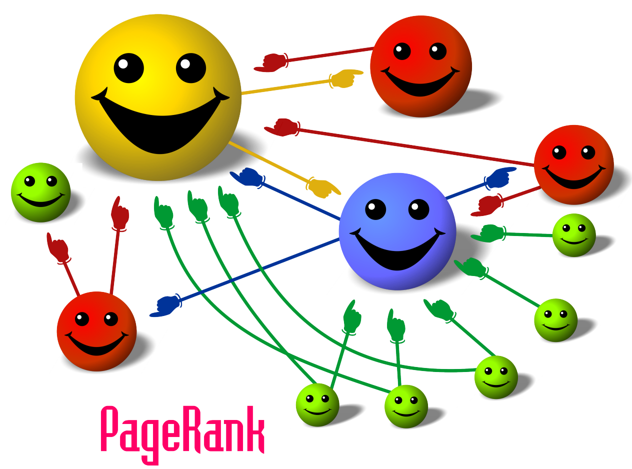
Frank den Hollander
Leiden University, The Netherlands

12th MSJ-SI,
Fukuoka, Japan, 31/07–09/08, 2019.

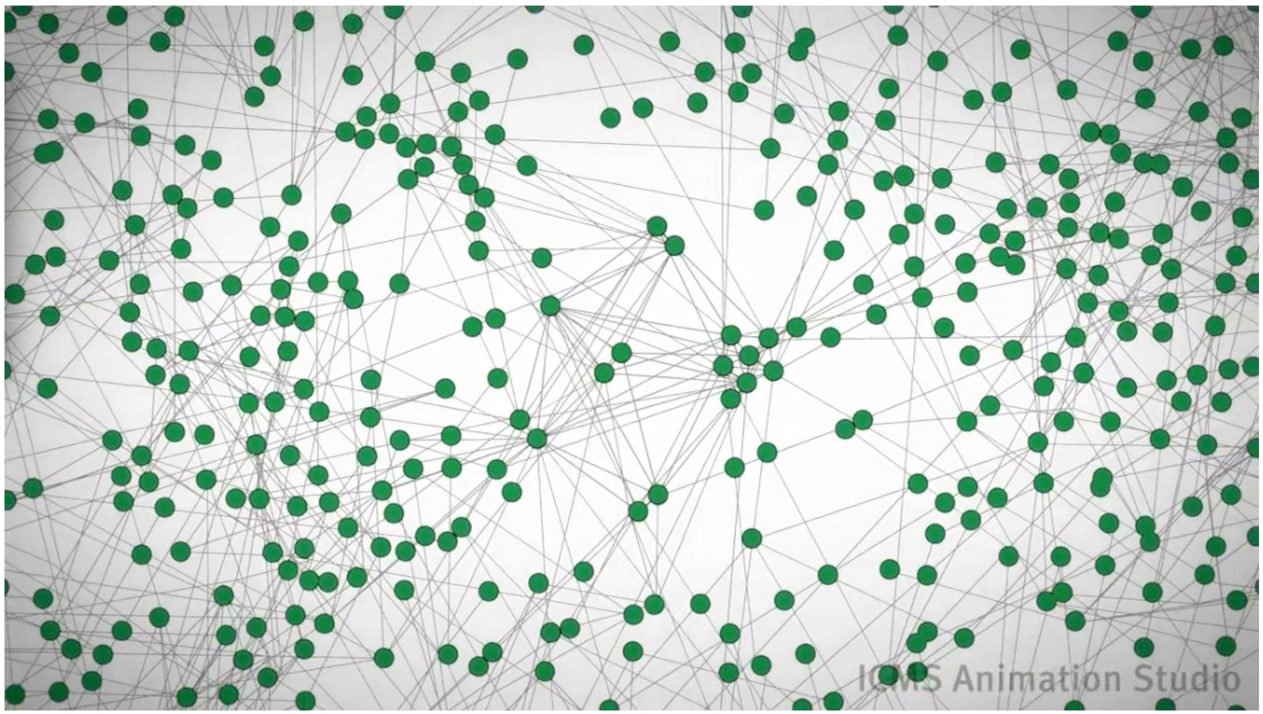
§ SEARCHING ON NETWORKS

Search algorithms on networks are important tools for the organisation of large data sets. A key example is Google PageRank, which assigns a weight to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of measuring its relative importance within the set.

The weights are assigned via exploration and are obtained recursively. A hyperlink counts as a vote of support: a page that is linked to by many pages with a high rank receives a high rank itself.



§ SEARCHING ON COMPLEX NETWORKS



▷ Networks are modelled as graphs, consisting of a set of vertices and a set of edges connecting pairs of vertices.

▷ Complex networks are modelled as random graphs, where the vertices and the edges are chosen according to some probability distribution.

▷ Search algorithms are modelled as random walks, moving along the network by randomly picking an edge incident to the vertex currently visited and jumping to the vertex at the other end.

KEY QUESTION

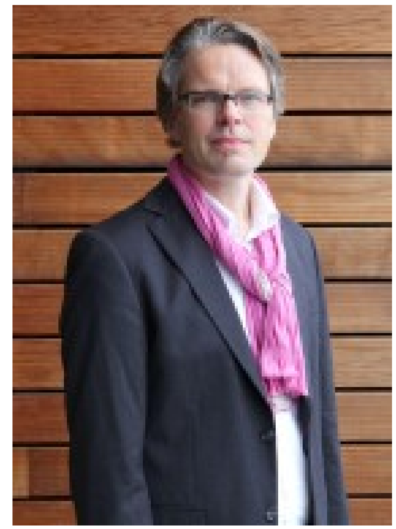
How long does it take the random walk to explore the random graph properly?

The answer to this question is important because it tells us how long the search algorithm must run.

The mixing time of a random walk is the time it needs to approach its stationary distribution.

For random walks on static random graphs, the mixing time has been the subject of intensive study. However, since many networks are dynamic in nature, it is natural to study random walks on dynamic random graphs.

This line of research is very recent in the mathematics literature.



Luca Avena | Hakan Guldas | Remco van der Hofstad

Papers:

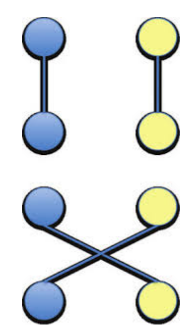
Annals of Applied Probability 2018

Stochastic Processes and Applications 2019

§ CONFIGURATION MODEL

The configuration model is a random graph with a prescribed degree sequence. It is popular because of its mathematical tractability and its flexibility in modelling real-world networks.

In this talk we consider a discrete-time dynamic version of the configuration model, where at each unit of time a certain fraction of the edges is rewired.



STATIC VERSION

Let $\mathcal{G}(\vec{d}_N)$ denote the set of all graphs on N vertices with a prescribed degree sequence

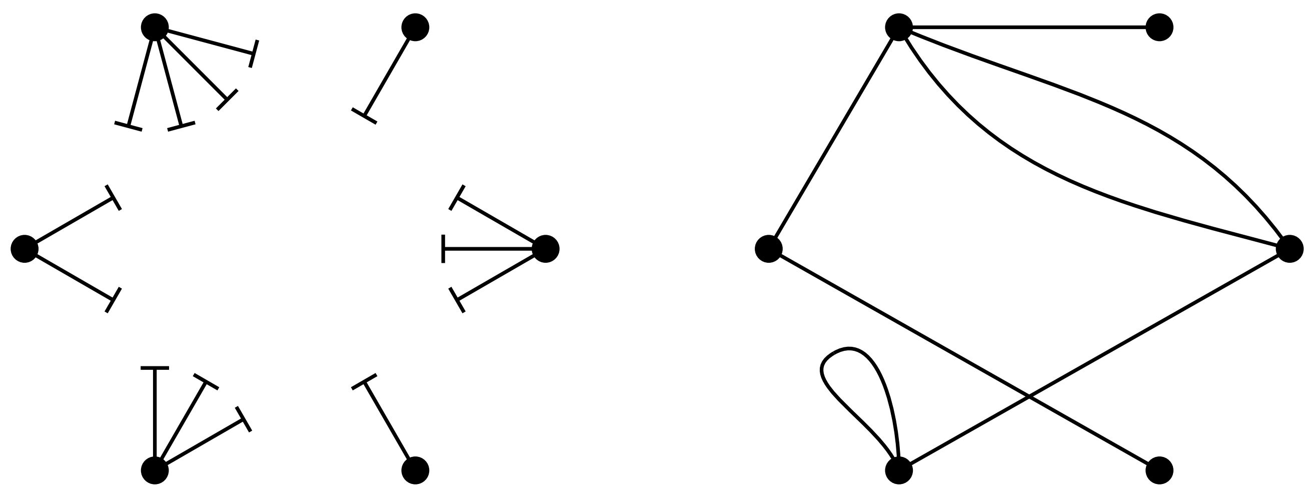
$$\vec{d}_N = (d_i)_{i=1}^N, \qquad \sum_{i=1}^N d_i = \text{ even.}$$

We draw a random graph uniformly from the set $\mathcal{G}(\vec{d}_N)$. The outcome may have self-loops and multiple edges. The stationary distribution of the random walk equals

$$\pi(i) = \frac{d_i}{\sum_{j=1}^N d_j}, \qquad 1 \leq i \leq N,$$

and does not depend on the outcome of the draw.

One way to generate the random graph is by randomly pairing half-edges:



Example with $N = 6$ and $\vec{d}_N = (1, 3, 1, 3, 2, 4)$

For random walk on the static configuration model, the mixing time is known to be

$$[1+o(1)]\, c \log N, \qquad N \to \infty,$$

with

$$\frac{1}{c} = \lim_{N\to\infty} \frac{\sum_{i=1}^{N} \log d_i}{\sum_{i=1}^{N} d_i},$$

subject to certain regularity assumptions on the degrees.

Lubetsky and Sly 2010

Ben-Hamou and Salez 2017

Berestycki, Lubetzky, Peres and Sly 2018

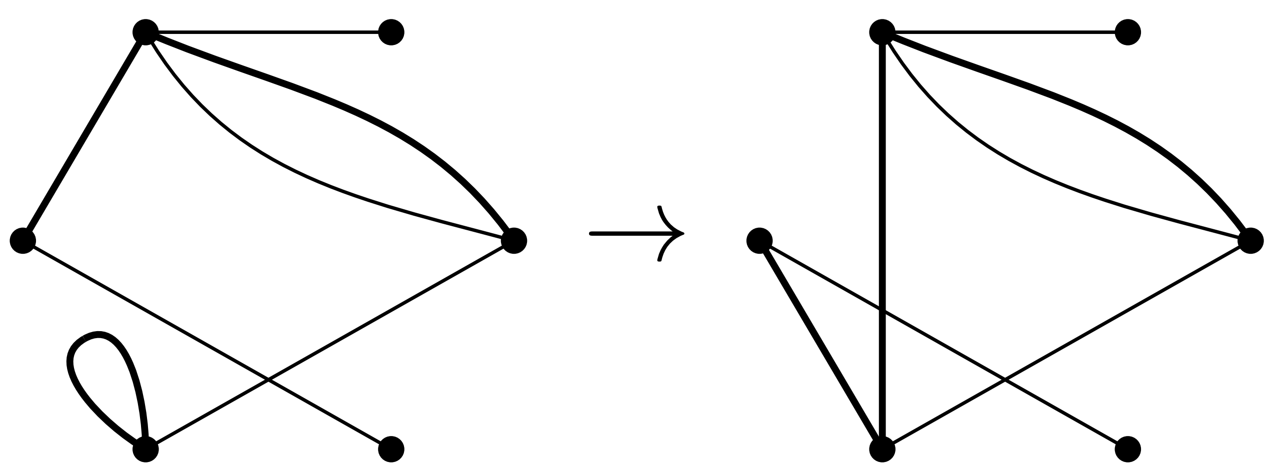
DYNAMIC VERSION

For fixed N, draw a starting graph η and a starting vertex i, and proceed as follows. At each time $t \in \mathbb{N}$:

1. Draw edges randomly with probability $\alpha_N \in (0, 1)$.

2. Rewire these edges by breaking them into half-edges and pairing these half-edges again randomly.

3. After the rewiring, let the random walk make a step to a randomly chosen neighbouring vertex.



Bold edges on the left are the ones chosen to be rewired.

Bold edges on the right are the newly formed edges.

KEY ASSUMPTIONS

- The degrees must be moderate, i.e., not too large.
- The random walk is non-backtracking, i.e., immediate jumps back along edges are not allowed.
- $\lim_{N\to\infty} \alpha_N = 0$, i.e., the dynamics is slow.

§ MIXING TIME

Let $\mathbb{P}_{\eta,i}$ denote probability with respect to the joint process of random graph and random walk with starting graph η and starting vertex i.

Let X_t denote the location of the random walk at time $t \in \mathbb{N}$, and write

$$\mathcal{D}_{\eta,i}(t) = \frac{1}{2} \sum_{j=1}^{N} |\mathbb{P}_{\eta,i}(X_t = j) - \pi(j)|$$

to denote total variation distance between the distribution of X_t and the stationary distribution π.

TRICHOTOMY

It turns out that there is are three regimes:

(1) $\lim_{N\to\infty} \alpha_N (\log N)^2 = \infty$
supercritical regime

(2) $\lim_{N\to\infty} \alpha_N (\log N)^2 = \beta \in (0, \infty)$
critical regime

(3) $\lim_{N\to\infty} \alpha_N (\log N)^2 = 0$
subcritical regime

MAIN THEOREM

With high probability, i.e., for a set of (η, i) *with probability tending to* 1 *as* $N \to \infty$, *the following hold.*

(1) supercritical regime:

$$\mathcal{D}_{\eta,i}(s/\sqrt{\alpha_N}) = e^{-s^2/2} + o(1), \quad s \in [0, \infty).$$

(2) critical regime:

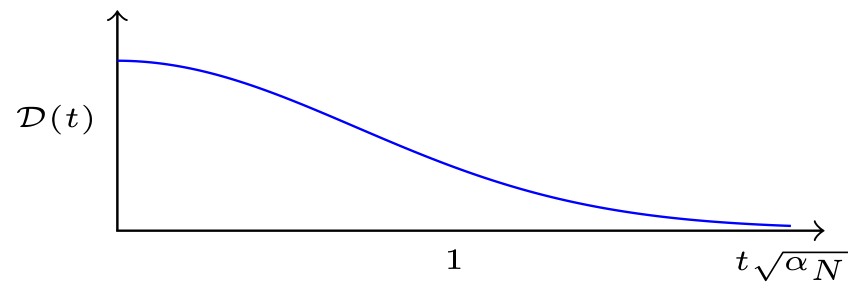
$$\mathcal{D}_{\eta,i}(s \log N) = \begin{cases} e^{-\beta s^2/2} + o(1), & s \in [0, c), \\ o(1), & s \in [c, \infty). \end{cases}$$

(3) subcritical regime:

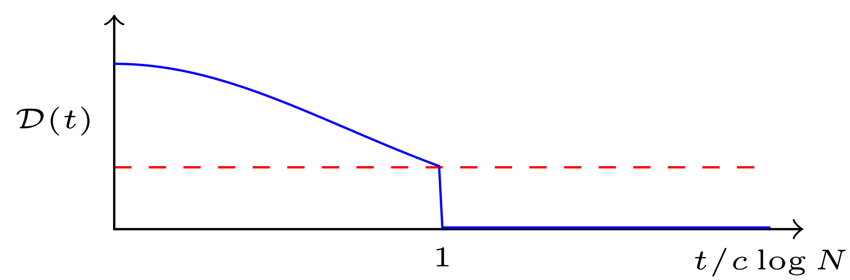
$$\mathcal{D}_{\eta,i}(s \log N) = \begin{cases} 1 - o(1), & s \in [0, c), \\ o(1), & s \in [c, \infty). \end{cases}$$

Here, c *is the constant in the static version.*

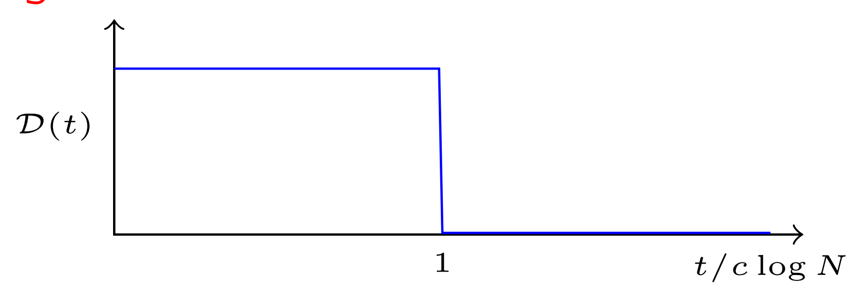
- supercritical regime:



- critical regime:



- subcritical regime:



REMARKS

- In the supercritical regime the mixing time is of order

$$1/\sqrt{\alpha_N} \ll \log N,$$

and does not depend on the degree sequence.

- In the critical regime and the subcritical regime the mixing time is of order $\log N$ and depends on the degree sequence.

- The proof is based on a stopping time argument: the first time the random walk moves along an edge that has been relocated is close to a strong uniform time.

§ FUTURE CHALLENGES

- ▷ What effect do hubs have on the mixing time?
- ▷ What happens when only edges touched by the random walk can be rewired?


Thank You