Complex Networks

Teachers: M. Emmerich, D. Garlaschelli, F. den Hollander.
Written examination: Wednesday, 31 January 2018, 14:00–17:00.

Open book exam: the lecture notes may be consulted, but no other material.

Answer each question on a separate sheet. Put your name, student number and the number of the question you are answering on each and every sheet. Provide full explanations with each of the answers!

Each question is weighted by a number of points, as indicated. The total number of points is 100. The final grade will be calculated as a weighted average: 30% for homework assignments and 70% for this exam.

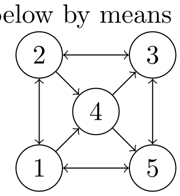
Success!

1. [**7 points**]
   Describe the structure of the World-Wide Web and explain why it has the features of a complex network.
2. Recall the definition of clustering coefficient and typical distance in a graph.
   - 2a. [**3 points**] Explain why the complete graph has clustering coefficient equal to 1 for all $n \geq 3$.
   - 2b. [**3 points**] Compute the typical distance of a square (4 vertices and 4 edges).
   - 2c. [**5 points**] Give a formula for the typical distance H_{L_n} when L_n is a linear graph with n vertices. Show that $\lim_{n\to\infty} \frac{1}{n} H_{L_n} = \frac{1}{3}$.
3.
   - 3a. [**4 points**] Draw all the possible outcomes of $\mathrm{PA}_2(1, \frac{1}{2})$, the preferential attachment graph with $n = 2$ vertices, $m = 1$ new vertices per iteration, and shift parameter $\delta = \frac{1}{2}$.
   - 3b. [**4 points**] What is the probability of each outcome?
   - 3c. [**5 points**] What is known about the scaling behaviour of $\mathrm{PA}_n(1, \frac{1}{2})$ as $n \to \infty$?
4. A real-world network $\mathbf{G}^*$ has N vertices and a constant degree sequence $\vec{k}^* = (k^*, \ldots, k^*)$.
   - 4a. [**5 points**] Consider the microcanonical ensemble of graphs having the same total number of links $L^* = L(\mathbf{G}^*)$ as the real network $\mathbf{G}^*$. Calculate the probability of generating a graph $\mathbf{G}$ in the ensemble as a function of its number $L(\mathbf{G})$ of links, with N and k^* appearing explicitly as the only parameters.

4b. [**5 points**] Consider the canonical ensemble of graphs having the same expected number of links $L^* = L(\mathbf{G}^*)$ as the number of links in the real network $\mathbf{G}^*$. Calculate the probability of generating a graph $\mathbf{G}$ in the ensemble as a function of its number $L(\mathbf{G})$ of links, with N and k^* appearing explicitly as the only parameters.

4c. [**5 points**] Compare each of the two ensembles considered above with the Erdos-Renyi random graph model with N vertices and connection probability $p^* = k^*/(N-1)$.

5. 5a. [**4 points**]
   Describe the graph below by means of a *graph formula* in `igraph`.

   5b. [**3 points**]
   What is the closeness centrality of node 4 and of node 2?

   5c. [**5 points**]
   What is the time complexity, in terms of $|V|$ and $|E|$, of computing the closeness centrality of a single node in a positively edge weighted graph using Dijkstra's algorithm? Consider the use of priority lists (inverted heap) and provide an asymptotic upper bound (in Big O notation) that is as close as possible.

6. Consider the problem of an epidemic with SI dynamics that spreads on an undirected circle graph of size N and contagiousness of the virus λ.

   6a. [**5 points**]
   Describe the structure of the generator matrix.

   6b. [**5 points**] Describe an efficient program that simulates the time it takes to infect the entire graph, starting from the initial state where only a single node is infected. You can use $\text{expDist}(\tau)$ to denote an exponentially distributed random number.

7. [**7 points**]
   Consider the configuration model $\text{CM}_n(D)$ with n vertices and degree sequence $D = (D_1, \ldots, D_n)$ whose components are i.i.d. random variables with probability distribution function f given by

$$f(1) = \tfrac{1}{3}, \quad f(2) = \tfrac{1}{6}, \quad f(4) = \tfrac{1}{2}, \quad f(k) = 0 \text{ otherwise.}$$

   A "hacker" removes each vertex of degree 4 with probability $q \in (0, 1)$, independently of other vertices. For what values of q does the "mutilated" random graph percolate for $n \to \infty$ (i.e., the largest connected component has a size of order n with a probability tending to 1 as $n \to \infty$)?

8. Consider the following model of an evolving graph with a fixed number N of vertices. The initial state of the graph is obtained by independently assigning to each vertex i a fitness value x_i randomly drawn from some probability distribution $\rho_0(x) \geq 0$ with support in the interval $(0, 1)$, and by connecting each pair of vertices with probability $p_{ij} = f(x_i, x_j)$, where $0 < f < 1$ is some function to be specified. Then, at each timestep the vertex with the minimum value of the fitness and its first neighbours in the graph are assigned new fitness values, independently drawn from the uniform distribution over the interval $(0, 1)$. If there are more vertices with the same fitness value, a random choice among these vertices is made. As soon as a vertex i receives a new fitness value x'_i, its connections to all other nodes $j \neq i$ are drawn anew with probability $p_{ij} = f(x'_i, x_j)$. The evolution described above is iterated indefinitely until a stationary state, characterized by a final fitness distribution $\rho_\infty(x)$, is reached.
Write the expression for $\rho_\infty(x)$ in the following cases:

   8a. [**4 points**] $\rho_0(x) = 1$ if $0 < x < 1$, i.e. the initial fitness is uniformly distributed in the interval $(0, 1)$, and $f(x_i, x_j) = c$, where $0 < c < 1$ is a constant.

   8b. [**4 points**] $\rho_0(x) = \delta(x - x_0)$, i.e. when all vertices have the same initial fitness value $x_i = x_0 \in (0, 1)$ $\forall i$, and $f(x_i, x_j) = c$, where $0 < c < 1$ is a constant.

   8c. [**4 points**] $\rho_0(x) = 1$ if $0 < x < 1$, i.e. the initial fitness is uniformly distributed in the interval $(0, 1)$, and $f(x_i, x_j) = cx_ix_j/(1 + cx_ix_j)$, where $c > 0$ is a constant.

   8d. [**4 points**] $\rho_0(x) = \delta(x - x_0)$, i.e. when all vertices have the same initial fitness value $x_i = x_0 \in (0, 1)$ $\forall i$, and $f(x_i, x_j) = cx_ix_j/(1 + cx_ix_j)$, where $c > 0$ is a constant.

9. The link-stub connection algorithm can be used to randomly create a graph with a desired degree distribution.

   9a. [**5 points**] Describe an efficient algorithm that creates a directed graph with maximal degree three and m edges, using the idea of repeated link-hub connection.

   9b. [**4 points**] What is the time complexity of your algorithm in the Big O notation?

SOLUTIONS

1. The WWW is a social network. The vertices of the WWW are electronic web pages, the edges are hyperlinks (or URLs) pointing from one web page to another. The WWW is therefore a directed network, since hyperlinks are not necessarily reciprocated. While Internet is physical, the WWW is virtual.

   With the rapid growth of the WWW, the interest in its properties is growing as well. It is of great practical importance to know what the structure of the WWW is, for example, to allow search engines like Goole Page Rank to explore WWW efficiently. The in-degrees have a power-law distribution with exponent $\tau_{\text{in}} \approx 2.1$, while the out-degrees have a power-law distribution with exponent $\tau_{\text{out}} \approx 2.5$. Thus, the WWW is scale free. It is also small world, with the typical distance growing only logarithmically in the size.

2a. In the complete graph K_n, all vertices are connected to each other, and so both the number of triangles Δ_{K_n} and the number of wedges W_{K_n} equals $\binom{n}{3}$. Hence $C_{K_n} = \Delta_{K_n}/W_{K_n} = 1$. We need $n \geq 3$ to make sure that triangles and wedges are possible.

2b. The square has 4 pairs of vertices at distance 1 and 2 pairs of vertices at distance 2. The typical distance therefore equals

$$\frac{(4 \times 1) + (2 \times 2)}{4 + 2} = \frac{4}{3}.$$

2c. The linear graph L_n has n vertices, labelled $V = \{1, \ldots, n\}$, and $n-1$ edges, labelled $E = \{(1,2), \ldots, (n-1, n)\}$. All vertices are connected to each other. Hence

$$\sum_{\substack{i,j \in V:\\ i \leftrightarrow j}} 1 = \sum_{1 \leq i < j \leq n} 1 = \tfrac{1}{2}n(n-1),$$

$$\sum_{\substack{i,j \in V:\\ i \leftrightarrow j}} d(i,j) = \sum_{1 \leq i < j \leq n} (j-i) = \tfrac{1}{6}n(n-1)(n+3).$$

   The typical distance equals $H_{L_n} = \frac{1}{3}(n+3)$. Hence $\lim_{n\to\infty} \frac{1}{n} H_{L_n} = \frac{1}{3}$.

3a. The two outcomes are drawn in Figure 5.3 of the lecture notes (last two pictures).

3b. The probabilities of the two outcomes are

$$\frac{1 + \frac{1}{2}}{3 + 2 \times \frac{1}{2}} = \frac{3}{8}, \qquad \frac{2 + \frac{1}{2}}{3 + 2 \times \frac{1}{2}} = \frac{5}{8}.$$

3c. It is know that as $n \to \infty$ the degree distribution converges to a limiting distribution f_{PA} given by

$$f_{\mathrm{PA}}(0) = 0, \qquad f_{\mathrm{PA}}(k) = \frac{5}{2}\frac{\Gamma(4)}{\Gamma(\frac{3}{2})}\frac{\Gamma(k+\frac{1}{2})}{\Gamma(k+4)}, \quad k \in \mathbb{N},$$

with Γ the Gamma function. For $k \to \infty$, $f_{\mathrm{PA}}(k)$ decays polynomially with exponent $\tau = \frac{7}{2}$.

4a. The number of links in $\mathbf{G}^*$ is $L^* = Nk^*/2$. There are $\binom{N(N-1)/2}{L^*} = \binom{N(N-1)/2}{Nk^*/2}$ graphs with N vertices and L^* links, and in the microcanonical ensemble each of these graphs is given the same probability, while graphs with a different number of links are given zero probability. So the probability of generating a graph $\mathbf{G}$ with $L(\mathbf{G})$ links is

$$P(\mathbf{G}) = \begin{cases} \dfrac{1}{\binom{N(N-1)/2}{Nk^*/2}} & \text{if } L(\mathbf{G}) = Nk^*/2 \\ 0 & \text{else} \end{cases}$$

4b. The canonical ensemble for a given number of links contains also the graphs that violate this number. The ensemble is generated by connecting each pair of nodes with probability equal to the observed link density, i.e. $p^* = 2L^*/N(N-1) = k^*/(N-1)$. Therefore the probability of generating a graph $\mathbf{G}$ with $L(\mathbf{G})$ links is

$$\begin{aligned} P(\mathbf{G}) &= (p^*)^{L(\mathbf{G})}(1-p^*)^{N(N-1)/2-L(\mathbf{G})} \\ &= [k^*/(N-1)]^{L(\mathbf{G})}[1-k^*/(N-1)]^{N(N-1)/2-L(\mathbf{G})}. \end{aligned}$$

4c. The microcanonical ensemble is different from the ER model, because the latter generates graphs with N vertices and any possible number of links L. The expected number of links in the ER model with connection probability $p^* = k^*/(N-1)$ is $\langle L\rangle = Nk^*/2$, which is equal to the only allowed number $L^* = Nk^*/2$ in the microcanonical ensemble.
The canonical ensemble coincides exactly with the ER model, because both generate links independently with probability $p^* = k^*/(N-1)$ (see point 4b).

5a.
```
g <- graph(c(1,2, 2,1, 2,3, 3,2, 1,5, 5,1, 3,5, 5,3, 3,4, 4,1, 2,4, 4,5))
g <- graph.formula(1-+2:4:5, 2-+3:1:4, 3-+2:5, 5-+1:3, 4-+1, 4-+5)
```

5b. The closeness centrality of node 2 is $\frac{1}{5}$, the closeness centrality of node 4 is $\frac{1}{4}$.

5c. All shortest path from that node have to be computed. Dijkstra's algorithm with priority lists can be executed. When it is continued until $T = \emptyset$, it computes all shortest paths in time $O(|E| + |V|\log|V|)$. The result is $1/\sum_{i=1}\mu(v_i)$.

6a. $a_{i,j} = \begin{cases} \lambda & \text{if } j = left(i) \\ \lambda & \text{if } j = right(i) \\ 2\lambda & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$

6b. The leave rate is constantly 2λ, and in each iteration one additional node gets infected. Thus:
For $i = 1$ to $|V|$ (totalTime = TotalTime + expDist($1/\lambda$));
Return totalTime.

7. The mutilated graph percolates for $n \to \infty$ if and only if $\bar{\nu} > 1$ with

$$\bar{\nu} = \frac{\sum_{k\in\mathbb{N}} k(k-1)\pi(k)f(k)}{\sum_{k\in\mathbb{N}} kf(k)}.$$

In our setting, we have $\pi(k) = 1$ for $k \neq 4$ and $\pi(4) = 1 - q$. Inserting the values of $f(k)$ for $k = 1, 2, 4$, we get

$$\bar{\nu} = \frac{(\frac{1}{3}\times 0) + (\frac{1}{6}\times 2) + (\frac{1}{2}\times 12(1-q))}{(\frac{1}{3}\times 1) + (\frac{1}{6}\times 2) + (\frac{1}{2}\times 4)} = \frac{2}{9} + 2(1-q).$$

Hence there is percolation if and only if $q < \frac{11}{18}$.

8a. The model conicides with the self-organized Bak-Sneppen model defined on a graph generated via the Fitness model (see Chapter 13). The initial condition for the fitness values (hence the choice of $\rho_0(x)$) is irrelevant, what matters is only the choice of $f(x_i, x_j)$. If f equals a constant c, this coincides with the random neighbour model. Then, using equations (13.27) and (13.28) of the Lecture Notes (with $p = c$), we can write

$$\rho_\infty(x) = \begin{cases} (\tau N)^{-1} & x \leq \tau \\ (c\tau N)^{-1} & x > \tau \end{cases}$$

where, for $N \to \infty$,

$$\tau = \frac{1}{1+cN} \to \begin{cases} 1 & \text{if} \quad cN \to 0 \\ (1+d)^{-1} & \text{if} \quad cN \to d > 0 \\ 0 & \text{if} \quad cN \to \infty \end{cases}$$

8b. Since $\rho_O(x)$ does not matter for determining $\rho_\infty(x)$, and since the choice of f is the same as in point 8a above, we retrieve the same expression for $\rho_\infty(x)$.

8c. In this case the choice of f is different, and coincides with that for the canonical configuration model, modulated by a constant c controlling the density of the graph. Using eqs. (13.29) and (13.30) (with $z = c$) of the Lecture Notes, we obtain

$$\rho(x) = \begin{cases} (\tau N)^{-1} & x \leq \tau \\ (\tau N)^{-1} + 2/(cN\tau^2 x) & x > \tau \end{cases}$$

where τ is

$$\tau = \sqrt{\frac{\phi(cN)}{cN}} \to \begin{cases} 1 & cN \to 0 \\ \sqrt{\phi(d)/d} & cN = d \\ 0 & cN \to \infty \end{cases}$$

Here $\phi(x)$ denotes the ProductLog function, defined as the solution of $\phi e^{\phi} = x$.

8d. Since $\rho_O(x)$ does not matter for determining $\rho_\infty(x)$, and since the choice of f is the same as in point 8c above, we retrieve the same expression for $\rho_\infty(x)$.

9a. Let n denote the number of nodes, V the set of nodes, and m the desired number of edges.

1. Produce a list L of $3|V|$ link stubs: $(l_{1,1}, l_{1,2}, l_{1,3}, ..., l_{n,1}, l_{n,2}, l_{n,3})$.
2. Produce a list Q of all pairs of link stubs. This has length $9|V|^2$.
3. Repeat.
4. $\quad S \leftarrow$ Apply Fisher Yates shuffle on Q. Stop at iteration m.
5. $\quad E \leftarrow$ first m pairs of S.
6. $\quad G \leftarrow (V, E)$.
7. Until (G does not contain self loops or multiple edges).
8. Return G.

9b. The algorithm has a time complexity of $O(|V|^2)$.