CocoNuTs
Complex Networks
@networksvox
Everything is connected

Complex Networks, CSYS/MATH 303
University of Vermont, Spring 2018
Assignment 3 • code name: "This is paintball"

Dispersed: Friday, February 2, 2018.
Due: Friday, February 16, by 11:59 pm, 2018.
Last updated: Tuesday, April 17, 2018, 11:00 am
Some useful reminders:
Deliverator: Peter Dodds
Office: Farrell Hall, second floor, Trinity Campus
E-mail: pdodds+coconuts@uvm.edu
Office hours: 10:05 am to 12:00 pm, Tuesday and Thursday
Course website: http://www.uvm.edu/pdodds/teaching/courses/2018-01UVM-303

All parts are worth 3 points unless marked otherwise. Please show all your workingses clearly and list the names of others with whom you collaborated.

Graduate students are requested to use LaTeX (or related TeX variant).

Email submission: PDF only! Please name your file as follows (where the number is to be padded by a 0 if less than 10 and names are all lowercase):
`CSYS303assignment%02d$firstname-$lastname.pdf` as in
`CSYS303assignment06michael-palin.pdf`

Please submit your project's current draft in pdf format via email. Please use this file name format (all lowercase after CSYS):
`CSYS303project-$firstname-$lastname-YYYY-MM-DD.pdf` as in
`CSYS303project-lisa-simpson-1989-12-17.pdf`

---

1. Okay, let's get back to the 6 networks we explored in the first assignment. Questions 2 through 4 will focus on them.

   Measure the degree-degree assortativity. This is the standard Pearson correlation coefficient and the focus is on links, and then the nodes at the end of each link.

   For undirected networks, we need to think about how we choose the ordering of an edge's two degrees when we perform the correlation. Which degree goes first? Or should we include both orderings? How about randomly choosing the ordering? Does it matter?

   For directed networks, various correlations are possible (in-in, in-out, etc.). For this question, measure the correlation of the in-degree of the source node and the out-degree of the destination node for each link.

2. Produce plots of the adjacency matrices.

3. Using a network visualization tool of your choice, produce plots of the networks (if possible, depending on size).

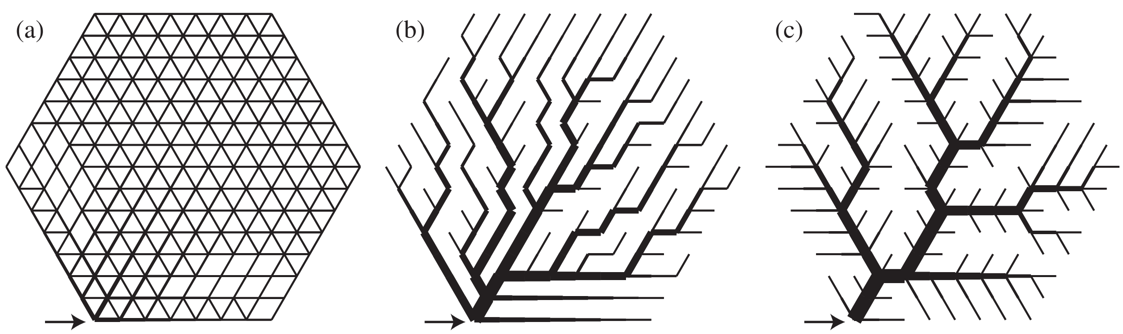
   For the smaller ones, please label the nodes numerically.

4. For river networks, basin areas are distributed according to $P(a) \propto a^{-\tau}$.

   Determine the exponent τ in terms of the Horton ratios R_n and R_s.

   Follow the same procedure shown in lectures for $P(\ell) \propto \ell^{-\gamma}$.

5. (3 + 3 + 3) Reproduce Bohn and Magnasco's Figs. 2a and 2b in [1]:



   Steps are given below but please read through the paper to understand how they set things up.

   The full team is encouraged to work together on Slack.

   (a) (3) Construct an adjacency matrix $\mathbf{A}$ representing the hexagonal lattice used in [1]. Plot this adjacency matrix.

   (b) (3 + 3) Run a minimization procedure to construct Figs. 2a and 2b which correspond to $\gamma = 2$ and $\gamma = 1/2$. Steps:

      i. Set each link's length to unity (the d_{kl}). The goal then reduces to minimizing the cost
      $$C = \sum_{k,l} |I_{kl}|^{\Gamma}$$
      where I_{kl} is the current on link kl and $\Gamma = 2\gamma/(\gamma + 1)$.

      ii. place a current source of nominal size i_0 at one node.

      iii. All other nodes are sinks, drawing a current of
      $$i_k = -\frac{i_0}{N_{\text{nodes}-1}}.$$

iv. Suggest setting $i_0 = 1000$ (arbitrary but useful value given the size of the network).

v. Generate an initial set of random conductances for each link, the $\{\kappa_{kl}\}$. These must sum to some global constraint as

$$K^\gamma = \sum_{k,l} \kappa_{kl}^\gamma.$$

Note: There seems to be no reason not to set $K = 1$ but the power of γ is a bit of a worry. (Also: we now have a lot of k types on deck.)

vi. Solve the following to determine the potential U at each node, and hence the current on each link using:

$$i_k = \sum_l \kappa_{kl}(U_k - U_l),$$

and then

$$I_{kl} = \kappa_{kl}(U_l - U_k).$$

Note: the paper erroneously has $I_{kl} = R_{kl}(U_l - U_k)$ below equation 4; there are a few other instances of similar miswritings of R_{kl} instead of κ_{kl}.

vii. Now, use scaling in equation (10) to compute a new set of $\{\kappa_{kl}\}$ from the I_{kl}. Everything boils down to

$$\kappa_{kl} \propto |I_{kl}|^{-(\Gamma-2)},$$

where the constant of proportionality is determined by again making sure $K^\gamma = \sum_{k,l} \kappa_{kl}^\gamma$.

Bonus: Please see reference 1 in [1] for a random connection to the next assignment's code name.

Some help—Let's sort out the key equation:

$$i_k = \sum_l \kappa_{kl}(U_k - U_l).$$

Each time we loop around through this equation, we know the i_k and the κ_{kl} and must determine the U_k. In matrixology, we love $\mathbf{A}\vec{x} = \vec{b}$ problems so let's see if we can fashion one:

$$i_k = \sum_l \kappa_{kl}(U_k - U_l)$$

$$= \sum_l \kappa_{kl} U_k - \sum_l \kappa_{kl} U_l$$

$$= U_k \sum_l \kappa_{kl} - \sum_l \mathbf{K}_{kl} U_l$$
$$= \lambda_k U_k - [\mathbf{K}\vec{U}]_k$$

where we have set $\lambda_k = \sum_l \kappa_{kl}$, the sum of the kth row of the matrix $\mathbf{K}$. We now construct a diagonal matrix Λ with the λ_k on the diagonal, and obtain:

$$\vec{i} = (\Lambda - \mathbf{K})\,\vec{U}.$$

The above is in the form $\mathbf{A}\vec{x} = \vec{b}$ so we can solve for $\vec{U}$ using standard features of R, Matlab, Python, ...(hopefully).

References

[1] S. Bohn and M. O. Magnasco. Structure, scaling, and phase transition in the optimal transport network. *Phys. Rev. Lett.*, 98:088702, 2007. pdf