P o C S What's The Story?

Principles of Complex Systems, CSYS/MATH 300
University of Vermont, Fall 2019
Assignment 6 • code name: Paradigms of Human Memory

Dispersed: Thursday, October 3, 2019.
Due: Friday, October 18, by 11:59 pm, 2019.
Some useful reminders:
Deliverator: Dr. Nick Allgaier (nicholas.allgaier@uvm.edu)
Assistant Deliverator: David Dewhurst (david.dewhurst@uvm.edu)
Office: To be disclosed
Office hours: To be emphatically determined through a democratic process
Course website: http://www.uvm.edu/pdodds/teaching/courses/2019-08UVM-300
Bonus course notes: http://www.uvm.edu/pdodds/teaching/courses/2019-08UVM-300/docs/dewhurst-pocs-notes.pdf

---

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.

Please obey the basic life rule: Never use Excel. Or any Microsoft product except maybe Xbox (which sadly will likely not help you here.)

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

email submission: 1. Please send to david.dewhurst@uvm.edu.
2. 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): `CSYS300assignment%02d$firstname-$lastname.pdf` as in `CSYS300assignment06michael-palin.pdf`

Please submit your project's current draft in pdf format via email. Please use this file name format (all lowercase after CSYS):
`CSYS300project-$firstname-$lastname-YYYY-MM-DD.pdf` as in
`CSYS300project-lisa-simpson-1989-12-17.pdf` where the date is the date of submission (and not, say, your birthdate).

---

1. The 1-d theoretical percolation problem:

   Consider an infinite 1-d lattice forest with a tree present at any site with probability p.

(a) Find the distribution of forest sizes as a function of p. Do this by moving along the 1-d world and figuring out the probability that any forest you enter will extend for a total length ℓ.

(b) Find p_c, the critical probability for which a giant component exists.

Hint: One way to find critical points is to determine when certain average quantities explode. Compute $\langle l \rangle$ and find p such that this expression goes boom (if it does).

2. Show analytically that the critical probability for site percolation on a triangular lattice is $p_c = 1/2$.

   Hint—Real-space renormalization gets it done.:

   Direct link: `http://www.youtube.com/watch?v=JlkbU5U7QqU`

3. (3 + 3)

   Coding, it's what's for breakfast:

   (a) Percolation in two dimensions (2-d) on a simple square lattice provides a classic, nutritious example of a phase transition.

   Your mission, whether or not you choose to accept it, is to code up and analyse the L by L square lattice percolation model for varying L.

   Take L = 20, 50, 100, 200, 500, and 1000.

   (Go higher if you feel L = 1000 is for mere mortals.)

   (Go lower if your code explodes.)

   Let's continue with the tree obsession. A site has a tree with probability p, and a sheep grazing on what's left of a tree with probability $1 - p$.

Forests are defined as any connected component of trees bordered by sheep, where connections are possible with a site's four nearest neighbors on a lattice.

Each square lattice is to be considered as a landscape on which forests and sheep co-exist.

Do not bagelize (or doughnutize) the landscape (no periodic boundary conditions—boundaries are boundaries).

(Note: this set up is called site percolation. Bond percolation is the alternate case when all links between neighboring sites exist with probability p.)

Steps:

i. For each L, run $N_{\textrm{tests}}$=100 tests for occupation probability p moving from 0 to 1 in increments of 10^{-2}. (As for L, you may use a smaller or larger increment depending on how things go.)

ii. Determine the fractional size of the largest connected forest for each of the $N_{\textrm{tests}}$, and find the average of these, $S_{\textrm{avg}}$.

iii. On a single figure, for each L, plot the average $S_{\textrm{avg}}$ as a function of p.

(b) Comment on how $S_{\textrm{avg}}(p; N)$ changes as a function of L and estimate the critical probability p_c (the percolation threshold).

Helpful reuse of code (intended for black and white image analysis): You can use Matlab's bwconncomp to find the sizes of components. Very nice.

4. (3 + 3)

(a) Using your model from the previous question and your estimate of p_c, plot the distribution of forest sizes (meaning cluster sizes) for $p \simeq p_c$ for the largest L your code and psychological makeup can withstand. (You can average the distribution over separate simulations.)

Comment on what kind of distribution you find.

(b) Repeat the above for $p = p_c/2$ and $p = p_c + (1 - p_c)/2$, i.e., well below and well above p_c.

Produce plots for both cases, and again, comment on what you find.

5. (3 + 3)

Repeat of the last question from Assignment 4, changing from $\gamma = 5/2$ to $\gamma = 3/2$. Now $1 < \gamma < 2$ so we should see a very different behavior.

You should be able to reuse everything you set up for Assignment 4.

Here's the question reprinted with γ switched.

For $\gamma = 3/2$, generate $n = 1000$ sets each of $N = 10$, 10^2, 10^3, 10^4, 10^5, and 10^6 samples, using $P_k = ck^{-3/2}$ with $k = 1, 2, 3, \ldots$

How do we computationally sample from a discrete probability distribution?

Hint: You can use a continuum approximation to speed things up. In fact, taking the exact continuum version from the first two assignments will work.

(a) For each value of sample size N, plot the maximum value of the $n = 1000$ samples as a function of sample number (which is not the sample size N). So you should have $k_{\max}$ for $i = 1, 2, \ldots, n$ where i is sample number. These plots should give a sense of the unevenness of the maximum value of k, a feature of power-law size distributions.

(b) For each set, find the maximum value. Then find the average maximum value for each N. Plot $\langle k_{\max} \rangle$ as a function of N and calculate the scaling using least squares.

Does your scaling match up with your theoretical estimate?

How to sample from your power law distribution (and kinds of beasts):

We now turn our problem of randomly selecting from this distribution into randomly selecting from the uniform distribution. After playing around a little, $k = 10^6$ seems like a good upper limit for the number of samples we're talking about.

Using Matlab (or some ghastly alternative), we create a cdf for P_k for $k = 1, 2, \ldots, 10^6$ and one final entry $k > 10^6$ (for which the cdf will be 1).

We generate a random number x and find the value of k for which the cdf is the first to meet or exceed x. This gives us our sample k according to P_k and we repeat as needed. We would use the exactly normalized $P_k = \frac{1}{\zeta(3/2)} k^{-3/2}$ where ζ is the Riemann zeta function.

Now, we can use a quick and dirty method by approximating P_k with a continuous function $P(z) = (\gamma - 1)z^{-\gamma}$ for $z \geq 1$ (we have used the normalization coefficient found in assignment 1 for $a = 1$ and $b = \infty$). Writing $F(z)$ as the cdf for $P(z)$, we have $F(z) = 1 - z^{-(\gamma-1)} = 1 - z^{-1/2}$. Inverting, we obtain $z = [1 - F(z)]^{-2}$. We replace $F(z)$ with our random number x and round the value of z to finally get an estimate of k.