PoCS What's The Story?

Principles of Complex Systems, Vols. 1, 2, & 3D
CSYS/MATH 300, 303, & 394
University of Vermont, Fall 2022
Assignment 07
Emergency B-Vord

Due: Friday, October 14, by 11:59 pm
https://pdodds.w3.uvm.edu/teaching/courses/2022-2023pocsverse/assignments/07/
Some useful reminders:
Deliverator: Prof. Peter Sheridan Dodds (contact through Teams)
Assistant Deliverator: Dylan Casey (contact through Teams)
Office: The Ether
Office hours: See Teams calendar
Course website: https://pdodds.w3.uvm.edu/teaching/courses/2022-2023pocsverse
Overleaf: LaTeX templates and settings for all assignments are available at
https://www.overleaf.com/project/631238b0281a33de67fc1c2b.

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

For coding, we recommend you improve your skills with Python, R, and/or Julia. The (evil) Deliverator uses (evil) Matlab.

Graduate students are requested to use LaTeX (or related TeX variant). If you are new to LaTeX, please endeavor to submit at least n questions per assignment in LaTeX, where n is the assignment number.

Assignment submission:

Via Blackboard.

Begin to think about projects.

See assignment 9 for instructions including details for the first presentation.

1. (3 + 3)

   You've earlier determined the theoretical scaling of the large sample of a power-law size distribution as a function of sample number.

   Let's see how well things match up with simulations.

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

How do we computationally sample from a discrete probability distribution?

Note: We examined some of these in class. See slides on power-law size distributions.

Perishing Monk Hint: You can use a continuum approximation to speed things up. See below.

(a) For each value of sample size N, sequentially create n sets of N samples. For each set, determine and record the maximum value of the set's N samples. (You can discard each set once you have found the maximum sample.)

   You should have $k_{\mathrm{max},i}$ for $i = 1, 2, \ldots, n$ where i is the set number. For each N, plot the n values of $k_{\mathrm{max},i}$ as a function of i.

   If you think of n as time t, you will be plotting a kind of time series.

   These plots should give a sense of the unevenness of the maximum value of k, a feature of power-law size distributions.

(b) Now find the average maximum value $\langle\rangle i k_{\mathrm{max},i}$ for each N.

   The steps again here are:
   1. Sample N times from P_k;
   2. Determine the maximum of the sample, k_{max};
   3. Repeat steps 1 and 2 a total n times and take the average of the n values of k_{max} you have obtained.

   Plot $\langle k_{\mathrm{max}} \rangle$ as a function of N on double logarithmic axes, and calculate the scaling using least squares. Report error estimates.

   Does your scaling match up with your theoretical estimate for $\gamma = 5/2$?

How to sample from your power law distribution (and similarly upsetting things):

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 $P_k = \frac{1}{\zeta(5/2)} k^{-5/2}$ 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 $F(z) = 1 - z^{-(\gamma-1)} = 1 - z^{-3/2}$. Inverting, we obtain $z = [1 - F(z)]^{-2/3}$. We replace $F(z)$ with our random number x and round the value of z to finally get an estimate of k.

2. (3 + 3 points) *Zipfarama via Optimization:*

   Complete the Mandelbrotian derivation of Zipf's law by minimizing the function

   $$\Psi(p_1, p_2, \ldots, p_n) = F(p_1, p_2, \ldots, p_n) + \lambda G(p_1, p_2, \ldots, p_n)$$

   where the 'cost over information' function is

   $$F(p_1, p_2, \ldots, p_n) = \frac{C}{H} = \frac{\sum_{i=1}^{n} p_i \ln(i+a)}{-g \sum_{i=1}^{n} p_i \ln p_i}$$

   and the constraint function is

   $$G(p_1, p_2, \ldots, p_n) = \sum_{i=1}^{n} p_i - 1 \quad (= 0)$$

   to find

   $$p_j = e^{-1-\lambda H^2/gC}(j+a)^{-H/gC}.$$

   Then use the constraint equation, $\sum_{j=1}^{n} p_j = 1$ to show that

   $$p_j = (j+a)^{-\alpha}.$$

   where $\alpha = H/gC$.

   3 points: When finding λ, find an expression connecting λ, g, C, and H.

   The Perishing Monks who have returned say the way is sneaky. Before collapsing, one monk mumbled something about substituting the form you find for $\ln p_i$ into H's definition (but do not replace p_i).

   Note: We have now allowed the cost factor to be $(j+a)$ rather than $(j+1)$.

3. (3 + 3) Carrying on from the previous problem:

   (a) For $n \to \infty$, use some computation tool (e.g., Matlab, an abacus, but not a clever friend who's really into computers) to determine that $\alpha \simeq 1.73$ for $a = 1$. (Recall: we expect $\alpha < 1$ for $\gamma > 2$)

   (b) For finite n, find an approximate estimate of a in terms of n that yields $\alpha = 1$.

   (Hint: use an integral approximation for the relevant sum.)

   What happens to a as $n \to \infty$?