P
o
C
S
What's
The
Story?

Principles of Complex Systems, Vols. 1, 2, & 3D
CSYS/MATH 6701, 6713, & a pretend number
University of Vermont, Fall 2024
"The seal is for marksmanship and the gorilla is for sand racing"
Assignment 06

Buster Bluth, Arrested Development, Afternoon Delight, S2E06.
Episode links: Wikipedia, IMDB, Fandom, TV Tropes.

Due: Monday, October 7, by 11:59 pm
https://pdodds.w3.uvm.edu/teaching/courses/2024-2025pocsverse/assignments/06/
Some useful reminders:
Deliverator: Prof. Peter Sheridan Dodds (contact through Teams)
Office: The Ether and/or Innovation, fourth floor
Office hours: See Teams calendar
Course website: https://pdodds.w3.uvm.edu/teaching/courses/2024-2025pocsverse
Overleaf: LaTeX templates and settings for all assignments are available at
https://www.overleaf.com/read/tsxfwwmwdgxj.

Some guidelines:

1. Each student should submit their own assignment.
2. All parts are worth 3 points unless marked otherwise.
3. Please show all your work/workings/workingses clearly and list the names of others with whom you ~~conspired~~ collaborated.
4. We recommend that you write up your assignments in LaTeX (using the Overleaf template). However, if you are new to LaTeX or it is all proving too much, you may submit handwritten versions. Whatever you do, please only submit single PDFs.
5. For coding, we recommend you improve your skills with Python, R, and/or Julia. **Please do not use any kind of AI thing.** The (evil) Deliverator uses (evil) Matlab.
6. There is no need to include your code but you can if you are feeling especially proud.

Assignment submission:

Via Brightspace (which is not to be confused with the death vortex of the same name).

Again: One PDF document per assignment only.

Please submit your project's current draft in pdf format via Brightspace four days after the due date for this assignment (normally a Friday). For teams, please list all team member names clearly at the start.

---

1. (3 + 3 + 3 points for each plot)

   Code up Simon's rich-gets-richer model.

   Plot Zipf distributions for $\rho = 0.10$, 0.01, and 0.001. and perform regressions to test $\alpha = 1 - \rho$.

   Run the simulation for long enough to produce decent scaling laws (recall: three orders of magnitude is good).

   Averaging over simulations will produce cleaner results so try 10 and then, if possible, 100.

   Note the first mover advantage.

2. (3 + 3 + 3 points) For Herbert Simon's rich-get-richer model of what we've called Random Competitive Replication, we found in class that the normalized number of groups in the long time limit, n_k, satisfies the following difference equation:

   $$\frac{n_k}{n_{k-1}} = \frac{(k-1)(1-\rho)}{1+(1-\rho)k} \tag{1}$$

   where $k \geq 2$. The model parameter ρ is the probability that a newly arriving node forms a group of its own (or is a novel word, starts a new city, has a unique flavor, etc.). For $k = 1$, we have instead

   $$n_1 = \rho - (1-\rho)n_1 \tag{2}$$

   which directly gives us n_1 in terms of ρ.

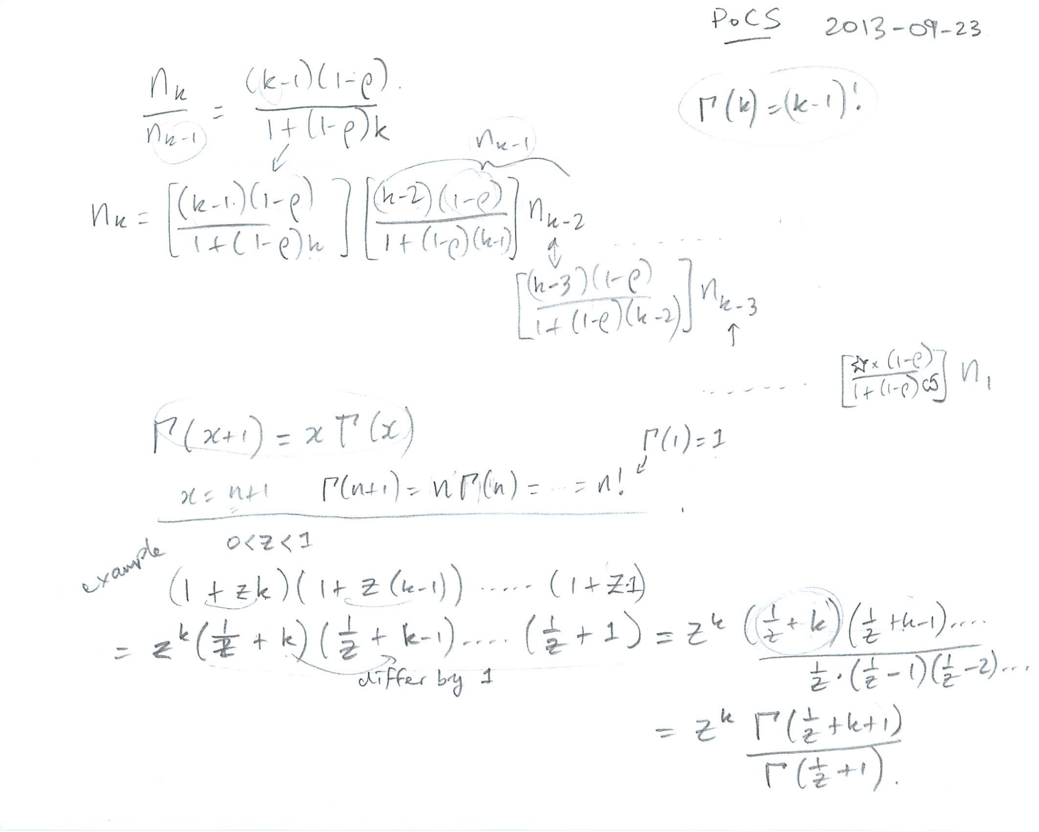
   (a) Derive the exact solution for n_k in terms of Gamma functions and ultimately the Beta function.

   (b) From this exact form, determine the large k behavior for n_k ($\sim k^{-\gamma}$) and identify the exponent γ in terms of ρ. You are welcome to use the fact that $B(x, y) \sim x^{-y}$ for large x and fixed y (or use Stirling's approximation directly on the Gamma functions that will appear).

   Note: Simon's own calculation is slightly awry. The end result is good however.

   Hint—Setting up Simon's model:
   http://www.youtube.com/watch?v=OTzI5J5W1K0

The hint's output including the bits not in the video:



PoCS 2013-09-23

$$\frac{n_k}{n_{k-1}} = \frac{(k-1)(1-\rho)}{1+(1-\rho)k}$$

$$\Gamma(k) = (k-1)!$$

$$n_k = \left[\frac{(k-1)(1-\rho)}{1+(1-\rho)k}\right]\underbrace{\left[\frac{(k-2)(1-\rho)}{1+(1-\rho)(k-1)}\right] n_{k-2}}_{n_{k-1}}$$

$$\left[\frac{(k-3)(1-\rho)}{1+(1-\rho)(k-2)}\right] n_{k-3}$$

$$\cdots\cdots \left[\frac{\star \times (1-\rho)}{1+(1-\rho)\star}\right] n_1$$

$$\Gamma(x+1) = x\,\Gamma(x) \qquad \Gamma(1) = 1$$

$$x = n+1 \qquad \Gamma(n+1) = n\,\Gamma(n) = \cdots = n!$$

example $0 < z < 1$

$$(1+zk)(1+z(k-1)) \cdots\cdots (1+z1)$$

$$= z^k\left(\frac{1}{z}+k\right)\left(\frac{1}{z}+k-1\right)\cdots\cdots\left(\frac{1}{z}+1\right) = z^k \frac{\left(\frac{1}{z}+k\right)\left(\frac{1}{z}+k-1\right)\cdots}{\frac{1}{z}\cdot\left(\frac{1}{z}-1\right)\left(\frac{1}{z}-2\right)\cdots}$$

differ by 1

$$= z^k \frac{\Gamma\left(\frac{1}{z}+k+1\right)}{\Gamma\left(\frac{1}{z}+1\right)}.$$

3. (3 points) What happens to γ in the limits $\rho \to 0$ and $\rho \to 1$? Explain in a sentence or two what's going on in these cases and how the specific limiting value of γ makes sense.

4. (6 + 3 + 3 points)

   In Simon's original model, the expected total number of distinct groups at time t is ρt. Recall that each group is made up of elements of a particular flavor.

   In class, we derived the fraction of groups containing only 1 element, finding

$$n_1^{(g)} = \frac{N_1(t)}{\rho t} = \frac{1}{2-\rho}$$

   (a) (3 + 3 points)
   Find the form of $n_2^{(g)}$ and $n_3^{(g)}$, the fraction of groups that are of size 2 and size 3.

   (b) Using data for James Joyce's Ulysses (see below), first show that Simon's estimate for the innovation rate $\rho_{\text{est}} \simeq 0.115$ is reasonably accurate for the version of the text's word counts given below.

Hint: You should find a slightly higher number than Simon did.

Hint: Do not compute ρ_{est} from an estimate of γ.

(c) Now compare the theoretical estimates for $n_1^{(g)}$, $n_2^{(g)}$, and $n_3^{(g)}$, with empirical values you obtain for Ulysses.

The data (links are clickable):

- Matlab file (`sortedcounts` = word frequency f in descending order, `sortedwords` = ranked words):
  https://pdodds.w3.uvm.edu/teaching/courses/2024-2025pocsverse/docs/ulysses.mat
- Colon-separated text file (first column = word, second column = word frequency f):
  https://pdodds.w3.uvm.edu/teaching/courses/2024-2025pocsverse/docs/ulysses.txt

Data taken from http://www.doc.ic.ac.uk/∼rac101/concord/texts/ulysses/.

Note that some matching words with differing capitalization are recorded as separate words.

5. (3 + 3)

Repeat the preceding data analysis for Ulysses for Jane Austen's "Pride and Prejudice" and Alexandre Dumas' "Le comte de Monte-Cristo" (in the original French), working this time from the original texts.

For each text, measure the fraction of words that appear only once, twice, and three times, and compare them with the theoretical values offered by Simon's model.

Download text (UTF-8) versions from https://www.gutenberg.org:

- Pride and Prejudice: https://www.gutenberg.org/ebooks/42671.
- Le comte de Monte-Cristo: https://www.gutenberg.org/ebooks/17989.

You will need to parse and count words using your favorite/most-hated language (Python, R, Perl-ha-ha, etc.).

Gutenberg adds some (non-uniform) boilerplate to the beginning and ends of texts, and you should remove that first. Easiest to do so by inspection for just two texts.

For a curated version of Gutenberg, see this paper by Gerlach and Font-Clos: https://arxiv.org/abs/1812.08092.

6. (3 + 3)

You've earlier determined the theoretical scaling of the largest 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 $\langle k_{\mathrm{max}} \rangle$ 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 of 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):

Because the tail of power-law size distributions can be so long, trying to sample from a discrete distribution can be either painfully slow or even computationally impossible. Brute force often works but not here.

We use a continuous approximation for P_k to make sampling both possible and fast.

We first approximate P_k with $P(z) = (\gamma - 1)z^{-\gamma}$ 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 $F(z) = 1 - z^{-(\gamma-1)} = 1 - z^{-3/2}$ when γ=5/2.

Inverting, we obtain $z = [1 - F(z)]^{-1/(\gamma-1)} = [1 - F(z)]^{-2/3}$ when γ=5/2.

We now replace $F(z)$ with our random number x and round the value of z to finally get an estimate of k.

In sum, given x is distributed uniformly on $[0, 1]$, then

$$k = \left[(1 - x)^{-2/3}\right]$$

is approximately distributed according to a power-law size distribution $P_k = ck^{-5/2}$ where $[\ \cdot\ ]$ indicates rounding to the nearest integer.