PoCS What's The Story?

Principles of Complex Systems, Vol. 1, CSYS/MATH 300
University of Vermont, Fall 2020
Assignment 04 • code name: A Fistful of Paintballs

Due: Friday, October 2, by 4:59 pm, 2020.
Relevant clips, episodes, and slides are listed on the assignment's page:
http://www.uvm.edu/pdodds/teaching/courses/2020-08UVM-300/assignments/04/
Some useful reminders:
Deliverator: Prof. Peter Sheridan Dodds (contact through Teams)
Assistant Deliverator: Michael Arnold (contact through Teams)
Office: The Ether
Office hours: Tuesdays, 12 to 12:50 pm; Wednesdays, 1:15 pm to 2:05 pm; Thursdays, 12 to 12:50 pm; all scheduled on Teams
Course website: http://www.uvm.edu/pdodds/teaching/courses/2020-08UVM-300

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.

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

Please submit your project's current draft in pdf format via Blackboard by the same time specified for this assignment. For teams, please list all team member names clearly at the start.

1. **Baby name frequencies in the US:**

   Plot the Complementary Cumulative Frequency Distributions and Zipf's law for the following:

   (a) Baby girl names in 1952.

   (b) Baby boy names in 1952.

   (c) Baby girl names in 2002.

   (d) Baby boy names in 2002.

As you did for the Google data set, fit regression lines and report values of γ and the Zipf exponent α.

We will revisit these distributions in following assignments.

Download:

Data for 1880 through 2018:

http://pdodds.w3.uvm.edu/permanent-share/pocs-babynames.zip (8.0M)

Files:

For each year, Zipf distribution of counts are stored in: `names-girlsYYYY.txt` and `names-boyYYYY.txt`.

For normalization to estimate rates, total number of births per year: `births_per_year.txt`. For this question, you do not need to determine rates, and this file is included for completeness.

For privacy, names with less than 5 counts are excluded.

Notes:

You should be able to re-use scripts from previous assignments.

Data is based on names registered through Social Security within the US.

Source:

Baby name dataset available here: https://catalog.data.gov/dataset?tags=baby-names. Separate dataset for total births available here: https://ssa.gov/oact/babynames/numberUSbirths.html.

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

   Show 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.

3. (3 + 3 + 3 points) For Herbert Simon's 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

$$n_1 = \rho - (1-\rho)n_1 \qquad (2)$$

which directly gives us n_1 in terms of ρ.

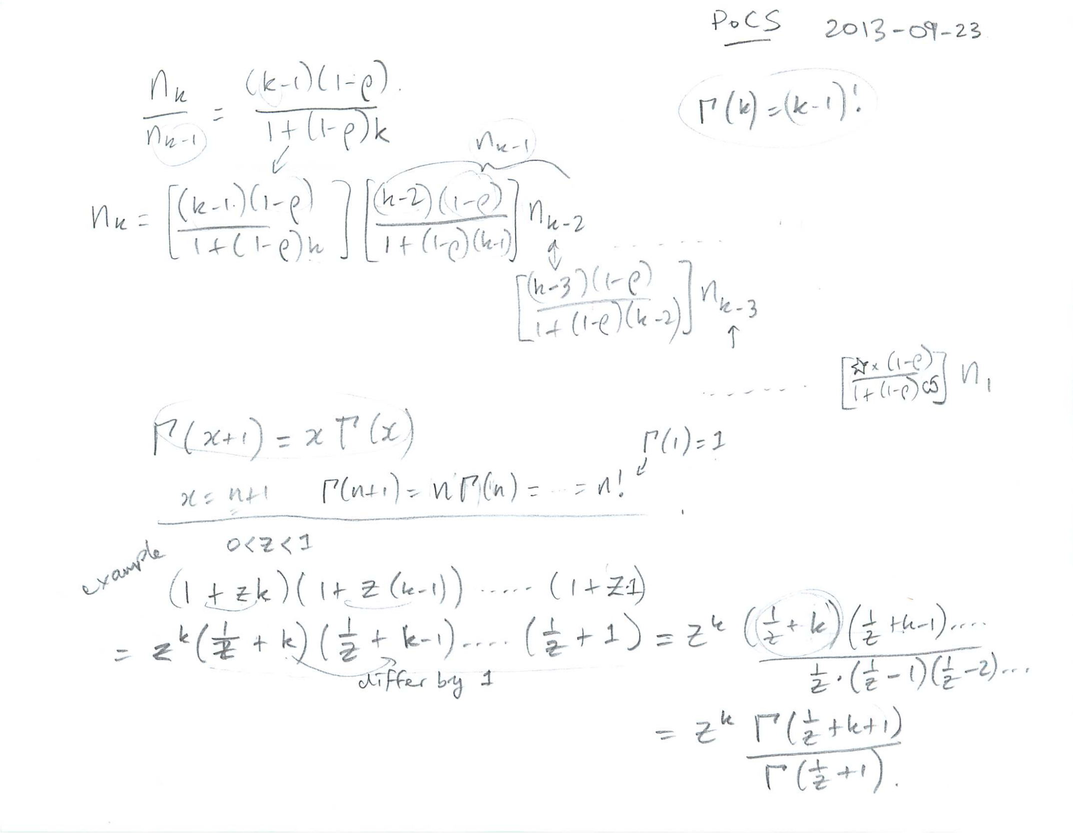
(a) Derive the exact solution for n_k in terms of gamma functions and ultimately the beta function.

(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 (use Stirling's approximation or possibly Wikipedia).

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

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

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

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

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

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

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

$$\Gamma(1) = 1$$

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

example $\quad 0 < z < 1$

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

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

differ by 1

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

4. 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.

5. (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):
     http://www.uvm.edu/pdodds/teaching/courses/2020-08UVM-300/docs/ulysses.mat
   - Colon-separated text file (first column = word, second column = word frequency f):
     http://www.uvm.edu/pdodds/teaching/courses/2020-08UVM-300/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.

6. (3 + 3)

Repeat the preceding analyses 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.

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

- Pride and Prejudice: https://www.gutenberg.org/ebooks/42671.
- Pride and Prejudice: https://www.gutenberg.org/ebooks/42671.

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.

7. (3 + 3)

   More on the peculiar nature of distributions of power law tails:

   Consider a set of N samples, randomly chosen according to the probability distribution $P_k = ck^{-\gamma}$ where $k \geq 1$ and $2 < \gamma < 3$. (Note that k is discrete rather than continuous.)

   (a) Estimate $\min k_{\max}$, the approximate minimum of the largest sample in the system, finding how it depends on N.

   (Hint: we expect on the order of 1 of the N samples to have a value of $\min k_{\max}$ or greater.)

   Hint—Some visual help on setting this problem up: http://www.youtube.com/watch?v=4tqlEuXA7QQ

   (b) Determine the average value of samples with value $k \geq \min k_{\max}$ to find how the expected value of $k_{\max}$ (i.e., $\langle k_{\max} \rangle$) scales with N.

   For language, this scaling is known as Heap's law.

8. (3 + 3)

   Let's see how well your answer for the previous question works.

   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.

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, 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 $k_{\max,i}$ for $i = 1, 2, \ldots, n$ where i is the set number. For each N, plot the n values of $k_{\max,i}$ 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 $\langle\rangle i k_{\max,i}$ 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 $\langle k_{\max} \rangle$ 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 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 $P_k = \frac{1}{\zeta(5/2)} k^{-5/2}$ 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.