



What's
The
Story?

Principles of Complex Systems, Vols. 1, 2, & 3D
CSYS/MATH 300, 303, & 394
University of Vermont, Fall 2022
Solutions to Assignment 06
The Truth Shall Make Ye Fret

Name: Kevin Kent

Conspirators: Shannon O'Connor, Daniel Forcade

1. 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 $k = 1, 2, 3, \dots$

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>

Solution:

See attached images on blackboard.



Notes:

- For language, this scaling is known as Heaps' law (Stigler's Law applies again).
- In a later assignment, we will test this scaling by (thoughtfully) sampling from power-law size distributions.

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.

Solution:

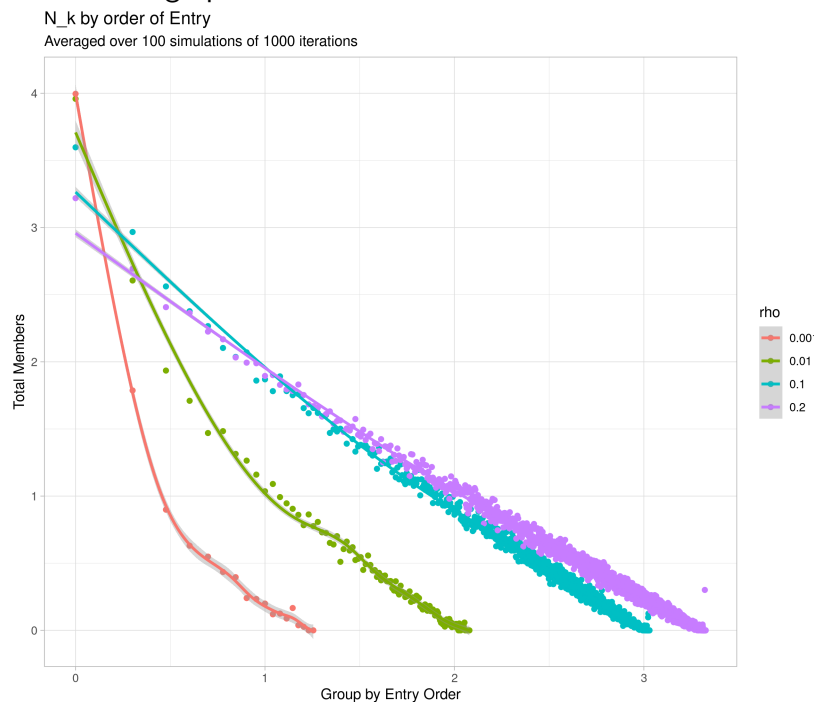
I created a rich-get-richer simulation using integers to represent unique entities and concatenated a list when I needed to add new entities. With a probability of ρ I added another unique item by adding an item with a name of one more than the max item (integer). Otherwise, I sampled once from the list.

For each simulation run, I ran it through 10000 steps. I did this 100 items for each ρ and aggregated the results by counting the number of occurrences for each type. For each ρ and type, I averaged over all the runs. Lastly, I found the rank by item and then did the log-log regression of average frequency on rank.

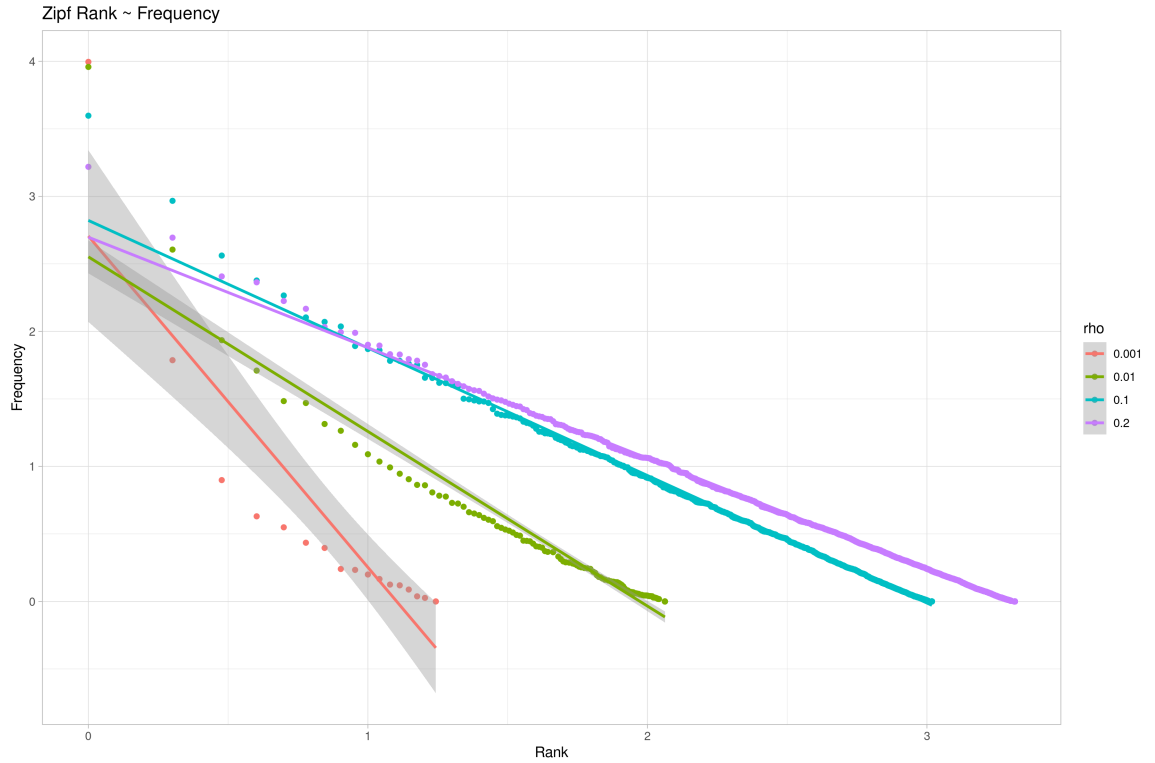
I ran an extra ρ value to see if the alpha became close to the expected value based on ρ as the innovation rate increased.

Below is a representation of the first-mover advantage, by showing the

First-mover graph:



Zipf graph:



Below are the results from fitting a linear regression on the log-transformed rank and frequency (averaged) by ρ , against the expected α given by $1 - \rho$. It is clear from my results that the expected zipf exponent that we derived using Simon's law is close to what it should be at higher innovation rates. However, when the innovation rate is lower and the first-mover advantage increases, this estimate begins to diverge drastically.

ρ	α	expected
0.001	2.45	1.00
0.01	1.29	0.99
0.10	0.94	0.90
0.20	0.82	0.80

□

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:

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

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

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

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=OTzl5J5W1K0>

Solution:

For part a and b please see attached images on blackboard.

□

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

Solution:

When ρ goes to 0 γ goes to 2 and when ρ goes to 1, gamma becomes undefined (1-1 in the denominator), so the limit diverges. We know from class that there is a relationship between γ and ρ :

$$\gamma = \frac{2-\rho}{1-\rho}$$

Thus by limiting γ to a certain range (between 1 and 2), we are keeping the ρ values in a range that makes sense for a probability of replication. Anything outside of that range would not be a valid probability.

□

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.

Solution:

Please see attached images on blackboard.

□

(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 .

Solution:

Using the provided data, I summarize the data to obtain the number of distinct words and total occurrences. In the case of a book, "time" is just the appearance of each word, thus the sum represents T , the total time. Rho is $\frac{N}{T}$. In this case I got a ρ of 0.1186 which is about 0.003 higher than Simon's estimate.

□

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

Solution:

Using the same procedure from above, I recalculated the proportion of observed groups with one, two and three observations. I also coded the formula I found above for N_1 , N_2 , and N_3 and calculated them as a function of ρ .

N_k	rho_data	rho_theo
N_1	0.56	0.53
N_2	0.16	0.17
N_3	0.07	0.08

□

The data (links are clickable):

- Matlab file (sortedcounts = word frequency f in descending order, sortedwords = ranked words):

<https://pdodds.w3.uvm.edu/teaching/courses/2022-2023pocsverse/docs/ulysses.mat>

- Colon-separated text file (first column = word, second column = word frequency f):

<https://pdodds.w3.uvm.edu/teaching/courses/2022-2023pocsverse/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 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> .

Solution:

To obtain ρ I read in the raw text files containing the book content, tokenized by word, and calculated word frequencies. I then summarized the dataset by calculating the number of distinct words and the total occurrences (which represent N and t , respectively). ρ is then N/t .

	book	ρ
Pride and Prejudice		0.05
Le comte de Monte-Cristo		0.10

After obtaining ρ I obtained the distribution of words by frequency and found the proportion of words that had a given frequency. Finally, I selected for words that had counts of 1, 2, or 3.

N_K	book	expected	prop_observed
N_1	Le comte de Monte-Cristo	0.53	0.51
N_2	Le comte de Monte-Cristo	0.17	0.15
N_3	Le comte de Monte-Cristo	0.08	0.08
N_1	Pride and Prejudice	0.51	0.38
N_2	Pride and Prejudice	0.17	0.14
N_3	Pride and Prejudice	0.08	0.10

□