Lab Session 1: Power Law distributions

Cristina Aguilera, Jesús M. Antoñanzas, Alex Carrillo
Cerca i Anàlisi de la Informació, Ciència i Enginyeria de Dades
UPC

September 2019

1 Distribution of family names

We begin by visually inspecting the data. After doing so, we are asked to plot the surnames' frequency with respect to their order. This plot shows a function that decreases very fast: we can only see a peak at the very first surnames and then it rapidly drops to 0. In order to better see its behaviour, we take the first 1000 samples and repeat the procedure, observing now the true character of the function. As we knew before, it tends to zero, but this time slower and with a shape very familiar: that of a power law, mainly because its tail is really long.
Now, we investigate this assumption. If the function were to be a power law, it would have this structure:

$$y = c \cdot (x + b)^a$$

We can, in very vague terms, forget about b, which allows us to approximate a power law to a linear function taking *log* on both sides:

$$\log y = a \cdot \log x + \log c$$

Applying this transformation to our data and plotting it, we can appreciate the linear nature of the plot (just the first terms behave oddly, again, it is a far approximation). This makes us think that, indeed, our data follows a power law. The next step is to find the constants a, b and c that best fit the approximation.

First, approximate a and c by solving a system of lineal equations composed by the log approximation of a power law evaluated at two large points (50000 and 70000, the reason being that low values are distorted because of the absence of b). Computing the linear system, we obtain the following values:

$$a = -1.5182, \quad \log c = 19.9820$$

From these results, we can see and deduce that as the function has a corresponding exponent of *approximately* -1, it follows Zipf's law.
These results are not quite satisfactory because, again, we are not taking into account the effect of the variable b. So, given the previous values of a and c, we minimize the mean squared error function for the variable b using scipy's *fmin* (reasons in **conclusions**), obtaining

$$b = 115.43$$

Plotting our approximated power law on top of the distribution of family names, we can see that it is a pretty close call. So, we would assume that, indeed, the latter follows a power law.

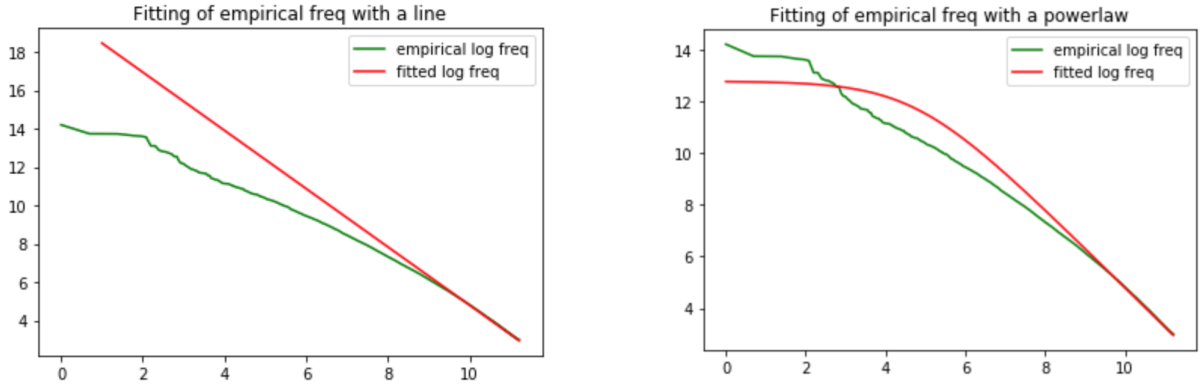


Figure 1: The empirical distribution fitted with our 2 parameter estimation [**left**] and with our estimated power law [**right**].

2 Distribution of river lengths

Using the same code as for **section 1**, we extract the length and the order of all rivers from the data set. Plotting them (*length* on the y axis and *order* on the x axis), one can see that *length* decreases very fast (does not mean it is an exponential decrease!). We also observe that the tail of this distribution is quite long with respect to other distributions (such as the exponential). The fact that the tail takes more or less half of the plot tells us that there are many rivers of short length, and although the curve has a strong similarity with a power law curve, we reckon there is not enough data to decide so just by looking at one simple plot. So, we compute the different parameters of a power law that better fit our data with the same methods as before:

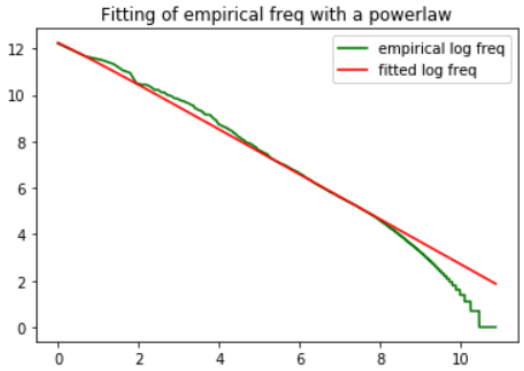
$$a = -0.55 \quad b = 3.63 \quad c = 17958.24$$

Plotting the curve, we see that it fits really well, so one can say that the data follows a power law.

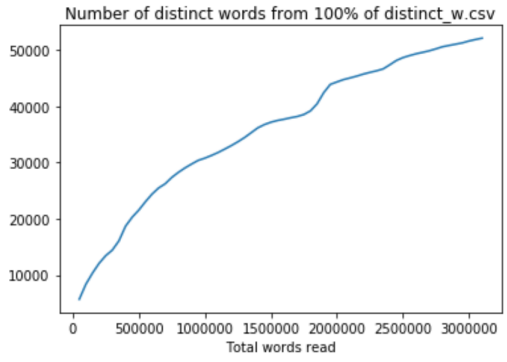
3 Words in text

Now, for another case study of Zip's Law, we collect all words from a directory of texts and save the number of occurrences of each one in a dictionary and from there to a .csv file. Then, we proceed as before creating a frequency plot from the .csv. We see the similarities with the plot of the first data set. It drops quickly to low frequencies, so its tail (words that appear few times) is very long and spans for more than 450.000 words. Examining the plot in more detail, we verify how rapidly it drops: taking into account 0.1% of the most frequent words, we see that between the word 40 and 50 it falls more than three quarters from the top frequency. Taking logarithms at both axis, it clearly resembles a line. In this case, there is no big influence of the variable b in the straightness of the line at low values because of the sheer volume of words in the dictionary.

For a second experiment, this time an experiment on Heaps' law, we modify the code so that for a given k, it saves the amount of distinct words having read ik total words.
Plotting the number of distinct words for every ik words, we see how the frequency of distinct words appearing in texts decreases as the number of total words processed increases. In spite of this general tendency, having read more than 3.000.000 words, there were still appearing new words, indicating that the appearance rate is not zero: it follows a power law.



(a) Fitted data (first experiment).



(b) Number of distinct words (second experiment).

Figure 2: Zip's & Heaps' Laws illustrated.

4 Conclusions

We have taken a glimpse at how some natural phenomena follow a power law distribution. One could assume, then, that many more also follow it (which is actually the case). Regarding what could be improved in this analysis, we think that the way the variables have been found can be improved, that is, using a more complex optimization algorithm. What we have used (*fmin*) performs the downhill simplex algorithm, which does not use first nor second derivatives and is regarded by some as not as great as some others. The reason for choosing it is because of the difficulties we have had trying methods like *steepest descent* or even *BFGS*: numerical errors arose when performing line search, which we think are related to our lack of previous study of the problem given that it is not the focus of the report.