Introduction to Computational Modeling in HCI: Assignment

Julien Gori (julien.gori@aalto.fi)

Fitts' law predicts the time MT required to select a target of size W that is located D away:

$$\mathrm{MT} = a + b\log_2(1 + D/W), \tag{1}$$

where $\mathrm{ID} = \log_2(1 + D/W)$ is the index of difficulty. We are going to use Fitts' law to model keyboard typing with a single finger. Throughout this work, it is assumed that the keyboard is a QWERTY one.

1 Instructions

1.1 Files and Software

You should have received the following files

- `assigmentSnippets.py`, which is the code you will need to complete for this assignment.
- `keystrokes.csv` which is a very small dataset of a person typing with a single finger
- `mostcommonwords.txt` which is the list of the 1000 most common words in English.
- `id.npy` which is the Numpy array that you should find in the first task. You can use it in your Python script using the `numpy.load()` function.

While you are given code to complete written in Python, you are free to use any programming language to perform the tasks. You can download Python at `https://www.python.org/downloads/`. You can also use an interactive Python notebook on Jupyter `https://jupyter.org/try` if you prefer things running online.

You will need a plotting library, I suggest using matplotlib `https://matplotlib.org/`, but you are again free to use anything else. You will be asked to compute linear regressions, for this you may need a statistical library. You can use the Scipy Stats library `https://docs.scipy.org/doc/scipy/reference/stats.html`. For those of you more familiar with statistical modeling syntax used e.g. in R, Statsmodels `https://www.statsmodels.org/stable/index.html` might be preferred.

For the first task, you will be asked to compute a matrix. If you don't succeed it will be given to you as a Numpy array. The tutorial at `https://numpy.org/` covers anything you should need regarding arrays.

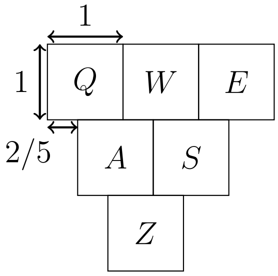

Figure 1: Extract of a QWERTY keyboard, and dimensions.

1.2 Logarithm

Throughout this assignment, you will use the logarithmic function log() several times. Here are some refreshers about the logarithmic function:

- The logarithmic function is the only function that verifies $\log(ab) = \log(a) + \log(b)$
- It follows that $\log(a^b) = b\log(a)$ ($a^b = a \times a \times \ldots \times a$ (b times))
- The logarithm is defined up to a base b: $\log_b(x) = \frac{log(x)}{log(b)}$. In this assigment, we use the base 2 logarithm: $\log_2(x) = \frac{\log(x)}{\log(2)}$. You can find it in Python in the Numpy `numpy.log2(x)` or math libraries `math.log(x,2)`.

2 Modeling the keyboard

To compute ID and use Fitts' law, we need to measure W and calculate D. We will consider that the keys are square with dimension $W = 1$ (see Fig. 1). Explain why this is justified, and no physical measuring is actually needed. (hint: study Equation (1)).

For two neighboring keys, say Q and W, we have $D = W = 1$, and ID$= \log_2(1 + D/W) = \log_2(2) = 1$. Compute the ID's for each possible pairs of keys. You will give the result as a matrix of ID's where the i$^{\text{th}}$ line and the j$^{\text{th}}$ column ID$_{i,j}$ is the ID to go from key i to j (for all 26 letters of the English alphabet — don't consider the SPACE key to make things simpler). Assume that each new line is shifted by 2/5th of a key to the right with regards to the one above it, see Fig. 1 for an example, using the QWERTY keyboard.

The first 25 entries of the table for the QWERTY keyboard should be:

$$\begin{array}{cc} & \begin{array}{ccccc} Q & W & E & R & T \end{array} \\ \begin{array}{c} Q \\ W \\ E \\ R \\ T \end{array} & \begin{bmatrix} 0.00 & 1.00 & 1.58 & 2.00 & 2.32 \\ 1.00 & 0.00 & 1.00 & 1.58 & 2.00 \\ 1.58 & 1.00 & 0.00 & 1.00 & 1.58 \\ 2.00 & 1.58 & 1.00 & 0.00 & 1.00 \\ 2.32 & 2.00 & 1.58 & 1.00 & 0.00 \end{bmatrix} \end{array} \tag{2}$$

Give the full table for the QWERTY keyboard.

3 Estimating Model parameters

I typed with a single finger on my keyboard the following sentence *thebrownfoxjumpsoverthegraydog* and recorded it together with corresponding timestamps in the file `keystrokes.csv`. I didn't type in the spaces, since we chose to not model it to keep things simpler.

Estimate Fitts' law parameter a and b of Equation 1 given the data in `keystrokes.csv`. You will need the table of ID's computed just before. If you didn't succeed in creating it, you can load it from `id.npy` as explained in Section 1.

Provide a plot, a measure of goodness of fit of your choice (e.g. r^2) and a way to assess the confidence in the parameter values (e.g. standard deviation of parameters, confidence intervals). You can fill in the code provided in `assigmentsSnippets.py`.

4 Most common words

We are going to estimate the average time it takes to type a word with a single finger on a keyboard. To do so, we use the 1000 most commonly used words in English, which you can find in the accompanying file `mostcommonwords.txt`. Assume that the time to go from one key to the other is given by Fitts' law Equation (1) with:

$$a = 0.03 \text{ seconds} \tag{3}$$

$$b = 0.12 \text{ seconds.} \tag{4}$$

Also consider that the first key is free, i.e. time is only counted after the first letter of the word has been pressed (one letter words take 0 seconds to type, two letter words take the time required to go from the first key to the second etc.)

Compute the time it takes to type the "average" word of the list with a single finger on this keyboard (use the table of ID computed before). Also plot the typing time for each word, as well as a histogram of the typing times.

5 Modeling Word Frequency

Some words are more frequent than others, which might change our evaluation. Zipf, a quantitative linguist, observed that the frequency of a word is inversely proportional to its rank in the frequency table. The most frequent word appears twice as much as the second most frequent word, three times as much as the third most frequent word etc. Let us call f_1 the frequency of the first item. What is the frequency of the second item f_2 with respect to f_1 ? what about the third item, and the k^{th} ?

We are going to convert these frequencies to probabilities. Show that the probability $p(k)$ of drawing the k^{th} most frequent word out of a corpus of N words is given by

$$p(k) = \frac{1/k}{\sum_{i=1}^{N} 1/i}. \tag{5}$$

Remember that probabilities are just frequencies that have been normalized so as to sum to 1. This law relating the k^{th} most frequent item in a list to $p(k)$ is called Zipf's law.

A more flexible version of Zipf's law adds a parameter s to tune the shape of Zipf's law:

$$p(k) = \frac{1/k^s}{\sum_{i=1}^{N} 1/i^s}. \tag{6}$$

Table 1 gives the frequencies for some k^{th} most frequent item occurrences in English, from a huge text corpus. Use this data to estimate the value of the parameter s. (You

Table 1: Word frequencies and ranks. From all the words in the this huge corpus, the most frequent one occurred 1e9 times. The tenth most frequent occurred 182e6 times. The one ranked 100 occurred 20e6 times.

Frequency	1e9	182e6	99e6	73e6	53e6	41e6	34e6	30e6	25.7e6	22.5e6	20e6
k	1	10	20	30	40	50	60	70	80	90	100

can either fit the model directly, or take advantage of the fact that $\log p(k)$ has a nice expression. Show the fitted model in a plot.

We can now account for the word frequencies. Consider $N = 1000$, and apply Zipf's law with the estimated s to compute the average word typing time. If you didn't succeed in estimating s use $s = 0.85$. Ignore the single letter words.

6 New keyboard layout

We are going to change the keyboard layouts, and see whether a simple change can increase single finger typing performance. The regular keys on the AZERTY keyboard are arranged in three lines, two lines with 10 characters and a third line with 6 characters. As a result, the keyboard is somewhat rectangular. However, a squared shape would be more efficient. A simple way to achieve this is to split the first two lines into extra lines. Instead of the layout

$$\begin{array}{r} qwertyuiop \\ asdfghjkl \\ zxcvbnm \end{array}$$

we would have the layout

$$\begin{array}{r} qwert \\ yuiop \\ qsdfg \\ hjkl \\ zxcvbnm \end{array}$$

How much time will a user using this new keyboard win if typing with a single finger ? Beyond the obvious fact that in real life, few people type with a single finger, why is changing the keyboard layout based on this analysis a bad idea ?

7 Grading

There are 5 tasks, the best achievable grade is 3. A 3 will be given to a student which completes at least 4 tasks correctly. A 2 will be given to a student which completes at least 3 tasks correctly. A 1 will be given to a student which completes at least 2 tasks correctly.