SINF115: Project –The Dribbble case

Peter VAN ROY Balint DAROCZY Xavier GILLARD

Context

In the world of today, social networks are ubiquitous: we all know the largest and most famous among them such as i.e. Facebook, Twitter or Tik Tok. Still, in spite of the dominant position of these platforms, there also exist some niche networks that are not meant to just connect friends with one another but attempt to fulfill a precise goal. This is for instance the case of GitHub, LinkedIn, ResearchGate, Thingiverse and many others.

In this assignment, you are going to work with snapshots of the *Dribbble*[1] social network. As you might have guessed, Dribbble is one such niche social network: it aims at connecting designers across the globe and let them share and advertise their creations. In that sense, Dribbble it resembles Deviant'Art which the most artistically inclined linuxian among you might already know and have used. But Dribbble targets a somewhat different audience: *professional designers* instead of free software amateurs. This is why the Dribbble platform has introduced features targeting that specific audience. Among those, it is worth mentioning the following features:

- A *Shot*, in Dribbble terminology, is a screen shot of some ongoing designer work. These shots may be liked and commented by other designers. behalf of which they may share shots, or react to those of others.
- The *Playbook*, is basically a designer's custom portfolio where he can share share and organize a gallery for his/her own work.

1 Format

The dataset contains two type of unique entities:

- User id("uid"): a unique integer identifier of the designer
- Shot id("sid"): a unique integer identifier of a shot.

Beside these there are two additional meta data:

- Timestamp: denotes the Linux "timestamp" of a specific interaction (desginer x follows designer y at time "ts", desginer x liked shot y at time "ts")

[1] `www.dribbble.com`

- Location: denotes the location of a designer. Note, these locations are self-declared thus very noisy.

The dataset contains three "csv" files:

- "designers.csv": row format ¡uid¿,¡location¿ means that designer "uid" self-declared "location".
- "shots.csv": row format ¡uid¿, ¡sid¿, ¡#likes¿ means that designer "uid" uploaded a design (shot) "sid" and it got ¡likes¿ number of likes
- "followers.csv": row format ¡uid_src¿,¡uid_dst¿,¡timestamp¿ means that designer "uid_src" followed designer "uid_dst" at time determined in Linux timestamp "timestamp".

2 In practice

For this project, you are allowed to use any language and library you like. For instance, if you feel comfortable with python, you might want to opt for the use of a library such as networkx[2] or networkit[3]. However, and regardless of the fact that these algorithms might have been readily implemented in these libraries, we request you to implement all algorithms seen during the class *by yourself*. You may of course use the available implementation, but for validation purpose only.

Also, this assignment must be completed by group of two students. If you do not manage to find a partner, please start by requesting partnership via the moodle forum. Then, if and only if you cannot find anyone else to work with, get in touch with us.

Unless specified otherwise, for each of the items below, we request you to explicit your quantitative results and then to explain how you implemented the required algorithm. Finally, you are requested to append your complete code as an appendix to your rapport.

2.1 Task 1: Required to pass

The bare minimum you need to do in order to have a chance at passing this project is to count:

1. The number of different components in the graph.
2. The number of bridges in the graph.
3. The number of local bridges in the graph.

The graph is identified in the "followers.csv". How can you interpret those number in the context of Dribbble ?

[2] `https://networkx.org/`
[3] `https://networkit.github.io/`

2.2 Task 2: If you aim at 14/20

List the top 20 most influential designers in the whole graph. Compare it to the total/average number of engagement (the sum/average of likes on shots made by the designer) of the designers.

2.3 Task 3: If you aim at 20/20

In this task, you will consider the members of the largest component only. Based on that sub-graph, we ask you to build the graph dynamically based on the timestamps.

1. Measure year after year how many triangles follow the triadic closure property and how much time past between one designer was connected with two others and all of them were connected (e.g. designer x is connected to designers y and z at time t_1 then after a while y and z will be connected at time t_2 thus the time difference is $t_2 - t_1$).

2. When the triadic closure property is fullfilled check if the location of the desginers are the same or not (if the location is "na", "none" etc. count it as unknown).

3. Visualize the distribution of the time difference and plot the histogram based on the number of months (e.g. there were 34 triangles which fullfilled the triadic closure property after 5 months as time difference). Similarly identify the top locations of the triangles and plot the histogram of the top 20 locations (e.g. New York has 356 triangles which follows triadic closure property).

4. Identify the average time difference of locations. (e.g. if all designers in a triangle are from New York (NY,NYC) then the average time is 6 months based on the 356 triangles in the city).

3 Deadlines

The assignment is due by Friday April, 30th 2021 at 23.59. It must be handed in on Moodle as a zip file containing both your report and your complete source code.