DATA301 Project Proposal - Twitter Analysis

Luka Foy

May 2023

1 Project Summary

The purpose of this project is to investigate the relationship between a Twitter user's profile features, and their influence within the app. In order to conduct this analysis, I will apply the PageRank and cosine similarity algorithms to the Stanford Twitter SNAP Dataset. This analysis should outline whether or not the most influential Twitter users share similar features. A conclusive outcome may be found useful by businesses or individuals looking to market themselves on Twitter, as it may give them some context to what type of content users are most engaged with.

2 Motivation

This research question is relevant to me as I have always been fascinated by the metrics which drive user engagement on social networks. To be able to work with real Twitter data and potentially draw meaningful conclusions is very motivating. Out of the algorithms we have covered, PageRank seemed like the biggest challenge for me personally to implement, hence why I have chosen it as my main algorithm, so I can gain a deeper understanding of parallel programming. The application of PageRank to a social media dataset seemed like a logical fusion of ideas, and was inspired partially by this Medium article. However, I didn't just want to apply PageRank, I was looking to extend my analysis of the dataset to draw a more meaningful conclusion than simply ranking Twitter accounts based on influence. Therefore, I am applying cosine similarity as well, to attempt to identify a relationship between the most influential Twitter accounts. The outcome of this project may be useful for both people driven by curiosity, or individuals / businesses who are looking to develop an influential Twitter account. If there is a relationship between page features and influence, this could guide the content they choose to post.

3 Background

My chosen dataset is the Stanford Twitter SNAP Dataset, which contains data of 81306 unique twitter users, contained in a .gz file. It includes 5 record files for each user (the user whom the record is centered on is called the "ego user" in the context of the dataset). The 5 records are as follows:

1. The circles file contains clusters of users connected with the ego user. I will ignore this for the purpose of this project, as the data it contains is not relevant to our analysis.
2. The edges file contains a set of graph edges (stored like tuples) for people who the ego user follows on twitter. These edges are directed, (a follows b), and we assume the ego user follows all users contained within this file.
3. The featnames file gives all the names and types of features which appear in the profiles of ALL users listed in the edges file. Features, in the context of Twitter, are hashtags and @ mentions of other users which are found on a profile.
4. The egofeat file is a vector of 0s and 1s which correspond to the ego user's profile features. 1 meaning that the user has this feature in their profile, and 0 meaning they don't.
5. The feat file is a set of vectors for each of the users which appear in the edges file. This data is the same as that in the egofeat file, in that 1s mean the user has the feature in their profile, and 0s mean they don't.

I am using both the PageRank and cosine similarity algorithms. PageRank determines the relative "importance" of an item within a network of other items. To do this, we take a graph as an input, and initialise all the nodes in the graph to a PageRank score of 1. Then, we create a teleportation matrix, which maps the probability of a person surfing the web clicking on a link (this probability is (1 - damping factor), which we will set to 0.15). This prevents both the spider trap and dead end problems, which will be explained in more detail in the Design & Methods section. Next, we divide the PageRank of a user by the number of external links it has (in this case, the number of other users the user follows), and these fractional values are distributed to those users who are followed by the original user. We then sum the matrix values and the contributions from other nodes to generate the user's PageRank. This process is repeated until the scores converge, at which point our PageRank is complete. Cosine similarity measures the likeness of two non-zero vectors. It's computed by taking the dot product of two vectors, then dividing by the product of their lengths. The result yields a value between -1 and +1, +1 meaning that the vectors are identical, 0 meaning that they are orthogonal, and -1 meaning they are the complete opposite.

4 Research Question

My chosen research question is "What is the relationship between a highly influential Twitter user's profile features, and their level of influence?". The precise nature of the question depends on my PageRank implementation, I may choose to adapt my question to a specific number of high influence users, in order to reduce computation times. I discuss this in more detail in my Design & Methods section.

The dataset is comprised of various data Twitter users' profile features and connections to each other, hence the question's relevance. The PageRank algorithm will help us to determine which users are highly influential - and cosine similarity will display similarities between these users' profile, thus determining what relationships exist between influential Twitter users.

5 Design & Methods

To answer my project question, I will use both the PageRank and cosine similarity algorithms. The edges records will be used as the input for my implementation of PageRank, which will use follows (a follows b) like links (website a links to website b). Given the size of our dataset, I'm anticipating we'll need to account for both the Spider trap and Dead end problems. In the context of the walk interpretation of PageRank, the spider trap problem occurs when two pages only link to each other (in our context, when two users only follow each other), causing the walker to get stuck in a loop. The dead end problem is pretty self explanatory, it occurs when a user does not follow anyone - so the algorithm has nowhere to go. As such, the algorithm will be implemented using a transportation matrix, with a damping factor of 0.85. Rather than guessing a number of iterations for PageRank to run through, I will set a convergence threshold of 0.0001 for the scores (meaning the algorithm stops once scores do not change by more than 0.0001) to prevent unnecessary running time. The results of PageRank will help answer the section of the question regarding a user's influence. Once the rankings are processed, depending on the effectiveness of my PageRank implementation, I will choose a number of high influence users' data as input to the cosine similarity algorithm. Then, the feat / egofeat records of the influential users will function as the input to the cosine similarity algorithm to determine whether a relationship exists between highly influential users' profile features.

Planned Project Timeline:

Week 8: Complete and submit proposal, create colab doc, import data and start on PageRank implementation.

Week 9: Continue PageRank implementation, bullet points for progress report.

Week 10: Finish PageRank (number of iterations), implement cosine similarity (anticipating this should be quite quick, as have working implementation already), finish and submit progress report.

Week 11: Running tests in cloud, code tweaking, drawing conclusions, draft bullet points for final report.

Week 12: Final code tweaks if necessary, write up and submit final report.

There are a couple of potentially limiting factors and technical difficulties that may impact this project. Firstly, given that assignments for other courses are typically posted with little notice, I don't have a full idea of how much time I can dedicate to this project. Hence, I might be under time pressure to complete it - which might impact the quality of my code/analysis. Secondly, as I

haven't implemented a PageRank algorithm before, there may be a steep learning curve. As such, I may need to adapt my project question on the fly. Finally, there may be inconclusive/uninteresting results from the dataset, leading to a difficult write up.

6 References

- Example PageRank implementation
- Similar project concept, helpful for gaining context
- Explanation of PageRank
- Inspiration for applying cosine similarity to social media data
- Example cosine similarity implementation
- Help for structuring research question