Project Submisson Document

Melody Horn, Fen Michael

December 2021

1 Introduction

Generally, when discussing influence on social networks, the primary metric of connection is follower/following. While this is sufficient for some issues, it is also important to analyze those who people interact with regularly. Professionals who utilize Twitter may follow certain people out of professional courtesy. Journalists may follow those they tend to write about, so as to notice more easily if they tweet something newsworthy. As such, we seek to determine if specific interactions, in the form of retweets and replies, are more useful in categorizing users than the general interaction of following. This would allow us to make predictions about how information propagates, analyze the differences between following and interaction clusters, etc. We originally hypothesized that clusters defined by following are noisy and abstract while clusters defined by interactions are more specifically clarified.

2 Data

For this project, we collected data using the Twitter API. We used Chelsea Manning (@xychelsea) as the starting user and collected verified users from following and interaction data. Our program was designed to gather up to 10,000 users, but for our experiment, we collected 2,069 verified users. Once the user information was collected, we created three graphs (based on following lists, retweets, and replies) using the 'python-louvain' package implementation of the Louvain method of community detection. We then found the modularity of each graph and compared them.

3 Algorithm

$$M = \frac{1}{2m}\sum_{i,j}(A_{ij} - p_{ij})\delta(c_i, c_j) = \frac{1}{2m}\sum_{i,j}(A_{ij} - \frac{k_i k_j}{2m})\delta(c_i, c_j) \qquad (1)$$

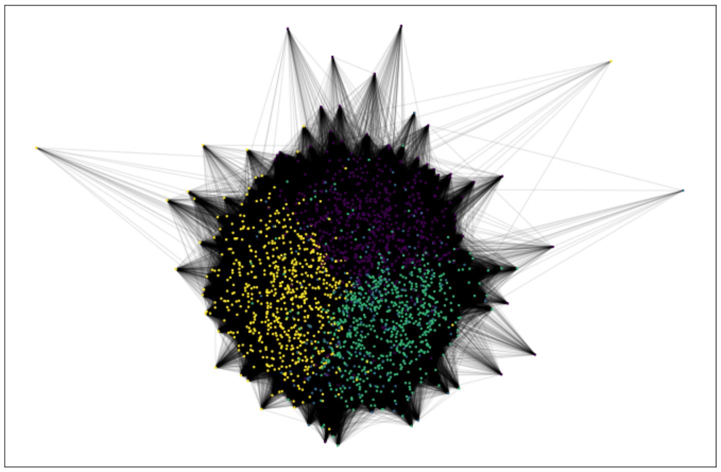
Figure 1: Louvain method equation

For this project, we utilized the Louvain method for community detection. This algorithm optimizes modularity using the equation from Figure 1. Modularity is a measure of the strength of the division between a graph's modules. In our case, the modules are the network communities. This is accomplished using the adjacency matrix A_{ij}, nodes i and j, communities c_i and c_j, and the delta function.

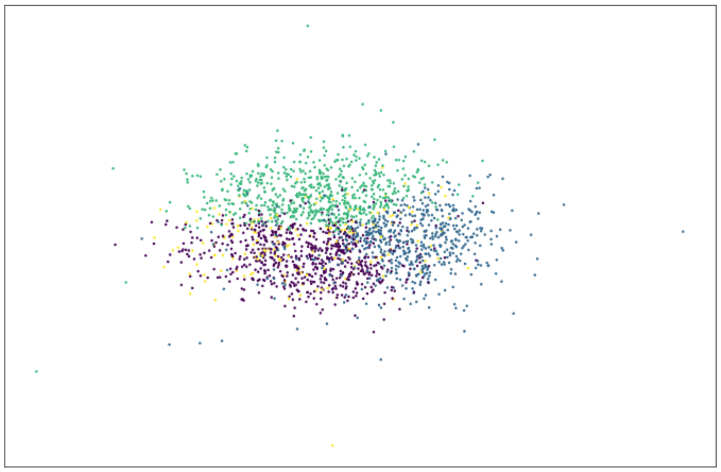
4 Experimental Results

Our data collection code ran for 4 days and gathered interaction data from 2069 users. This was accomplished by gathering data from user ids. We chose this because more data can be collected in one call using user ids (rather than user screen names).

We initially hypothesized that retweets would have the highest modularity and following would have the lowest. After analysis, we learned that replies [Figure 4a, b] had the highest modularity with a value of 0.8037, indicating the strongest division between node communities. Following [Figure 2a, b] had the lowest modularity of 0.1058. Retweets [Figure 3a, b] had a moderate modularity of 0.4206. Our hypothesis was incorrect, but we were correct in assuming following data would have the lowest modularity of the three sets.

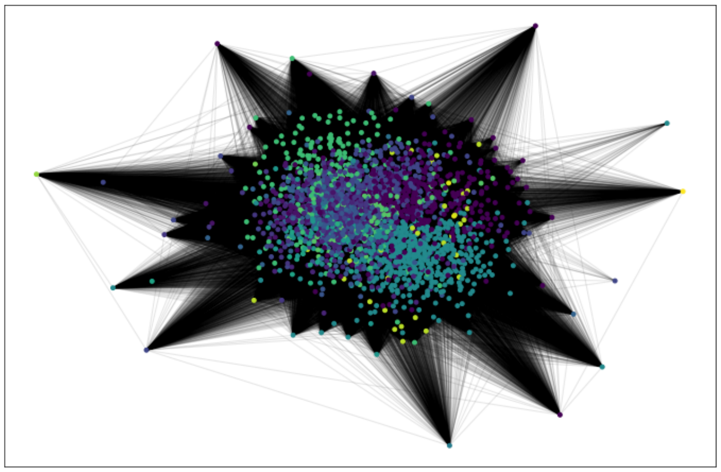


(a) With edges.



(b) Without edges.

Figure 2: Graph from following data.


(a) With edges.

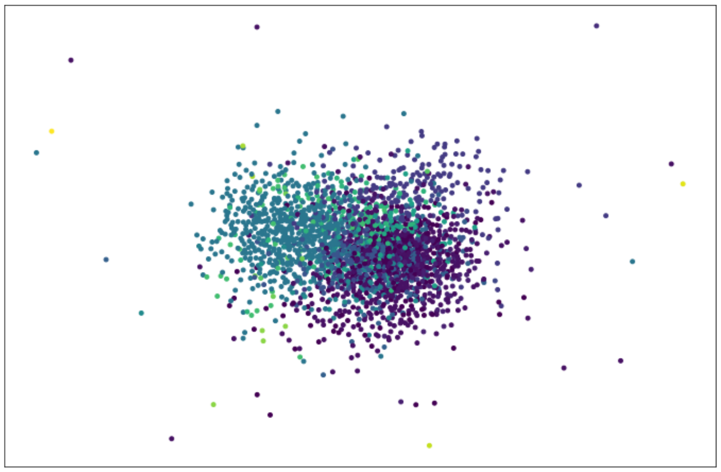
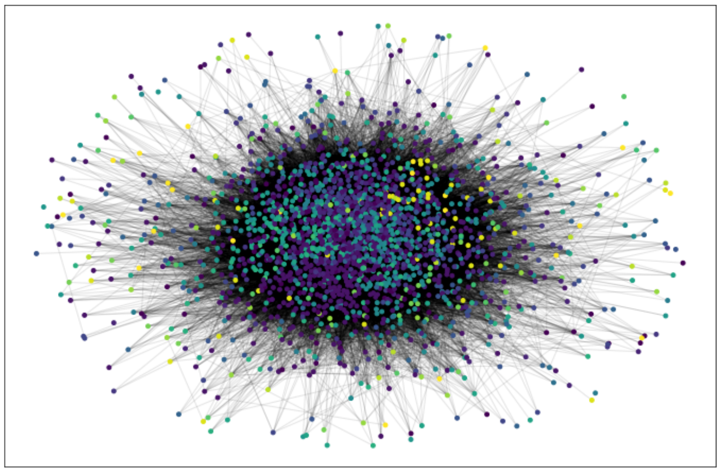
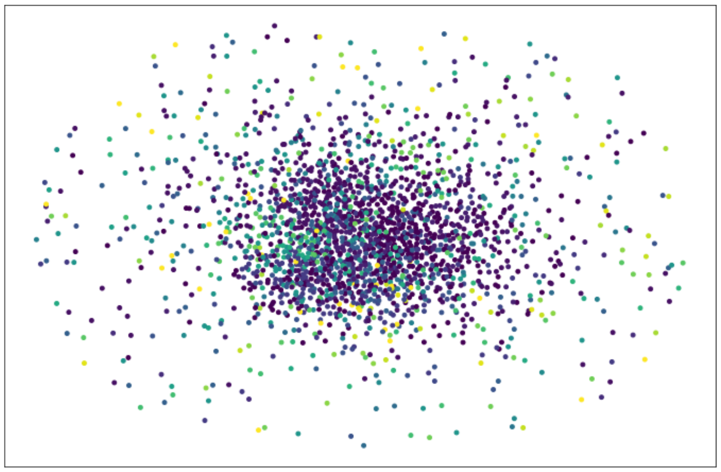

(b) Without edges.

Figure 3: Graph from retweet data.



(a) With edges.



(b) Without edges.

Figure 4: Graph from reply data.

5 Discussion

Through our experiment, we learned that reply data shows the highest modularity of the information we collected. Additionally, the modularity of the following data was unexpectedly low. This shows that users tend to reply to the same set of people, but follow users nearly at random.

If we were to perform this experiment again, we would analyze the modularity of other groups and determine if the differences in modularity are consistent across all Twitter communities. For examples, we could expand our current code to allow for both verified and non-verified users and examine changes.

6 Conclusion

During the project, our team divided the work to ensure a smooth process. Fen Michael wrote the inital drafts of the code and the project paper. Melody refined the code and set up the infrastructure to run the data gathering continuously.