CSC 407 Term Project Report

Michael McCulloch & Ronald S King

Fall 2019

# I. Introduction

This project analyzes dynamic Face-to-Face Interaction Network data made available by [SNAP](https://snap.stanford.edu/data/comm-f2f-Resistance.html). The dataset represents 62 games, both unweighted and weighted, with various groups of individuals playing the [Resistance game](https://en.wikipedia.org/wiki/The_Resistance_(game)). Each game consists of 5-8 players, lasts between 45-60 minutes, and is provided in two different versions. Unweighted versions of games represent who-looks-at-whom networks in a binary manner, a player may only look at a single player at time, while weighted versions contain the probabilities associated with where players are most likely to be looking in the immediate future. The data for both game types is arranged in .csv files in a network folder with a network list file containing the number of players for each game at a one higher directory level.

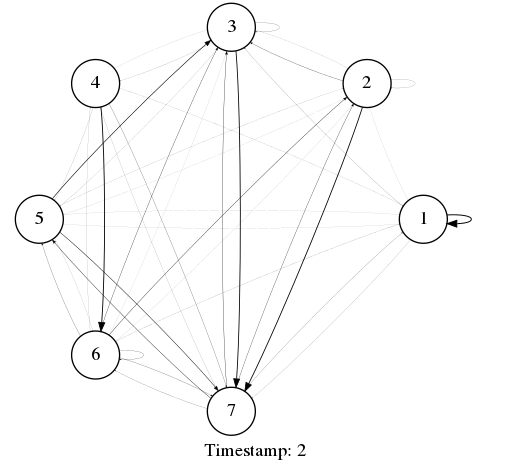
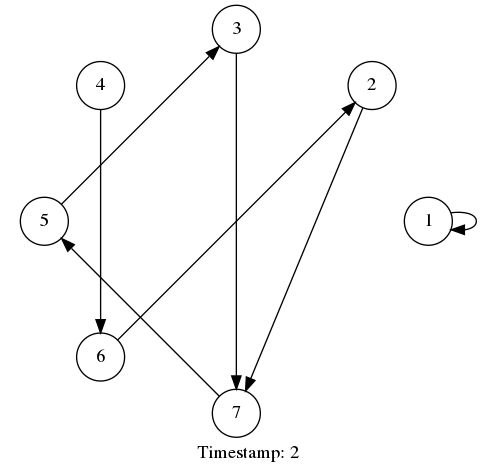
II. Designs and Methods

Visualizing Face-to-Face Interactions

To visualize the interactions among players we created a command-line tool, which used Python, Graphviz, PyGraphviz, and OpenCV. This tool allowed us to view the interactions between nodes as the games progress, analyze games, and optionally save video. The tool’s options and examples of visualized network data can be seen below.

# 

The image above shows the command line options for our tool. Below are two examples of visualize output. The image on the left if from the second time from the unweighted game file network0.csv and the image on the right is from the weighted version at the same time. The probabilities of node edges are represented by varying widths of the edges and the arrows associated with them.



# Predicting Face-to-Face Interactions

# Originally we explored if there was a connection between bi-directional communication between nodes and an increase in either of their in-degree centrality in the near-term; but there didn’t seem to be connection from the games analyzed and it was time consuming processing the dataset—so we moved-on to try to predict the game’s overall interactions from a samples of games. We explored whether it was possible to predict which nodes would have the highest and lowest in-degree centrality based on various sample sizes. Simulating the Spread of a Rumor

A method for simulating the potential spread of a rumor was developed using the network interactions as a basis for opportunistic spreading of the rumor. Initially, a player was assumed to have knowledge of a rumor. The player number and the probability value (p) were passed into the program as a parameter. Even time that this player interacted with another, the program assumed that this was an opportunity to spread the rumor. If the p value was greater than a randomly generated number, then the rumor was spread to the new player. This players’ future interactions with other players also became opportunities to spread the rumor further. At the conclusion of the game, the number players to which the rumor had spread along with the initial p value provided were recorded in an external .xlsx data file. Multiple iterations of the game were run for each p value to obtain a reliable average. Just as with the face-to-face interaction network modeling, a weighted version of the rumor spreading was developed. With this version, the p value was compared proportionately to weighted values in the data at each time segment instead of the p value as a whole.

# III. Results and Discussions

# Predicting Face-to-Face Interactions

Using a sample size of half a game, the highest ranked node in the samples—where nodes are ranked from highest to lowest average in-degree centrality—were in the top-two for the game 93% of the time across all games. The lowest ranked node in the samples was in one of the two lowest ranked nodes between 98-100% of the time.

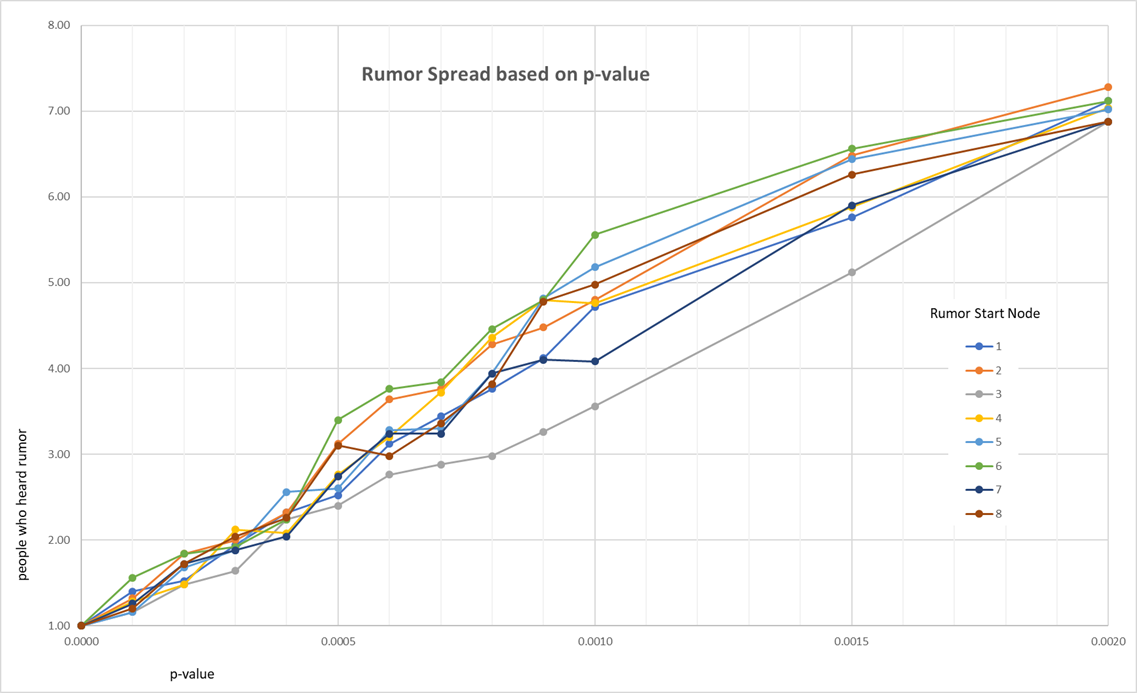
For a sample size of quarter of each game, the highest ranked node in the sample was one of the two highest ranked overall between 94-97% of the time­ and the lowest ranked node in the sample was in one of the two lowest overall 95-97% of the time.

For a sample size of a tenth of a game the results from the highest ranked in the sample matched one of the two highest overall 89-94% and one of the two lowest 81-94% of the time. Being able to predict one of the two highest and lowest ranked nodes from a tenth of a game is interesting; but it made us wonder how frequently the highest and lowest ranked nodes in a sample were also the highest and lowest overall. It turns out with only a tenth of a game the highest ranked node in the sample was only the overall highest ranked 77% of the time and the lowest 68% of the time. For the sample and overall ranks to match for the highest ranked node, the sample size needed to be half of a game. Unfortunately, even with such a sample size the lowest ranked nodes matched only 68% of the time.

The ability to predict interactions among networks nodes with a small fraction of data is useful and interesting. It is often the case that decisions need to be made without all the data—either because acquiring the data is prohibited by cost, not available, or doesn’t exist yet. Using the advertising industry and social networks as an example, it is infeasible to process all the data to determine who might be the most or least important, influential, etc. But from samples of network data advertisers can make better decisions regarding who to partner with and who to avoid.

Simulating the Spread of a Rumor

Our methodology assumed that each interaction between two players was an opportunity to propagate the rumor further. This assumption, along with the large number of time segments in each game (most games have more than 6,000 time segments), resulted in a very high/quick spreading of the rumor. As a result, a comparatively low p value was required. The graph below depicts the average propagation count for various p values. The results are based on average for 50 iterations of each p value and each initial player starting the rumor.



When p value was 0, obviously no rumor propagation occurred, but when p was increased to 0.06, typically full propagation of rumor occurred, regardless of which player initially started the rumor. Values between 0.0 and 0.6 generated rumor propagations in near direct proportion to the p value.

An interesting observation was that the propagation rate was generally slower when the rumor started with player 3. This would imply that player 3 had less interaction with other players (more interaction with the laptop) and therefore, had less opportunity to spread the rumor. Similarly, the propagation rate was generally faster when the rumor started with player 6, implying greater opportunity for propagation through more player interaction.

Using the weighted version of the game file as a basis for the network interactions and rumor spreading potential yielded similar results. The weighted game data is peppered with multiple time segments that contain a very high probability of one player interacting with another player. This disproportionate value frequently yielded propagation of the rumor to another player in most occurrences. Therefore, a very small p value was required to get interesting results.

# IV. Conclusions

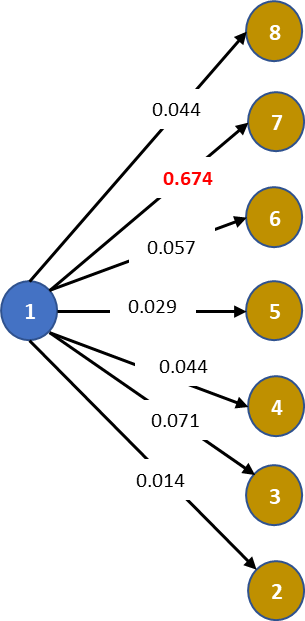
# Predicting Face-to-Face Interactions

Individual interactions are complex and often the interaction is driven by external forces not captured in this study data. However, based solely on the data presented, it appears that a sample of the data could be used to predict the overall interaction. Similar to the Pareto principle, a large number of the interactions appear to be governed by a small subset of interactions. Based on these findings, it appears that the network data could be used for accurate predictions.

Simulating the Spread of a Rumor

While the games may be suitable as a simulation of interaction network, the methodology used for discerning rumor spread offered to much opportunity. In our base methodology, a rumor could be spread during a single 1/3 second interaction. A better approach would have been for us to only use intervals where there was sustained interaction with the same player as an opportunity for rumor propagation. For example, anytime player A interacted with player B for more than X time segments, this presents an opportunity for spreading of the rumor. If this methodology had been applied and only large blocks of similar time segments were considered, then larger p values would have been required to fully propagate rumor to all players.

With the weighted network data, a similar problem also existed. The time segments of high interaction probability facilitated almost certain spread of the rumor between players. For example, the graphic below shows one time segment were probability is heavily weighted to a specific player.



With such a high weighted time interval, propagation is likely. One suggested variant would be to require multiple successful interactions between players before the rumor could be considered spread. These interactions could come from multiple other players rather than the same player and the accumulative effect of repeated exposure to the rumor would be a satisfactory condition for rumor propagation. This approach may not yield consistent results, but will is more likely to represent the rumor spreading in a practical environment.

V. References

No external sources or references were used in the development of the program solution or interaction methodologies. However, in addition to the python 3.8.0 documentation website (<https://docs.python.org/3/>), various sources were used for general tutorial as well as specific syntactical code inquiries. Most notable were [https://www.w3schools.com/python/default.asp](https://www.w3schools.com/python/default.asp%20) and the book “Python Crash Course” by Eric Matthes. One question was posed to the stack overflow website: <https://stackoverflow.com/questions/35372700/whats-0xff-for-in-cv2-waitkey1>.