Weight Loss Prediction Using Social Network Analysis

Seung Min Baek[1], Danielle Lesinski[2], Nicholas Grimaldi[3]

Vanderbilt University[1,2,3]
seung.min.baek@vanderbilt.edu[1], danielle.s.lesinski@vanderbilt.edu[2], nicholas.r.grimaldi@vanderbilt.edu[3]

ABSTRACT

In recent years, there have been many social networking applications aimed at helping users lose weight and providing outlets to interact with others also trying to manage their weight. The social networking aspect of these applications allows users to connect with other users' profiles. As the users track their individual progress towards their weight loss goals, they can share their achievements and view the progress of users they are connected with on the platform. Our group conducted research into the possibility of the social aspects of weight loss applications creating contagion in weight loss. While numerous studies have shown how obesity and weight gain can spread from social connections, we wanted to explore how a weight loss application could have the opposite effect. We used data from the BOOHEE weight management operator and compared user interactions through the *NetworkX* library in Python. We used feature extraction to merge social activities between users to ultimately create graph networks for testing and evaluating our hypothesis. Applying graph convolutional network algorithms and training them with five different batches of dataset values allowed us to receive a maximum weight loss prediction of 71.8% accuracy. These findings allowed us to confidently show that weight loss applications like BOOHEE and the social interactions that take place on the platform can impact a user's weight loss.

KEYWORDS

Social Network Analysis, Weight Loss, BOOHEE, Graph Convolutional Network

I. INTRODUCTION

Obesity is an ever-growing issue facing our society today; from 2000 to 2020, the percentage of obese citizens in the United States has increased from 31% to over 40%. While this condition may be hereditary in some respects, there is still a great need for a solution to this dangerous condition. Some may argue that weight loss may even be contagious, as there is a tendency for people of a certain weight to be friends with others of the same weight. Traditional methods like dieting and exercising may fail some people due to a lack of consistency or other issues. As obesity becomes more prevalent, many studies have been conducted to gain a better understanding of how obesity cases continue to increase.

An article in *The New England Journal of Medicine* titled "The Spread of Obesity in a Large Social Network over 32 Years" evaluated a social network of over 12,000 people assessed for the Framingham Heart Study [1]. This social network allowed them to consider whether weight gain in an individual could be linked to similar weight gain in that individual's family and friends. Looking at the clusters of an obese individual's social ties provided the researchers with statistics related to an individual's chance of obesity. Some observations were that there is a 37% likelihood that a spouse would become obese, given the weight gain of the other spouse. Additionally, they found that "a person's chances of becoming obese increased by 57% (95% confidence interval [CI], 6 to 123) if he or she had a friend who became obese in a given interval" (Christakis and Fowler 2007). Due to the research by Christakis and Fowler having been conducted in 2007, it could not concern itself with the future possibility of the role that social networking apps play in the obesity epidemic.

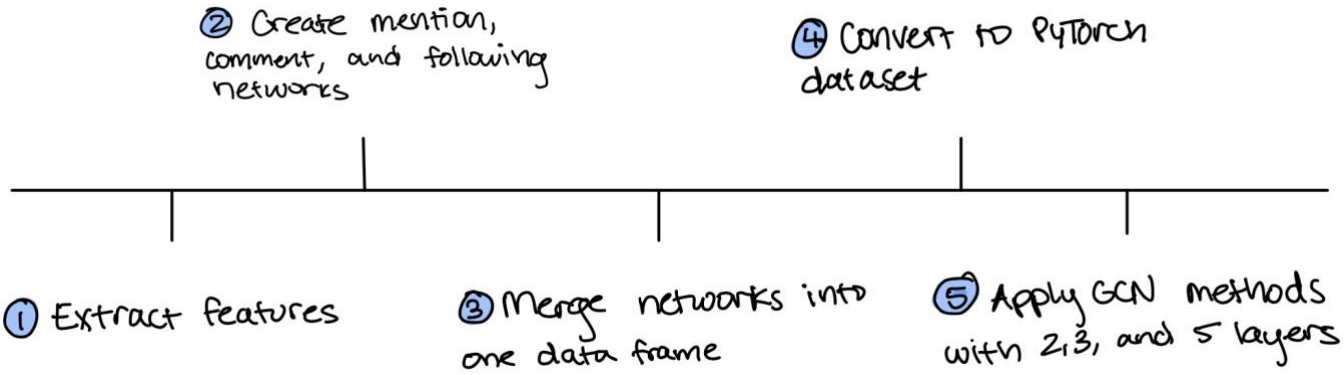
Countless applications have been created with the purpose of advocating and pushing for weight loss through dieting and/or exercise. Generally, these applications provide guides and routines for adopting and leading a healthier lifestyle. Not only that, often a social networking aspect is included to allow users ways to communicate and encourage others through a shared weight loss journey. Sometimes the social component is even advertised as a method for creating friendly competition through tracking activity and calorie goals of a user's social connections on the app.

Through this paper and project, we applied social network analysis to online weight loss application datasets so we could explore the feasibility of weight loss being contagious. This project was designed after the efforts done by Zhiwei Wang, Tyler Derr, Dawei Yin, and Jiland Tang

in their paper on "Understanding and Predicting Weight Loss with Mobile Social Networking Data [2]. We used the BOOHEE Weight Loss dataset provided by our Professor, Tyler Derr. BOOHEE is a social media application popular in China in which users can create a profile, track their weight loss progress, and interact with other users. Users can follow other users, post, comment on fellow users' posts, and mention other users. Users also register their height, weight, birthday, beginning and current weight, as well as target weight. This dataset includes data for nearly 10 million users, which will allow us to apply methods to a large population and draw strong conclusions, experimenting with data for a wide variety of users with different weight classes and social connections. We explored the communities formed within the datasets and their properties so we could predict weight loss using not only the user's information but also that of their social network neighborhood. Understanding these claims will help build better weight loss management applications and help more people lose weight, bettering the health of millions of people.

II. METHODS

Figure I: Process Map



A. Reading Data

We began by creating a Python script that could read our large datasets. This step is crucial since we need the data to run feature extraction and ultimately create networks for our tests. The .csv files were read using the *Pandas* library, which was imported into our Python script. The following BOOHEE dataset files were included in our project:

- *friend_consolidated.csv*
- *comment_rela.csv*
- *mention_rela.csv*
- *post_consolidated.csv*
- *users_consolidated.csv*

This initial step created some concerns due to the large nature of the dataset files. One of the largest files in the provided BOOHEE data, *friend_consolidated.csv*, was 3.44 GB in size. Due to the size of the file, the data was consolidated into a variable 'users_consolidated' and was then deleted using 'del' to free RAM memory. It was also crucial to fill in 'NaN' data values to prevent improperly classified nodes when applying GCN methods later. Finally, small housekeeping tasks were done to convert data types for time values and correct index labeling. Below is a sample of the information contained in the BOOHEE dataset.

Table I: BOOHEE Statistics [2]

Description	BOOHEE Data
# of users	9,967,290
# of following links	127,425,354
# of mentions	26,502,741
# of comments	25,960,524

B. Feature Extraction

After reading the data, we created our user feature matrix. Feature extraction is an important step since we need to discern the users from each other and, most importantly, their weight values. We decided to create several new features to be included in our feature matrix. We created 'diet_plan_duration,' determining how fast (in days) the user wants to reach their desired weight, and 'diet_plan_amount,' determining how much the user wants to lose. We created a label category called 'weight_loss_status' displaying whether or not the user has reached their target weight; if the user's current weight is less than their target weight, 'weight_loss_status' will be 1. Otherwise, 'weight_loss_status' will be 0. We extracted several more features as well, including the number of posts a user makes, the number of comments received, the number of comments a user makes, the number of followers a user has, the number of users a user follows, the number of received mentions, and the number of mentions made by the user. All of these features were then normalized with the

sklearn StandardScaler Python package, which removes the mean and scales each feature to standard variance. We then converted our categories to the necessary correct data types. From the altered dataset, we extracted our feature matrix containing user IDs, gender, age, height, diet plan duration, diet plan amount, BMI, number of posts, comments received, comments given, follower count, following count, mentions received, and mentions given. We also extracted a label matrix of user IDs and weight loss status. We then extracted the user IDs of commenting interactions, listing what user commented on what user's post and creating a dataset containing the commenting activity. We did the same for following and mentioning activities.

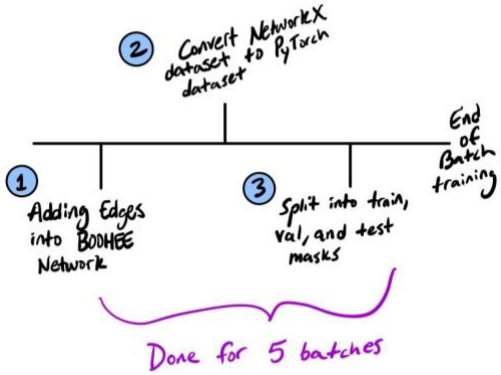
C. Creating and Merging Networks

We then merged all social activities (commenting, following, and mentioning) into one dataframe so that we can create our undirected graph network. Since we want our graph to be undirected, we dropped all duplicate rows.

D. Converting to PyTorch

This merged dataset was converted into a NetworkX graph using the nx.from_pandas_edgelist() method, which directly converts DataFrames to NetworkX graphs. However, since the original Boohee dataset contains an enormous amount of data (total number: 52,968,252), we decided to use the idea of batch processing. We split the dataset into 5 different datasets with a size of 200,000 for each and planned to utilize 5 different batches into training. See Figure II for a visualization of this process.

Figure II: Process map for batch processing



We then generated a dictionary of attributes from our features matrix, and a label dictionary from our label matrix. We then used these dictionaries to set node attributes and labels in our graph network. Finally, we converted our NetworkX networks to a PyTorch Geometric dataset using the method from_networkX(). Initially, we struggled with the formatting; our data is in a different format than the format used in our previous homework assignments, leading us into new territory. However, we were able to readjust and eventually succeed in converting the data. We then split this dataset into our training set, testing set, and validation set. We decided to train on 80% of the data, and reserve 10% of data for the training set and 10% of data for the validation set.

III. GCN METHODS AND EXPERIMENTS

Next, we applied the Graph Convolutional Network algorithm to our dataset. Since the number of layers in GCN implies the number of hops, we first predicted 2 or 3 layers of Convolutional Networks that would give the best result and proceeded with our experiments with another number of layers (2,3,7). The small-world network graph tells us most of the nodes can be reached by a small number of hops, and we believed 7 hops would capture insignificant relations with given nodes in this network.

Before training our GCN Models, we need to find hyperparameter values for more accurate models. We performed experiments on tuning learning rate and weight decay values to obtain the best performance hyperparameter values. We proceeded with experiments on learning rate with values 0.0001, 0.01, and 0.03 and weight decay with values 0.0001, 0.001, 0.01, and 0.1. Interestingly, we discovered the fact that insufficient learning rate size disturbs models from converging, and large weight decay value returns the worst prediction accuracy. Figures III, IV, and V display the accuracies for modifying the learning rate and figures VI, VII, and VIII display the accuracies for modifying the weight decay. We proceeded to choose the parameters that return the most accurate values, learning rate 0.03 and weight decay 0.0001.

Figure III: Accuracy for learning rate of 1e-03

```
Learning rate 1e-05: 0.45504508372692143
```

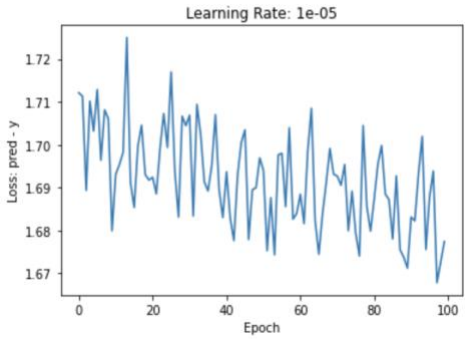


Figure IV: Accuracy for learning rate of 0.01

```
Learning rate 0.01: 0.6713324745956777
```

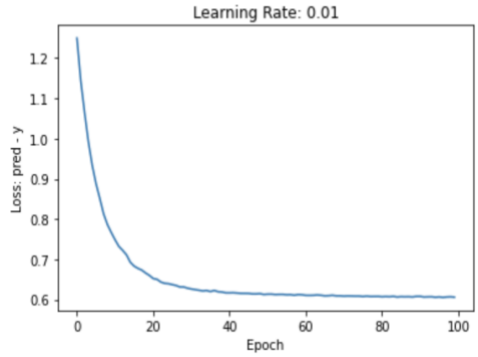
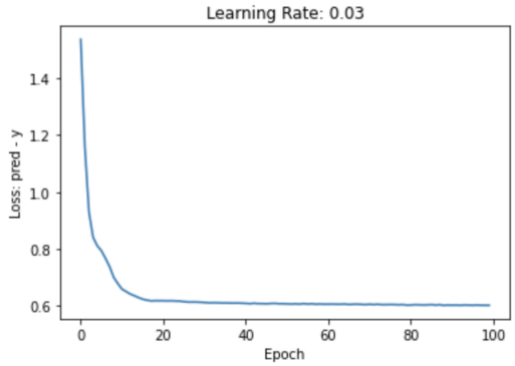


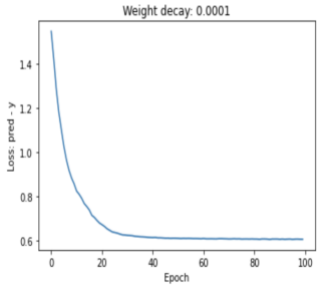
Figure V: Accuracy for learning rate of 0.03

```
Learning rate 0.03: 0.6733934449692286
```

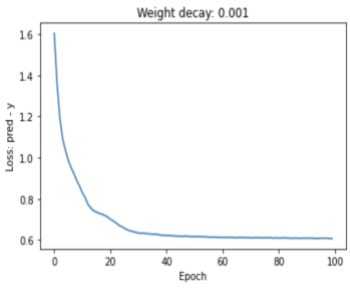


Figures VI, VII, VIII, and IX

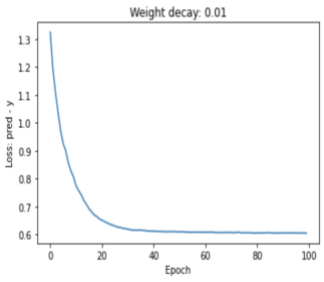
```
Weight decay 0.0001: 0.6717552887364208
```



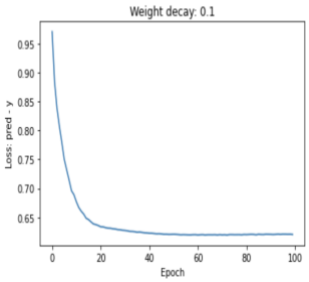

```
Weight decay 0.001: 0.6700400228702116
```




```
Weight decay 0.01: 0.6699256718124642
```




```
Weight decay 0.1: 0.6650371640937679
```



We then trained our GCN Models (2,3,7) to maximize our prediction with 5 different batches of the dataset and 200 epochs for training for each batch. For these models, we used the aforementioned learning rate and weight decay parameters of 0.03 and 0.0001, respectively. As we expected, GCN with 3 layers predicted weight loss status most accurately with 71.144% while GCN with 2 layers prediction accuracy was 70.812%, and the GCN with 7 layers prediction accuracy was 70.585%, as shown in the graph below. We had to confirm our GCN model by utilizing network attributes with well-predicted weight loss status, and we decided to compare it with the classic classification machine learning algorithm, Logistic Regression. We used *sklearn* Logistic Regression with only users' features, and it predicted weight loss status with 70.88%, which is less than our 3 different layers of GCN. It proves our GCN models' prediction accuracy outperforms machine learning algorithms which do not take advantage of network systems. Figure X displays a GCN model with 2 layers, Figure XI displays a GCN model with 3 layers, and Figure XII displays a GCN model with 7 layers.

Figure X

```
2 layer Accuracy: 0.7081260573138496
```

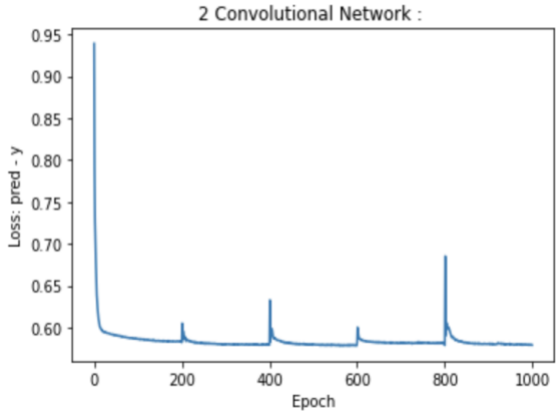


Figure XI

```
3 layer Accuracy: 0.7114409785664907
```

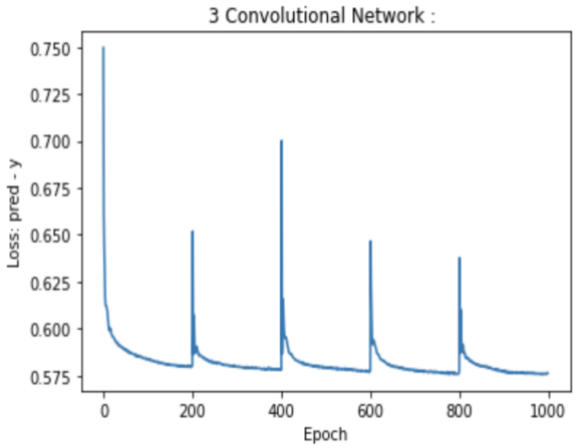
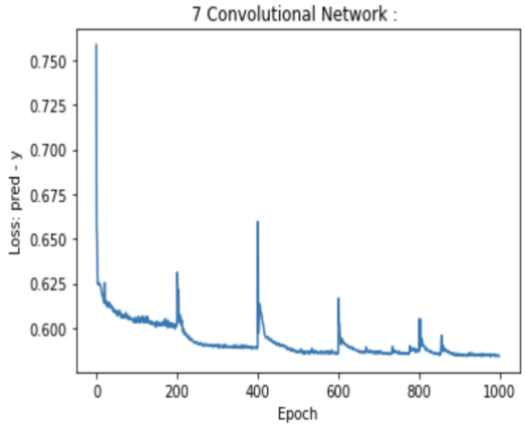


Figure XII

```
7 layer Accuracy: 0.7058515890400608
```



To increase the accuracy of these results, we proceeded to use more data points through batch processing. Through a batch process, we could separate the data into five randomly chosen datasets. This increased the total number of data points from 200,000 to 1,000,000. The process went as follows:

1. Adding edges into BOOHEE network
2. Convert NetworkX dataset to PyTorch dataset
3. Split into train, validation, and test masks
4. End of batch training. Train the next batch if applicable

This process was repeated for each of the five batches of 200,000 data points. We found that by adopting this method, the accuracy of our experiments increased to 71.14%. While the accuracy increase is helpful, it is not surprising due to the fact that we utilized more points in our training.

IV. CHALLENGES

A. Reading and extracting from large files

When loading our data in, we initially used code similar to code used on previous projects. However, those projects used much smaller datasets; we quickly realized that it needed to be modified in order to handle the massive amount of information in some of the datasets. While this code worked well with about 28 KB of information, the "friend_consolidated.csv" file in our weight loss datasets contains about 3.45 GB of information. Running the previous code without any changes, we saw that the code was running for roughly 20 minutes without creating a graph for one of the given datasets. A quick inspection into these files revealed that over 10 millions lines needed to be iterated through to create edges for our graph. With this massive change in size, we needed to incorporate a faster way than the iterate method to create edges. Quick research on this issue helped show that it would be exponentially faster to turn the read dataset into a dictionary. With this new variable, we now use a for loop to go through the rows in the dictionary and add an edge with the needed information. Putting this to the test helped reduce the process of creating a graph from the "weight_record_consolidated" file down to thirty seconds. Despite this change, larger files like "friend_consolidated.csv" still took several minutes to create a graph from.

To make the data processing even faster, the data was consolidated into a variable 'users_consolidated' and was then deleted using 'del' to free RAM memory. We then filled in 'NaN' data values to prevent improperly classified nodes when applying GCN methods later. Lastly, the data was cleaned up to ensure correct data types for time values were used, and indexes were labeled correctly.

B. Converting Pandas Dataframe to PyTorch

Our merged dataset consisted of a Pandas DataFrame, which is similar to a table with rows and columns. We knew that we wanted to get our data into a form that can be usable by PyTorch, so that we can run the GCN algorithm and perform our experiments. However, we were initially stuck on how to do this. We eventually figured out that we could covert the DataFrame into a NetworkX graph using the from_pandas_edgelist() function provided by NetworkX. From the NetworkX graph, it would be easier to covert into a torch.geometric.data instance using the PyTorch from_networkX() function, which creates a torch.geometric.data instance from a NetworkX graph. Initially, our plan did not work as intended; after research and trial and error, we resolved the problem by explicitly setting node attributes in the NetworkX graph before attempting the PyTorch conversion. Although the error took a significant amount of time to resolve, this often happens when handling raw data. The setback did not throw off our

timeline and we were able to continue on to our experiments without issue.

C. Merging feature matrix and network index

Our next challenge came when we attempted to merge the feature matrix and the network index. This step created some difficulties due to all of our information needing to be in one dataframe when creating the network index. After completing our feature extraction, the matrix with all of our features had the information stored across numerous dataframes. Using the 'concat' method in the Pandas library, we were able to easily concatenate the information of mentions, comments, and follows. From here, we dropped all of the duplicate information and shaped the variable to store all of the information inside one dataframe. We then utilized NetworkX's ability to set node attributes with the 'set_node_attributes' function provided in the library. Now, all of our features were successfully merged into our network.

D. Applying GCN to dataset

When initially applying graph convolutional networks to our dataset, our nodes were being classified to 0, and the model was filled with 'NaN' values. This was a problematic discovery because it barred us from testing and evaluating the data to create weight loss predictions. However, after searching through the code, it appeared that the problem came from how the feature matrix was created. When extracting the features from reading the given dataset files, there were some users who had 'NaN' values. These values were used in sections like posts, followers, following, comments, comments received, or mentions. A 'NaN' value in one of these sections meant that the user was not engaged in the following category. This was handled and interpreted by our code as being equivalent to a numeric value of 0 since the user had nothing to contribute to these categories. With this issue out of the way, our network was complete and ready for tests.

V. RESULTS

Through our methods which ultimately involved extracting users' features and creating a network on which we could apply GCN methods on, we received a maximum weight loss prediction accuracy of 71.8%. Logistic regression predicted with a lower accuracy of 70.88%; therefore, incorporating social activity into our prediction model using GCN is proven to be a better method of prediction, meaning that social activity does play a role in weight loss prediction. Using the edge index from the dataset, we can predict whether or not a user will lose weight. While these findings may not prove our hypothesis of weight loss being contagious through social weight loss applications, it provides enough information to hint at a correlation between a user's social activity on BOOHEE and their possible subsequent weight loss. Our findings have allowed us to say that by examining a user's follows, followings, comments, and mentions, we can accurately predict their weight loss. This means that these features make a direct connection to how accurately we can predict weight loss and therefore make a connection to a user's weight loss.

There are more factors and questions to consider moving forward in this field of research. Interaction on social media platforms like BOOHEE is more intricate than statistical numbers relating to the amount of followers, following, or the number of posts, comments, and mentions. With these platforms being designed to allow users to send motivational posts and comments, future work could consider how these factors play a role in a user's weight loss.

VI. CONCLUSIONS

In conclusion, this research paper attempts to investigate the often theorized observation that social patterns influence weight distribution using social network analysis. First, we find a dataset from a popular social media app, BOOHEE, containing user data like age, weight, and gender, along with social activity like commenting, following, posting, and mentioning. We then extracted user data along with user social activity and created mentioning, following, and commenting networks. These networks were then merged into one DataFrame. From the Pandas DataFrame, we converted our data into a NetworkX graph and then into a PyTorch dataset. From this dataset, we split it into training, testing, and validation sets. We then applied the GCN algorithm, experimenting with 2, 3, and 5 layers; these were trained with 5 different batches and 100 epochs of training for each batch. As expected, the GCN model with 3 layers performed the best, predicting with an accuracy of 71.67%. The GCN model outperformed Logistic Regression models, demonstrating the power of using network analysis in prediction models.

Our findings indicate that a correlation between social activity and weight loss does exist and that social activity can predict weight loss with an accuracy of 71.67%. While we may not be able to state whether weight loss is "contagious," as we are unable to differentiate whether users lose weight due to interaction with others or simply choose to interact more with those of a similar weight class, our results still indicate that a correlation exists. In the future, we would like to investigate whether social activity has the same predictive power on weight gain; this study may help us understand more about the idea of weight "contagion." We would also like to study how much weight users lost, along with what factors were associated with higher weight loss. It would also be beneficial to help users lose weight

by providing weight loss recommendations based on our correlation findings.

References

[1] Nicholas A. Christakis, M.D., Ph.D., M.P.H., and James H. Fowler, Ph.D. 2007. *The Spread of Obesity in a Large Social Network over 32 Years*. The New England Journal of Medicine. DOI: 10.1056/NEJMsa066082

[2] Zhiwei Wang, Tyler Derr, Dawei Yin, and Jiliang Tang. 2017. Understanding and Predicting Weight Loss with Mobile Social Networking Data. In Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17), 10 pages. DOI: 10.1145/nnnnnnn.nnnnnnn