Analysis of Large Scale Social Networks Analytics Project

Friend Recommendation Algorithm
for Social Networks

Adnan Kaan Ekiz
Andrés Reverón Molina

adnankaan.ekiz@student.kuleuven.be
andres.reveronmolina@student.kuleuven.be

May 2020

Contents

1 Introduction

Social Network Analysis is considered to be the analysis and measurement of the structural properties of networks of interdependent dyadic relationships [1]. These can be interpersonal relationships like friendships, relationships, work-related connections, or any kind of connection that tie individuals together in terms of the social aspect. These relationships are analyzed through the use of networks created with the help of graph theory.

For this project, we decided to use social network analysis to analyze a network that consists of friends lists from Facebook. Data is collected through surveys using the Facebook app and a friend network of 10 different individuals is used to generate the graph. The used dataset only includes nodes and edges to connect the provided nodes. Furthermore, it has been anonymized in a way that only shows the relationship between different nodes. No further data about the individuals (e.g. interests, liked pages, comments) is present.

Using this network, the main objective is to come up with an efficient and robust recommendation algorithm by extracting different metrics from the graph and using them to calculate a *recommendation score* for each possible node to be recommended. In addition to these metrics, we plotted several visualizations of key aspects of the graph, such as degree distribution, which helped us understand the underlying structure of the social network.

As the dataset does not contain personal information, the proposed friend recommendation algorithm is based on the structural properties inherent to social networks, i.e. the topological characteristics, the information and the metrics given by complex network theory.

2 Methodology

To analyze the dataset, a Python notebook is created in order to show our procedure and results step by step. All graph operations and the relevant used metrics are performed using the NetworkX library [2]. The different measures and descriptive statistics used include: betweenness centrality, degree centrality, eigenvector centrality, closeness centrality, different community structures, modularity, cluster coefficient, etc. in addition to the basic information of our graph such as the number of nodes and edges and average degree of the network. These metrics gave us a chance to understand the network and its properties.

In order to have a better understanding of the implemented recommendation algorithm and why certain features are used more actively, it is important to mention these metrics in more detail. Hence, some key metrics will be further explained in the following sections by emphasizing how it can be used in the implemented recommendation algorithm:

2.1 Betweenness Centrality

In graph theory, betweenness centrality is a measure in the graph that is calculated based on the shortest path between pairs of nodes. We know that for every pair of nodes, there is at least one shortest path such that number of edges in this path between these 2 nodes is considered to be minimum [3]. Based on this fact, betweenness centrality for each vertex is the number of these shortest paths passing through that vertex. The formula of the betweenness centrality is shown below:

$$g(x) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to t and $\sigma_{st}(v)$ is the number of shortest paths that happens to be passing from node v. This metric is considered to be important during the implementation of the recommendation algorithm since nodes with high betweenness centrality offer a chance to extract the valuable individuals since these individuals will be the crossing point of many different relations between different node pairs. Therefore, recommending a more important individual will not only increase the chance of offering more recommendations but it might also lead the given individual to different communities and individuals in a faster way as we know that this individual is the crossing point of many shortest paths.

2.2 Degree Centrality

Another centrality measure to use in this analysis is called degree centrality. It is often the easiest centrality measure to compute and it shows simply the degree of a given vertex. Sometimes this measure can be converted into a 0-1 scale. In such cases the highest degree node in the graph will have 1 degree centrality whereas lowest degree centrality in the network will be closer to 0.

Degree centrality allows us to analyze how many connections or relations a certain given individual has. An individual who has a high degree centrality might be connected with people at the center of the network, this will most probably cause that individual also has a high betweenness centrality; but an individual might also be far off on the edge of the network, and this will most probably cause that individual to have a low betweenness centrality since not too many shortest paths will pass through the node [4].

Overall, this metric will be useful for the recommendation algorithm in terms of finding the possible recommendations that have a lot of friends but it does not tell us whether this individual is at the heart of the network or not. Therefore, we reach to the conclusion that this metric should be used with other centrality measures such as closeness centrality or betweenness centrality to truly understand the importance of an individual in the given network.

2.3 Eigenvector Centrality

It is a measure of the influence of a node in the network. Relative scores are assigned to all nodes in a network based on the connections to high scoring nodes that contribute more to the score of the node in question than equal connections to low scoring nodes [4]. For a given graph G = (V,E) with $\|V\|$ number of vertices and let A be the adjacency matrix of the graph G where $a_{v,t}$ is 1 if vertex v and t have an edge and 0 if v and t do not have an edge. We define the eigenvector centrality as following where M(v) is a set of neighbors of v and λ is constant:

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{x,t} x_t$$

This metric proved to be useful in a sense that it does not show how many individuals one node is connected to, but the importance of those individuals based on the influential potential of their connections. The same idea can also be used in the recommendation system's implementation. Some users might prefer seeing individuals that are connected to popular or more "important" individuals in the recommendation system since these individuals are most likely to lead the person to different communities or different friends.

2.4 Closeness Centrality

In a social network, closeness centrality is a measure of closeness to other nodes in a network. It is calculated as the reciprocal of the sum of the length of the shortest paths between a given node and all the other nodes in the graph [4]. Therefore, the bigger the closeness centrality of a node is, the closer it is to the center of the network and easier to access to all nodes in a given network.

In a recommendation system, when we are measuring the degree of an individual it is usually not known where that individual is located in the network as it is mentioned above.

When individuals are located in the center of a network and assuming they have access to all nodes in general, individuals with higher closeness centrality become more important as they have access to different parts of the network and make it easier to connect with other people. Hence, using closeness centrality as one the metrics for our algorithm will improve the quality of the recommendations.

2.5 Community Extraction

In the study of network analysis, a network is said to have a community structure if individuals in a given community can easily be grouped with each other. This implies that the network can naturally be divided into different groups of nodes. Inside of a community these connections tend to be dense and other connections are much more sparse between different communities. This

is a concept which is closely related to the term *cluster*, used frequently in the field of Machine Learning. This also gives the definition of **modularity**, and it is designed to calculate the strength of a division of networks in terms of the connections that it forms. Overall, if a list of individuals has high modularity it will form dense connections between each other and sparse connections with any other individuals outside of the network [5].

Another general definition is that if two different individuals are in the same community, these individuals are most likely to be connected with each other. For the same reason, community analysis has been done on the social network to distinguish the nodes into different communities. All individuals are labeled to be in a certain community such that the modularity of the nodes in these communities is maximized. To achieve this goal, a modularity maximization approach called the *Louvain Method* is used to greedily optimize the modularity score of the communities. Since using a brute approach is almost impossible for a large number of nodes in terms of complexity, using a heuristic approach for this task has proved to be time-efficient even though it might not give the best result.

2.6 Cluster Coefficient

Just like modularity, the clustering coefficient is a measure of the degree of which nodes in a graph are more likely to cluster together. It is mentioned in the literature that in most of the real networks and in particular social networks, nodes tend to create tightly knit groups characterized by a relatively high density of ties [6]. However, it does not always mean that having high modularity will result in also having a high clustering coefficient due to the difference in calculation. Graphs that have a complete balanced bipartite structure has high modularity but very little clustering coefficient.

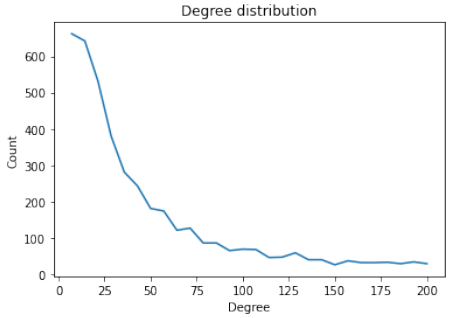
Moreover, since all the relations in this social graph are considered to have an undirected relationship (Facebook friendships), the clustering coefficient might also work well with the given structure. This metric might also be used to understand the density of the communities by calculating the ratio of closed triplets to all open and closed triplets in a given list of individuals. The formula can be seen as the following:

$$C = \frac{N_{opentriplets}}{N_{alltriplets}}$$

3 Data

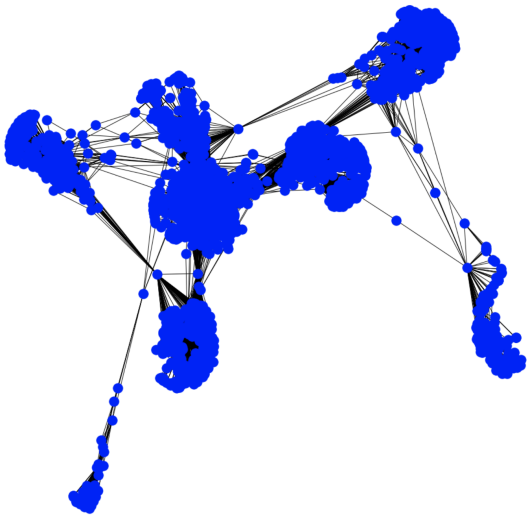
The metrics given above are used to have an understanding of the dataset and to generate the recommendation algorithm. As we stated in the introduction section, the dataset [7] used in this project is an anonymized version of the aggregated friends network of 10 different individuals. Hence, the dataset is considered as a social network, where nodes represent the different individuals in the network, and edges are used to represent the friendship between these individuals.

The network consists of 4039 nodes/individuals and 88.234 edges/friendships. The average degree is 43.69, which corresponds to the average number of friends that a person has.

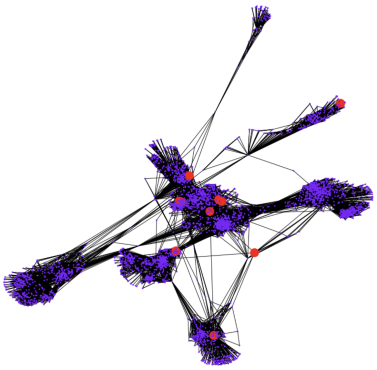


Although this data volume is relatively small compared to more complex real social networks, it has a number of nodes and edges sufficient to cause slowdowns in the computation of certain metrics such as betweenness, as all the shortest paths have to be calculated.

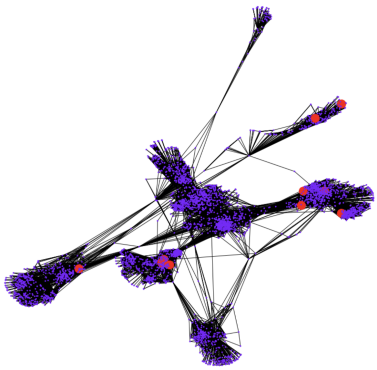
A simple visualization of the structure of the network, where we can already distinguish clearly separated clusters is as follows:



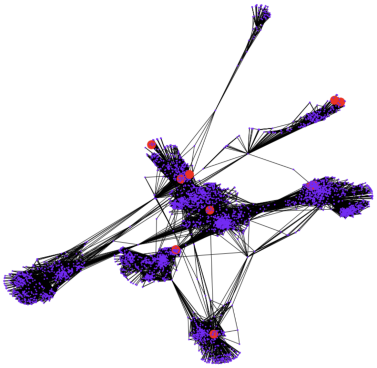
Different centrality measures are then used to generate the nodes with top 10 betweenness, degree, eigenvector, and closeness centrality.



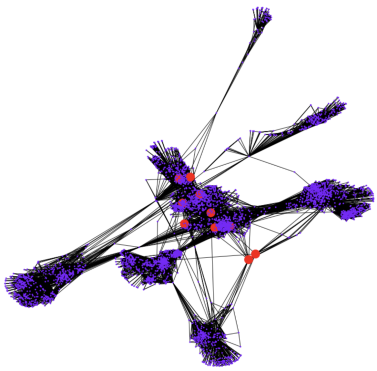
(a) 10 highest betweenness centrality



(b) 10 highest eigenvector centrality



(c) 10 highest degree centrality



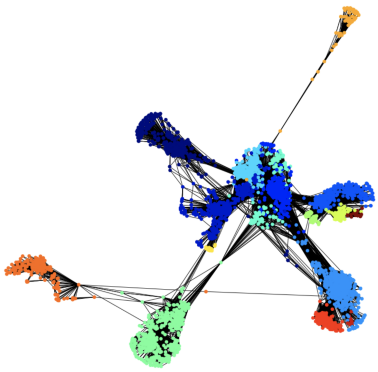
(d) 10 highest closeness centrality

Based on the results, it is possible to draw several conclusions regarding the data and what these metrics represent.

- Even though the nodes with the highest betweenness centrality are mostly located in the central areas in the graph, we still see that if a cluster has a substantial amount of nodes, there might be a node with a high value of betweenness centrality in the peripheral areas since this node is located in the shortest path of many different node pairs.
- Having a high degree centrality does not always mean that it also has a high betweenness centrality. It can be seen from the network that nodes might have high betweenness centrality but also has a low degree centrality or vice versa.
- Top 10 nodes that have the highest eigenvector centrality seems to be having a connection with the important nodes(high betweenness, high degree) in our graph.
- Another important conclusion was the location of the nodes in terms of closeness centrality. We see that all nodes are located relatively in the center of the network and eccentricity of these nodes mostly tends to have a low value compared to other nodes in the network. We

can define the **eccentricity** as the length of the longest path from a given node to any other node in the network.

It was also important to extract the different communities in the graph. For that reason, all communities are extracted by the Louvain Method in a way that will maximize the modularity score for a given list of nodes.



As a result, even though the dataset contains friend list of 10 individuals, 16 different communities are extracted from the network, which implies that some sub-communities are shared between individuals. Nodes in these communities are mostly connected with each other, and have a small number of connections with other communities.

4 Algorithm

The implemented algorithm is based on the *friend of friend(FoF)* approach. The initial possible recommendation nodes are extracted from the graph in a way that these nodes will be the FoF of a given node. Then, the total number of mutual friends is determined for each and every possible recommendation for later use as one of the deciding metrics.

In addition to the number of mutual friends, all the centrality measures explained above are taken into account and combined to calculate a recommendation score. However, the use of computing-intensive metrics such as betweenness can be deactivated. This produces less optimal recommendations, but speeds up the computation enormously.

Besides, the community value is used as a flag to prioritize the individuals in the same community or individuals from a different community. Our main motivation to use such an option is to offer various selections to the users, since some individuals might want to connect with people from another community or vice versa. However, more often than not, users will tend to add people from their own community.

The output of the algorithm is a list with N node identifiers, which are the recommended nodes. The number of recommendations to generate is configurable.

5 Results

5.1 Validity

Without user feedback and usage analytics, a friend recommendation algorithm remains a subjective system. We feel like the metrics used accurately align with recommendations that will probably interest users. However, this claim cannot be proved without statistics of user feedback.

In order to scientifically prove the validity of the proposed algorithm, we would need to put it into practice in a real system, and collect the user's usage. With this data, the weights given to the different metrics used in our system could be iteratively improved until the system produces attractive recommendations.

Talking about external validity does not really make sense in this case as a new dataset will obviously get different recommendations.

5.2 Reliability

Our algorithm uses some specific metrics and structures to propose recommendations. Given a network in a particular state, these recommendations will remain consistent no matter the number of times that the algorithm is run. However, changing the methodology will give different recommendations: adding new metrics or changing the weight given to the existing metrics(in our case they all have the same weight) will produce different results.

5.3 Scalability

The proposed algorithm relies on some centrality measures, specifically betweenness and closeness, which are quite computationally heavy. Due to this, the recommender system would slow down significantly if used with a bigger dataset. To help improve scalability, an option is provided which disables computation of these heavy-load metrics, thus improving the time complexity and allowing the algorithm to work fast enough with big datasets.

From the point of view of space complexity, the algorithm only makes calculations based on friends of friends, and this number remains stable around a mean value of 330 FoF. Of course some outliers are present, but in average, the space complexity will need to store and analyze metrics from 330 nodes.

6 Individual Contribution

All tasks were developed in collaboration with each other, spending a similar amount of time on the project.

References

[1] Steketee, M., Miyaoka, A., Spiegelman, M. (2015) "International Encyclopedia of the Social & Behavioral Sciences (Second Edition)","Social Network Analysis", pp. 461-467

[2] NetworkX. https://networkx.github.io/

[3] Betweenness centrality. https://en.wikipedia.org/wiki/Betweenness_centrality

[4] Centrality measures. https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/

[5] Modularity. https://en.wikipedia.org/wiki/Betweenness_centrality

[6] Holland, P. W., Leinhardt, S. (1971). "Transitivity in structural models of small groups".Comparative Group Studies. 2 (2): 107–124.

[7] Dataset. https://snap.stanford.edu/data/egonets-Facebook.html

[8] Golbeck, J. (2015) "Introduction to Social Media Investigation", "Analyzing networks", pp. 221-235

[9] A graph-based friend recommendation system using Genetic Algorithm.