Abstract

Networks are graphical representations of subjects and their relationships with each other. When we talk about social networks, it is common that the nodes are the representation of individuals and the links are the connections between them.

In this paper we analyze the main characteristics of a network of Facebook friends. Using an ego-network model, 10 main nodes ('ego' nodes) and their groups of friends ('alter' nodes) are represented.

The creation of a network with a random network model was achieved thanks to Gilbert's model, which succeeded in asserting the properties described in the theory.

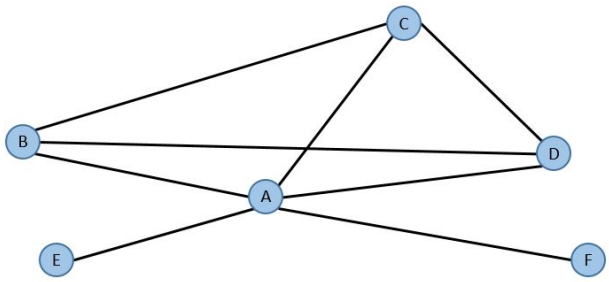
The study of this and other networks can be useful in understanding the communication of human beings, the ways in which they interact and how individuals can be influential or not within a network.

Introduction

The network was taken from the Stanford Network Analysis Project (SNAP), which was published in 2012 by J. McAuley and J. Leskovec. [1]

According to the information provided on the SNAP website, the network being worked on is an *ego network*.

According to the Georgia State University Research Guides, an *ego network* can be defined as a local network that has a protag/central node that is referred to as an *ego*. The nodes connected to the ego are called *alters*. Since the alters can only be referred to by the ego, ego networks are said to be "perceived" or "cognitive" networks. [2]



	Node A	Node B	Node C	Node D	Node E	Node F
Node A	0	-	-	-	-	-
Node B	1	0	-	-	-	-
Node C	1	1	0	-	-	-
Node D	1	1	1	0	-	-
Node E	1	0	0	0	0	-
Node F	1	0	0	0	0	0

Figure 1.1 - Ego Network with ego node A

The analysis of this type of networks is useful for the detection of ties between individuals for the purposes of support, resource access and information dissemination. Specifically, the analysis of alters allows to know how they act as influences on the network ego.

In the paper *Learning to Discover Social Circles in Ego Networks*, conducted by Stanford's scientists, a method was created to find social circles across 10 ego networks. The results concluded that the new method was correctly able to identify overlapping circles as well as sub-circles (circles within circles). "The model naturally learns the social dimensions that lead to a social circle." And found out that "membership to the same community provides the strongest signal that edges will form, while profile data provides a weaker (but still relevant) signal". [3]

The topic of ego networks and how nodes interact with each other is of particular importance to social networking companies and users alike. On the companies' side, an algorithm capable of successfully segmenting a user's contacts leads to the user having contact with those people he or she is most interested in. On the other hand, users can update their lists (or circles of friends) on their social networks automatically.

The dataset is in plain text format. Using Python's *networkx* module, the following graphical representation of the network was obtained:

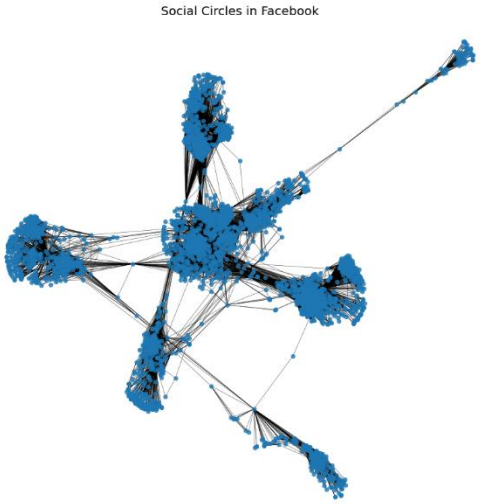


Figure 1.2 Graphic representation of Facebook's Ego Network

It is worth mentioning that the network shown consists of 10 ego networks, so there are 10 *egos*, while all the other nodes would be *alters*.

When talking about a network of friends on Facebook, it can be observed that we are dealing with a non-directed network. The study also tested its models with Twitter and Google+ datasets, which are represented by directed networks.

Network Characteristics

The Python module *networkx* provides tools to compute many of the features quickly. Other features were taken directly from the page where the dataset was published (SNAP).

Table 2.1 shows the main characteristics of the network:

Table 2.1. Main characteristics of the network

Characteristic	Measure
Number of nodes	4039
Number of links	88234
Average Path Length	3.69 = 4
Clustering Coefficient	0.6055
Ego nodes	10
Alter nodes	4029
Average Degree <K>	43.6910
Density	0.01082
Diameter	8

Although it may seem that the network being worked on is a **densely** connected one (judging by the number of links in relation to the number of nodes), the opposite is true. This is mentioned by Barabasi [4] who emphasizes that "real networks are sparse". Later, with the **degree distribution** of the nodes, we will see how this is true.

Although the **average degree** is 44, it is known in advance that the mean is not a reliable representation of the true degrees of the nodes within the network. Even so, the data tells us that, on average, each node is connected to 44 other nodes. In the context of the network, this means that, on average, all individuals have 44 friends/contacts added on Facebook.

The **average path length** is 4, which can be understood to mean that, regardless of who you are in the network, you will be (on average) 4 people away from any node in the network. This is a measure I trust more if you keep in mind that it is a relatively small network (4039) nodes grouped in social circles that are connected to each other by a few nodes. The most “isolated” nodes are 8 people (nodes) away from their farthest pair (measure described by the **diameter** of the network).

Centrality Measure(s)

For the social network used, 3 measures of centrality were chosen, namely:

- Degree Centrality.

Degree centrality was chosen because it is one of the easiest centrality measures to calculate and can give us a good first idea about the most influential nodes within the network [5][6]. It is worth mentioning that those nodes with the highest centrality score are expected to be the ego nodes, since their very definition implies that they are already important themselves.

According to Cambridge Intelligence, degree centrality assigns an importance score based simply on the number of links held by each node. [7]

Since it is an undirected network, the nodes have only one score.

- Eigenvector Centrality.

Eigenvector centrality provides us with a different condition for the centrality score to remain high. Although the number of links between nodes is important, it must also be considered whether the nodes to which it connects are important in themselves. [6]

The eigenvector centrality x_i of node i is defined to be proportional to the sum of the centralities of i's neighbors. [6]

$$x_i = \kappa^{-1} \sum_{\substack{\text{nodes } j \text{ that are} \\ \text{neighbors of } i}} x_j,$$

Figure 3.1. Eigenvector Centrality formula for an undirected network

This is helpful in determining the most influential nodes within a network. Of course, given the nature of the chosen network, it could be expected (again) that the most influential nodes are the *ego* nodes, and that the score of the *alter* nodes is directly linked to their connection to the number of *ego* nodes.

- Closeness Centrality.

Finally, closeness centrality allows us to know how close a node is to all the other nodes in the network. This is a measure of centrality of interest for this paper, since it allows us to see which nodes are more influential at the "global" level.

$$C(x) = \frac{N}{\sum_y d(y,x)}.$$

Figure 3.2. Closeness Centrality formula for an undirected network

Table 3.1 shows the 10 nodes with the highest centrality score according to the centrality measures described above.

Table 3.1. Nodes with highest centrality score

Centrality Measure	Nodes with highest score
Degree	107, 1684, 1912, 3437, 0, 2543, 2347, 1888, 1800, 1663.
Eigenvector	1912, 2266, 2206, 2233, 2464, 2142, 2218, 2078, 2123, 1993.
Closeness	107, 58, 428, 563, 1684, 171, 348, 483, 414, 376.

According to [5] degree centrality is a natural measure in social networks. It makes sense to think that the most important nodes are those with the largest number of connections within a network.

What does this metric tell us within an *ego network*? This metric returns us (with a very high probability) to *ego* nodes per se. We must remember that the definition of an *ego network* is one in which there are "central" nodes. The dataset consists of 10 people who act as *egos*, so the friends of these nodes are considered. On the other hand, the *alter* nodes are tried to be visualized as possible friends of the *ego* nodes, they play a secondary role. In this way, the list of total friends of the *alters* is not considered. However, there is a possibility that some *alters* are friends with many contacts of the *egos*.

The eigenvector centrality provides additional information by taking into account the centrality of the other nodes and not only the one whose centrality is to be known. [5] In this case we might expect the nodes with the highest scores to be:

- *Ego nodes* that are friends of other *egos*.
- *Alters* who have many *egos* as friends.

The closeness centrality allows to know the node centrality based on the average shortest paths between a node and the others. This is a good metric to determine possible bridges between *alters*: while less important alters must pass through a "bridge" node to reach other *egos*, the *alters* that serve as a bridge can be considered the ones that most people know. It could also help us to know which nodes know the most people.

Degree Distribution and Models of Networks.

Degree Distribution

A graph of the degree distribution is shown below:

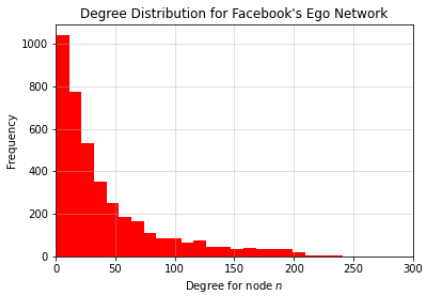


Figure 4.1. Degree distribution for Facebook Network

A trend as described by [4] can be observed, in which real networks tend to have few highly connected nodes, while most have few connections to other nodes. In this case, this means that most people have few friends, while a small percentage have a large number of friends. Again, it is vitally important to note that this behavior is noticeable because we are working in an ego-network, in which we care about certain main nodes, while we have no information about the actual number of friends of the alter. However, if we had all the friends of both types of nodes, it is highly probable that the distribution would remain the same.

I did not consider it necessary to standardize this data, as I was only interested in its distribution.

Models of Networks

The network used is suitable for modeling using a Random Network Model. As mentioned in [4] [5] [6], the Gilbert Random Model requires two parameters to create a random graph: the number of nodes n and a probability p that serves as a conditional to know if a pair of nodes will have a connection to each other or not.

Zafarani [8] offers a quick way to calculate the value of the variable p using the following formula:

$$p = \frac{c}{n-1}.$$

Figure 4.2. Formula for calculating p

Where:

- c is the expected degree of the graph.
- n is the number of nodes.
- p is the probability that two nodes are connected.

However, p can also be obtained with the value of the network density. The *networkx* module allows a quick calculation of the value.

Having calculated p, we can now calculate the expected degree of the random network.

$$c = (n-1)p,$$

Figure 4.3. Formula for calculating c

For the second estimated value (expected number of edges) it is possible to use the formula described by Zafarani. [8]

$$\textit{The expected number of edges in } G(n, p) \textit{ is } \binom{n}{2} p.$$

Figure 4.4. Expected number of edges

With the two parameters necessary for the creation of a random network using the Gilbert model, the following graph was generated:

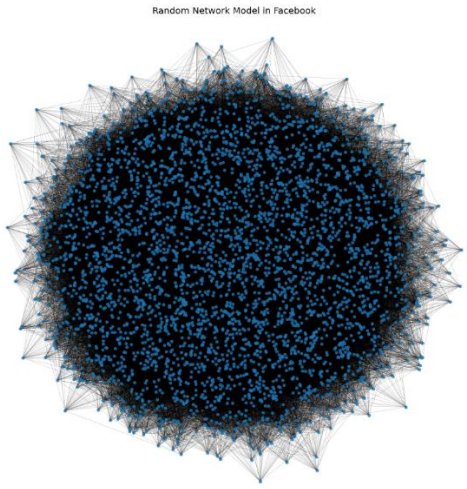


Figure 4.5. Random Network Model for Facebook network

Table 4.1 shows the expected values against the actual values obtained after the creation of the random network:

Parameter	Expected Value	Value Obtained
Number of Links	89285	89267
Average Degree	44.2114 = 44	44

It is easy to see that the expected values are almost equal to the actual values obtained, demonstrating the behaviors described by both Gilbert and Erdős–Rényi. [4]

As mentioned by Barabasi, real networks are sparse, which means that $\langle k \rangle << N$. In this case the degree distribution of the randomly generated network is expected to follow a Poisson

distribution trend, since N is sufficiently large. The degree distribution plot of the random network is shown below.

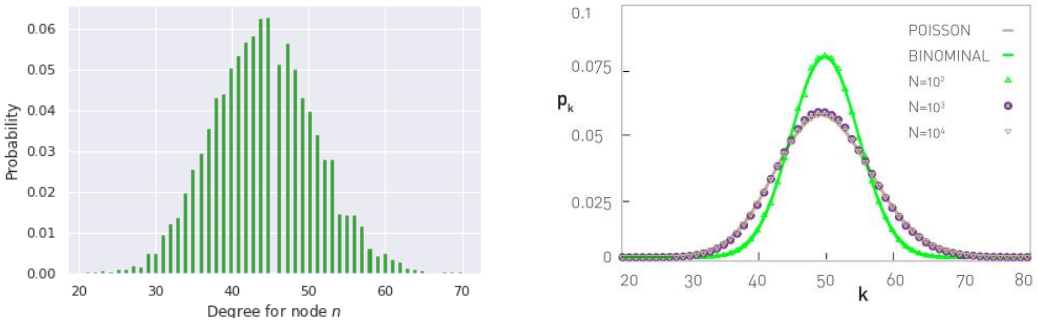


Figure 4.6. Degree distribution for random network and expected degree distribution

The degree distribution closely resembles a Poisson distribution. A peak can be observed at $\langle k \rangle = 44$, which is the average degree.

Community Detection.

The Louvain algorithm was used for community detection. [ref] provides an explanation of the algorithm.

> *"The Louvain algorithm is an agglomerative algorithm, which works by taking single nodes and joining them into groups, then joining groups with other groups, and so forth, to find the configuration with highest modularity. Initially, each node is placed in a separate group on its own. Then one performs a node-moving procedure akin to that of Section 14.2.2, although not identical. In this procedure, one goes through each node in turn and moves that node to another group chosen such that the modularity of the complete system is increased by the largest amount. If no move increases the modularity, then the node stays where it is. Also, to make things faster, one only ever considers moving a node into a group that contains at least one of its neighbors. When all nodes have been considered and potentially moved, one repeats the process, and continues to do so until there are no more moves that increase the modularity. This ends the first round of the algorithm. On the next round, one carries out the same procedure again, but now instead of moving nodes, one moves whole groups. That is, one treats the groups found on the previous round as the units of the algorithm and moves them, in their entirety, from group to group to increase the modularity, stopping when no further increase is possible. And so the algorithm proceeds, through as many rounds as are necessary until one reaches a configuration where there are no moves at all that will increase the modularity. This final configuration is then taken as the community division of the network."* [5]

One of the reasons for choosing this algorithm over other options for community detection is its speed. Louvain's algorithm has an $O(nlog(n))$ runtime.

The *networkx* module allows the detection of communities through the *community_louvian* function. A user-defined function was performed to determine the colors and the position in the network with the communities found, the result is shown in the following image.

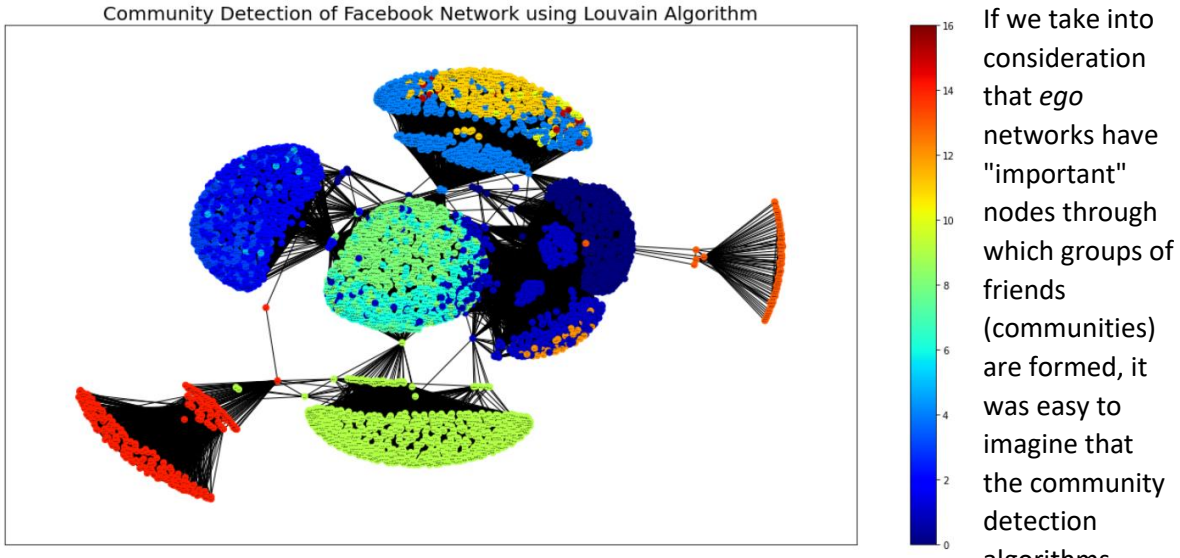


Figure 5.1. Community Detection using Louvain Algorithm

If we take into consideration that *ego* networks have "important" nodes through which groups of friends (communities) are formed, it was easy to imagine that the community detection algorithms would find approximately 10 communities. It could be said that the result found in the algorithms did not

come as a great surprise to me. In fact, from the beginning, a tendency of the nodes to group into clusters could be visualized, only that they were not exactly 10 (they seemed to be 8).

Another interesting algorithm for the detection of communities in networks is the Girvan-Newman algorithm. An attempt was made to run the function included in the networkx module. This algorithm has O(n) complexity, which makes it somewhat slow compared to the previous one. If we consider the possible combinations that the algorithm must perform, one can notice that the process is quite slow. The generation of Figure 5.2 took approximately 2 hours and 30 minutes.

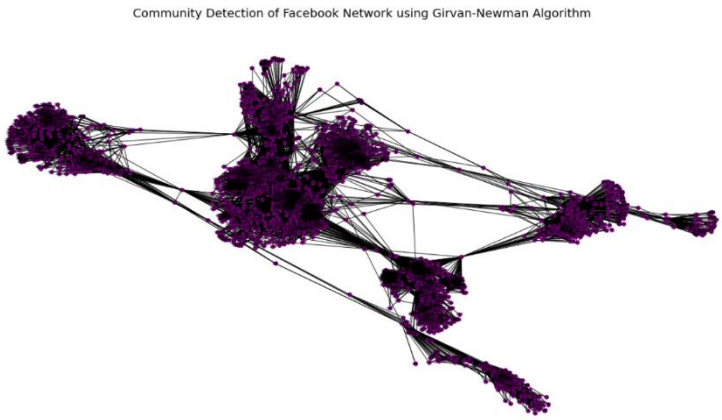


Figure 5.2. Community Detection using Girvan-Newman Algorithm

Regardless of the time to complete the execution of the algorithm, I consider that the results are similar to those found by the Louvain algorithm.

During the execution of this algorithm there were some errors when assigning the color to the communities. Given the long run time, I decided to leave it as shown. While the 10 communities I expected to find cannot be visualized, about 8 can be distinguished by the groupings formed by the algorithm.

Conclusions & Future Work.

Perhaps the most interesting observation is the realization that while *ego* nodes play a crucial role and are of great importance in the network, *alter* nodes that serve as bridges between groups of friends also share some importance. The various centrality metrics agreed on some nodes, but in the judgment of the writer of this paper, it is the eigenvector centrality that demonstrates the indicated centrality of nodes within the network.

Studying a social network from a scientific approach allowed a better understanding of social relationships and their characteristics in the real world. As well described by Barabasi in the Network Science book, social networks are sparse. The metrics obtained confirm the theory described in the books.

Although some of the results were not unexpected given the nature of the network, it should be noted that the process of calculating them was not.

The random network model proposed by Gilbert proved to be accurate in simulating the behavior of the original network.

One of the major impediments (especially in the community detection section) was the size of the network. It would be interesting to test other community detection algorithms with smaller networks but with other characteristics, such as weights and addresses.

Reference(s):

[1] J. McAuley and J. Leskovec, “Social circles: Facebook,” Learning to Discover Social Circles in Ego Networks. 2012.

[2] J. Walker, “GSU library research guides: Network analysis: Ego networks,” 2019.

[3] J. McAuley and J. Leskovec, “Learning to discover social circles in ego networks,” in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, 2012, pp. 539–547.

[4] A.-L. Barabasi, Network Science. Cambridge, England: Cambridge University Press, 2016.

[5] F. Menczer, S. Fortunato, and C. A. Davis, A First Course in Network Science. Cambridge, England: Cambridge University Press, 2020.

[6] M. Newman, Networks. London, England: Oxford University Press, 2018.

[7] A. Disney, “Social network analysis: Understanding centrality measures,” Cambridge-intelligence.com, 02-Jan-2020. [Online]. Available: https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/. [Accessed: 28-Jul-2021].

[8] H. Liu, M.-A. Abbasi, and R. Zafarani, Social Media Mining: An Introduction. Cambridge, England: Cambridge University Press, 2014.