# An Investigation Into Modeling With Networks

Math380 Spring 2018 Project - Code available at
https://github.com/philiphossu/Modeling-With-Networks

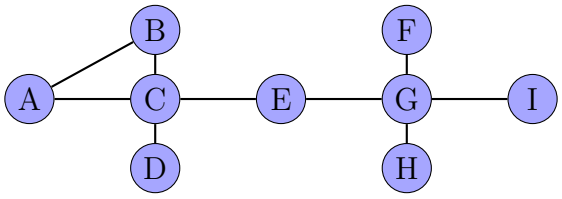
*Philip Hossu, Paolo Ratti Tamayo, Jiateng Sun*

*April 26, 2018*

## Contents

## 1. Introduction

Graphs are some of the most versatile and interesting structures in mathematics. A graph G is characterized as an ordered pair of vertices and edges, $G = (V(G), E(G))$. The edges in a graph imply the existence of a relationship between two vertices or objects. They can be directed (implying a relationship based on ordering) or undirected where the edges have no orientation. Edges can also be weighted, having several real world applications like distance, cost, etc. These structures evidently can represent countless applications which have discrete and related objects. A simple undirected graph which will operate as our running example is shown below.




Influence is the capacity to produce an effect in indirect or intangible ways. If one wants to be successful in their field, they must share their ideas and persuade their peers to accept them. This idea is present in all types of industries and specializations. We analyze several networks including the Erdos1 co-author network and an actors collaboration network with multiple influence models and discuss their results.

In the following project report, we analyze how influence can be determined in a graph and describe our exploration into mathematical modeling with networks. We follow mathematical notation given by *A First Course in Mathematical Modeling* by Giordano, Fox and Horton (2013). There exist several known measures to explore the importance of a vertex in a graph, including degree centrality, betweenness centrality, etc. We will model various networks, analyze them, find who/what has the most influence in the network, and describe the reasons behind this result. We seek to combine real world scenarios with some of these mathematical ideas to accurately describe the concept of influence in graphs. Furthermore we will argue how these methods can show steps to rapidly boost ones own influence.

The prompt asked us to consider a number of specific tasks, briefly outlined below:

- Build and analyze the Erdos1 co-author network using the data from:

  https://files.oakland.edu/users/grossman/enp/Erdos1.html.

  The Erdos1 co-author network is characterized by the authors who have collaborated with the famous mathematician, Paul Erdos. Rather than including every entry in this data file, we were asked to consider a network where each author has collaborated directly with Erdos (omitting Erdos himself). After building this network, we are asked to explore the graph and find some interesting features and properties.

- Define and study a critical measurement by which to determine the influence of authors in the network we created in part a.

- Gather data, build, and analyze a network showing the relationship between some

foundational papers in the emerging field of network science. Apply the influence measures used for parts a,b and discuss their effectiveness. Also, discuss methodology and other factors surrounding this network.

- Gather data, build, and analyze a real life scenario which can be modeled as a network. Again apply influence measures and discuss the results, shortcomings, external factors, etc.
- Discuss how influence and impact can be used in real life situations. Consider business decisions, improving influence, selecting a graduate school etc.

To address the above tasks, we utilized the R programming language and our knowledge from various math and computer science courses.

## 2. Statement and Analysis of the Problem

While our project has a number of specific requirements discussed in the introduction, essentially the problem we are trying to approach is how to measure and interpret the influence of a vertex within a network. We define three influence measures to address this problem, discussed in depth in the latter sections. Once these are defined, we collect and clean multiple data sets, create multiple networks, and analyze these networks under the applicable metrics.

While the field of graph theory is certainly not young, we noticed a wealth of more recent works geared towards our topic and more general social network analysis. We began by finding a modern survey of various graph centrality measures, notably the work titled *Centrality Measures in Networks* (Bloch and Jackson 2016). Rather than focusing on a concrete example, this work provides a much needed overview of several fundamental graph measurements and their properties. This publication helped us determine what different types of graph metrics exist and how they are calculated. This was not the only publication which we considered to help us learn about various graph metrics. The work by Segarra and Ribeiro titled *Stability and Continuity of Centrality Measures in Weighted Graphs* gave us an insight into some of the advantages and disadvantages of metrics like Eigenvector centrality and Betweenness centrality (2015).

Another work which we used to help shape our understanding of the problem was the comment released by Karwa and Petrovic titled *Coauthorship and citation networks for statisticians* (2016). This paper looks directly at how co-author networks and citation networks can be analyzed which is very relevant to our project. Regrettably, the networks and paper which Karwa and Petrovic are commenting on are more advanced than what we are able to make. Consequently, their analysis is also more advanced. As we will discuss, our Erdos1 co-author network is slightly oversimplified. An edge in our network implies the presence of two authors collaborating at least once, but the exact number is unknown. We are not able to give weights to our edges. Regardless, we were able to take a number of pointers from this paper. For a complex network, the paper suggests that a metric like counting the degree of each node is far too simple. For our simple network, such a measure may be more appropriate.

A number of publications also exist which analyze specific co-author networks. One such work is *Co-authorship and citation networks in Spanish history of science research* by Osca-Lluch, et al (2009). The author's data allows them to look at papers published by various journals, number of papers published by Spanish scientists over time, types of publications, etc. While we did not have this type of data readily accessible, the paper also looks at co-author collaboration and the impact of these publications. The paper interestingly notes that a very high percentage of authors have no collaboration whatsoever. They create a visualization of their graph where the thickness of each edge represent number of collaborations between two authors. To analyze the impact of a publication, the authors weigh in on using a quantitative measurement vs a qualitative one. They argue that a qualitative approach can be effective due to its ability to take into consideration the expertise, reach, and prestige of a researcher and their written work separately from the journal which they published in.

Despite that our networks were based on much more simple data sets, it was interesting and useful to see how others have approached similar co-author and citation networks. This problem is not totally novel, but it's apparently getting increasing amounts of attention, as we were able to find several publications referencing co-author and citation networks in the last 10 to 15 years.

## 3. Description of the Model

We selected three primary models by which to determine influence in a graph. While these metrics are not entirely novel, we explain their formulation both mathematically and conceptually.

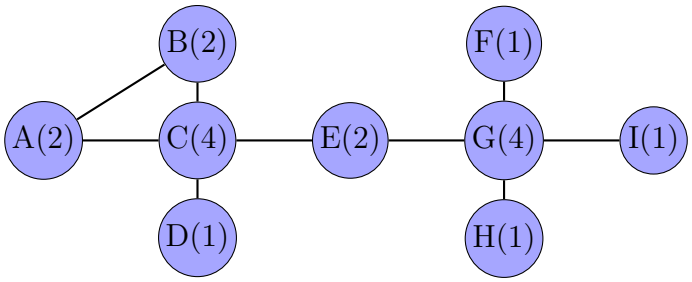
### 3.1 Total Degree

The degree of a vertex v, commonly denoted $deg(v)$, represents the number of edges incident to the vertex. This calculation can vary slightly depending on if the graph is directed or undirected. For the purpose of our application, we choose our measurement to be the total degree of a vertex, where total degree is defined as follows:

For a directed graph $G = (V(G), (E(V))$, Total Degree of a vertex v = $\sum edges\ in\ (v) + \sum edges\ out\ (v)$ where edges in (v) represents the incident edges directed into a vertex v and edges out (v) represents the incident edges directed out of the vertex v.

For an undirected graph, the Total Degree of a vertex v is simply $\sum edges\ incident\ to\ (v)$.

We show an example calculation of the total degree of each node in a simple graph below. Note that the degree of each vertex is written inside the vertex next to the label.




By the total degree metric, we would say that the most important/influential vertices in this graph would be C and G, each having a degree of 4. These vertices have the most direct connections to the other vertices in the network.

Another way to visualize this graph and consider the degree of each vertex is via an adjacency matrix. This is a V x V matrix where the value of each entry denotes the presence (1) or absence (0) of an edge between the two corresponding vertices. The value in each entry can be modified to contain the edge weight. The corresponding adjacency matrix for our running example is shown below.

$$\begin{array}{c} \begin{matrix} A & B & C & D & E & F & G & H & I \end{matrix} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \end{matrix} \end{array}$$

From this adjacency matrix, we can confirm the total degree counts of each vertex. Looking at vertex G for example, taking the summation of the G row and G column both yield the degree of 4 which we were previously able to calculate visually. In a directed graph, a similar approach can be taken but accounting for the fact that the matrix would no longer be symmetric about the main diagonal.

We selected the total degree metric due to its simplicity and relevance in the example networks which we will consider. However, this metric doesn't always reflect the real world definition of influence.

### 3.2 Eigenvector Centrality

The eigenvector centrality of a vertex in network is a more advanced measurement which can help determine influence of a vertex. While this measurement also suggests that the importance of a vertex depends on the vertices it is directly connected to, it adds a layer of

complexity. Rather than considering the number of incident edges to a vertex, eigenvector centrality suggests that the importance of a vertex is relative to the importance of the vertices it is connected to (Ruhnau 2000).

This metric is based on the general eigenvector problem, defined as follows: The eigenvalues and eigenvectors of a square mxm matrix are the scalar values $\lambda$ and vectors $\vec{x}$ respectively that are the solutions to:

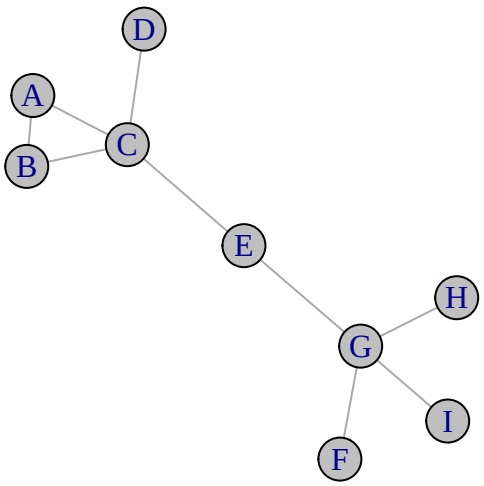
$$A\vec{x} = \lambda\vec{x}$$

Now, applying this problem specifically to graph centrality, the eigenvector centrality is defined as follows: Given a graph $G = (V(G), E(G))$ and its corresponding adjacency matrix A, the score of a vertex $X_v$ is calculated as follows with M(v) representing the set of neighbors of the vertex v $\in V(G)$. The specific value of lambda can be calculated via power iteration.

$$X_v = 1/\lambda \sum_{t \in M(v)} X_t$$

We utilize the igraph package in R which provides an easy to use interface through which to perform this calculation (Csardi and Nepusz 2006). The following code block demonstrates the calculation on our example using igraph.

```
# Loading Package
library('igraph')
# Creating edge list, vertex list
example_edges <- data.frame(x=c("A","A","B","C","C","E","G","G","G"),
                    y=c("B","C","C","D","E","G","F","I","H"))
example_vertices <- data.frame(x=c("A","B","C","D","E","F","G","H","I"))
# Generating network
net <- graph_from_data_frame(d=example_edges,vertices=example_vertices,
                             directed=F)
# Plotting network
plot(net,vertex.size=20,margin=0,edge.arrow.size=0.3,vertex.color="gray75")
```



```r
# Calculate Eigenvector Centrality of each Vertex
EGvals <- eigen_centrality(net)$vector
sort(EGvals, decreasing=TRUE)
```

```
##         C         A         B         E         G         D         F
## 1.0000000 0.6980301 0.6980301 0.6254604 0.5214969 0.4110823 0.2143781
##         H         I
## 0.2143781 0.2143781
```

Eigenvector centrality has several advantages over using a more simple approach like the total degree of a vertex. A vertex having many incident edges doesn't necessarily have a higher eigenvector centrality, which differs from total degree centrality and arguably more accurately measures the importance (influence) of nodes. We can see in the above example that using eigenvector centrality, the importance of the nodes is different from the degree approach. Even though vertices D and H both have degree 1, eigenvector centrality acknowledges that D is the more important vertex because it's connected to C which is also a more important vertex in the graph. Eigenvector centrality recognizes that vertices in an interconnected component are more important than vertices which are weakly connected. We also see that even though E has a degree of 2 compared with the degree 4 of G, eigenvector centrality finds E to be the more important vertex because it is connected to very important components.

The primary disadvantage with Eigenvector centrality is that it's geared towards undirected graphs which consequently have symmetric adjacency matrices. Eigenvector centrality is very applicable to our undirected network scenarios like the Erdos1 network and the actor collaboration network. However, when it comes to the citation network, we will use a third metric to determine the influence of a node.

### 3.3 Alpha Centrality

Alpha centrality adds a new consideration to Eigenvector centrality to allow the metric to be applied to directed graphs which have asymmetric adjacency matrices (Bonacich and Paulette 2001). Given a graph $G = (V(G), E(G))$ and its corresponding adjacency matrix A, alpha centrality is defined as follows where e is the importance given to a vertex and $\alpha$ is a constant:

$$\vec{x} = (I - \alpha A^T)^{-1}\vec{e}$$

Alpha centrality begins by giving every vertex a starting random positive amount of influence. The algorithm divides the influence of each node and updates the relative influence of its outward neighbors while also receiving influence from its inward neighbors based on the alpha coefficient. As the alpha parameter decreases, the significance of a directed edge is larger in the calculation. This process repeats until the graph reaches a steady state, through power iteration. Once this is achieved, we are returned the relative ranks of the vertices in the graph based on their centrality scores.

# 4. Analysis and Testing of the Model

## 4.1 Aside: Data Cleaning & Network Creation Using iGraph Package

Before we could test any of our influence metrics, we had to collect, clean, and import data into R. The raw data for the Erdos1 collaborator network (found at https://files.oakland.edu/users/grossman/enp/Erdos1.html) is a 1 column by ~19000 row document that contains one name in each row. The authors who collaborated directly with Erdos have no spaces before them. Each direct collaborator is followed by a variable number of entries which begin with a set number of spaces. Each of these represent a collaborator of the direct collaborator of Erdos. We had to create an algorithm to parse this file, the pseudo code of which can be seen below:

```
Initialize currentMain ← "", Initialize df to store all the vertices
Open data file
currentMain ← First Level 1 Collaborator (first line in file)
while File is not empty do
    Read next line
    if Line is empty then
        Level 1 collaborator has no more Level 2 collaborators
        currentMain ← Next Level 1 Collaborator
    else
        Create an entry with currentMain and his next collaborator
        Append the new entry to df
    end if
end while
Close file
return df
```

To create the actual networks in R, we utilize the iGraph package available at http://igraph.org/r/. In order to create an igraph network object, we need to provide a two column edge list, where the first column represents the "from" vertex and the second column represents the "to" vertex. We also must create and specify a vertex list containing all of the unique vertices. Each of these data frames can have subsequent columns containing extra data about each edge/vertex like weight, etc. Given these pieces of information, we can create, visualize, and run calculations on directed and undirected networks.

In the following sections we focus primarily on describing the results of our testing with irrelevant code blocks being run but hidden from view. Our full code can be found at https://github.com/philiphossu/Modeling-With-Networks.

## 4.2 Exploring the Erdos1 Co-author Network

Here, we create the Erdos1 co-author network after cleaning and importing the data. We make this network undirected since our data only shows some form of collaboration between

authors. Prior to testing our metrics on this network, we explore some of the interesting properties of this network as suggested in the project requirements.

```
erdos1_network <- graph_from_data_frame(d=result, vertices=nodes,
                                        directed=F)
erdos1_network <- simplify(erdos1_network, remove.multiple = TRUE)
length(V(erdos1_network)) # Number of vertices in the graph
```

```
## [1] 474
```

```
length(E(erdos1_network)) # Number of edges in the graph
```

```
## [1] 1663
```

We see that this network consists of 474 vertices and 1663 edges – a relatively large network. It is worth noting that our network does not contain all 511 Erdos1 authors because we go through our data frame and remove all the edges which are not between Erdos1 authors as instructed.

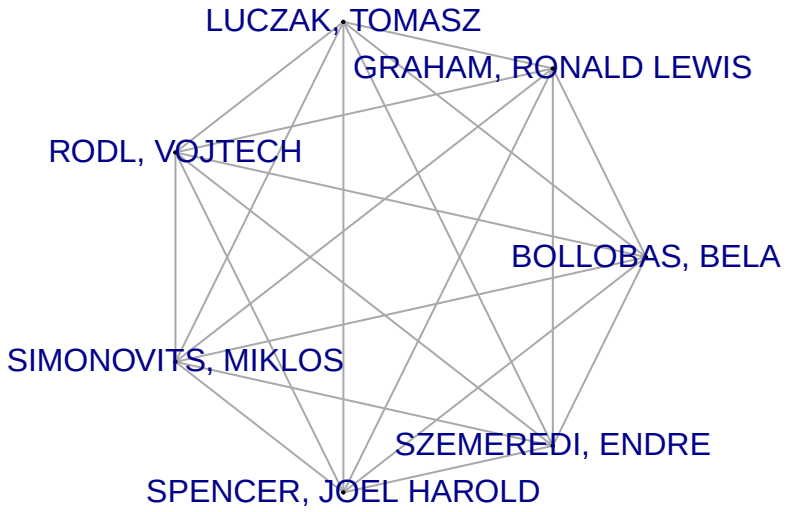
The first aspect of the graph we look at is the presence of cliques. A clique is a subset of vertices where each vertex is connected to every other vertex. A simple example of a clique would be a triangle which has three vertices connected by three edges. In terms of our network, the real world meaning of a clique would be a set of authors who all collaborated with each other at some point in their careers. Simply calculating the total number of cliques larger than 3 in our network returns in a very large data set.

However, we can also find only the largest cliques present in the graph and plot one of these largest cliques, as shown in the code segment below.

```
# List of biggest cliques
biggestCliquesFound <- largest_cliques(erdos1_network)
# Number of largest cliques present
length(biggestCliquesFound)
```

```
## [1] 11
```

```
# Visualizing one of the large cliques
clique1 <- biggestCliquesFound[[1]]
cliqueNetwork <- induced.subgraph(graph=erdos1_network,vids=clique1)
plot(cliqueNetwork, edge.arrow.size=.01,vertex.size=1,margin=0,
     layout=layout_in_circle, vertex.label.family="Helvetica")
```




Another property which we explored was the diameter of the graph. The diameter of a graph is the longest path present in the graph.

```
get.diameter(erdos1_network) # Show the longest path
```

```
## + 11/474 vertices, named, from 86da9a0:
##  [1] BLEICHER, MICHAEL NATHANIEL FEJES TOTH, LASZLO
##  [3] MAKAI, ENDRE, JR.           PACH, JANOS
##  [5] FUREDI, ZOLTAN              RUBEL, LEE ALBERT
##  [7] SHIELDS, ALLEN LOWELL       PIRANIAN, GEORGE
##  [9] BAGEMIHL, FREDERICK         GILLMAN, LEONARD
## [11] HENRIKSEN, MELVIN
```

While this metric doesn't have a lot of real world meaning in terms of determining influence, it's still an interesting graph property to explore. It's possible that the authors on either end of this longest path would have very different academic works but are somehow connected through this longest path due to their co-authors.

Edge density is also an interesting calculation relevant to our network. This is defined as the ratio of number of edges in the graph over all of the possible edges, and is calculated below.

```
edge_density(erdos1_network) # Show the density calculation result
```

```
## [1] 0.01483484
```

Given our 1663 total edges and 474 vertices, the result of this calculation is accurate. We see here that compared to the possible connectedness of the graph, the Erdos1 co-author network is fairly sparse.

The last property we considered was the clustering coefficient in the graph. The clustering coefficient measures locally how connected the vertices are.

```
# Show the clustering coefficient calculation result
transitivity(erdos1_network)
```

```
## [1] 0.2245325
```

This number shows the probability that an author's co-authors have collaborated together. This metric also shows us how connected the vertices in the graph are, and given the result, we see that this probability is not high.

### 4.3 Testing the Erdos1 Co-author Network

Since we created this network in the previous section, we can begin by calculating the total degree of each vertex, shown in the code segment below.

```
totalDegree <- degree(erdos1_network, v = V(erdos1_network), mode = c("all")
                      ,loops = TRUE, normalized = FALSE)
head(sort(totalDegree, decreasing=TRUE))
```

```
##        ALON, NOGA M.       BOLLOBAS, BELA GRAHAM, RONALD LEWIS
##                   54                   44                   44
##        HARARY, FRANK        RODL, VOJTECH       FUREDI, ZOLTAN
##                   44                   43                   40
```

According to the total degree method, the most important authors in the Erdos1 co-author network are NOGA M. ALON, BELA BOLLOBAS, RONALD LEWIS GRAHAM, etc. This model suggests that the authors who have the highest degree, or most collaborators, are the most important. However, as shown in our running example, the total degree metric doesn't necessarily give the full picture.

Next, we can calculate the eigenvector centrality, our second model, which assigns importance to an author based on the importance of the authors who they are connected with.

```
eigenValues <- eigen_centrality(erdos1_network)$vector
head(sort(eigenValues, decreasing=TRUE))
```

```
##        ALON, NOGA M.        RODL, VOJTECH       BOLLOBAS, BELA
##            1.0000000            0.8852907            0.8226504
##       FUREDI, ZOLTAN GRAHAM, RONALD LEWIS      GYARFAS, ANDRAS
##            0.7855001            0.7646224            0.7001373
```

Based on the eigenvector centrality model, the most important Erdos1 co-authors are NOGA M. ALON, VOJTECH RODL, BELA BOLLOBAS, etc.

We can also create a visualization of the network where the color of each vertex represents its eigenvector centrality, shown in Figure 1. The cooler the color, the lower the eigenvector centrality (red is most important, blue is least important).

We see some similarities and differences between the outputs of our two models. We look specifically at the rankings of "VOJTECH RODL" and "RONALD LEWIS GRAHAM" in the degree model vs the eigenvector model. In the degree model, Ronald Lewis Graham is considered more important or influential because he has 44 connections to other authors vs

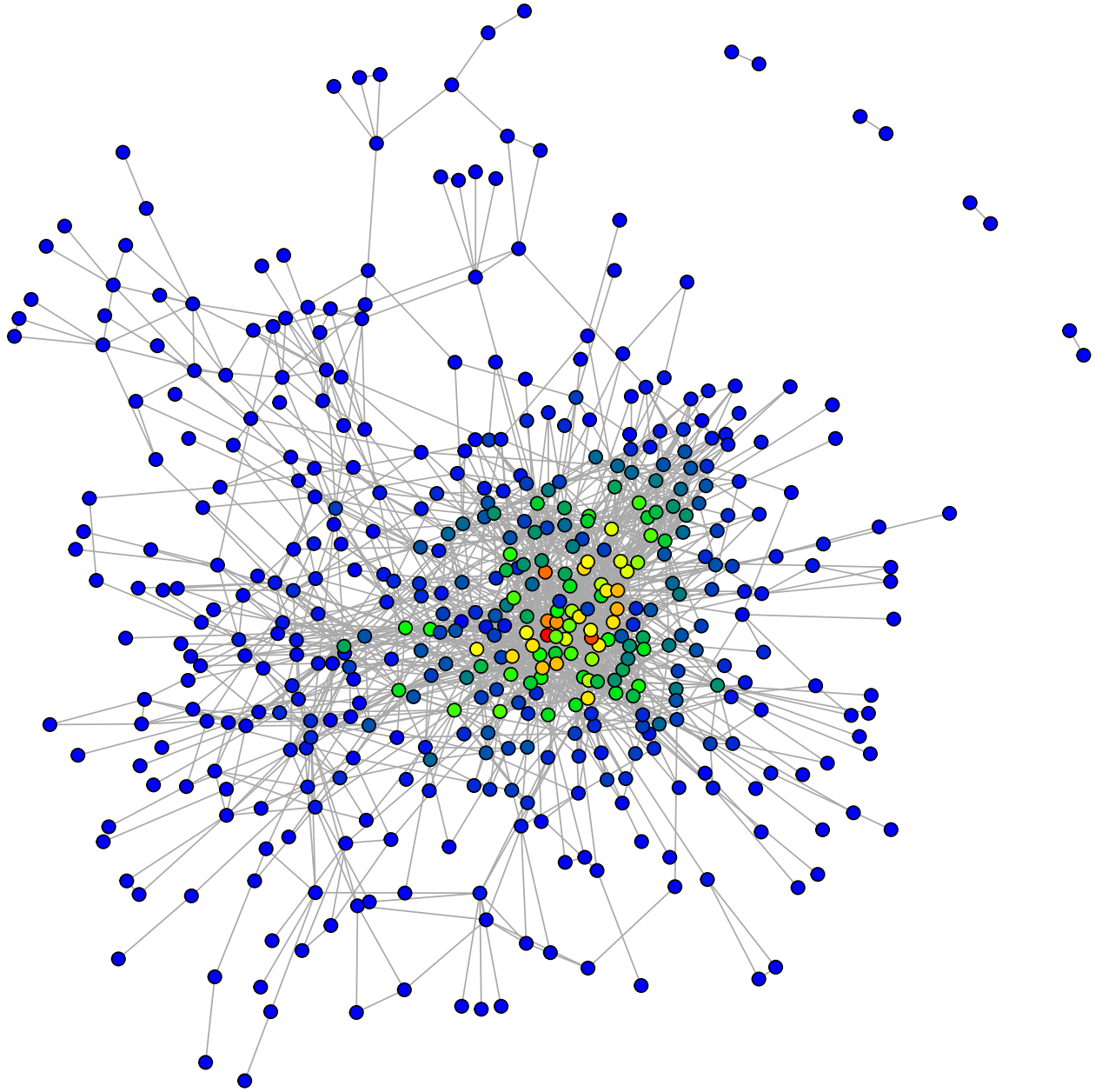


Figure 1: Eigenvector Centrality Heatmap for Erdos1 Network

the 43 connections of Vojtech Rodl. However, the eigenvector centrality shows that Rodl is in fact the more influential figure in this network because the individuals he is connected with are themselves more influential. The code below shows that in terms of eigenvector centrality, Graham's connections are less important than Rodl's. We find Graham's and Rodl's respective neighborhoods and then take the mean of their neighbor's eigenvector centrality.

```
graham_connections <- which(names(eigenValues) %in% neighbors(erdos1_network
                      ,"GRAHAM, RONALD LEWIS", mode = c("total"))$name)
# Find eigenvector centrality values for Graham neighborhood
mean(eigenValues[graham_connections])
```

```
## [1] 0.3475813
```

```
rodl_connections <- which(names(eigenValues) %in% neighbors(erdos1_network
                      ,"RODL, VOJTECH", mode = c("total"))$name)
# Find eigenvector centrality values for Rodl neighborhood
mean(eigenValues[rodl_connections])
```

```
## [1] 0.4117935
```

From the above output, it's clear that even though Rodl's degree is lower, his influence is larger in the network.

Alpha centrality calculation is not relevant for this network because the Erdos1 graph's edges are undirected.

### 4.4 Testing the Citation Network

To build this network, we first had to manually create a csv citation data set consisting of a directed edge list. We started with the paper titled *On Random Graphs* by Paul Erdos and Alfred Renyi. Using Google Scholar, we looked at all the papers which cited this work. Due to the large number of papers which cited this and subsequent papers, we decided to scale down this network. Since many of the papers on our list had between 500 and 3000 citations, we chose to scale down the data by a factor of 500. In other words, if paper A was cited 3000 times, we award it 6 edges to the top papers which cited it. By scaling the network this way, we were able to maintain the behavior of the data in a manageable way. The data set which we created is available in the included citationNetwork.csv file.

We create the directed citation network using this data in the following code.

```
citation_network <- graph_from_data_frame(d=citation_edges[,1:2],
                      vertices=citation_vertices, directed=T)
length(V(citation_network)) # Number of vertices in the graph
```

```
## [1] 81
```

```r
length(E(citation_network)) # Number of edges in the graph
```

```
## [1] 107
```

We see that the citation network consists of 81 vertices and 107 edges. In this network, the vertices represent each paper, and a directed edge represents a paper citing (or being cited by) another paper.

Now, we can apply our first model to this network. The following code returns the total degree for each vertex in the network. We then sort the results in decreasing order, and display the six vertices with the highest total degree.

```r
total_citation_degree <- degree(citation_network, v = V(citation_network),
                        mode = c("all"), loops = TRUE, normalized = FALSE)
head(sort(total_citation_degree, decreasing=TRUE))
```

```
##                          The structure and function of complex networks
##                                                                      38
##                               Complex networks: Structure and dynamics
##                                                                      20
##                                        Community detection in graphs
##                                                                      17
##                                         Social and economic networks
##                                                                      10
##                                     Nonparametric statistical inference
##                                                                       9
## Random graphs with arbitrary degree distributions and their applications
##                                                                       8
```

According to the total degree method, *The Structure and Function of Complex Networks (TSFCN)* is the most influential paper in the network, since it cites and it is cited the most times. However, as explained before, total degree does not take into consideration the influence of each citation, just the presence of one. To do this, we apply our second model which calculates eigenvector centrality for each vertex in the network, and display the top entries in decreasing order.

```r
eigen_values <- eigen_centrality(citation_network)$vector
head(sort(eigen_values, decreasing=TRUE))
```

```
##                          The structure and function of complex networks
##                                                                1.0000000
##                               Complex networks: Structure and dynamics
##                                                                0.7344580
##                                        Community detection in graphs
##                                                                0.3945499
##                                                       On random graphs
##                                                                0.3940448
```

```
##                                                          Evolution of networks
##                                                                      0.3927205
## Random graphs with arbitrary degree distributions and their applications
##                                                                      0.3012425
```

According to the eigenvector centrality, *TSFCN* is also the most influential paper in the citation network. A plot for the network is provided in Figure 2, where the vertices are colored based on their eigenvector centrality. A higher eigenvalue is mapped to a warmer color.

```
plot(citation_network, vertex.color=graphCol, edge.arrow.size=1,
     edge.size=NA,vertex.label=NA,vertex.size=5,margin=0)
```

*TSFCN* (shown in red) is connected to 38 papers which is the reason that it has a really large influence according to the total degree model. Because of the relative influence of these 38 papers, eigenvector centrality also finds this paper to be very important. However, we want to take into consideration the direction of these 38 edges to get a better gauge on the influence of the paper. *TSFCN* has 34 papers which cited it (in degree) and 4 papers which it cites (out degree). Alpha centrality, our third model, splits the influence *TSFCN* has over the four papers that it cites, giving those a higher influence.

The following code calculates the alpha centrality values for the citation network. We select the alpha parameter to be 0.5.

```
citation_alpha <- alpha_centrality(citation_network,
    nodes = V(citation_network), alpha = 0.5, loops = FALSE)
head(sort(citation_alpha, decreasing=TRUE))
```

```
##                                                              On random graphs
##                                                                      85.87500
##                                                          Evolution of networks
##                                                                      41.25000
##                                The structure and function of complex networks
##                                                                      38.16667
## Random graphs with arbitrary degree distributions and their applications
##                                                                      38.00000
##                                      Complex networks: Structure and dynamics
##                                                                      30.83333
##                                               Community detection in graphs
##                                                                       7.50000
```

According to the alpha centrality, *On Random Graphs* is the most influential paper in the network. Due to this, we can see it was colored red in Figure 3. This result makes sense because of the way in which we built the citation network. We expect *On Random Graphs* to be the most influential paper because it's the paper which we chose to build our network on. All papers in the network point to this work, both directly and indirectly.

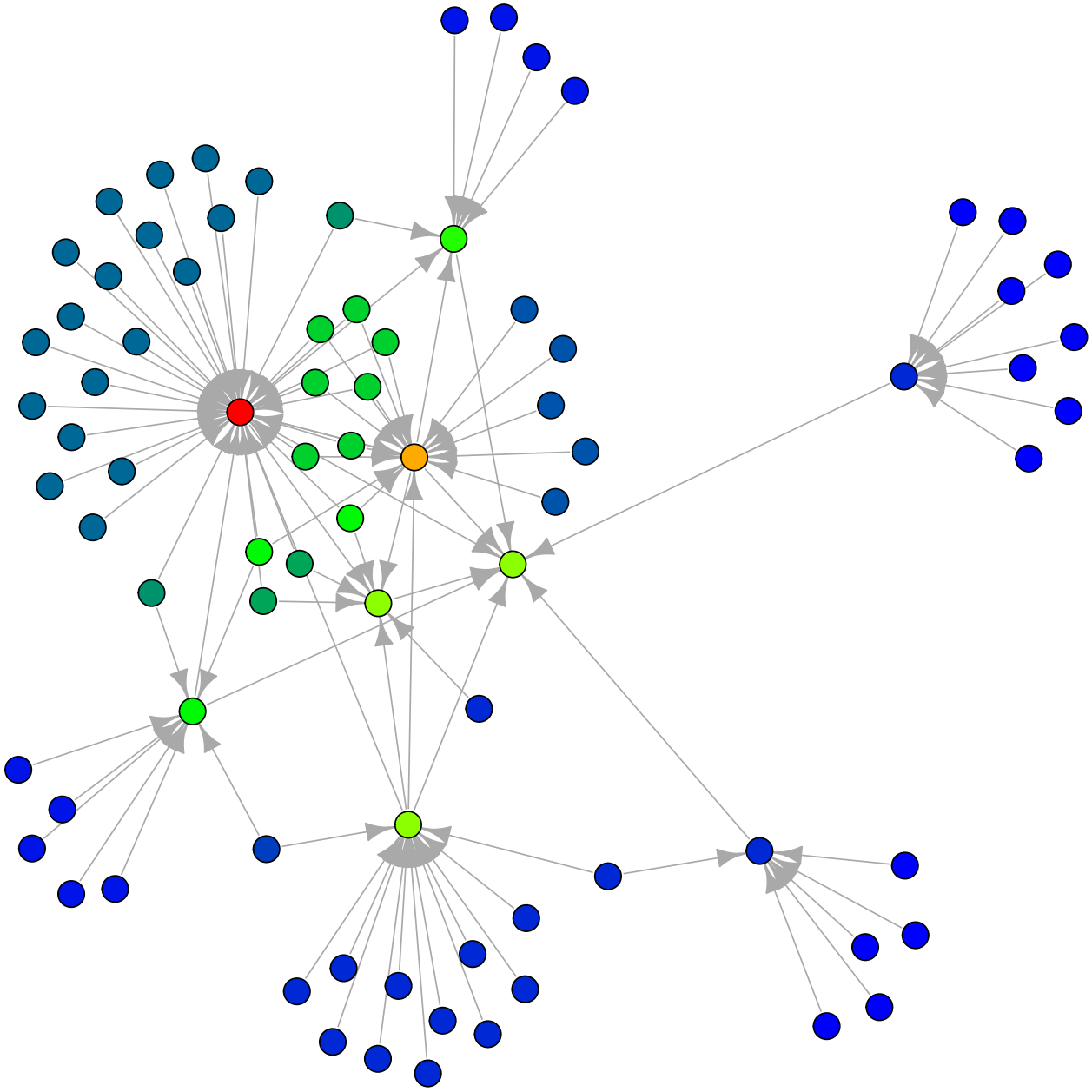


Figure 2: Eigenvector Centrality Heatmap for Citation Network

```
plot(citation_network, vertex.color=graphCol, edge.arrow.size=1,
     edge.size=NA,vertex.label=NA,vertex.size=5,margin=0)
```

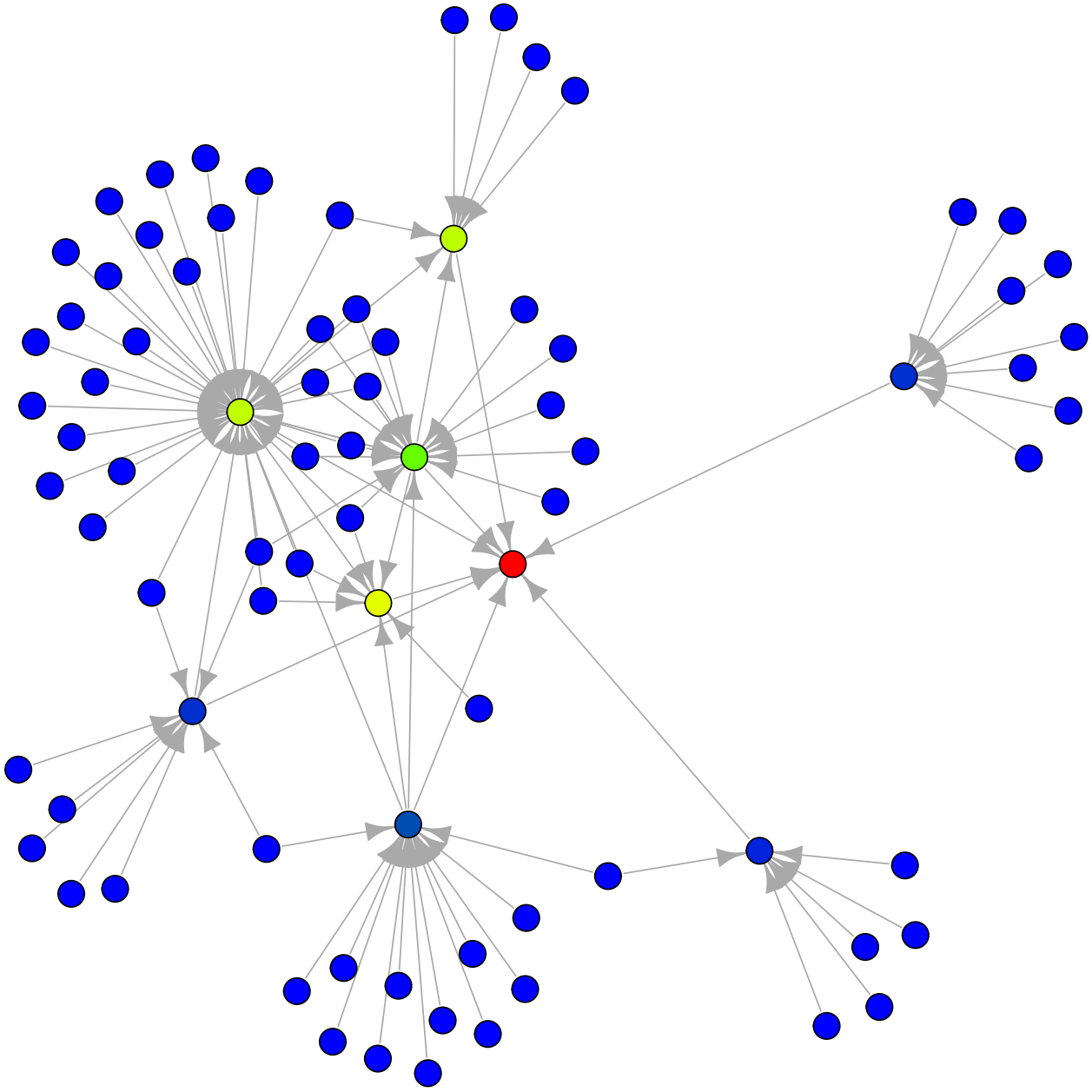


Figure 3: Alpha Centrality Heatmap for Citation Network

### 4.5 Testing the Actors Network

After importing and processing the data for this graph (omitted from our report), we begin by creating the actors collaboration network. The data we used is available at https://www.kaggle.com/sudheersankar/movie-metadata/data. This is a sizable data set

which contains entries about various modern movies. We clean this data to focus on just the actors who are collaborating in a certain film. The data set contains other interesting information such as critic ratings, Facebook likes, etc. which could be utilized in further analysis.

```
actorsNetwork <- graph_from_data_frame(d=actorCollabDF,
vertices=actorsNodes, directed=F)
length(V(actorsNetwork)) # Number of vertices in the graph
```

```
## [1] 6256
```

```
length(E(actorsNetwork)) # Number of edges in the graph
```

```
## [1] 29286
```

The number of vertices in the actor/collaborator network is 6256 and the number of edges is 29286. Figure 4 shows a visualization of this large network.

```
plot(actorsNetwork, vertex.color="cornflowerblue", edge.arrow.size=.001,
     edge.size=0.1,vertex.label=NA,vertex.size=2,margin=0)
```

Similar to the analysis of the Erdos co-author network, we start with a primary approach of determining each actor's influence: each node's degree represents that actor's influence.

```
totalDegree <- degree(actorsNetwork, v = V(actorsNetwork), mode = c("all"),
                      loops = TRUE, normalized = FALSE)
head(sort(totalDegree, decreasing=TRUE))
```

```
## Robert De Niro Morgan Freeman   Bruce Willis     Matt Damon  Steve Buscemi
##            200            172            148            148            140
##    Johnny Depp
##            132
```

Our calculation suggests that Robert De Niro has collaborated with 200 actors, Morgan Freeman has collaborated with 172 actors, etc. In this model, Mr. De Niro has the most influence within this network.

As mentioned before, the total degree model awards one centrality point for every link a node receives, but not all vertices are equivalent. In other words: just because a node has more connections, it does not mean that it has more influence. Instead, a node is important if it is linked to other important nodes, which is eigenvector centrality's thesis.

```
eigenValues <- eigen_centrality(actorsNetwork)
head(sort(eigenValues$vector, decreasing=TRUE))
```

```
## Morgan Freeman Robert De Niro      Brad Pitt   Bruce Willis    Johnny Depp
##      1.0000000      0.9949732      0.8605449      0.6409109      0.6276712
##   J.K. Simmons
##      0.6044213
```

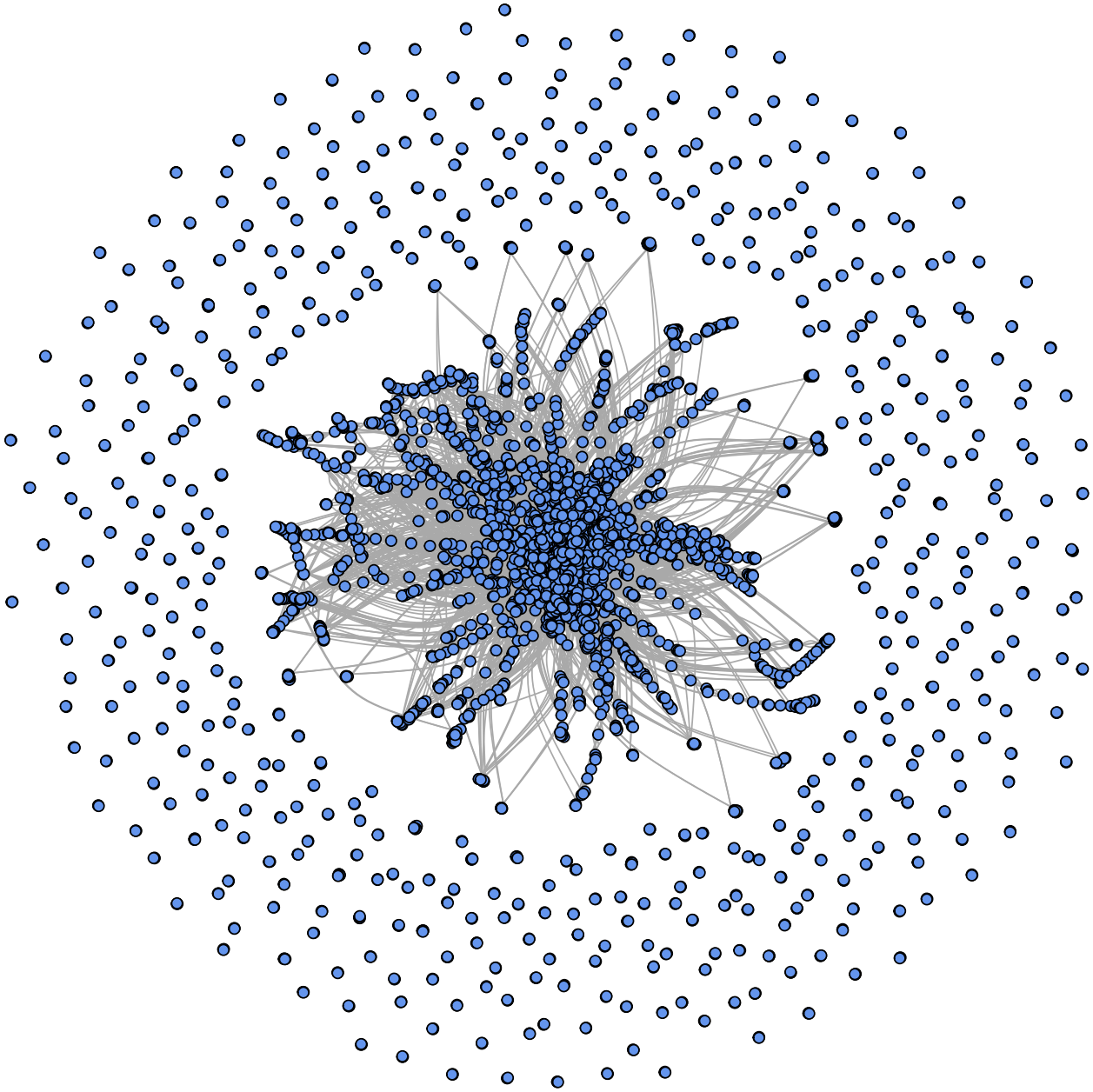


Figure 4: Actors Network

According to the result of our eigenvector centrality calculation, Morgan Freeman has the most influence with 1.0000000 followed by Robert De Niro with 0.9949732.

We can see that although Robert De Niro has collaborated with 28 more actors than Morgan Freeman, Morgan Freeman has a higher eigenvector centrality score which means Morgan Freeman is more influential than Robert De Niro. To justify this result, we find Freeman's and De Niro's respective neighborhoods and then take the mean of their neighbor's eigenvector.

```
MorgenFreeman <- which(names(eigenValuesNum) %in% neighbors(actorsNetwork,
"Morgan Freeman", mode = c("total"))$name)
# Find eigenvector centrality values for Freeman neighborhood
mean(eigenValuesNum[MorgenFreeman])
```

```
## [1] 0.2383885
```

```
RobertDeNiro <- which(names(eigenValuesNum) %in% neighbors(actorsNetwork,
"Robert De Niro", mode = c("total"))$name)
# Find eigenvector centrality values for De Niro neighborhood
mean(eigenValuesNum[RobertDeNiro])
```

```
## [1] 0.2042668
```

From this calculation, Morgan Freeman's collaborators has a higher average eigenvector centrality score, which means his collaborators, on average, are more influential than Robert De Niro's. This results in Morgan Freeman having a higher eigenvector centrality score although he collaborated with fewer actors.

Alpha centrality calculation is not relevant for this network because the Actor Collaborator graph's edges are undirected.

# 5. Results and Quality of the Models

## 5.1 Discussion on Erdos Co-author Network

Our entire analysis of this network relies on the assumption that an authors influence can be determined by who they have collaborated with. We applied the total degree model and the eigenvector centrality model to the Erdos1 co-author network. As previously shown, both models produced meaningful results. The authors which the models have selected to be most influential somewhat overlapped, but the order in which these authors were ranked changed slightly. This change in rank is due to the different calculations and assumptions which each model performs to determine influence.

The total degree model is primarily strong in terms of simplicity of computation and interpretation. The results simply show how many edges are adjacent to each vertex and we ordered these in descending order to see the most important authors. In the context of the total degree method, an author who has more co-authors arguably has more influence on his/her circle because of the increased number of interactions with researchers who could be

from different fields, backgrounds, nationalities, etc. However, the strength for this model can also arguably be its weakness. The assumption that more collaborators implies more influence may be an oversimplification of the real scenario. If an author collaborates 100 times with authors who only end up writing one academic work in their lives, this is arguably not as influential as an author who collaborated 50 times with well known academics.

Eigenvector centrality addresses this dilemma in a way which is applicable to our undirected co-author network. The advantage of this method is to take into consideration the importance of an author's collaborators when assigning influences. However, the complex computation in this method makes the results more difficult to interpret. The actual values resulting from the eigenvector centrality computation don't have much meaning in terms of the application. We know the result consists of numbers between 0 and 1, but all they tell us is the relative importance rankings of the authors. While this is the question we sought to answer, we get less supporting information and have to perform further analysis to determine why these rankings were assigned. This is in contrast to the clear results of the total degree model which simply returns the number of collaborators an author had.

It is worth noting that our results are based on a simple network in which we only know the presence of a collaboration between two authors. Having a value of the number of times which authors collaborated would have given us a more interesting analysis and more largely favored the Eigenvector centrality model.

### 5.2 Discussion on Citation Network

Our analysis of the Citation Network is based on two major assumptions. First, the influence of a paper can be determined by the number and importance of papers which cited it. Second, the scaling of the citation network maintains the behavior of the original network. We applied the total degree model, the eigenvector centrality model and the alpha centrality model to the citation network. As shown in the test section, because of some of the properties of the citation network, the alpha centrality model provided the most meaningful results.

While the total degree model results are simple to understand, they are not very meaningful in terms of the citation network. It does not take into consideration the influence of the papers being cited. It also does not consider the direction of the edges, so a paper that has cited 1000 papers will be as influential as a paper that has been cited 1000 times, which is not correct.

The eigenvector centrality model addresses the first problem by taking into consideration the relative influence of all the neighbor vertices of a paper. Unfortunately, it does not address the second issue, since eigenvector centrality is primarily used in undirected graphs, and the citation network is a directed graph.

The alpha centrality model accounts for all the problems, because it can handle directed edges. Unlike the eigenvector centrality model, in order to determine the influence of a paper, alpha centrality considers the influence of the papers which cites it. Because the citation network we used was built around 1 paper (*On Random Graphs*), we expected this paper to

have the highest influence in the network, and the results obtained with the alpha centrality model matched our expectations.

In the context of our testing, alpha centrality is the optimal influence measure for a directed network. Even though the results were accurate, the citation network used was a simple network. A more complex citation network would have allowed us to further test and demonstrate the effectiveness of the alpha centrality model in directed graphs.

### 5.3 Discussion on Actors Network

The discussion is based on the assumption that an actor's influence is affected by the actors who they have collaborated with. We analyzed this network with total degree model and the eigenvector centrality model. Both models show similar results but the ranking of the actors vary. This is because we have different definitions of influence in the models which result in different methods of calculation.

The total degree model definition of influence is simple – the more times an actor has collaborated with others, the more influential that actor is. Since each edge between two nodes represents a collaboration, the calculation is simply getting the total degree of each node. The total degree analysis falls short for two reasons:

1. If an actor has collaborated with few actors but for many times, that actor should not gain more influence but this model counts duplicate collaborations towards influence.
2. The definition is not accurate enough. In other words, it does not reflect the real world definition of influence.

To solve these problems we applied the the eigenvector centrality model. In this model, the definition of influence is more sensible to the real world – the more influential an actor's collaborators are, the more influential that actor is. Eigenvector centrality distributes influence among nodes through edges.

### 5.4 Overall Conclusions

Overall, the total degree model is an intuitive approach for a simple network. It has the simplest output to understand, as the degree of each vertex is easily relatable to the real world scenario which the network models. However, this approach fails to deliver meaningful results on more complex networks. This is because influence does not simply depend on the direct number of connections in a neighborhood. Rather, we must consider the influence of each neighbor, and the eigenvector centrality model shows this. We illustrate this through our example analysis of the Erdos1 co-author network and the Authors/Collaboration network. However, when considering a directed graph, the alpha centrality model provides the most meaningful results. This is demonstrated through our testing of the academic paper citation network.

## 6. Future Work

While our project covers three models for influence in a static snapshot of a network, an additional interesting investigation would be to see how accurate our influence metrics are for predicting future states of a network. Take for example our paper citation network. We determined that of our models, alpha centrality is the most comprehensive metric for determining influence of an academic work. Given more time and resources, it would be interesting to do the following:

1. Create a larger scale citation network
2. Run models on the network
3. After some time has passed (perhaps a year) create an updated version of the network, adding the new papers which cite already present papers in the network
4. Analyze changes in network and re-run metrics

Performing the above process would give us some interesting insight into how effective our centrality measures were in predicting the future behavior of a graph of this kind. Could we treat the value obtained from alpha centrality or eigenvector centrality as a "probability" that a new edge will be attached to a vertex? Intuitively we may assume that there is a relationship, but further testing and analysis would be required to have concrete evidence of this.

Another interesting future project would be to analyze networks which have more complex attributes. In many of our networks, we are analyzing simple edges and vertices. However, having more advanced metadata in terms of the edges and vertices could reveal interesting properties of the graphs and/or shortcomings of our models. We may be able to further validate the results of our graphs given a more in depth data set.

A final future goal would be to use the knowledge which we have gained from this project (and potentially a future course on graph theory) to build an original model to define influence in a network. Our report contains a comprehensive analysis of three leading approaches, but it would be interesting to revisit this problem given more experience and develop our own influence metric.

## 7. References

Bloch, Francis, and Matthew Jackson. 2016. "Centrality Measures in Networks." *SSRN Electronic Journal*. http://dx.doi.org/10.2139/ssrn.2749124.

Bonacich, Phillip, and Elena Paulette. 2001. "Eigenvector-Like Measures of Centrality for Asymmetric Relations." *Social Networks* 23 (3). https://doi.org/10.1016/S0378-8733(01)00038-7.

Csardi, Gabor, and Tamas Nepusz. 2006. "The Igraph Software Package for Complex

Network Research." *InterJournal*. http://igraph.org.

Giordano, Frank, William Fox, and Steven Horton. 2013. "A First Course in Mathematical Modeling 5th Edition." *Cengage Learning*.

Karwa, Vishesh, and Sonja Petrović. 2016. "Coauthorship and Citation Networks for Statisticians (Comment)." *ArXiv Archives*. https://arxiv.org/abs/1608.06667.

Osca-Lluch, Julia, Elena Velasco, Mayte López, and Julia Haba. 2009. "Co-Authorship and Citation Networks in Spanish History of Science Research." *Scientometrics* 80 (2). Springer Netherlands. https://doi.org/10.1007/s11192-008-2089-5.

Ruhnau, Britta. 2000. "Eigenvector-Centrality — a Node-Centrality?" *Social Networks* 22 (4). https://doi.org/10.1016/S0378-8733(00)00031-9.

Segarra, Santiago, and Alejandro Ribeiro. 2015. "Stability and Continuity of Centrality Measures in Weighted Graphs." *IEEE Transactions on Signal Processing* 64 (3). IEEE Signal Processing Society. https://doi.org/10.1109/TSP.2015.2486740.