An Investigation Into Modeling With Networks

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

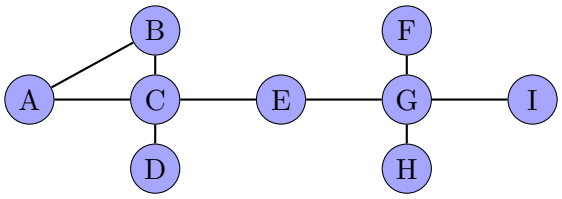
Philip Hossu, Paolo Ratti Tamayo, Jiateng Sun

April 13, 2018

---

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. A significant portion of this project focuses on the widespread influence of a famous mathematician, Paul Erdos, and his large co-author network.

In the following project report, we analyze how influence can be determined in a graph and describe our exploration into mathematical modeling with networks. 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 two critical measurements 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 does have 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 are 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 general social network analysis. We began by finding a modern survey of various graph centrality measures, notably the work by Bloch, et al. titled *Centrality Measures in Networks* [CITE]. 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 Santiago Segarra and Alejandro 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 [CITE].

Another work which we used to help shape our understanding of the problem was the comment released by Vishesh Karwa and Sonja Petrovic titled *Coauthorship and citation networks for statisticians* [CITE]. 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 simply implies the presence of these two authors collaborating at least once, but the exact number is unknown. We not able to give weights to our edges since our data set simply shows some form of collaboration. 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 [CITE]. The authors 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 edge thicknesses represent number of collaborations. To analyze the publication impact, 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 publication separately from the publishing journal.

Despite that our networks were based on much more simple data sets, it was interesting and useful to see how others have approached similar analyses of co-author and citation networks. This problem

is not totally novel, but it's apparently getting increasing amount of attention in recent years, 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 two primary metrics by which to determine influence in a graph. While these metrics are not entirely novel, we explain their formulation mathematically and conceptually.

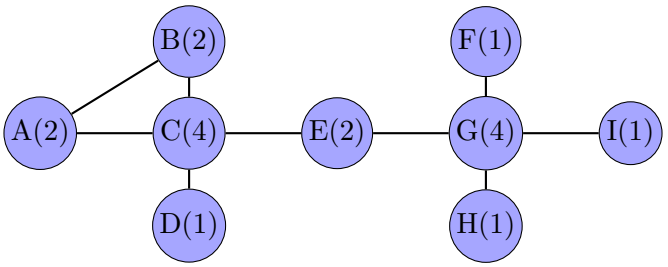
3.1 Total Degree

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 to define 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 or absence of an edge between the two corresponding vertices. The value in each entry can be modified to contain the edge weight, but this is not applicable in the networks we are analyzing. The corresponding adjacency matrix for our running example is shown below.

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

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 metric due to its simplicity and relevance in the example networks which we will consider. More reasoning will be present in the sections which follow, but a subset is mentioned here. The first network which we will analyze, the Erdos1 co-author network, is a simple undirected network which shows authors who have collaborated with the mathematician Paul Erdos. We can apply the total degree metric to help us determine influence of authors in this network because more collaborators will imply that an author has more influence. If a vertex (representing an author) has a higher degree, we could assume that their works are more diverse and their influence is greater. The second network which we will analyze, the citation network, is a directed network which shows academic papers which have cited each other. Our metric will also be relevant in this scenario, as a paper which has a higher degree will mean that more papers have cited it, making it more influential. The third network is a movie/actors network. Total degree has a solid meaning here too, as more connections (edges) will imply that the actor has worked with a larger number of other actors, meaning that they have more of an influence in their industry.

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 just 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 with [CITE].

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 λ and vectors $\mathbf{x}$ respectively that are the solutions to:

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

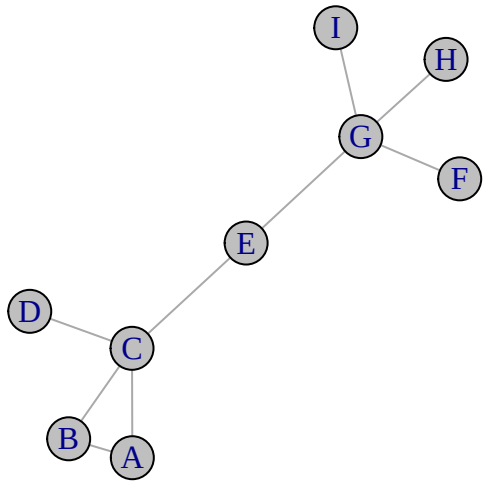
Now, applying this problem specifically to graph centrality, the eigenvector centrality is defined as follows: Given a graph $G = (V(G), E(G))$ and its corresponding adjacency matrix A, the score of

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.

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

We utilize the igraph package in R which provides an easy to use interface through which to perform this calculation. The following code block demonstrates the calculation 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")
```




```
# 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
##         I         H
## 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 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 quite different from the counting degrees 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 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 acts as a crucial bridge between the two pieces of the graph.

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 movie/actor network. However, when it comes to the citation network, we will use a third metric to determine the influence of a node. [CITE]

3.3 Alpha Centrality

Alpha centrality is an advancement to Eigenvector centrality to allow the metric to be applied to directed graphs which have asymmetric adjacency matrices. 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:

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

Alpha centrality starts by giving every vertex a starting, random positive amount of influence. Each node splits its influence evenly and divides it amongst its outward neighbors, receieving from its inward neighbors. This process repeats until the graph reaches a steady state, perhaps through power iteration.

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 spend some significant time collecting, cleaning, and importing 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 Dr. Paul 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 pseudocode 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 for download 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. However, our full code can be found at https://github.com/philiphossu/Modeling-With-Networks.

4.2 Exploring the Erdos1 Co-author Network

CHANGE THE HEADERS ON THIS CODE SEGMENT BEFORE RUNNING

Here, we create the Erdos1 co-author network after cleaning and importing the data. This network makes the most sense to be undirected since the data which we have 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)
length(V(erdos1_network)) # Number of vertices in the graph
```

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

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

```
## [1] 3325
```

We see that this network consists 474 of vertices and 3325 of edges – it's a relatively large network. It is worth noting that our network does not contain all 511 erdos1 authors because go through our data frame and remove all the edges which are not between Erdos1 authors as instructed.

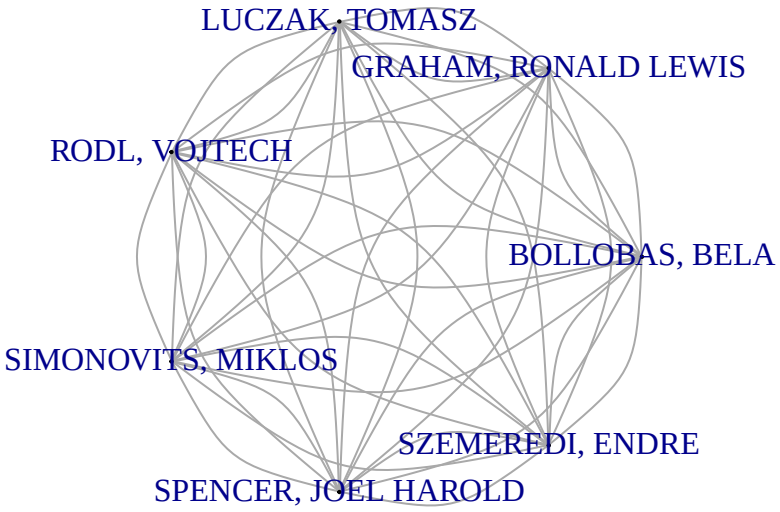
The first aspect of the graph we looked at is the presence of cliques. A clique is a subset of vertices where each vertex is connected to every other vertex. A simple example of a clique would be a triangle which has three vertices connected by three edges. In terms of our network, the real world meaning of a clique in our network 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 data set far larger than is valuable.

However, we can also find only the largest cliques present in the graph and plot one of these largest cliques.

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

```
## [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)
```



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

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

```
## + 11/474 vertices, named, from ca94ee1:
##  [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 interesting to look at. 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.

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

```
## [1] 0.02966075
```

Given our 3325 total edges and 474 vertices, the result of this calculation is correct. We see here that compared to the possible connectedness of the graph, the Erdos1 co-author network is fairly sparsely connected.

The last property we considered was the clustering coefficient in the graph. The clustering coefficient measures locally how connected the vertices are. The definition of clustering is fairly fluid as clustering algorithms take several parameters, but generally speaking clustering seeks to find groups of similar vertices.

```
transitivity(erdos1_network)
```

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

4.3 Testing the Erdos1 Co-author Network

4.4 Testing the Citation Network

4.5 Testing the Actors Network

5. Results and Quality of Model

6. References

To be completed at a later time: (APA)

Paper about calculating eigenvector centrality, why this measurement is stable, etc: https://arxiv.org/pdf/1410.5119.pdf

Paper with a whole bunch of graph metrics and explanation about them: https://arxiv.org/pdf/1608.05845.pdf

Paper about analyzing a co-author network of Spanish scientists: https://link.springer.com/article/10.1007/s11192-008-2089-5

Paper with IIT author regarding co-author and citation networks: https://arxiv.org/pdf/1608.06667.pdf

Paper about why eigenvector is not good for directed and definition of alpha centrality http://www.leonidzhukov.net/hse/2016/networks/papers/bonacich2001.pdf

7. Appendix