Contents

1 Introduction to Centrality Measures

Centrality measures are used to deduce the impact of different nodes in a graph. It is a way to measure how important one node is compared to another. There are many different measures of centrality due to the wide array of problems that centrality is used in. In our paper we will discuss four different types of centrality, including vertex (degree) centrality, closeness centrality, eigenvector centrality, and betweenness centrality. We will begin by defining these measures in terms of weighted and unweighted undirected graphs, then expand our investigation to include directed graphs.

Throughout our initial explorations of centrality and community structures, we will be focusing on four simple unweighted graphs:

1. The cycle graph on five vertices, C_5 shown below

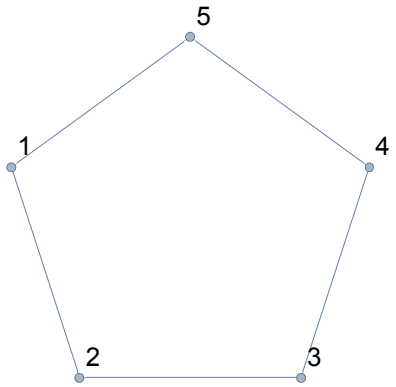


Figure 1: The cycle graph, C_5

2. The complete graph on five vertices, K_5

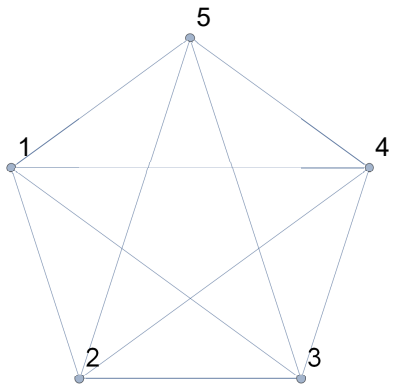


Figure 2: The complete graph, K_5

3. The five-node bow tie graph,

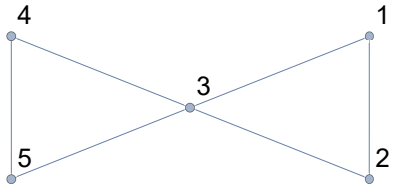


Figure 3: The bowtie graph, B_5

4. The seven-node bow tie graph

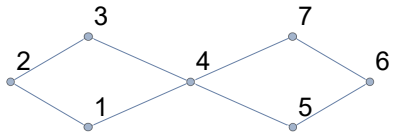


Figure 4: The bowtie graph, B_7

1.1 Degree Centrality

The degree centrality is defined as the degree of a specific node. This can vary depending depending on the type of graph in question.

Unweighted Graph Centrality For an unweighted graph, the degree centrality is defined as the number of links adjacent to a node. For example, the degree centrality of each vertex in the cycle graph of Figure (1) is $C_D(v) = 2$.

For an unweighted, directed graph, there are three distinctions of degree centrality:

1. In-Degree Centrality- Only the edges pointing towards the node are taken into account

2. Out-Degree Centrality - Only the edges pointing away from the node are taken into account

3. In-Out-Degree Centrality - All edges, regardless of direction, are factored into the measurement

Weighted Graph Centrality In a weighted graph, degree centrality is still only defined for the links adjacent to a node. However, degree centrality is now the sum of the weights of adjacent edges. For example, in Figure 5 of the weighted bowtie, $C_D(3) = 16$. The same distinctions of degree centrality also apply to a directed, weighted graph.

1. In-Degree Centrality- Only the edge weights of the links pointing toward the node are included in the sum

2. Out-Degree Centrality - Only the edge weights of the links pointing toward the node are included in the sum

3. In-Out-Degree Centrality - All edges, regardless of direction, are factored into the sum of the adjacent edge weights

Degree centrality is somewhat limited in its application because it only measures which nodes have the most connections. However, an example of its use is determining interactions in a social network. If Instagram's network was modeled by an unweighted, undirected graph, a high degree centrality would represent a person with a large amount of connections. Therefore, they interact with many people and are more central to the graph.

1.2 Betweenness Centrality

The *betweenness centrality* is a measure of centrality within a connected graph that quantifies how many shortest paths must pass through a particular node. For a given node v, $\sigma_{s,t}$ denotes the number of shortest paths between unique nodes s and t while $\sigma_{s,t}(v)$ denotes the total number of

shortest paths between s and t that pass through node v (Brandes 2001). The calculation of betweenness centrality is expressed as

$$C_B(v) = \sum_{v \neq s \neq t} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}.$$

A higher betweenness score for a node implies that the node acts as a bridge from one portion of the graph to another. For example, destinations within a city can be represented as nodes and roads connecting them can be represented as edges. A node with a high betweenness centrality is akin to a stop along a road that everyone was pass by to get to a certain location. By treating destinations within a city as nodes and roads as edges between them, one can, for example, begin to determine bottlenecks and busy roads by measuring betweenness centrality, as well as under-utilized roads.

Unweighted Betweenness Graph Centrality

The formula provided above does not define what we mean by "shortest path". In an unweighted graph, the shortest path is determined by counting the number of edges between two nodes. The path with the least amount of edges is the shortest. In Figure (4), for example, there are two shortest paths from node 3 to node 6 both of length 3. Here is a table that shows the calculations for node 1 in the B_5 graph from Figure (3):

s $\rightarrow$ t	$\sigma_{s,t}(1)$	$\sigma_{s,t}$	$\frac{\sigma_{s,t}(1)}{\sigma_{s,t}}$
$2 \rightarrow 3$	0	1	0
$2 \rightarrow 4$	0	1	0
$2 \rightarrow 5$	0	1	0
$3 \rightarrow 4$	0	1	0
$3 \rightarrow 5$	0	1	0
$4 \rightarrow 5$	0	1	0
		Total:	0

Since we are focusing on node 1, we do not include it in the paths we consider. Each pair of vertices considered only has 1 shortest path, and none of them pass through node 1. So, the sum of $\sigma_{s,t}$ is 6 and the sum of $\sigma_{s,t}(1)$ is 0. Therefore, since $\frac{0}{1} = 0$, $C_B(1) = 0$. Note that $C_B(i) = 0$ for $i = 2, 4, 5$ as well, due to symmetry.
We examine $C_B(3)$ to compare with node 1 since node 3 appears to be more central to the graph.

s → t	$\sigma_{s,t}(3)$	$\sigma_{s,t}$	$\frac{\sigma_{s,t}(3)}{\sigma_{s,t}}$
1 → 2	0	1	0
1 → 4	1	1	1
1 → 5	1	1	1
2 → 4	1	1	1
2 → 5	1	1	1
4 → 5	0	1	0
		Total:	4

As we can see, node 3 has a much higher betweenness score and is acting like a bridge for the graph much more than node 1 is.

Weighted Graph Betweenness Centrality

Cite THIS? From? Take the example of the bowtie graph of five vertices, except this time let's apply weights to each of the edges connecting the nodes. This is more likely to resemble real-world situations in which certain nodes have stronger connections with each other than with others, such as the strength of friendships in a group of friends. The process of calculating the betweenness centralities of the nodes in a weighted graph is conceptually similar to the procedure for unweighted graphs.

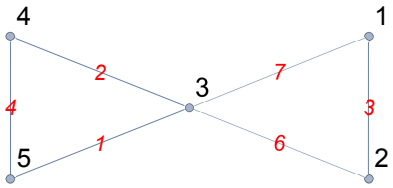


Figure 5: Weighted bowtie graph with 5 nodes, $B_W(5)$

To calculate betweenness centrality for a weighted graph, we do not need to change our process too much. We just need to modify our interpretation of the shortest path from one node to another. Also, instead of calculating $\frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$ like we did previously, we will just specify yes or no as to whether or not the node v is contained in the path from s to t. In a weighted graph, the shortest path between to node has the lowest or highest sum of edge weights (depending on the problem's context). Take the example of the bowtie graph of five vertices, except this time let's apply weights to each of

the edges connecting the nodes.

Let us examine the table below where we evaluate the betweenness centrality of $B_W(5)$ where higher weight is better:

s $\rightarrow$ t	Thru 1	2	3	4	5
1 $\rightarrow$ 2	N	N	Y	N	N
1 $\rightarrow$ 3	N	N	N	N	N
1 $\rightarrow$ 4	N	N	Y	N	N
1 $\rightarrow$ 5	N	N	Y	Y	N
2 $\rightarrow$ 3	N	N	N	N	N
2 $\rightarrow$ 4	N	N	Y	N	N
2 $\rightarrow$ 5	N	N	Y	Y	N
3 $\rightarrow$ 4	N	N	N	N	N
3 $\rightarrow$ 5	N	N	N	Y	N
4 $\rightarrow$ 5	N	N	N	N	N
Total:	0	0	5	3	0

So, as we can see, measuring betweenness of weighted graphs will produce somewhat different numbers because the edge weights are being taken into account. Based on these calculations, node 3 is the most central to this graph with node 4 close behind.

1.3 Closeness Centrality

sThe *closeness centrality* is defined as the reciprocal of the farness measure of a node (Bavelas 1950). More specifically, it is calculated by averaging the shortest path distances from a node to all other nodes.

Unweighted Graph Closeness Centrality

For unweighted graphs, basic closeness centrality is notationally defined as

$$C_C(v) = \frac{1}{\sum_x d(v,x)},$$

where we are measuring node v's centrality. The shortest distance from node v to node x is represented as $d(v,x)$.

This function for closeness centrality gets us close to what we need, but it is common to normalize these measurements across the graph by multiplying by $N-1$, which is one less than the total number of nodes, N, in the graph.

This makes it easier to compare centrality measures between different nodes and other graphs as well. So, the formula is slightly modified to now be

$$C_C(v) = \frac{N-1}{\sum_x d(v,x)}$$
$$\approx \frac{N}{\sum_x d(v,x)} \text{ (as N becomes large).}$$

Here is a table that shows the calculations for node 3 in the B_5 graph from Figure (3):

From → To	Shortest Distance
3 → 1	1
3 → 2	1
3 → 4	1
3 → 5	1
Total:	4

Since $N = 5$ for this graph, $N - 1 = 4$. Also, the sum of $d(x, y)$ will be 4 as well. So, we get the closeness measure by evaluating $\frac{4}{4}$, which equals 1. Thus, the closeness measure for node 3 in this graph is 1. The measurements for closeness centrality of this graph are summarized at the end of the chapter.

Weighted Graph Closeness Centrality

This method is only slightly modified when dealing with weighted graphs. In the case of weighted edges, we will follow a similar approach as explained in the section about betweenness centrality to determine a shortest path between two nodes. This means that $d(v, x)$ is calculated by adding up edges' weights for the paths that are between nodes v and x (Brandes 2001). In our case, we will consider that lower edge weight is better. When dealing with flow problems, for example, edge weights can represent resistance. So, the lower the weight the better.

We will now calculate closeness for node 3 and 1 of the weighted graph to compare their measures of centrality. The process is very similar to unweighted closeness, but to calculate the shortest paths we must sum the edges and determine which path is best. For $B_W(5)$, higher is better so the table will look like this:

From → To	"Shortest" Distance
$3 \to 1$	9
$3 \to 2$	10
$3 \to 4$	5
$3 \to 5$	6
Total:	30

From → To	"Shortest" Distance
$1 \to 2$	13
$1 \to 3$	9
$1 \to 4$	14
$1 \to 5$	15
Total:	51

So, since $N = 5$, we can calculate node three's closeness to be $\frac{4}{30} \approx .133$. Node one's closeness is $\frac{4}{51} \approx .078$. This suggests that node three is more central to the graph in terms of closeness. This process works similarly for the cases when lower weight is better. When determining the meaning of the closeness measure, one must rely heavily on the context of the problem at hand. Depending on the situation, a higher closeness centrality will equate to the node being more central, while in other situations a lower closeness measure will equate to a node being more central.

1.4 Eigenvector Centrality

The *eigenvector centrality* is the measure of how a node influences a graph (Newman 2010). This means that not only are direct connections taken into account, but a node's influence depends upon how connected and influential the nodes connected to it are. A simple approach to calculating the eigenvector centrality of the nodes of a graph is to convert the graph to an adjacency matrix and find its eigenvalues and corresponding eigenvectors. Take the eigenvector corresponding to the largest eigenvalue and scale it so the maximum value is 1. Take the dominant eigenvector and divide its entries by the maximum entry. Each entry of this calculated vector corresponds to each node's centrality measure. For example, entry three of the vector corresponds to the centrality measure for node three. Here is an example of the process using the graph shown in Figure (3):

We will first construct the adjacency matrix of B_5:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

The eigenvalues are

$$\{2.56155, -1.56155, -1, -1, 1\},$$

which correspond respectively to the eigenvectors

$$\left\{ \begin{bmatrix} 1 \\ 1 \\ 1.56 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ -2.56 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

The largest positive eigenvalue is 2.56155. So, we will take the eigenvector corresponding to this eigenvalue, sometimes referred to as the "dominant eigenvector."

The result is

$$\begin{bmatrix} 0.18 \\ 0.18 \\ 0.28 \\ 0.18 \\ 0.18 \end{bmatrix}.$$

Consequently, the eigenvector centrality measure for node three, for example, is 0.28. It is the highest which indicates that this node has the most influence in this graph.

Eigenvector centrality was used to build the original model for search engines to find influential web pages. Google's PageRank algorithm is built upon this concept and is used to make sure that only relevant and deeply-linked web pages are served to users.

1.5 Katz Centrality

The *Katz centrality* is an improvement upon the aforementioned eigenvector centrality. In most directed networks, the in-degree of a node is much more relevant than its out-degree. Katz centrality resolves issues that arise when calculating eigenvector centrality on directed graphs with nodes that have no incoming edges (Newman 2010). Normally, these nodes would have an eigenvalue centrality of zero, but this can skew the centrality measure of the nodes it points to. A node that is pointed to by many of these "zero" nodes will have a very low eigenvector centrality, even though they are important and pointed to by many other nodes. Katz centrality fixes this issue by giving a "free" centrality value to all nodes, regardless of their degree (Newman 2010). Therefore, we define the Katz centrality of node i to be

$$x_i = \alpha \sum_j A_{ij} x_j + \beta,$$

where α and β are adjustable parameters.

Even though it was originally designed to fix problems discovered with eigenvector centrality, Katz centrality can be used on undirected graphs as well. As Newman (2010) says, "It allows a vertex that has many neighbors to have high centrality regardless of whether those neighbors themselves have high centrality." Referring back to the equation to calculate Katz centrality, it can be rewritten in matrix notation as

$$\vec{x} = \alpha A x + \beta(\vec{1}).$$

As Freeman proposed in his essay, β is usually set to 1. Setting β to 1 and solving for $\vec{x}$ we obtain

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

He also proposed that $\alpha < \frac{1}{\lambda}$ where λ is the highest positive eigenvalue of the adjacency matrix of the graph. This is due to a few different reasons. The role of α is to provide a "...balance between the eigenvector term and the constant term" (175). If α was very large, the centrality measurements would be skewed. The β term would be insignificant and we would obtain essentially a scaled eigenvector centrality. If α becomes too small, the β parameter would take over and all nodes would have relatively the same centrality. Finally, if $\alpha = \frac{1}{\lambda}$, the centralities diverge. This occurs because the matrix $I - \alpha\ A$ is not invertible at this value of α; its determinant is 0. So, we choose an α such that $\alpha < \frac{1}{\lambda}$, but very close to this value.

Unweighted and Weighted Katz Centrality

Katz centrality is calculated the same for weighted and unweighted graphs. The only difference is the entries of the adjacency matrix. So, we will only show one example of a Katz centrality calculation. Recall the bowtie graph on five vertices in figure 4. Using the matrix equation above, we will solve for its Katz centrality. The largest positive eigenvalue for its adjacency matrix was $\lambda = 2.56$. Therefore, α must be less than $\frac{1}{2.56}$. We will then let $\alpha = .352$ to obtain

$$(I - 0.352A)^{-1}\vec{1} =$$

$$\begin{pmatrix} 8.77 \\ 8.77 \\ 13.34 \\ 8.77 \\ 8.77 \end{pmatrix}.$$

This vector is each node's Katz centrality. This is summed up in the table below.

Node	Measure
1	8.77
2	8.77
3	13.34
4	8.77
5	8.77

These values make sense because node 3 is obviously more central to the graph and should have a higher value. Also, since the graph is undirected, nodes 1, 2, 4, and 5 are all symmetric. So, their centralities should all match. Katz centrality once again fixed issues of eigenvector centrality, but it has issues of its own. One major issue of Katz centrality can be described by Newman (2010):

"If a vertex with high Katz centrality has edges pointing to many others then those others also get high centrality...One could argue that this is not always appropriate...For instance, the famous Yahoo! web directory might contain a link to my webpage, but it also has links to millions of other pages. Yahoo! is an important website, and would have high centrality by any sensible measure, but should I therefore be considered very important by association?"

Newman (2010) goes on to argue that the answer is no. This is because in certain contexts, like Internet links, websites' centrality is arbitrarily inflated if they have one link to a few important sites like Yahoo!. However, Yahoo! has links to millions of sites. Since Yahoo! points to so many different websites, Freeman argues that its influence on each of these websites' centrality should be diluted, or distributed. This Problem is solved by implementing PageRank centrality.

1.6 PageRank Centrality

To address the issue presented in the previous section, Katz centrality can be modified to aid in the dilution of high centrality scores across insignificant nodes (Newman 2010). Concretely, nodes that have a high out-degree will have their influence distributed equally over all their targets. In practice, the amount of centrality an adjacent node would receive from a high-centrality node would be the origin-node's centrality divided by the number of recipients (represented by the out-degree of the origin-node). The result is that all nodes adjacent to a high-centrality node will only receive a portion of that node's centrality.

This centrality measure is known as Google's PageRank algorithm, which was designed to deliver the most relevant content to users based on both users' search queries as well as the respective *scores* of the pertinent websites. As described Newman (2010), for a network of n websites with adjacency matrix A, the PageRank of each website is stored in the vector $\vec{x}$ below:

$$\vec{x} = (I - \alpha AD^{-1})^{-1}\beta\vec{1}, \tag{1}$$

In the above equation, D is the $n \times n$ diagonal matrix containing the out-degrees of each node along the main diagonal. For nodes that have an out-degree of zero, we artificially set that entry in the diagonal matrix to 1. This guarantees that the matrix D is invertible. **INCLUDE NEWMAN EQUATION** β is a scalar that is conventionally set to 1 and is multiplied by the n-dimensional $\vec{1}$ vector. Lastly, α is some scalar between 0 and 1 that is strictly less than the inverse of the largest eigenvalue of AD^{-1}. Newman reports that Google has traditionally used $\alpha = .85$ for the algorithm, although the reasoning behind this is not well understood (177). For the sake of the demonstration below, we will follow Google's lead and let $\alpha = .85$.

Let's revisit B_5, the bowtie graph on five vertices. Recall that we already constructed its adjacency matrix in the section on Eigenvector Centrality.

In order to create the diagonal matrix D, we are going to sum the entries in each column of A and then populate each diagonal entry with these sums representing each node's out-degree. Thus,

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

Computing D^{-1} is easily done by taking the reciprocal of each entry along the diagonal. Substituting D^{-1} along with I_5 and the adjacency matrix corresponding to B_5 into equation 1, we obtain

$$\vec{x} = \begin{bmatrix} 1 & -.425 & -.2125 & 0 & 0 \\ -.425 & 1 & -.2125 & 0 & 0 \\ -.425 & -.425 & 1 & -.425 & -.425 \\ 0 & 0 & -.2125 & 1 & -.425 \\ 0 & 0 & -.2125 & -.425 & 1 \end{bmatrix}^{-1} \cdot \beta\vec{1}.$$

Simplifying the above expression, we find

$$\vec{x} = \begin{bmatrix} 5.67 \\ 5.67 \\ 10.64 \\ 5.67 \\ 5.67 \end{bmatrix}.$$

Recall that the vector $\vec{x}$ hosts the PageRank of each node, where the entry in the first row corresponds to the PageRank of the first node and follows down the rows chronologically. We observe that node 3 has the highest PageRank centrality, corroborating what we have previously concluded using other centrality measures. The PageRank algorithm will generally compute more reliable centrality scores given that it takes more factors into account throughout the process.

1.7 Weighted Graphs

All the centrality measures we discussed are very useful. Each one summarizes a node's importance in different aspects. However, we have only talked about these in terms of unweighted graphs. When it comes to most applications, graphs are weighted. This means that one node's connection to a

neighbor could be stronger than that node's connection to another neighbor. For example, a person on facebook has two different friends. Typically, they will not interact with each one equally. The person may like friend one's photos more frequently than the other. Therefore, their connection is stronger with the friend than friend two. The edge connecting the person and friend one should be more important than the edge connecting the person and friend two. Hence, it will have a larger weight. In other contexts, like flow, edge weights represent resistance. Lower edge weight is better here. The context of a problem will let us distinguish whether low or high edge weight is desired. We will explain a few of the centrality measure previously described above in terms of weight graphs. The first is betweenness centrality.

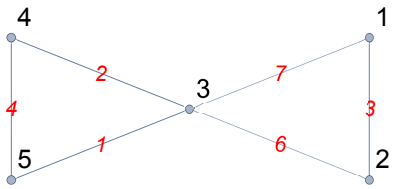


Figure 6: Weighted bowtie graph with 5 nodes, $B_W(5)$

To calculate betweenness centrality for a weighted graph, we do not need to change our process too much. We just need to modify our interpretation of the shortest path from one node to another. Also, instead of calculating $\frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$ like we did previously, we will just specify yes or no as to whether or not the node v is contained in the path from s to t. In a weighted graph, the shortest path between to node has the lowest or highest sum of edge weights (depending on the problem's context). Take the example of the bowtie graph of five vertices, except this time let's apply weights to each of the edges connecting the nodes.

Let us examine the table below where we evaluate the betweenness centrality of $B_W(5)$ where higher weight is better:

s → t	Thru 1	2	3	4	5
1 → 2	N	N	Y	N	N
1 → 3	N	N	N	N	N
1 → 4	N	N	Y	N	N
1 → 5	N	N	Y	Y	N
2 → 3	N	N	N	N	N
2 → 4	N	N	Y	N	N
2 → 5	N	N	Y	Y	N
3 → 4	N	N	N	N	N
3 → 5	N	N	N	Y	N
4 → 5	N	N	N	N	N
Total:	0	0	5	3	0

So, as we can see, measuring betweenness of weighted graphs will produce somewhat different numbers because the weights are being taken into account. Based on these calculations, node 3 is the most central to this graph with node 4 close behind.

We will now calculate closeness for node 3 and 1 of the weighted graph to compare their measures of centrality. The process is very similar to unweighted closeness, but to calculate the shortest paths we must sum the edges and determine which path is best. For $B_W(5)$, higher is better so the table will look like this:

From → To	"Shortest" Distance
3 → 1	9
3 → 2	10
3 → 4	5
3 → 5	6
Total:	30

From → To	"Shortest" Distance
1 → 2	13
1 → 3	9
1 → 4	14
1 → 5	15
Total:	51

So, since $N = 5$, we can calculate node three's closeness to be $\frac{4}{30} \approx .133$. Node one's closeness is $\frac{4}{51} \approx .078$. This suggests that node three is more central to the graph in terms of closeness. This process works similarly for the cases when lower weight is better. LEAVE THIS OPEN ENDED. SAY THIS DEPENDS ON THE CONTEXT OF THE PROBLEM BEING SOLVED

CAN WE TAKE THIS WEIGHTED SECTION OUT SINCE THEY ARE EXPLAINED EARLIER?

1.8 Summary of Centrality Measures for Unweighted Graphs

We will now give centrality measures for Figures (1), (2), (3), (4), and (1.2).

- C_5

Measure of Centrality	Node 1	2	3	4	5
Degree	2	2	2	2	2
Betweenness	1	1	1	1	1
Closeness	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$
Eigenvector	0.2	0.2	0.2	0.2	0.2
Katz 3.34	3.34	3.34	3.34	3.34	
PageRank	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$

  In any cycle graph, each node will have the same centrality measures as each other node. This is due to the symmetry of the graph and indicates that no one node has more influence than any other.

- K_5

Measure of Centrality	Node 1	2	3	4	5
Degree	4	4	4	4	4
Betweenness	0	0	0	0	0
Closeness	1	1	1	1	1
Eigenvector	0.2	0.2	0.2	0.2	0.2
Katz	-2.47	-2.47	-2.47	-2.47	
PageRank	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$	$.\overline{66}$

  Since the K_5 graph is completely connected, every node will have the same measurements. This means in terms of centrality that each node is just as central, or influential, to the graph as any other node.

- B_5

Measure of Centrality	Node 1	2	3	4	5
Degree	2	2	4	2	2
Betweenness	0	0	4	0	0
Closeness	$.\overline{66}$	$.\overline{66}$	1	$.\overline{66}$	$.\overline{66}$
Eigenvector	0.18	0.18	0.28	0.18	0.18
Katz	8.73	8.73	13.26	8.73	8.76
PageRank	5.67	5.67	10.64	5.67	5.67

In the case of the bowtie graph on five nodes, node three is going to claim the largest centrality measures of all the other nodes.

- B_7

Measure of Centrality	Node 1	2	3	4	5	6	7
Degree	2	2	2	4	2	2	2
Betweenness	2	0.5	2	10	2	0.5	2
Closeness	$.\overline{54}$	0.29	$.\overline{54}$	0.75	$.\overline{54}$	.29	$.\overline{54}$
Eigenvector	0.14	0.11	0.14	0.22	0.14	0.11	0.14
Katz	6.57	5.61	6.57	10.23	6.57	5.61	6.57
PageRank	5.9	6.02	5.9	11.03	5.9	6.02	5.9

As can be seen in the table, node 4 is the most central node of all. Since it is in the "middle" of the graph, it is connected and close to all other nodes, very influential, and acts as a bridge between one side of the graph and the other. Thus, all of its centrality measures will be very high.

1.9 One Size Does NOT Fit All

It is easy to fall into the trap of measuring every type of centrality when analyzing a graph. S.P. Borgatti (2005), discussed the issue of thinking that all measures can be applied in every context. In fact, centrality is not context-free. Let us first define a few keywords needed before discussing centrality assumptions. A *trail* is a sequence of incident links in which no link is repeated. A *path* is a sequence in which both links and nodes are not repeated. A *walk* is an unrestricted sequence. In short, "all paths are trails and trails are walks, but not every walk is a trail and not every trail

is a path." (Borgatti 2005). A geodesic path is a shortest path between two nodes.

Borgatti (2005) also describes three major typologies for flow of information through a graph:

1. Parallel Duplication - Information can be passed simultaneously along multiple effected edges within a graph. Examples of this include: Internet DNS server, email broadcast, and attitude influencing.

2. Serial Duplication - Information is passed from one node to another without the chance of it being passed to itself again later. Information is replicated and passed on while some is maintained by previous node in sequence. Examples of this include: mitotic reproduction, viral infection, gossip, and emotional support.

3. Transfer - Information is an indivisible object that can only be owned by one node a time. Information travels along trails, so links are not repeated usually. Examples of this include: package delivery and money or goods exchange.

Borgatti (2005) provides us with these tables that describe different examples along with which measures of centrality that were found to be relevant:

	Parallel Duplication	Serial Duplication	Transfer
Geodesics	–	Mitotic reproduction	Package delivery
Paths	Internet DNS Server	Viral infection	Mooch
Trails	E-mail broadcast	Gossip	Used goods
Walks	Attitude influencing	Emotional support	Money exchange

Now we can see which centrality measures match up to these processes. Note that much of the second table is missing. Borgatti (2005) also argues that most of the sociologically interesting processes cannot be accurately measured using the major measures of centrality. He made special mention of the fact that are still no measures appropriate for measuring infection and gossip processes, which could be very important and an area of future study.

	Parallel Duplication	Serial Duplication	Transfer
Geodesics		Closeness	Closeness Betweenness
Paths	Closeness Degree		
Trails	Closeness Degree		
Walks	Closeness Degree Eigenvector		

1.10 Conclusion

Academics and researchers are more in less in agreement with respect to the definitions of these centrality measures and the processes through which they are calculated as they pertain to undirected, unweighted graphs. These centrality measures are critical tools in evaluating the influence and importance of nodes in a graph, and they generally complement each other to create a more robust picture of the connections within a graph. Throughout the chapter, we attempted to show how certain centrality measures are relevant to different problems and used illustrative examples to demonstrate their strengths and shortcomings. For example, the degree centrality of a node does not take into account the influence or importance of the nodes to which it is connected. In summary, these measures are useful in the analysis of small-scale graphs. However, as we will see shortly, the lines begin to blur as the graphs become larger and more intricate, as real world networks tend to be.

2 Modularity and Community Finding

2.1 Introduction

Graph modularity is a measure used to determine the quality of detected communities within a network. Newman (2010) defines it as a measure of the extent to which "like is connected to like" within a network. Communities are partitions of a graph that move nodes into separate groups based on some sort of agreed characteristic such as node-link connectivity or edge weights. For example, in a social network, it is likely for communities to arise among people who attend the same schools, work at the same companies, or share the same social circles. Therefore, a partition of a network that best defines these communities will have a higher modularity then a partition that does not preserve these communities. Modularity can be used as an objective function that can be optimized to find the best number of communities at the highest quality of modularity within each community.

2.2 Examples of Current Algorithms

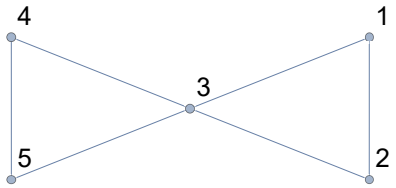
Current algorithms have been proposed by many researchers such as Clauset, Newman, Moore, Pons, Latapy, Wakita, and Tsurumi. All of these work relatively well at maximizing modularity in a decent amount of time. Problems arise as the number of nodes increases to that of hundreds of thousands of nodes and beyond, which is common today, these algorithms become too slow. For example, the Arxiv network's maximum modularity was calculated by CNM in 3.6 seconds. Here is a table of performances from 2 on various network sizes to illustrate the issue (Blondel et. al., 2008):

	Arxiv	Internet	Web nd.edu	Web uk-2005	Web WebBase 2001
Nodes/Links	9k/24k	70k/351k	325k/1M	39M/783M	118M/1B
CNM	.772/3.6s	.692/799s	.927/5034s	-/-	-/-
PL	.757/3.3s	.729/575s	.895/6666s	-/-	-/-
WT	.761/0.7s	.667/62s	.898/248s	-/-	-/-
Louvain	.813/0s	.781/1s	.935/3s	.979/738s	.984/152mn

It is obvious to see that Louvain's algorithm is much more robust in that it can achieve a higher modularity in each scenario, but also finish in a fraction of the time. We will be exploring this algorithm in combination with Newman's definition of modularity in the next section.

2.3 Modularity Algorithm

In order to illuminate how modularities of individual communities are computed, we will experiment with several different partitions of the unweighted bowtie graph on five vertices, B_5, pictured below:



To do these calculations, we will introduce a slight modification to **Newman's** formula for computing modularity. Newman (2010) described the modularity of an unweighted network according to a predetermined community partitioning as

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(i,j)$$

where m represents the total number of edges within the graph, A_{ij} contains the connections between nodes i and j, $k_i k_j$ is the product of the degrees of nodes i and j, and $\delta(i,j)$ determines whether or not nodes i and j belong to the same community. More specifically, $\delta(i,j) = 1$ when i and j belong to the same community and equals 0 otherwise. Newman's formula is described in terms of summations, which can have limitations. However, his formula can be expressed quite effectively in terms of matrices, which are excellent tools for containing vast quantities of data. The equivalent formula in terms of matrices is

$$Q = \frac{1}{2m} \operatorname{Tr} \left[S^\intercal B S \right]. \tag{2}$$

For an unweighted network, m is similarly defined as the number of edges within a graph. S is a matrix whose dimensions are determined by the number of nodes (rows) in a network and the number of communities (columns) defined by a specific partitioning. The construction of S is dependent on the predetermined partitioning of the network. Therefore, if node i belongs to community k, we would put a 1 in the ik^{th} entry. If node i does *not* belong to community k, we would place a 0 in that spot. B is the modularity matrix equivalent to $A_{ij} - \frac{k_i k_j}{2m}$. To find B, we subtract the

probability matrix, whose ij^{th} entry contains the probability $\frac{k_i k_j}{2m}$ that there is an edge connecting nodes i and j, from the adjacency matrix. Computing the modularity of a network according to a specific partitioning should yield a number between -1 and 1. The modularity of a network with the optimal partitioning will be greater than all the other modularities, while still being less than 1. We will describe how to construct S and B using the example of the unweighted bowtie graph on five vertices.

We will first construct the modularity matrix, B. The ij^{th} entry of the adjacency matrix again will contain a 1 if there exists an edge between nodes i and j. The entries of the probability matrix are found by computing the product of the degrees of nodes i and j and dividing that number by $2m$, or the number of edge ends, or "stubs", within the network. For example, the probability that nodes 1 and 3 are connected by an edge is $(2 * 4)/12 = 2/3$. Thus, the modularity matrix B of the bowtie graph on five vertices is

$$B = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 1/3 & 1/3 & 2/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 2/3 & 1/3 & 1/3 \\ 2/3 & 2/3 & 4/3 & 2/3 & 2/3 \\ 1/3 & 1/3 & 2/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 2/3 & 1/3 & 1/3 \end{bmatrix}$$

$$= \begin{bmatrix} -1/3 & 2/3 & 1/3 & -1/3 & -1/3 \\ 2/3 & -1/3 & 1/3 & -1/3 & -1/3 \\ 1/3 & 1/3 & -4/3 & 1/3 & 1/3 \\ -1/3 & -1/3 & 1/3 & -1/3 & 2/3 \\ -1/3 & -1/3 & 1/3 & 2/3 & -1/3 \end{bmatrix}.$$

To find the optimal partitioning, we conventionally assume there are no community structures within the network and assign each node to its own community. In that situation, the matrix S would look like

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Once this "base" modularity is computed, we begin to shrink the number of communities by grouping nodes into the same community and testing to see if this grouping improves the overall modularity of the network.

Substituting all these matrices into equation (2) will tell us the modularity of B_5 for which each node belongs to its own community. We obtain a modularity of $-.22$, which indicates that it is unlikely that the above partitioning effectively reflects the actual communities within the network. Recall that once you compute the modularity matrix B, the only adjustment that needs to be made to calculate the modularities of other partitions is to the S matrix. Observe what happens to the matrix S when we group nodes 1 and 2 within the same community. We obtain the matrix

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

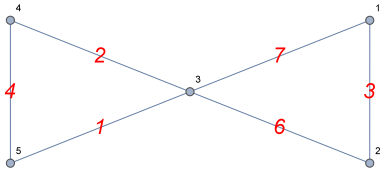
which, upon entering S into the modularity formula, would yield a modularity of $-.11$. This number is greater than the modularity calculated when each node belonged to its own community, suggesting that the above partitioning is superior to our initial partitioning. Going forward, we would continue to group nodes into the same community to see how the modularity is effected at each iteration. If we were to continue with the calculations, we would find that the optimal partitioning of the network occurs when nodes 1,2, and 3 belong to the same community and when 4 and 5 belong to a community.

This would correspond to the following S matrix:

$$S = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

This partitioning produces a modularity of .11. Given the simplicity and the symmetry of the bowtie graph on five vertices, it is unsuprising that it is difficult to define communities within the network. As the graph grows larger and more complicated, however, calculating the modularity of different partitionings of the network can offer useful information as to the connectivity of the network and who the principal players are. In the following section, we will explore what happens when we add edge weights to the connections between two nodes. See the Table on Summary of Modularities to compare how different partitions of B_5 result in different modularities.

We can modify our algorithm slightly in order to use it on weighted graphs. The steps will be the same; we will group nodes together in different communities until we obtain an optimized partition based on modularity. Only our interpretation of m and k_i will change. In the unweighted graph calculation, m was defined as the total number of edges and $k_i k_j$ was the product of the degree of nodes i and j. For a weighted graph, m is the the sum of all the edge weights of the graph and k is the sum of the edge weights surrounding a node. So, $k_i k_j$ is the product of edge weights surrounding nodes i and j. The entires of the adjacency matrix will also contain the actual edge weights between nodes. We will show an example using a weighted version of the five node bowtie graph, displayed below.



The adjacency matrix of this weighted bowtie is

$$A = \begin{bmatrix} 0 & 3 & 7 & 0 & 0 \\ 3 & 0 & 6 & 0 & 0 \\ 7 & 6 & 0 & 2 & 1 \\ 0 & 0 & 2 & 1 & 4 \\ 0 & 0 & 1 & 4 & 1 \end{bmatrix}.$$

Now that we have the graph'adjacency matrix, we can calculate the modularity matrix, B. By the assigned weights of the bowtie, $m = 23$. The matrix B is defined the same as in the previous unweighted bowtie calculation. It is the difference between the adjacency matrix and the probability matrix, whose entries are the probabilities $\dfrac{k_i k_j}{2m}$. This is shown below.

$$B = \begin{bmatrix} 0 & 3 & 7 & 0 & 0 \\ 3 & 0 & 6 & 0 & 0 \\ 7 & 6 & 0 & 2 & 1 \\ 0 & 0 & 2 & 1 & 4 \\ 0 & 0 & 1 & 4 & 1 \end{bmatrix} - \begin{bmatrix} 100/46 & 90/46 & 160/46 & 60/46 & 50/46 \\ 90/46 & 81/46 & 144/46 & 54/46 & 45/46 \\ 160/46 & 144/46 & 256/46 & 96/46 & 80/46 \\ 60/46 & 54/46 & 96/46 & 36/46 & 30/46 \\ 50/46 & 45/46 & 80/46 & 30/46 & 25/46 \end{bmatrix}$$

$$= \begin{bmatrix} -100/46 & -48/46 & -162/46 & -60/46 & -50/46 \\ 48/46 & -81/46 & 132/46 & -54/46 & -45/46 \\ 162/46 & 132/46 & -256/46 & -4/46 & -34/46 \\ -60/46 & -54/46 & -4/46 & -36/46 & 154/46 \\ -50/46 & -45/46 & -34/46 & 154/46 & -25/46 \end{bmatrix}.$$

From this point on, the steps are the same as before. We start by assuming that each node is in its own community. So, S_1 is the identity matrix of five dimensions. We apply our algorithm for modularity, which is $Q = \frac{1}{2m} \operatorname{Tr}[S^\intercal BS]$. The modularity for this partition is $-.24$, which is very close to what we had for this grouping in the unweighted graph as well. The succeeding steps entail just redefining the S matrix to match the community distribution being investigated. In the end, the optimal modularity value is obtained from two communities. Nodes 1 and 2 are in the same community and nodes 3, 4, and 5 are in the same community. No other partitions give us a greater modularity value. This is the same optimal grouping obtained by the unweighted graph as well.

2.4 Louvain's Modularity Algorithm

Louvain's algorithm is a two phase process in which each is iterated repeatedly.

We start in phase one with a graph of N nodes. Each node is its own community. Within the first phase each node, i, is iterated over and its neighbors are examined for the change in modularity if node i were to be placed in its neighbors' communities. The grouping that produces the maximum positive modularity gain is performed. If no positive gain exists, then node i remains in its current community. This will quickly start partitioning nodes into like communities. This stops once no individual can be moved to increase modularity. The change in modularity is calculated with the given

formula, which is expressed by Blondel et. al (2008):

$$\Delta Q = \left[\frac{\sum_{in} + k_i^C}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right],$$

where $\sum_{in}$ is the sum of the weights of the links inside of a community C, k_i^C is the sum of the weights of the links from i to nodes in C, $\sum_{tot}$ is the sum of the weights of the links incident to the nodes in C, k_i is the sum of the weights of the links incident to node i, and m is the sum of the weights of all the links in the network. **MAKE A WALKTHROUGH FOR THIS!!**

In phase two, a new network is built who nodes are now the communities found in the previous phase. To accomplish this, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Links between nodes within the same community lead to self-loops for this node in the new network. We can then apply the steps in the first phase on this new network and repeat the phases until modularity can no longer increase.

Louvain listed several advantages of this algorithm over previous ones:

1. The steps are intuitive and easy to implement.
2. The outcome is unsupervised (no human input) and therefore not biased.
3. The algorithm is extremely fast; can be completed in linear time.
4. Past algorithms have trouble finding small communities due to resolution problems. This algorithm does not have this problem due to its bottom-up approach.

The algorithm works, or can be adapted, for directed and undirected graphs. It works for weighted and unweighted graphs. In an unweighted graph, one would treat every edge as having a weight of 1.

2.5 Louvain's Algorithm for Directed Graphs

For directed graphs, the behavior is exactly the same. There are only a few changes that need to be made to the algorithm. These changes were

described by Nicolas Dugue and Anthony Perez (2015).For the undirected case,

$$\begin{aligned}\Delta Q &= \left[\frac{\sum_{in}+k_i^C}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \\ &= \frac{k_i^C}{2m} - \frac{\sum_{tot}\cdot k_i}{2m^2}.\end{aligned}$$

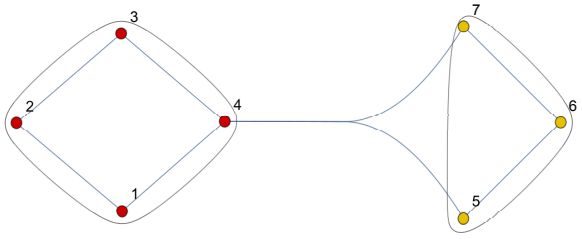
This will slightly change to the following for the directed case:

$$\Delta Q_d = \frac{k_i^C}{m} - \left[\frac{k_i^{out}\cdot\sum_{tot}^{in}+k_i^{in}\cdot\sum_{tot}^{out}}{m^2}\right]$$

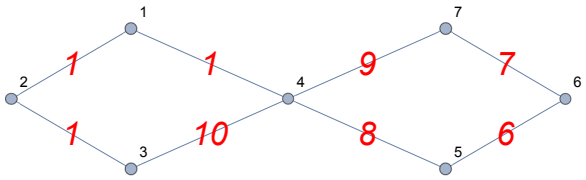
where $\sum_{tot}^{in}$ and $\sum_{tot}^{out}$ represents the number in- and out-going edges respectively in regards to community C. Essentially, this change now accounts for the fact that not only a relationship exists between two nodes, but also the direction of that relationship. The same process holds true when applying Louvain's algorithm for directed graphs.

2.6 Comparing Effects of Weighted Edges

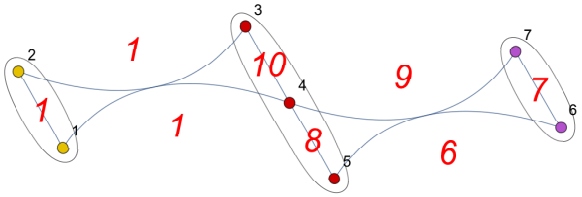
We will now observe the effect of weighted edges in hard community detection and modularity. In Figure (4), we have the unweighted seven node bowtie graph. Using the process described in this section, we can determine that the graph must be split into two communities to achieve a maximum modularity of 0.21875.



Now, we will apply weights to the edges in this contrived example to prove our point.



We determined the modularity of this graph is 0.108978. The modularity was cut in half, but, due the weights of the edges, it was more appropriate to split the network into three communities. This must be done to maintain the strong relationships.



2.7 Summary of Modularities for Unweighted B_5

Partition	Modularity
{1},{2},{3},{4},{5}	-.22
{1,2},{3},{4},{5}	-.11
{1,3},{2},{4},{5}	-.17
{1,2,3},{4},{5}	0
{1,2},{3,4},{5}	-.056
{1,2},{3},{4,5}	0
{1,2},{3,5},{4}	-.056
{1,2,3,4},{5}	-.056
{1,2,3,5},{4}	-.056
{1,2,3},{4,5}	.11

2.8 Soft-Community Finding and SVD

Singular value decomposition is a method to soft-partition networks through matrix factorization. Soft-partitioning is a useful tool in network community detection because it, unlike hard-partitioning, allows for the existence of overlapping communities. Recall that when we hard-partition a network, we assign each node to exactly one community. This can be effective when

trying to understand the general connectivity of a network, but presents challenges when trying to partition a network comprised of nodes that belong equally to more than one community. We saw this when attempting to optimize the modularity of the bowtie graph on five nodes; the center node was torn between the communities formed by each half of the bowtie. Although the bowtie graph is a relatively trivial example, especially when considering the complexity of most real-world networks, it is revealing of the drawbacks of hard partitioning. This is where soft-partitioning and singular value decomposition comes in handy.

Matrix factorization lends itself to a wide range of applications, in large part because the operation is not exclusive to square matrices. In general terms, when we define the singular value decomposition of a matrix A, we factor A into the product of three matrices: $A = U\Sigma V^T$. Consequently, when A is an $m \times n$ matrix, U is necessarily an $m \times m$ matrix, Σ is an $m \times n$ matrix, and V^T is an $n \times n$ matrix. Note that U and V^T are always square matrices. This process translates almost seamlessly when applied to networks, but with a slight modification which we will discuss momentarily. For the sake of comprehension, however, we will demonstrate how the singular value decomposition is executed using the example of the 3×2 matrix

$$A = \begin{bmatrix} 7 & 1 \\ 0 & 0 \\ 5 & 5 \end{bmatrix}.$$

It is fitting to discuss matrix Σ first, as it hosts the singular values of matrix A. Since matrix A has a rank of 2, the first 2 diagonal entries of Σ will contain the singular values of A, which are found by taking the square root of the positive eigenvalues of the square matrix A^TA. In this case, those eigenvalues are $\lambda_1 = 90$ and $\lambda_2 = 10$ (disregarding the third eigenvalue of A^TA, $\lambda_3 = 0$), corresponding to singular values $\sigma_1 = \sqrt{90} = 3\sqrt{10}$ and $\sigma_2 = \sqrt{10}$. The remaining entries are populated with zeros. Thus,

$$\Sigma = \begin{bmatrix} 3\sqrt{10} & 0 \\ 0 & \sqrt{10} \\ 0 & 0 \end{bmatrix}.$$

The matrix U is formed by computing the normalized eigenvectors of the square matrix A^TA and assigning them to the columns of U. We will include the eigenvector corresponding to the eigenvalue $\lambda_3 = 0$ because we want

the columns of U to form a basis of $\mathbb{R}^3$. This implies that the vectors of U are orthonormal. Matrix V^T is constructed by finding the normalized eigenvectors of the square matrix AA^T, taking the transpose of each vector and assigning each transposed vector to the rows of V^T. Observe that the row vectors of V^T are likewise orthonormal. As a result,

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}, \text{ and } \quad V^T = \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix}.$$

Finally, the singular value decomposition of matrix A is

$$A = \begin{bmatrix} 7 & 1 \\ 0 & 0 \\ 5 & 5 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} 3\sqrt{10} & 0 \\ 0 & \sqrt{10} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix}.$$

Recall the matrix A is a rank 2 matrix, since it has at most two linearly independent vectors. Considering the rank of a matrix that is being factored into its singular value decomposition can be useful, particularly when analyzing large matrices. In many situations, we may want to simply give an approximation of a matrix of rank r by describing it as the sum of matrices $\sigma_1 \vec{u_1} \vec{v_1}^T + \sigma_2 \vec{u_2} \vec{v_2}^T + \cdots + \sigma_j \vec{u_j} \vec{v_j}^T$, where $j \leq r$. We would call this sum a rank j approximation of a matrix A. It is possible that a rank j approximation gives a pretty good approximation of matrix A, even when j is significantly smaller than r.

In the world of networks, we need to introduce a special type of matrix called the Laplacian, denoted L. The Laplacian is a modification to the adjacency matrix of a graph, made necessary by the need to compute eigenvalues and eigenvectors of the matrix A^TA and AA^T. Were we to try to factor the adjacency matrix of a network, we would find that A^TA and AA^T are not invertible matrices, due to the zero entries along the main diagonal. Thus, we construct the Laplacian matrix by adding the adjacency matrix of a graph, A, with the matrix D. Matrix D hosts the total degrees of each node along the main diagonal and contains zeros everywhere else.

For this exploration, consider the bowtie graph on seven nodes. The Lapla-

cian matrix corresponding to B_7 would look like

$$L = \begin{bmatrix} 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 4 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 2 \end{bmatrix}.$$

$L^T L$ and LL^T are both invertible matrices, and thus will pose no problem when computing eigenvalues and eigenvectors.

Suppose wanted a rank 2 approximation of matrix L. Thus, we will only be using the two largest eigenvalues to compute our singular values and the matrices U, Σ, and V^T. Allowing Mathematica to do the heavy-lifting, we observe that the matrices U, Σ, and V^T are

$$U = \begin{bmatrix} -.28 & .35 \\ -.16 & .5 \\ -.28 & .35 \\ -.79 & 0 \\ -.28 & -.35 \\ -.16 & -.5 \\ -.28 & -.35 \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} 5.41 & 0 \\ 0 & 3.41 \end{bmatrix},$$

$$V^T = \begin{bmatrix} -.28 & -.16 & -.28 & -.79 & -.28 & -.16 & -.28 \\ -.35 & .5 & .35 & 0 & -.35 & -.5 & -.35 \end{bmatrix},$$

resulting in the rank 2 approximation

$$L \approx \begin{bmatrix} .85 & .85 & .85 & 1.21 & 0 & -.35 & 0 \\ .85 & 1 & .85 & .71 & -.35 & -.71 & -.35 \\ .85 & .85 & .85 & 1.21 & 0 & -.35 & 0 \\ 1.21 & .71 & 1.21 & 3.41 & 1.21 & .71 & 1.21 \\ 0 & -.35 & 0 & 1.21 & .85 & .85 & .85 \\ -.35 & -.71 & -.35 & .71 & .85 & 1 & .85 \\ 0 & -.35 & 0 & 1.21 & .85 & .85 & .85 \end{bmatrix}.$$

Clearly, a rank 2 approximation is not the best approximation we can get, but its advantage lies in the fact that the size of matrices U, Σ, and V^T

have smaller dimensions than those of a higher rank approximation. This has important applications in data storage and image compression.

While a rank 2 singular value decomposition of L has shortcomings, it opens the door for another exploration: cosine similarity. Constructing the cosine similarity matrix is what will ultimately allow us to discern soft community structures within the network. The cosine similarity matrix is constructed by find the matrix products $U\Sigma$ and $V\Sigma$ (note here that we are using matrix V, which is simply the transpose of V^T) and then taking the dot products .....

3 Real Examples of Network Analysis

Now, we will use the topics and ideas we have explored earlier in the paper to analyze three applications of network analysis. These examples will each take a unique approach. The goal will be to interpret the results of the various measurements for the reader.

3.1 Grand Valley's Sidewalk Network

Grand Valley's campus can easily be represented as a network of its sidewalks. We collected this dataset ourselves by identifying the main exits to buildings and using map software to find the distance between buildings using yards as our units. So, the nodes are the buildings on campus and the edges are the sidewalks that directly connect various buildings. To simplify the network, even though you can technically walk from Mackinac Hall straight to the Kirkhof Center without entering a single building along the walk, these nodes would not be directly connected, because the walker would have to pass by many other buildings to get there. The major rule of thumb when creating the dataset was: if a walker leaving building A must pass another building, B, to get to the destination, C, then A and C should not be directly connected. Sometimes this rule is unclear because the setup of buildings on campus is a bit convoluted.

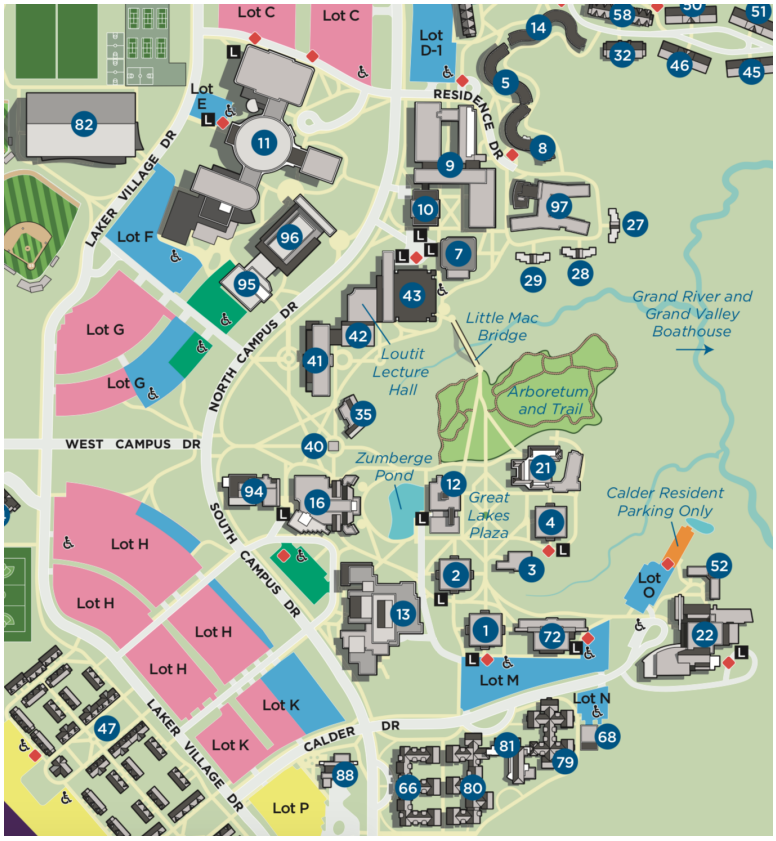


Figure 7: Grand Valley's Allendale Campus

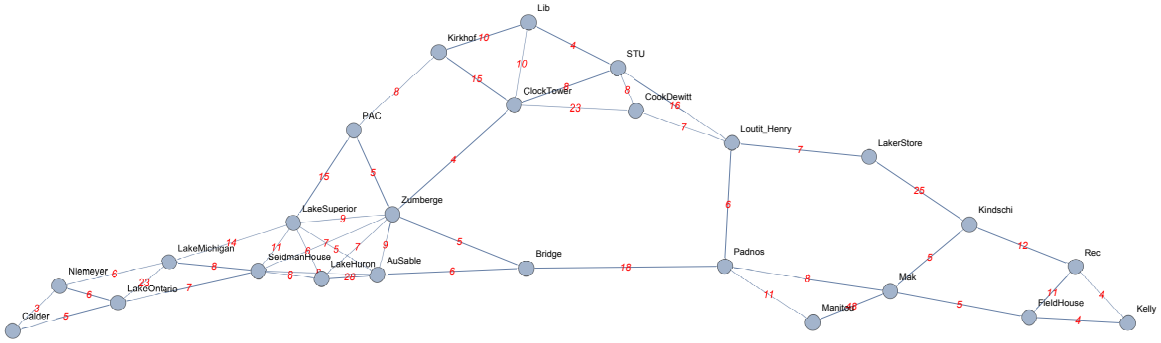


Figure 8: Grand Valley's Allendale Campus Graph with inverse edge labels scaled by 1000 and rounded for easier display. A higher number represents a stronger relationship.

Figure (7) shows Grand Valley's Allendale Campus[1]. It is cropped to display only the classroom buildings. The network analysis consists only of those nodes. By using the distance mapping software along the sidewalks and creating a weighted edgelist, we get the following representation in Figure (8).

For this network, a strong relationship between two nodes is defined as the inverse of the distance between them. We will use inverse weights for measures of centrality and modularity to accurately find the most central nodes and the most modular communities. We will use the normal distances as weights when we begin to find the shortest path between certain nodes and finding the shortest tour through the graph.

Centrality
All previously discussed measures of centrality were calculated for this network with the exception of Katz centrality. We were unable to find an implemented method for calculating Katz for weighted graphs within Mathematica. Through observing these major measures of centrality, we can conclude Mackinac Hall is a very influential place on campus (its eigenvector and PageRank centralities are the highest). Also, based on degree, closeness, and betweenness centralities, we can conclude that Zumberge is very central to the network in terms of physical location.

Degree Centrality

[1] https://www.gvsu.edu/homepage/files/pdf/maps/allendale.pdf

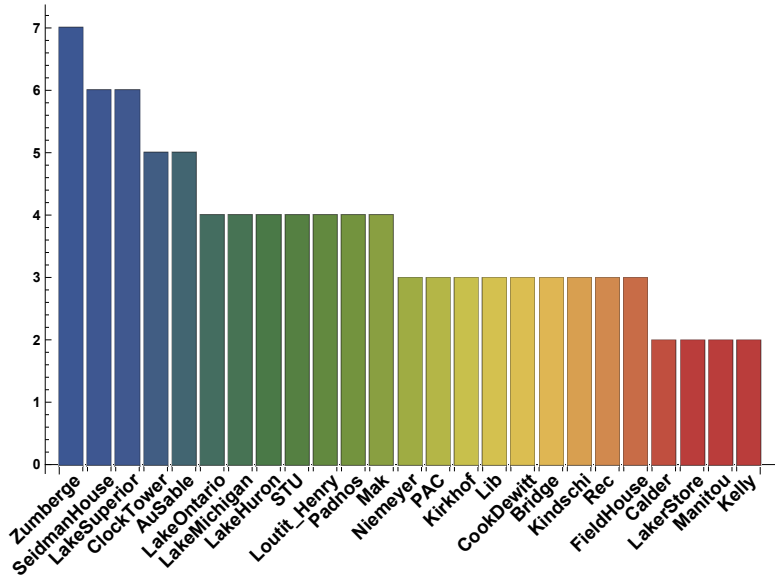


Figure 9: Bar chart of node degree centrality

Closeness Centrality

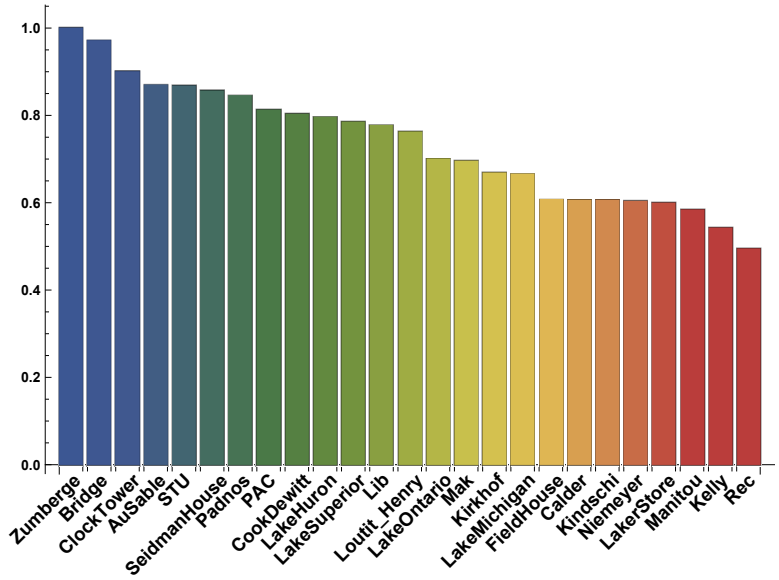


Figure 10: Bar chart of node closeness centrality

Betweenness Centrality

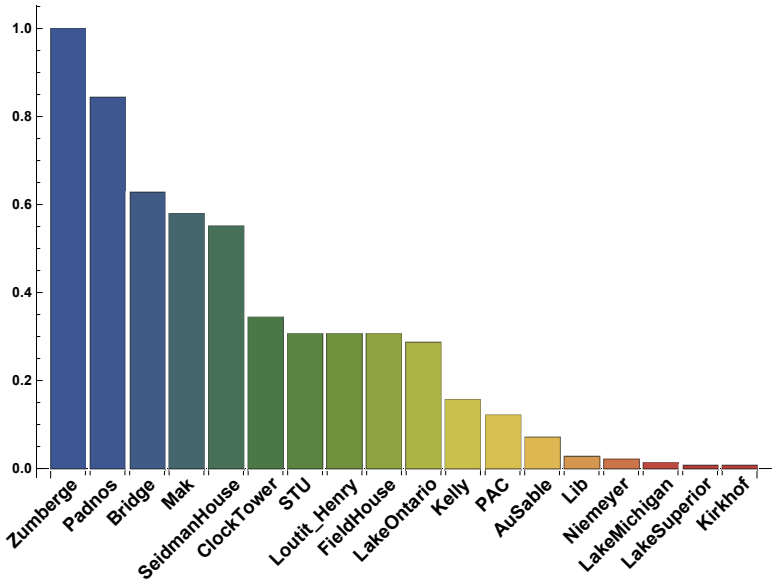


Figure 11: Bar chart of node betweenness centrality (nodes with values of zero are omitted.)

Eigenvector Centrality

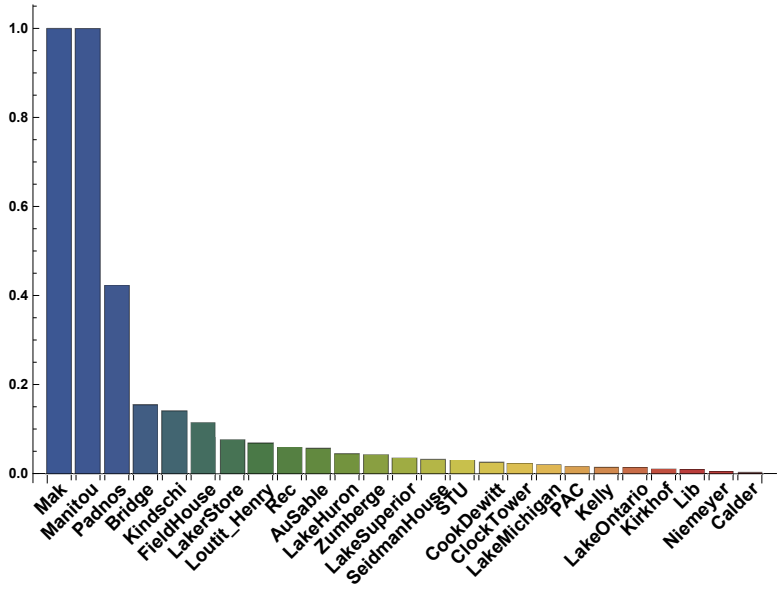


Figure 12: Bar chart of node eigenvector centrality

PageRank Centrality

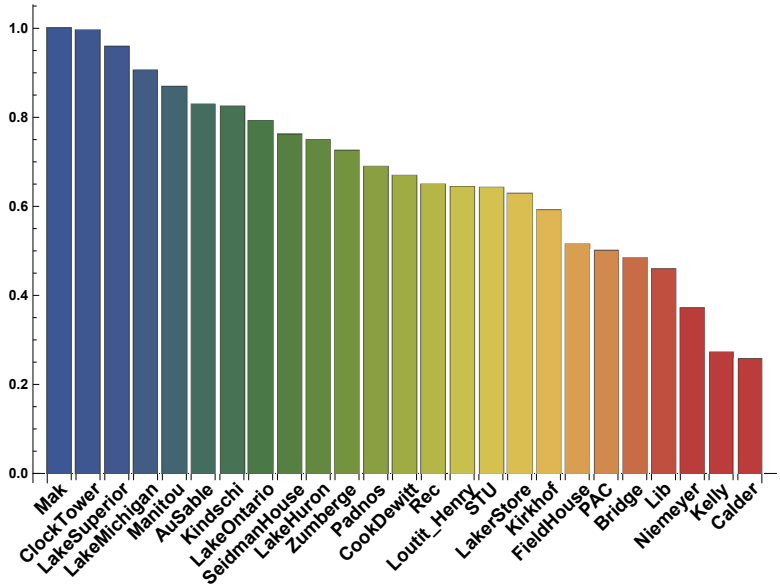


Figure 13: Bar chart of node PageRank centrality

Hard Community Detection

Using modularity, we can partition this network into communities of buildings. The implementation we used does take edge weights into account when calculating the overall modularity of the set of communities. Since the edges and edge weights relied on the physical connection of buildings by sidewalks, it makes sense that hard community detection splits the nodes into roughly North, East, West, South, and Central campus. The best achievable modularity was about 0.623, which gives us five distinct communities within the network. See Figure (14) for reference.

Figure 14: Campus Communities

Soft Community Detection

Figure (15) represents a rank 10 singular value decomposition approximation that is then reordered by cosine similarity. We then plot this on a matrix plot where an orange block represents a strong cosine similarity and a blue block represents a weak cosine similarity. The reordering moves similar nodes closer together toward the diagonal.

While generating this plot, we chose rank 10 because this maintains 70 percent of the singular values in the singular value decomposition. This them maps to about 10 soft communities within the network. It is easy to see the hard communities within this plot along with more information regarding how some nodes could belong in multiple communities. This occurs for nodes such as the ones that are contained within the block from Loutit_Henry to Ausable. Zumberge is not in the same community as Lib, Kirkhof, and PAC

in the hard communities, but within the soft communities, there still exists a relationship. This can be verified by the relatively close proximity that Zumberge is to Kirkhof (just across the pond).

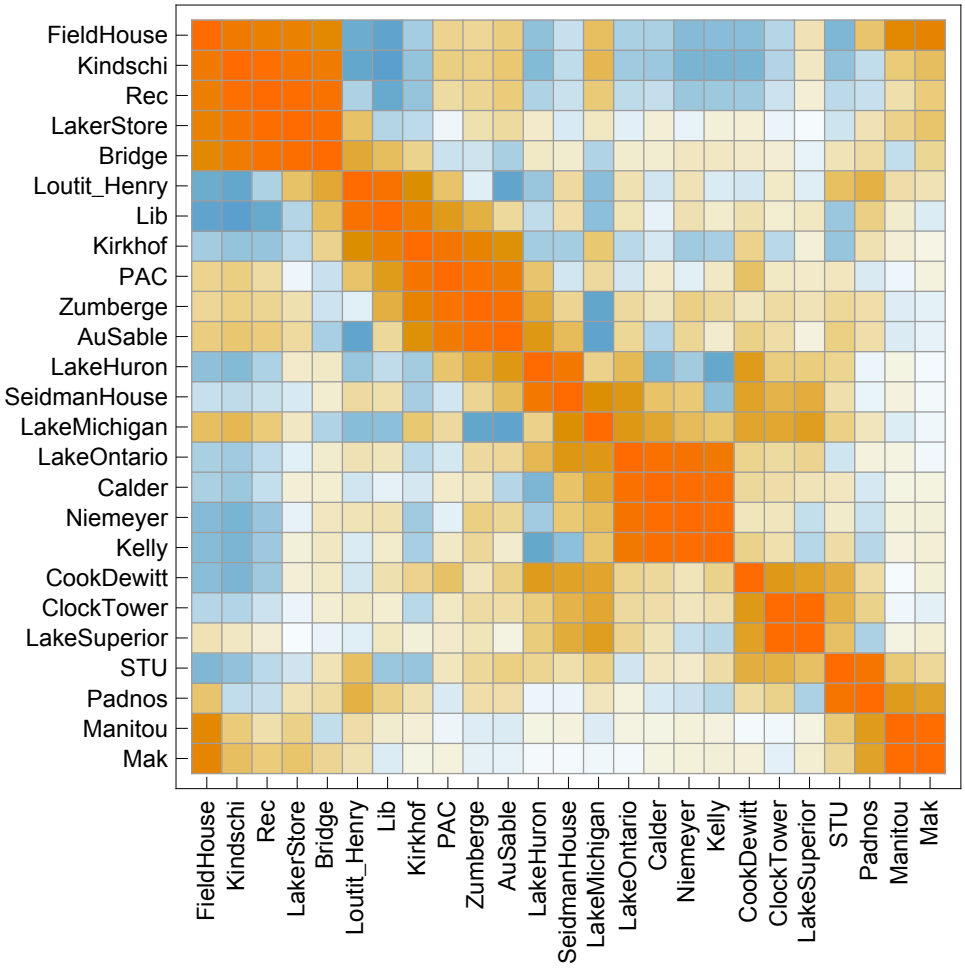


Figure 15: Soft Community Detection using Singular Value Decomposition

Shortest Paths and Tours

Since we decided to represent GV's campus as a network where the edge weights are based on actual distances, this led us to analyze the network as a Hamiltonian graph. This property means that a path exists where each node will be visited only once. Finding the shortest path that satisfies

that criterion is generally computationally intractable for large networks, but ours is small enough to find such a path (see Figure (17)). This tour is approximately 1.7 miles in length. A possible use or exploration for this type of analysis would be for GVSU to optimize the route tours should take to get done in the shortest amount of time to allow more tours in the schedule. Another interesting feature that can be extracted from our network is a shortest path between two nodes. Again, since the edge weights have accurate distances associated with them, we can provide an accurate path between two buildings to optimize travel time. A possible user or exploration for this calculation would be for new students that need to find the best way around campus (see Figure (16)).

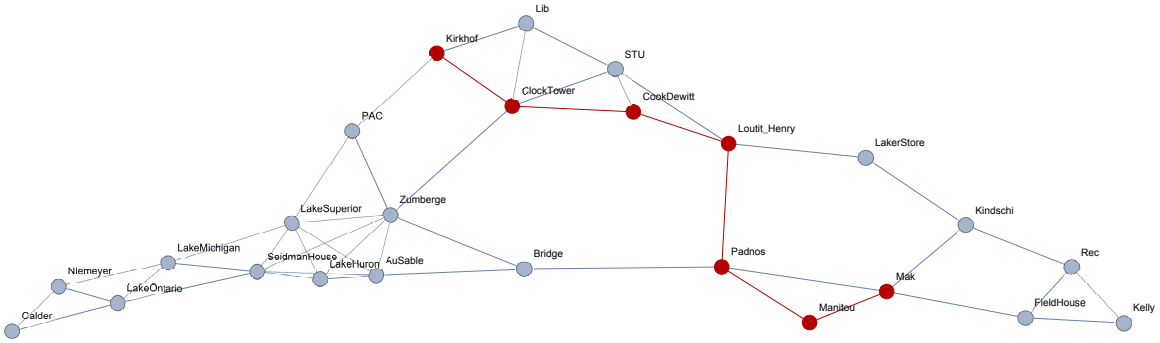


Figure 16: The shortest path from Kirkhof to Mackinac

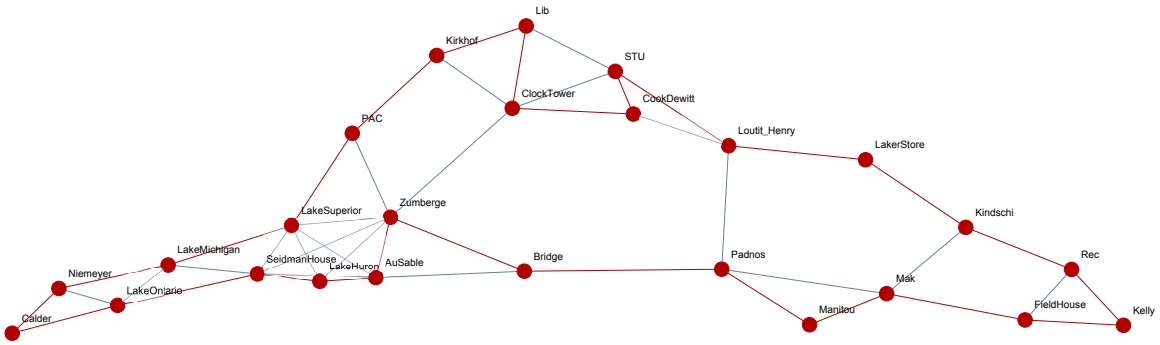


Figure 17: The shortest tour around campus

3.2 INSERT NAME OF NETWORK HERE

3.3 Madrid Train Bombing Terrorist Network

On March 14, 2004, ten explosions occurred simultaneously upon four commuter trains while en route to Atocha station in Madrid, Spain's capital city. Claiming 192 lives, the attack was the deadliest in Spanish history, and the worst to have occurred in Europe in fifteen years. Politicians and the media were quick to accuse the ETA, a well-known homegrown terrorist group from the Basque Country in northern Spain, but as evidence rolled in it became increasingly clear that the attack was likely connected to an outside terrorist organization. After several years of investigation, a Spanish court found 21 people guilty of either masterminding the attack, executing it, or aiding in the preparation of the attacks.

The following network, described in Brian Hayes's "Connecting the Dots: Can the Tools of Graph Theory and Social-Network Studies Unravel the Next Big Plot" published by American Scientist, illustrates the relationships or interactions between 70 individuals presumed to have some connection to the attack. These connections were made according to a range of criteria varying from friendships or familial ties, interactions with suspected terrorists or terrorist sympathizers, co-participants in war or relationships to al-Qaeda or Osama Bin Laden. Consequently, many of the connections are coincidental and do not necessarily imply the direct involvement of each of the individuals in the network in the advancement of the plot. These nodes are not superfluous, however, as they help to paint a more intricate picture of the goings-on behind a terror attack of this scale.

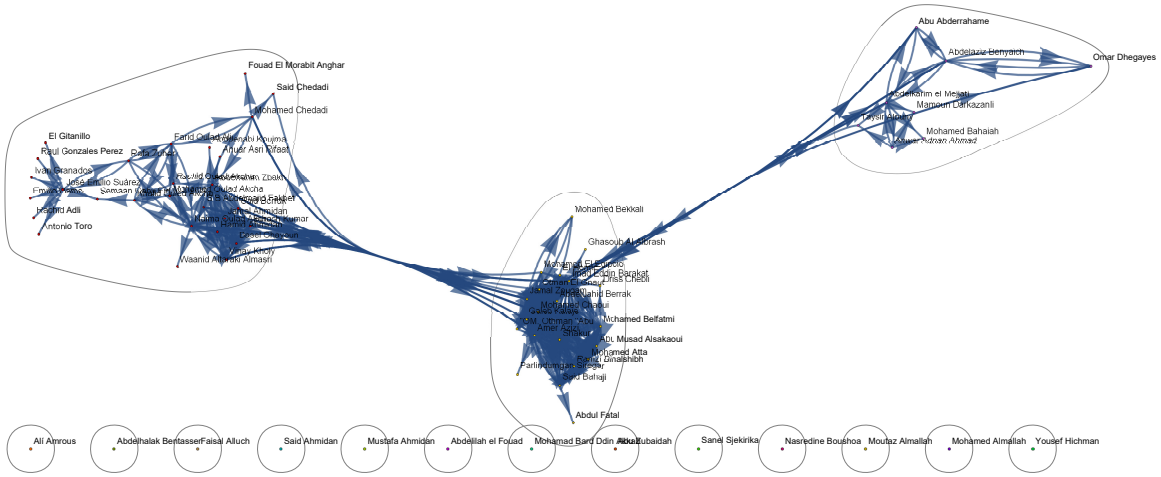


Figure 18: Community structures

Figure 23 displays the detected community structures within the network and is particularly helpful in understanding how the data was compiled. We can see that there are three principal communities to which the vast majority of nodes belong. Additionally, there are 13 dissociated nodes each belonging to their own community. These nodes largely correspond to individuals who were not immediately involved in the attack, but had been suspected of terrorist activity or observed making comments that suggested that they were sympathetic to terrorism. Ali Amrous, the bottom left-most node, for example, had been detained and interrogated in San Sebastian for threatening an attack on Madrid. He was also suspected of being a member of al-Qaeda, although there was not enough evidence to suppose that he knew of the attack beforehand. There are some inconsistencies though, particularly when we focus our attention on the Almallah brothers, who were both accused of hosting meetings in an apartment that they owned in Madrid in which they provided resources and information to al-Qaeda recruits. There is evidence that suggests that they had links to a number of the terrorists charged with having committed the attack. These discrepancies may be explained by the way in which the data was collected and organized. It is also possible that at the time this

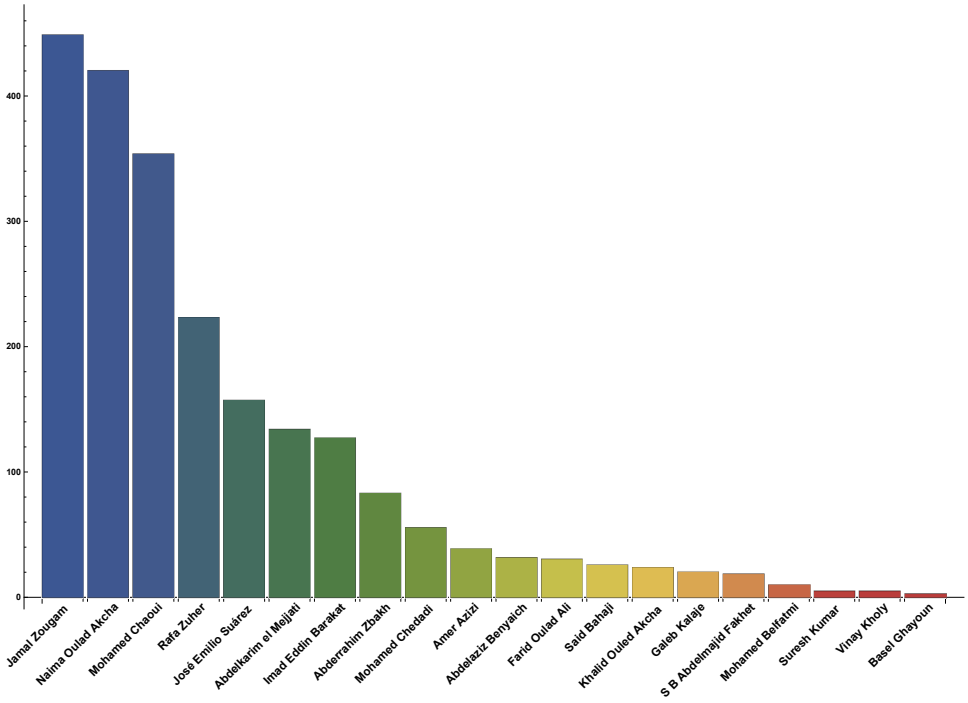


Figure 19: Bar chart of node betweenness centrality

Betweenness centrality is a measure that quantifies how many shortest paths from node i to node j pass through node c. It follows that a node with a high betweenness score would be central to the network in that many of the other nodes in the network would be connected to each other through it. In the context of a terrorist network, an individual with a high betweenness score would be someone who is directly connected to many people who are not themselves directly connected. They are likely a critical component of the scheme or, in terms of communities, a linking factor between communities. Naima Oulad Akcha, the only woman charged in the case, was the sister of two prime suspects, Mohamed and Rachid Oulad Akcha, who were thought to have helped place the bombs on the trains. Imad Eddin Barakat, better known as Abu Dahdah, has a high betweenness score likely due to his elite role in the Madrid faction of al-Qaeda. Thus, his connections to others within the network may not be consequence of a direct relationship with them, but instead co-participance in a terror organization or previous acts of terror.

It is unsurprising that Jamal Zougam, charged with helping to mastermind and execute the attack for which he was sentenced to 40,000 years in prison, has the highest betweenness centrality. He was the connection between young amenable Moroccan migrants and al-Qaeda and was responsible for developing the terrorist cell based in Madrid. Among other unsurprising results, Mohamed Chaoui, half-brother to Zougam, was another principal actor in the attack. The brothers, along with their business parter Mohamed Bekkali, were arrested when a mobile phone that was supposed to have detonated one of bombs and failed to do so was discovered.

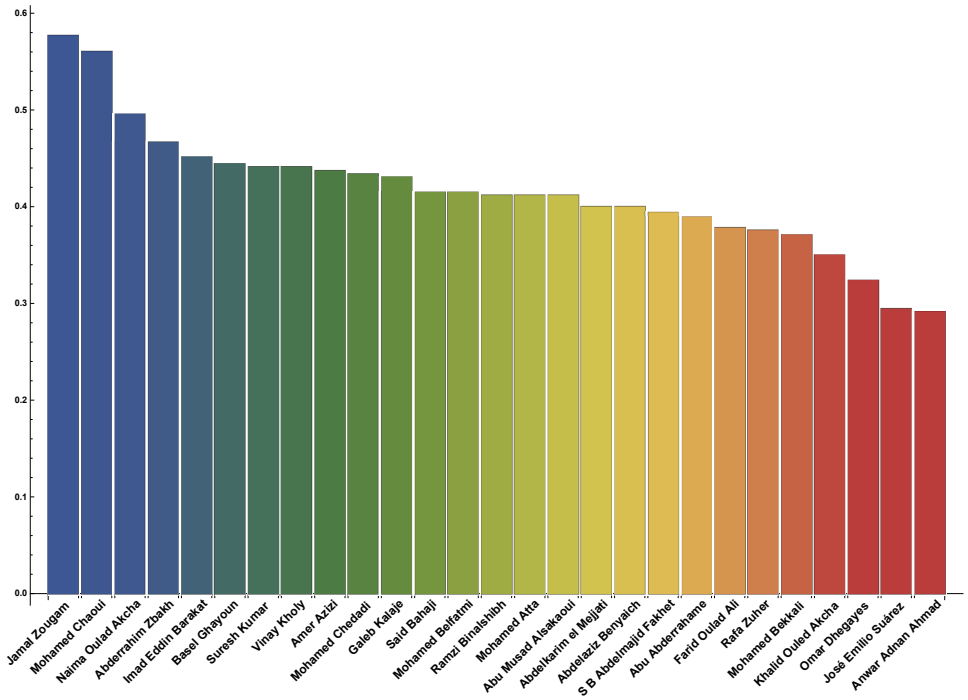


Figure 20: Bar chart of node closeness centrality

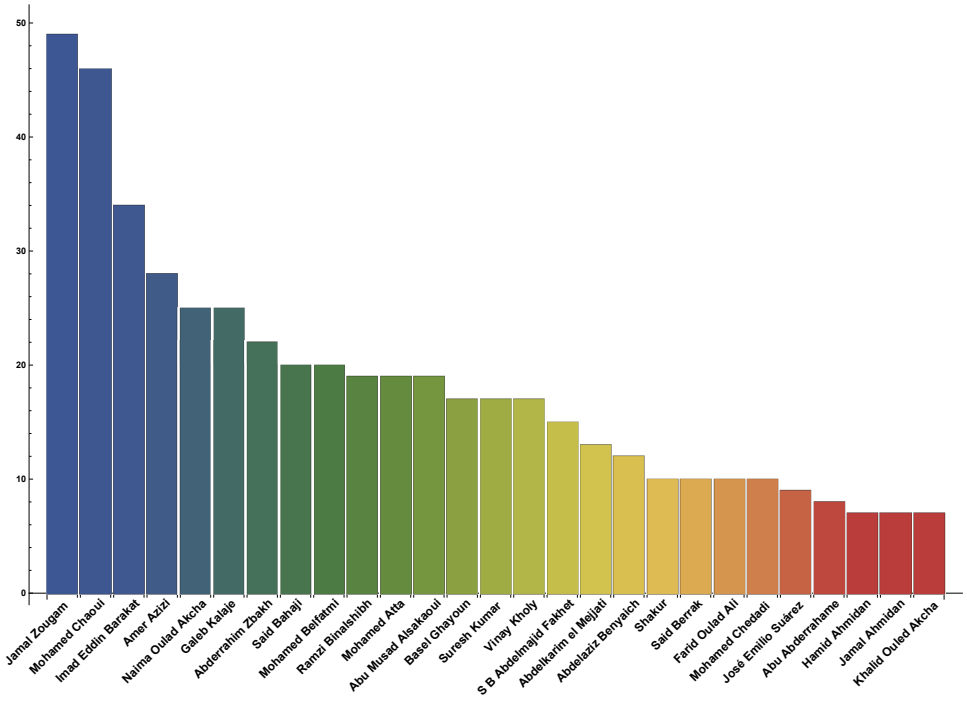


Figure 21: Bar chart of node degree centrality

Figure 21 produces similar results to the other measures. Jamal Zougam has the largest number of connections within the network, due to his role in implementing the attack as well as his position as recruiter.

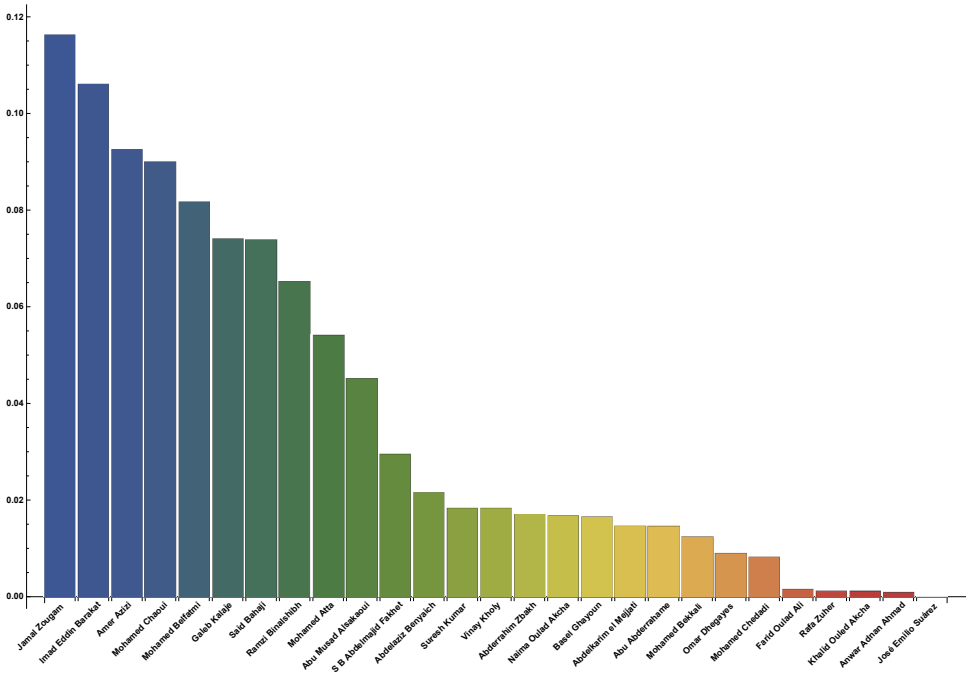


Figure 22: Bar chart of node eigenvector centrality

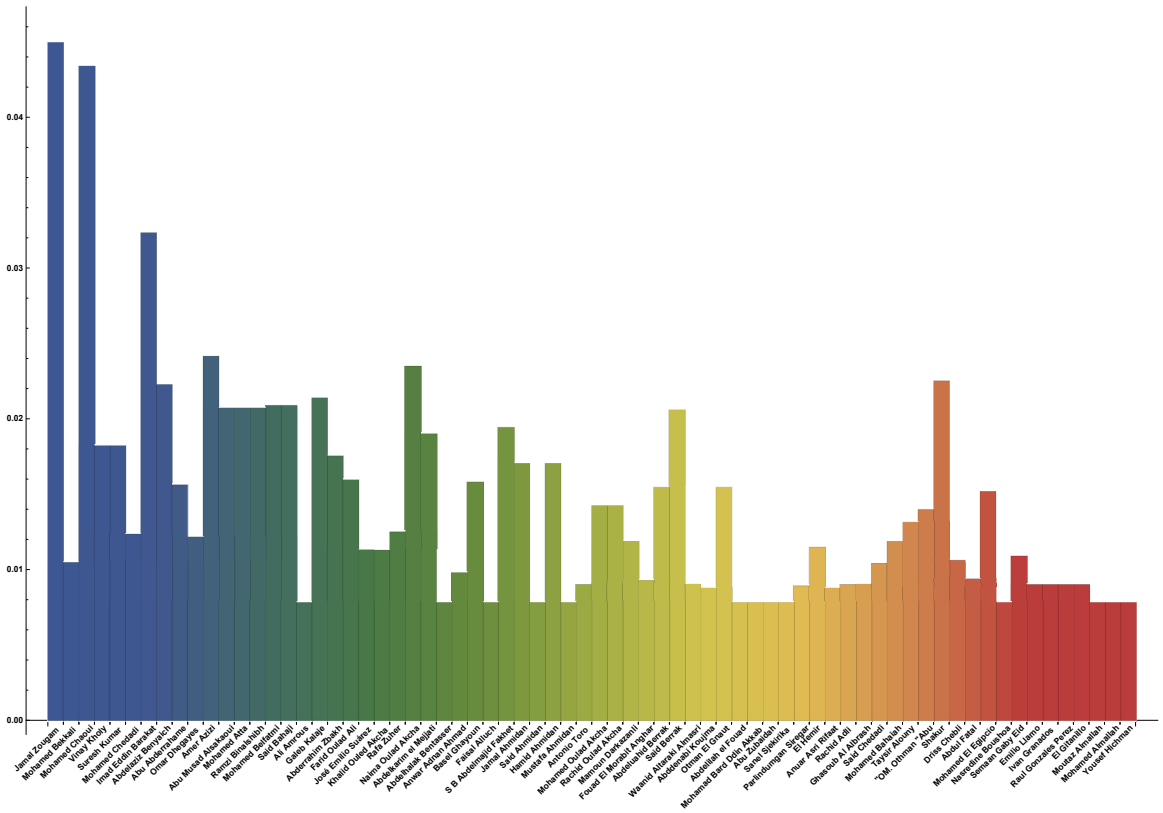


Figure 23: Community structures

4 Works Cited

Bavelas, A. (1950). Communication patterns in task-oriented groups. *J. Acoust. Soc. Am 22(6)*, 725730.

Blondel, V.D. Guillaume, J.L., Lambiotte, & R., Lefebvre, E. (2008). Fast unfolding of communities in large networks. Universite catholique de Louvain, Louvain-la-Neuve, Belgium.

Borgatti, S. J. (2005). Centrality and Network flow. *Social Networks, 27*, 55-71.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology 25*, 163177. doi:10.1080/0022250x.2001.9990249.

Newman M.E.J. (2010). *Networks: An Introduction.* Oxford, New York: Oxford University Press

Perez, A.P. & Dugue, Nicolas

Nicolas Dugue, Anthony Perez. Directed Louvain: maximizing modularity in directed networks. Universite d'Orleans. 2015.