This chapter is from *Social Media Mining: An Introduction*.
By Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu.
Cambridge University Press, 2014. Draft version: April 20, 2014.
Complete Draft and Slides Available at: http://dmml.asu.edu/smm

Chapter 3

Network Measures

In February 2012, Kobe Bryant, the American basketball star, joined Chinese microblogging site Sina Weibo. Within a few hours, more than 100,000 followers joined his page, anxiously waiting for his first microblogging post on the site. The media considered the tremendous number of followers Kobe Bryant received as an indication of his popularity in China. In this case, the number of followers *measured* Bryant's popularity among Chinese social media users. In social media, we often face similar tasks in which measuring different structural properties of a social media network can help us better understand individuals embedded in it. Corresponding measures need to be designed for these tasks. This chapter discusses measures for social media networks.

When mining social media, a graph representation is often used. This graph shows friendships or user interactions in a social media network. Given this graph, some of the questions we aim to answer are as follows:

- Who are the central figures (influential individuals) in the network?
- What interaction patterns are common in friends?
- Who are the *like-minded* users and how can we find these similar individuals?

To answer these and similar questions, one first needs to define *measures* for quantifying centrality, level of interactions, and similarity, among other qualities. These measures take as input a graph representation of a social interaction, such as friendships (adjacency matrix), from which the measure value is computed.

To answer our first question about finding central figures, we define measures for *centrality*. By using these measures, we can identify various types of central nodes in a network. To answer the other two questions, we define corresponding measures that can quantify interaction patterns and help find like-minded users. We discuss centrality next.

3.1 Centrality

Centrality defines how important a node is within a network.

3.1.1 Degree Centrality

In real-world interactions, we often consider people with many connections to be important. Degree centrality transfers the same idea into a measure. The degree centrality measure ranks nodes with more connections higher in terms of centrality. The degree centrality C_d for node v_i in an undirected graph is

$$C_d(v_i) = d_i, \tag{3.1}$$

where d_i is the degree (number of adjacent edges) of node v_i. In directed graphs, we can either use the in-degree, the out-degree, or the combination as the degree centrality value:

$$\begin{aligned} C_d(v_i) &= d_i^{\text{in}} \qquad \textit{(prestige)}, & (3.2)\\ C_d(v_i) &= d_i^{\text{out}} \qquad \textit{(gregariousness)}, & (3.3)\\ C_d(v_i) &= d_i^{\text{in}} + d_i^{\text{out}}. & (3.4) \end{aligned}$$

Prominence or Prestige

When using in-degrees, degree centrality measures how popular a node is and its value shows *prominence* or *prestige*. When using out-degrees, it measures the *gregariousness* of a node. When we combine in-degrees and out-degrees, we are basically ignoring edge directions. In fact, when edge directions are removed, Equation 3.4 is equivalent to Equation 3.1, which measures degree centrality for undirected graphs.

The degree centrality measure does not allow for centrality values to be compared across networks (e.g., Facebook and Twitter). To overcome this problem, we can normalize the degree centrality values.

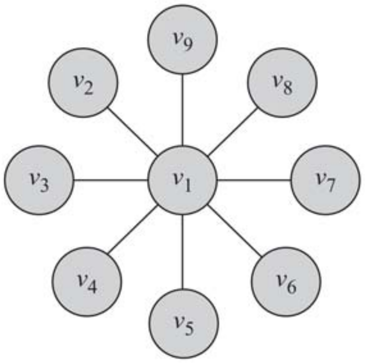


Figure 3.1: Degree Centrality Example.

Normalizing Degree Centrality

Simple normalization methods include normalizing by the maximum possible degree,

$$C_d^{\text{norm}}(v_i) = \frac{d_i}{n-1}, \tag{3.5}$$

where n is the number of nodes. We can also normalize by the maximum degree,

$$C_d^{\max}(v_i) = \frac{d_i}{\max_j d_j}. \tag{3.6}$$

Finally, we can normalize by the degree sum,

$$C_d^{\text{sum}}(v_i) = \frac{d_i}{\sum_j d_j} = \frac{d_i}{2|E|} = \frac{d_i}{2m}. \tag{3.7}$$

Example 3.1. *Figure 3.1 shows a sample graph. In this graph, degree centrality for node v_1 is $C_d(v_1) = d_1 = 8$, and for all others, it is $C_d(v_j) = d_j = 1$, $j \neq 1$.*

3.1.2 Eigenvector Centrality

In degree centrality, we consider nodes with more connections to be more important. However, in real-world scenarios, having more friends does not by itself guarantee that someone is important: *having more important friends* provides a stronger signal.

Eigenvector centrality tries to generalize degree centrality by incorporating the importance of the neighbors (or incoming neighbors in directed graphs). It is defined for both directed and undirected graphs. To keep track of neighbors, we can use the adjacency matrix A of a graph. Let $c_e(v_i)$ denote the eigenvector centrality of node v_i. We want the centrality of v_i to be a function of its neighbors' centralities. We posit that it is proportional to the summation of their centralities,

$$c_e(v_i) = \frac{1}{\lambda} \sum_{j=1}^{n} A_{j,i} c_e(v_j), \tag{3.8}$$

where λ is some fixed constant. Assuming $\mathbf{C}_e = (C_e(v_1), C_e(v_2), \ldots, C_e(v_n))^T$ is the centrality vectors for all nodes, we can rewrite Equation 3.8 as

$$\lambda \mathbf{C}_e = A^T \mathbf{C}_e. \tag{3.9}$$

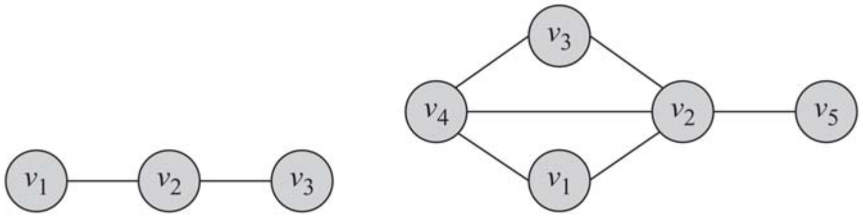
This basically means that $\mathbf{C}_e$ is an eigenvector of adjacency matrix A^T (or A in undirected networks, since $A = A^T$) and λ is the corresponding eigenvalue. A matrix can have many eigenvalues and, in turn, many corresponding eigenvectors. Hence, this raises the question: which eigenvalue–eigenvector pair should we select? We often prefer centrality values to be positive for convenient comparison of centrality values across nodes. Thus, we can choose an eigenvalue such that the eigenvector components are positive.[1] This brings us to the *Perron-Frobenius* theorem.

Perron-Frobenius Theorem

Theorem 3.1 (Perron-Frobenius Theorem)**.** *Let $A \in \mathbb{R}^{n \times n}$ represent the adjacency matrix for a [strongly] connected graph or $A : A_{i,j} > 0$ (i.e. a positive n by n matrix). There exists a positive real number (Perron-Frobenius eigenvalue) $\lambda_{\max}$, such that $\lambda_{\max}$ is an eigenvalue of A and any other eigenvalue of A is strictly smaller than $\lambda_{\max}$. Furthermore, there exists a corresponding eigenvector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ of A with eigenvalue $\lambda_{\max}$ such that $\forall v_i > 0$.*

Therefore, to have positive centrality values, we can compute the eigenvalues of A and then select the largest eigenvalue. The corresponding eigenvector is $\mathbf{C}_e$. Based on the Perron-Frobenius theorem, all the components of $\mathbf{C}_e$ will be positive, and this vector corresponds to eigenvector centralities for the graph.

[1] This constraint is optional and can be lifted based on the context.



(a) A three node graph

(b) A five node graph

Figure 3.2: Eigenvector Centrality Example.

Example 3.2. *For the graph shown in Figure 3.2(a), the adjacency matrix is*

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{3.10}$$

Based on Equation 3.9, we need to solve $\lambda\mathbf{C}_e = A\mathbf{C}_e$, or

$$(A - \lambda I)\mathbf{C}_e = 0. \tag{3.11}$$

Assuming $\mathbf{C}_e = [u_1\ u_2\ u_3]^T$,

$$\begin{bmatrix} 0-\lambda & 1 & 0 \\ 1 & 0-\lambda & 1 \\ 0 & 1 & 0-\lambda \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3.12}$$

Since $\mathbf{C}_e \neq [0\ 0\ 0]^T$, the characteristic equation is

$$det(A - \lambda I) = \begin{vmatrix} 0-\lambda & 1 & 0 \\ 1 & 0-\lambda & 1 \\ 0 & 1 & 0-\lambda \end{vmatrix} = 0, \tag{3.13}$$

or equivalently,

$$(-\lambda)(\lambda^2 - 1) - 1(-\lambda) = 2\lambda - \lambda^3 = \lambda(2 - \lambda^2) = 0. \tag{3.14}$$

So the eigenvalues are $(-\sqrt{2}, 0, +\sqrt{2})$. We select the largest eigenvalue: $\sqrt{2}$. We compute the corresponding eigenvector:

$$\begin{bmatrix} 0-\sqrt{2} & 1 & 0 \\ 1 & 0-\sqrt{2} & 1 \\ 0 & 1 & 0-\sqrt{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3.15}$$

Assuming $\mathbf{C}_e$ *vector has norm 1, its solution is*

$$\mathbf{C}_e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ \sqrt{2}/2 \\ 1/2 \end{bmatrix}, \tag{3.16}$$

which denotes that node v_2 *is the most central node and nodes* v_1 *and* v_3 *have equal centrality values.*

Example 3.3. *For the graph shown in Figure 3.2(b), the adjacency matrix is as follows:*

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.17}$$

The eigenvalues of A *are* $(-1.74, -1.27, 0.00, +0.33, +2.68)$*. For eigenvector centrality, the largest eigenvalue is selected:* 2.68*. The corresponding eigenvector is the eigenvector centrality vector and is*

$$\mathbf{C}_e = \begin{bmatrix} 0.4119 \\ 0.5825 \\ 0.4119 \\ 0.5237 \\ 0.2169 \end{bmatrix}. \tag{3.18}$$

Based on eigenvector centrality, node v_2 *is the most central node.*

3.1.3 Katz Centrality

A major problem with eigenvector centrality arises when it considers directed graphs (see Problem 1 in the Exercises). Centrality is only passed on when we have (outgoing) edges, and in special cases such as when a node is in a directed acyclic graph, centrality becomes zero, even though the node can have many edges connected to it. In this case, the problem can be rectified by adding a bias term to the centrality value. The bias term β is added to the centrality values for all nodes no matter how they are situated in the network (i.e., irrespective of the network topology). The

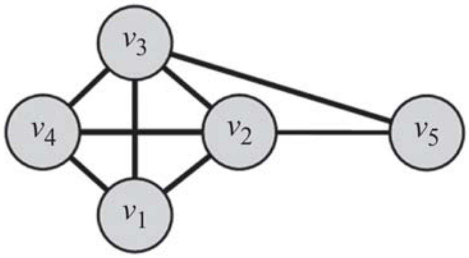


Figure 3.3: Katz Centrality Example.

resulting centrality measure is called the *Katz centrality* and is formulated as

$$C_{\text{Katz}}(v_i) = \alpha \sum_{j=1}^{\text{n}} A_{j,i} C_{\text{Katz}}(v_j) + \beta. \tag{3.19}$$

The first term is similar to eigenvector centrality, and its effect is controlled by constant α. The second term β, is the bias term that avoids zero centrality values. We can rewrite Equation 3.19 in a vector form,

$$\mathbf{C}_{\text{Katz}} = \alpha A^T \mathbf{C}_{\text{Katz}} + \beta \mathbf{1}, \tag{3.20}$$

where **1** is a vector of all 1's. Taking the first term to the left hand side and factoring C_{Katz},

$$\mathbf{C}_{\text{Katz}} = \beta(\mathbf{I} - \alpha A^T)^{-1} \cdot \mathbf{1}. \tag{3.21}$$

Since we are inverting a matrix here, not all α values are acceptable. When $\alpha = 0$, the eigenvector centrality part is removed, and all nodes get the same centrality value β. However, once α gets larger, the effect of β is reduced, and when $\det(\mathbf{I} - \alpha A^T) = 0$, the matrix $\mathbf{I} - \alpha A^T$ becomes non-invertible and the centrality values diverge. The $\det(\mathbf{I} - \alpha A^T)$ first becomes 0 when $\alpha = 1/\lambda$, where λ is the largest eigenvalue[2] of A^T. In practice, $\alpha < 1/\lambda$ is selected so that centralities are computed correctly.

Divergence in Centrality Computation

Example 3.4. *For the graph shown in Figure 3.3, the adjacency matrix is as*

[2] When $\det(\mathbf{I} - \alpha A^T) = 0$, it can be rearranged as $\det(A^T - \alpha^{-1} I) = 0$, which is basically the characteristic equation. This equation first becomes zero when the largest eigenvalue equals α^{-1}, or equivalently $\alpha = 1/\lambda$.

follows:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} = A^T. \tag{3.22}$$

The eigenvalues of A are $(-1.68, -1.0, -1.0, +0.35, +3.32)$. *The largest eigenvalue of A is $\lambda = 3.32$. We assume $\alpha = 0.25 < 1/\lambda$ and $\beta = 0.2$. Then, Katz centralities are*

$$\mathbf{C}_{Katz} = \beta(\mathbf{I} - \alpha A^T)^{-1} \cdot \mathbf{1} = \begin{bmatrix} 1.14 \\ 1.31 \\ 1.31 \\ 1.14 \\ 0.85 \end{bmatrix}. \tag{3.23}$$

Thus, nodes v_2, and v_3 have the highest Katz centralities.

3.1.4 PageRank

Similar to eigenvector centrality, Katz centrality encounters some challenges. A challenge that happens in directed graphs is that, once a node becomes an authority (high centrality), it passes **all** its centrality along **all** of its out-links. This is less desirable, because not everyone known by a well known person is well known. To mitigate this problem, one can divide the value of passed centrality by the number of outgoing links (out-degree) from that node such that each connected neighbor gets a fraction of the source node's centrality:

$$C_p(v_i) = \alpha \sum_{j=1}^{n} A_{j,i} \frac{C_p(v_j)}{d_j^{\text{out}}} + \beta. \tag{3.24}$$

This equation is only defined when d_j^{out} is nonzero. Thus, assuming that all nodes have positive out-degrees ($d_j^{\text{out}} > 0$)[3], Equation 3.24 can be reformulated in matrix format,

$$\mathbf{C}_p = \alpha A^T D^{-1} \mathbf{C}_p + \beta \mathbf{1}, \tag{3.25}$$

[3] When $d_j^{\text{out}} = 0$, we know that since the out-degree is zero, $\forall i, A_{j,i} = 0$. This makes the term inside the summation $\frac{0}{0}$. We can fix this problem by setting $d_j^{\text{out}} = 1$ since the node will not contribute any centrality to any other nodes.

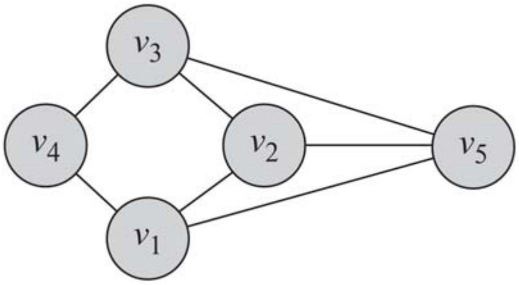


Figure 3.4: PageRank Example.

which we can reorganize,

$$\mathbf{C}_p = \beta(\mathbf{I} - \alpha A^T D^{-1})^{-1} \cdot \mathbf{1}, \tag{3.26}$$

where $D = diag(d_1^{\text{out}}, d_2^{\text{out}}, \ldots, d_n^{\text{out}})$ is a diagonal matrix of degrees. The centrality measure is known as the *PageRank* centrality measure and is used by the Google search engine as a measure for ordering webpages. Webpages and their links represent an enormous web-graph. PageRank defines a centrality measure for the nodes (webpages) in this web-graph. When a user queries Google, webpages that match the query and have higher PageRank values are shown first. Similar to Katz centrality, in practice, $\alpha < \frac{1}{\lambda}$ is selected, where λ is the largest eigenvalue of $A^T D^{-1}$. In undirected graphs, the largest eigenvalue of $A^T D^{-1}$ is $\lambda = 1$; therefore, $\alpha < 1$.

PageRank and Google Web Search

Example 3.5. *For the graph shown in Figure 3.4, the adjacency matrix is as follows,*

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{3.27}$$

We assume $\alpha = 0.95 < 1$ and $\beta = 0.1$. Then, PageRank values are

$$\mathbf{C}_p = \beta(\mathbf{I} - \alpha A^T D^{-1})^{-1} \cdot \mathbf{1} = \begin{bmatrix} 2.14 \\ 2.13 \\ 2.14 \\ 1.45 \\ 2.13 \end{bmatrix}. \tag{3.28}$$

Hence, nodes v_1 and v_3 have the highest PageRank values.

3.1.5 Betweenness Centrality

Another way of looking at centrality is by considering how important nodes are in connecting other nodes. One approach, for a node v_i, is to compute the number of shortest paths between other nodes that pass through v_i,

$$C_b(v_i) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}}, \tag{3.29}$$

where σ_{st} is the number of shortest paths from node s to t (also known as *information pathways*), and $\sigma_{st}(v_i)$ is the number of shortest paths from s to t that pass through v_i. In other words, we are measuring how central v_i's role is in connecting any pair of nodes s and t. This measure is called *betweenness centrality*.

Betweenness centrality needs to be normalized to be comparable across networks. To normalize betweenness centrality, one needs to compute the maximum value it takes. Betweenness centrality takes its maximum value when node v_i is on all shortest paths from s to t for any pair (s, t); that is, $\forall\ (s, t),\ s \neq t \neq v_i,\ \frac{\sigma_{st}(v_i)}{\sigma_{st}} = 1$. For instance, in Figure 3.1, node v_1 is on the shortest path between all other pairs of nodes. Thus, the maximum value is

$$C_b(v_i) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}} = \sum_{s \neq t \neq v_i} 1 = 2\binom{n-1}{2} = (n-1)(n-2). \tag{3.30}$$

The betweenness can be divided by its maximum value to obtain the normalized betweenness,

$$C_b^{\text{norm}}(v_i) = \frac{C_b(v_i)}{2\binom{n-1}{2}}. \tag{3.31}$$

Computing Betweenness

In betweenness centrality (Equation 3.29), we compute shortest paths between all pairs of nodes to compute the betweenness value. If an algorithm such as Dijkstra's is employed, it needs to be run for all nodes, because

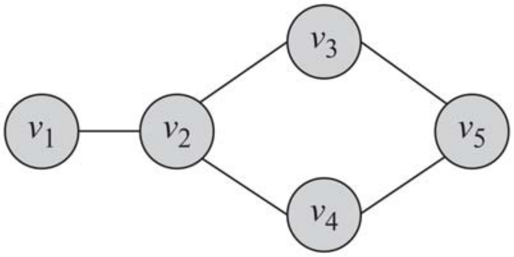


Figure 3.5: Betweenness Centrality Example.

Dijkstra's algorithm will compute shortest paths from a single node to all other nodes. So, to compute all-pairs shortest paths, Dijkstra's algorithm needs to be run $|V| - 1$ times (with the exception of the node for which centrality is being computed). More effective algorithms such as the Brandes' algorithm [45] have been designed. Interested readers can refer to the bibliographic notes for further references.

Example 3.6. *For Figure 3.1, the (normalized) betweenness centrality of node v_1 is*

$$C_b(v_1) = 2\binom{8}{2}, \tag{3.32}$$

$$C_b^{\text{norm}}(v_1) = 1. \tag{3.33}$$

Since all the paths that go through any pair $(s, t), s \neq t \neq v_1$ pass through node v_1, the centrality is $2\binom{8}{2}$. Similarly, the betweenness centrality for any other node in this graph is 0.

Example 3.7. *Figure 3.5 depicts a sample graph. In this graph, the betweenness centrality for node v_1 is 0, since no shortest path passes through it. For other nodes, we have*

$$\begin{aligned} C_b(v_2) &= 2 \times (\underbrace{(1/1)}_{s=v_1,t=v_3} + \underbrace{(1/1)}_{s=v_1,t=v_4} + \underbrace{(2/2)}_{s=v_1,t=v_5} + \underbrace{(1/2)}_{s=v_3,t=v_4} + \underbrace{0}_{s=v_3,t=v_5} + \underbrace{0}_{s=v_4,t=v_5}) \\ &= 2 \times 3.5 = 7, \end{aligned} \tag{3.34}$$

$$\begin{aligned} C_b(v_3) &= 2 \times (\underbrace{0}_{s=v_1,t=v_2} + \underbrace{0}_{s=v_1,t=v_4} + \underbrace{(1/2)}_{s=v_1,t=v_5} + \underbrace{0}_{s=v_2,t=v_4} + \underbrace{(1/2)}_{s=v_2,t=v_5} + \underbrace{0}_{s=v_4,t=v_5}) \\ &= 2 \times 1.0 = 2, \end{aligned} \tag{3.35}$$

$$C_b(v_4) = C_b(v_3) = 2 \times 1.0 = 2, \tag{3.36}$$

$$\begin{aligned} C_b(v_5) &= 2 \times (\underbrace{0}_{s=v_1,t=v_2} + \underbrace{0}_{s=v_1,t=v_3} + \underbrace{0}_{s=v_1,t=v_4} + \underbrace{0}_{s=v_2,t=v_3} + \underbrace{0}_{s=v_2,t=v_4} + \underbrace{(1/2)}_{s=v_3,t=v_4}) \\ &= 2 \times 0.5 = 1, \end{aligned} \tag{3.37}$$

where centralities are multiplied by 2 because in an undirected graph $\sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}} = 2 \sum_{s \neq t \neq v_i, s<t} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$.

3.1.6 Closeness Centrality

In closeness centrality, the intuition is that the more central nodes are, the more quickly they can reach other nodes. Formally, these nodes should have a smaller average shortest path length to other nodes. Closeness centrality is defined as

$$C_c(v_i) = \frac{1}{\bar{l}_{v_i}}, \tag{3.38}$$

where $\bar{l}_{v_i} = \frac{1}{n-1} \sum_{v_j \neq v_i} l_{i,j}$ is node v_i's average shortest path length to other nodes. The smaller the average shortest path length, the higher the centrality for the node.

Example 3.8. *For nodes in Figure 3.5, the closeness centralities are as follows:*

$$C_c(v_1) = 1 \,/\, (\,(1+2+2+3)/4\,) = 0.5, \tag{3.39}$$

$$C_c(v_2) = 1 \,/\, (\,(1+1+1+2)/4\,) = 0.8, \tag{3.40}$$

$$C_c(v_3) = C_b(v_4) = 1 \,/\, (\,(1+1+2+2)/4\,) = 0.66, \tag{3.41}$$

$$C_c(v_5) = 1 \,/\, (\,(1+1+2+3)/4\,) = 0.57. \tag{3.42}$$

Hence, node v_2 has the highest closeness centrality.

The centrality measures discussed thus far have different views on what a central node is. Thus, a central node for one measure may be deemed unimportant by other measures.

Example 3.9. *Consider the graph in Figure 3.6. For this graph, we compute the top three central nodes based on degree, eigenvector, Katz, PageRank, betweenness, and closeness centrality methods. These nodes are listed in Table 3.1.*

As shown in the table, there is a high degree of similarity between most central nodes for the first four measures, which utilize eigenvectors or degrees: degree

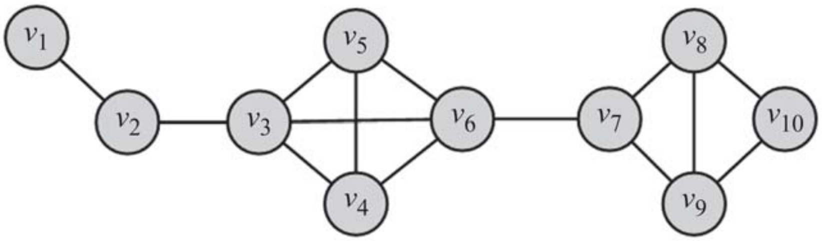


Figure 3.6: Example for All Centrality Measures.

Table 3.1: A Comparison between Centrality Methods

	First Node	Second Node	Third Node
Degree Centrality	v_3 or v_6	v_6 or v_3	$v \in \{v_4, v_5, v_7, v_8, v_9\}$
Eigenvector Centrality	v_6	v_3	v_4 or v_5
Katz Centrality: $\alpha = \beta = 0.3$	v_6	v_3	v_4 or v_5
PageRank: $\alpha = \beta = 0.3$	v_3	v_6	v_2
Betweenness Centrality	v_6	v_7	v_3
Closeness Centrality	v_6	v_3 or v_7	v_7 or v_3

centrality, eigenvector centrality, Katz centrality, and PageRank. Betweenness centrality also generates similar results to closeness centrality because both use the shortest paths to find most central nodes.

3.1.7 Group Centrality

All centrality measures defined so far measure centrality for a single node. These measures can be generalized for a group of nodes. In this section, we discuss how degree centrality, closeness centrality, and betweenness centrality can be generalized for a group of nodes. Let S denote the set of nodes to be measured for centrality. Let $V - S$ denote the set of nodes not in the group.

Group Degree Centrality

Group degree centrality is defined as the number of nodes from outside the group that are connected to group members. Formally,

$$C_d^{\text{group}}(S) = |\{v_i \in V - S | v_i \text{ is connected to } v_j \in S\}|. \qquad (3.43)$$

Similar to degree centrality, we can define connections in terms of out-degrees or in-degrees in directed graphs. We can also normalize this value. In the best case, group members are connected to all other nonmembers. Thus, the maximum value of $C_d^{\text{group}}(S)$ is $|V - S|$. So dividing group degree centrality value by $|V - S|$ normalizes it.

Group Betweenness Centrality

Similar to betweeness centrality, we can define group betweenness centrality as

$$C_b^{\text{group}}(S) = \sum_{s \neq t, s \notin S, t \notin S} \frac{\sigma_{st}(S)}{\sigma_{st}}, \tag{3.44}$$

where $\sigma_{st}(S)$ denotes the number of shortest paths between s and t that pass through members of S. In the best case, all shortest paths between s and t pass through members of S, and therefore, the maximum value for $C_b^{\text{group}}(S)$ is $2\binom{|V-S|}{2}$. Similar to betweenness centrality, we can normalize group betweenness centrality by dividing it by the maximum value.

Group Closeness Centrality

Closeness centrality for groups can be defined as

$$C_c^{\text{group}}(S) = \frac{1}{\bar{l}_S^{\text{group}}}, \tag{3.45}$$

where $\bar{l}_S^{\text{group}} = \frac{1}{|V-S|} \sum_{v_j \notin S} l_{S,v_j}$ and l_{S,v_j} is the length of the shortest path between a group S and a nonmember $v_j \in V - S$. This length can be defined in multiple ways. One approach is to find the closest member in S to v_j:

$$l_{S,v_j} = \min_{v_i \in S} l_{v_i,v_j}. \tag{3.46}$$

One can also use the maximum distance or the average distance to compute this value.

Example 3.10. *Consider the graph in Figure 3.7. Let $S = \{v_2, v_3\}$. Group degree centrality for S is*

$$C_d^{\text{group}}(S) = 3, \tag{3.47}$$

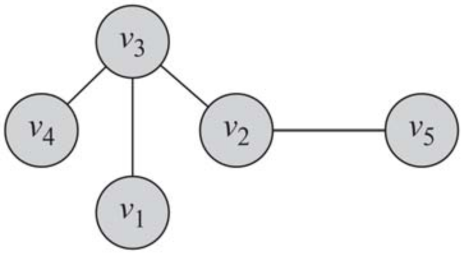


Figure 3.7: Group Centrality Example.

since members of the group are connected to all other three members in $V - S = \{v_1, v_4, v_5\}$. The normalized value is 1, since $3/|V - S| = 1$. Group betweenness centrality is 6, since for $2\binom{3}{2}$ shortest paths between any two members of $V - S$, the path has to pass through members of S. The normalized group betweenness is 1, since $6/(2\binom{|V-S|}{2}) = 1$. Finally, group closeness centrality – assuming the distance from nonmembers to members of S is computed using the minimum function – is also 1, since any member of $V - S$ is connected to a member of S directly.

3.2 Transitivity and Reciprocity

Often we need to observe a specific behavior in a social media network. One such behavior is linking behavior. Linking behavior determines how links (edges) are formed in a social graph. In this section, we discuss two well-known measures, *transitivity* and *reciprocity*, for analyzing this behavior. Both measures are commonly used in directed networks, and *transitivity* can also be applied to undirected networks.

3.2.1 Transitivity

In transitivity, we analyze the linking behavior to determine whether it demonstrates a transitive behavior. In mathematics, for a transitive relation R, $aRb \wedge bRc \rightarrow aRc$. The *transitive linking* behavior can be described as follows.

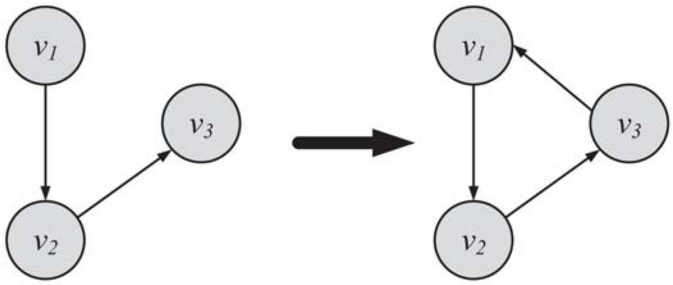


Figure 3.8: Transitive Linking.

Transitive Linking

Let v_1, v_2, v_3 denote three nodes. When edges (v_1, v_2) and (v_2, v_3) are formed, if (v_3, v_1) is also formed, then we have observed a transitive linking behavior (*transitivity*). This is shown in Figure 3.8.

In a less formal setting,

> *Transitivity is when a friend of my friend is my friend.*

As shown in the definition, a transitive behavior needs at least three edges. These three edges, along with the participating nodes, create a triangle. Higher transitivity in a graph results in a denser graph, which in turn is closer to a complete graph. Thus, we can determine how close graphs are to the complete graph by measuring transitivity. This can be performed by measuring the *[global] clustering coefficient* and *local clustering coefficient*. The former is computed for the network, whereas the latter is computed for a node.

Clustering Coefficient

The clustering coefficient analyzes transitivity in an undirected graph. Since transitivity is observed when triangles are formed, we can measure it by counting paths of length 2 (edges (v_1, v_2) and (v_2, v_3)) and checking whether the third edge (v_3, v_1) exists (i.e., the path is closed). Thus, clustering coefficient C is defined as

$$C = \frac{|\text{Closed Paths of Length 2}|}{|\text{Paths of Length 2}|}. \tag{3.48}$$

Alternatively, we can count triangles

$$C = \frac{(\text{Number of Triangles}) \times 6}{|\text{Paths of Length 2}|}. \tag{3.49}$$

Since every triangle has six closed paths of length 2, we can rewrite Equation 3.49 as

$$C = \frac{(\text{Number of Triangles}) \times 3}{\text{Number of Connected Triples of Nodes}}. \tag{3.50}$$

In this equation, a triple is an ordered set of three nodes, connected by two (i.e., open triple) or three (closed triple) edges. Two triples are different when

- their nodes are different, or
- their nodes are the same, but the triples are missing different edges.

For example, triples $v_iv_jv_k$ and $v_jv_kv_i$ are different, since the first triple is missing edge $e(v_k, v_i)$ and the second triple is missing edge $e(v_i, v_j)$, even though they have the same members. Following the same argument, triples $v_iv_jv_k$ and $v_kv_jv_i$ are the same, because both are missing edge $e(v_k, v_i)$ and have the same members. Since triangles have three edges, one edge can be missed in each triple; therefore, three different triples can be formed from one triangle. The number of triangles are therefore multiplied by a factor of 3 in the numerator of Equation 3.50. Note that the clustering coefficient is computed for the whole network.

Example 3.11. *For the graph in Figure 3.9, the clustering coefficient is*

$$\begin{aligned} C &= \frac{\textit{(Number of Triangles)} \times 3}{\textit{Number of Connected Triples of Nodes}} \\ &= \frac{2 \times 3}{2 \times 3 + \underbrace{2}_{v_2v_1v_4, v_2v_3v_4}} = 0.75. \end{aligned} \tag{3.51}$$

The clustering coefficient can also be computed locally. The following subsection discusses how it can be computed for a single node.

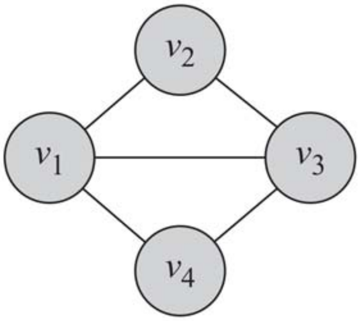


Figure 3.9: A Global Clustering Coefficient Example.

Local Clustering Coefficient

The local clustering coefficient measures transitivity at the node level. Commonly used for undirected graphs, it estimates how strongly neighbors of a node v (nodes adjacent to v) are themselves connected. The coefficient is defined as

$$C(v_i) = \frac{\text{Number of Pairs of Neighbors of } v_i \text{ That Are Connected}}{\text{Number of Pairs of Neighbors of } v_i}. \quad (3.52)$$

In an undirected graph, the denominator can be rewritten as $\binom{d_i}{2} = d_i(d_i - 1)/2$, since there are d_i neighbors for node v_i.

Example 3.12. *Figure 3.10 shows how the local clustering coefficient changes for*

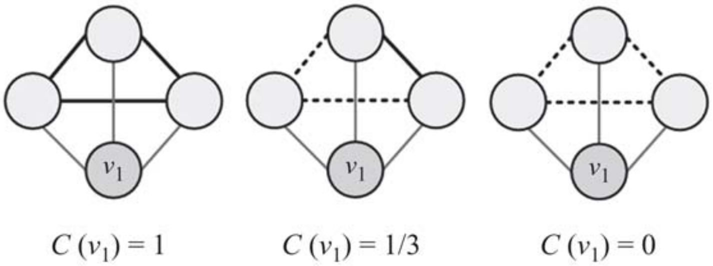


Figure 3.10: Change in Local Clustering Coefficient for Different Graphs. Thin lines depict connections to neighbors. Solid lines indicate connected neighbors, and dashed lines are the missing connections among neighbors.

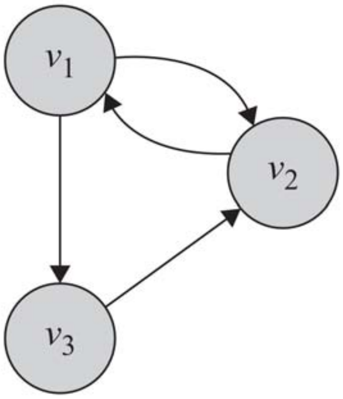


Figure 3.11: A Graph with Reciprocal Edges.

node v_1. Thin lines depict v_1's connections to its neighbors. Dashed lines denote possible connections among neighbors, and solid lines denote current connections among neighbors. Note that when none of the neighbors are connected, the local clustering coefficient is zero, and when all the neighbors are connected, it becomes maximum, $C(v_i) = 1$.

3.2.2 Reciprocity

Reciprocity is a simplified version of transitivity, because it considers closed loops of length 2, which can only happen in directed graphs. Formally, if node v is connected to node u, u by connecting to v exhibits reciprocity. On microblogging site Tumblr, for example, these nodes are known as "mutual followers." Informally, reciprocity is

If you become my friend, I'll be yours.

Figure 3.11 shows an example where two nodes (v_1 and v_2) in the graph demonstrate reciprocal behavior.

Reciprocity counts the number of reciprocal pairs in the graph. Any directed graph can have a maximum of $|E|/2$ pairs. This happens when all edges are reciprocal. Thus, this value can be used as a normalization factor. Reciprocity can be computed using the adjacency matrix A:

$$R = \frac{\sum_{i,j,i<j} A_{i,j} A_{j,i}}{|E|/2},$$

$$
\begin{aligned}
&= \frac{2}{|E|}\sum_{i,j,i<j} A_{i,j}A_{j,i},\\
&= \frac{2}{|E|}\times\frac{1}{2}\mathrm{Tr}(A^2),\\
&= \frac{1}{|E|}\mathrm{Tr}(A^2),\\
&= \frac{1}{m}\mathrm{Tr}(A^2),
\end{aligned}
\tag{3.53}
$$

where $\mathrm{Tr}(A) = A_{1,1} + A_{2,2} + \cdots + A_{n,n} = \sum_{i=1}^{n} A_{i,i}$ and m is the number of edges in the network. Note that the maximum value for $\sum_{i,j} A_{i,j}A_{j,i}$ is m when all directed edges are reciprocated.

Example 3.13. *For the graph shown in Figure 3.11, the adjacency matrix is*

$$
A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{3.54}
$$

Its reciprocity is

$$
R = \frac{1}{m}\mathrm{Tr}(A^2) = \frac{1}{4}\mathrm{Tr}\left(\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}\right) = \frac{2}{4} = \frac{1}{2}. \tag{3.55}
$$

3.3 Balance and Status

A signed graph can represent the relationships of nodes in a social network, such as friends or foes. For example, a positive edge from node v_1 to v_2 denotes that v_1 considers v_2 as a friend and a negative edge denotes that v_1 assumes v_2 is an enemy. Similarly, we can utilize signed graphs to represent the social status of individuals. A positive edge connecting node v_1 to v_2 can also denote that v_1 considers v_2's status higher than its own in the society. Both cases represent interactions that individuals exhibit about their relationships. In real-world scenarios, we expect some level of consistency with respect to these interactions. For instance, it is more plausible for a friend of one's friend to be a friend than to be an enemy. In signed graphs, this consistency translates to observing triads with three positive edges (i.e., all friends) more frequently than ones with two positive edges and one negative edge (i.e., a friend's friend is an enemy). Assume we observe a signed graph that represents friends/foes or social status. Can we measure the consistency of attitudes that individual have toward one another?

To measure consistency in an individual's attitude, one needs to utilize theories from social sciences to define what is a consistent attitude. In this section, we discuss two theories, *social balance* and *social status*, that can help determine consistency in observed signed networks. Social balance theory is used when edges represent friends/foes, and social status theory is employed when they represent status.

Social Balance Theory

This theory, also known as *structural balance theory*, discusses consistency in friend/foe relationships among individuals. Informally, social balance theory says friend/foe relationships are consistent when

Structural Balance Theory

> *The friend of my friend is my friend,*
> *The friend of my enemy is my enemy,*
> *The enemy of my enemy is my friend,*
> *The enemy of my friend is my enemy.*

We demonstrate a graph representation of social balance theory in Figure 3.12. In this figure, positive edges demonstrate friendships and negative ones demonstrate enemies. Triangles that are consistent based on this theory are denoted as *balanced* and triangles that are inconsistent as *unbalanced*. Let w_{ij} denote the value of the edge between nodes v_i and v_j. Then, for a triangle of nodes v_i, v_j, and v_k, it is consistent based on social balance theory; that is, it is balanced if and only if

Balanced and Unbalanced Triangles

$$w_{ij}w_{jk}w_{ki} \geq 0. \tag{3.56}$$

This is assuming that, for positive edges, $w_{ij} = 1$, and for negative edges, $w_{ij} = -1$. We observe that, for all balanced triangles in Figure 3.12, the value $w_{ij}w_{jk}w_{ki}$ is positive, and for all unbalanced triangles, it is negative. Social balance can also be generalized to subgraphs other than triangles. In general, for any cycle, if the product of edge values becomes positive, then the cycle is socially balanced.

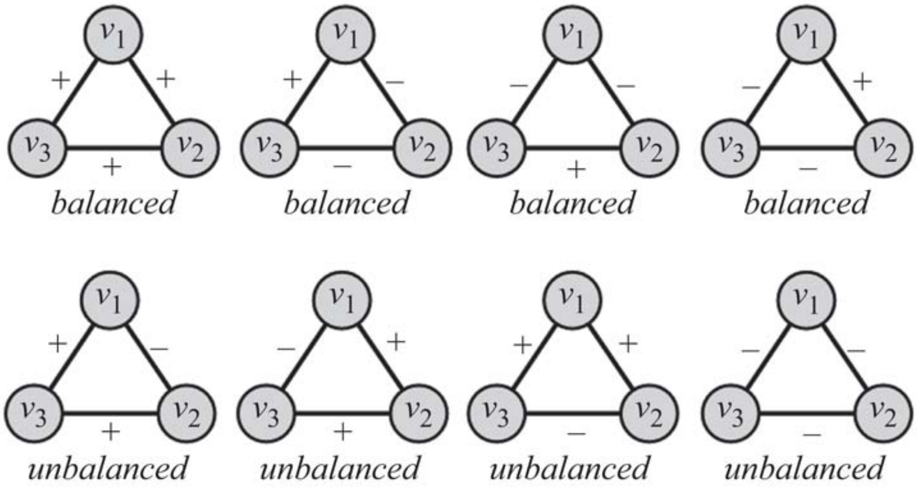


Figure 3.12: Sample Graphs for Social Balance Theory. In balanced triangles, there are an even number of negative edges.

Social Status Theory

Social status theory measures how consistent individuals are in assigning status to their neighbors. It can be summarized as follows:

> *If X has a higher status than Y and Y has a higher status than Z, then X should have a higher status than Z.*

We show this theory using two graphs in Figure 3.13. In this figure, nodes represent individuals. Positive and negative signs show higher or lower status depending on the arrow direction. A directed positive edge from node X to node Y shows that Y has a higher status than X, and a negative one shows the reverse. In the figure on the left, v_2 has a higher status than v_1 and v_3 has a higher status than v_2, so based on status theory, v_3 should have a higher status than v_1; however, we see that v_1 has a higher status in our configuration.[4] Based on social status theory, this is implausible, and thus this configuration is *unbalanced*. The graph on the right shows a balanced configuration with respect to social status theory.

[4] Here, we start from v_1 and follow the edges. One can start from a different node, and the result should remain the same.

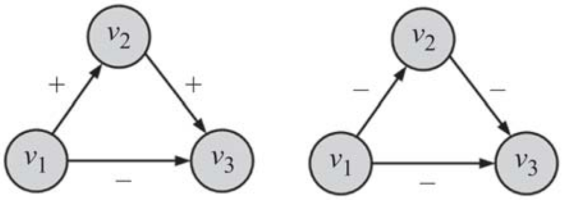


Figure 3.13: Sample Graphs for Social Status Theory. The left-hand graph is an unbalanced configuration, and the right-hand graph is a balanced configuration.

In the example provided in Figure 3.13, social status is defined for the most general example: a set of three connected nodes (a triad). However, social status can be generalized to other graphs. For instance, in a cycle of n nodes, where $n - 1$ consecutive edges are positive and the last edge is negative, social status theory considers the cycle balanced.

Note that the identical configuration can be considered balanced by social balance theory and unbalanced based on social status theory (see Exercises).

3.4 Similarity

In this section, we review measures used to compute similarity between two nodes in a network. In social media, these nodes can represent individuals in a friendship network or products that are related. The similarity between these connected individuals can be computed either based on the network in which they are embedded (i.e., *network similarity*) or based on the similarity of the content they generate (i.e., *content similarity*). We discuss content similarity in Chapter 5. In this section, we demonstrate ways to compute similarity between two nodes using network information regarding the nodes and edges connecting them. When using network information, the similarity between two nodes can be computed by measuring their *structural equivalence* or their *regular equivalence*.

3.4.1 Structural Equivalence

To compute structural equivalence, we look at the neighborhood shared by two nodes; the size of this neighborhood defines how similar two nodes are. For instance, two brothers have in common sisters, mother, father, grandparents, and so on. This shows that they are similar, whereas two random male or female individuals do not have much in common and are not similar.

The similarity measures detailed in this section are based on the overlap between the neighborhoods of the nodes. Let $N(v_i)$ and $N(v_j)$ be the neighbors of nodes v_i and v_j, respectively. In this case, a measure of node similarity can be defined as follows:

$$\sigma(v_i, v_j) = |N(v_i) \cap N(v_j)|. \tag{3.57}$$

For large networks, this value can increase rapidly, because nodes may share many neighbors. Generally, similarity is attributed to a value that is bounded and is usually in the range $[0, 1]$. Various normalization procedures can take place such as the Jaccard similarity or the cosine similarity:

Jaccard Similarity and Cosine Similarity

$$\sigma_{\text{Jaccard}}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}, \tag{3.58}$$

$$\sigma_{\text{Cosine}}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{\sqrt{|N(v_i)||N(v_j)|}}. \tag{3.59}$$

In general, the definition of neighborhood $N(v_i)$ excludes the node itself (v_i). This leads to problems with the aforementioned similarities because nodes that are connected and do not share a neighbor will be assigned zero similarity. This can be rectified by assuming nodes to be included in their neighborhoods.

Example 3.14. *Consider the graph in Figure 3.14. The similarity values between nodes v_2 and v_5 are*

$$\sigma_{\text{Jaccard}}(v_2, v_5) = \frac{|\{v_1, v_3, v_4\} \cap \{v_3, v_6\}|}{|\{v_1, v_3, v_4, v_6\}|} = 0.25, \tag{3.60}$$

$$\sigma_{\text{Cosine}}(v_2, v_5) = \frac{|\{v_1, v_3, v_4\} \cap \{v_3, v_6\}|}{\sqrt{|\{v_1, v_3, v_4\}||\{v_3, v_6\}|}} = 0.40. \tag{3.61}$$

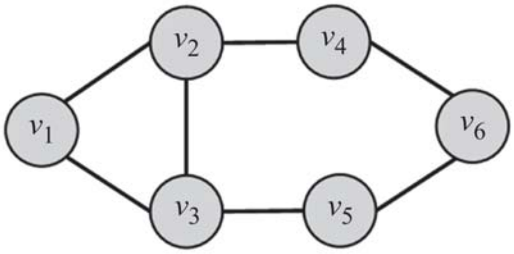


Figure 3.14: Sample Graph for Computing Similarity.

A more interesting way of measuring the similarity between v_i and v_j is to compare $\sigma(v_i, v_j)$ with the expected value of $\sigma(v_i, v_j)$ when nodes pick their neighbors at random. The more distant these two values are, the more significant the similarity observed between v_i and v_j ($\sigma(v_i, v_j)$) is. For nodes v_i and v_j with degrees d_i and d_j, this expectation is $\frac{d_i d_j}{n}$, where n is the number of nodes. This is because there is a $\frac{d_i}{n}$ chance of becoming v_i's neighbor and, since v_j selects d_j neighbors, the expected overlap is $\frac{d_i d_j}{n}$. We can rewrite $\sigma(v_i, v_j)$ as

$$\sigma(v_i, v_j) = |N(v_i) \cap N(v_j)| = \sum_k A_{i,k} A_{j,k}. \tag{3.62}$$

Hence, a similarity measure can be defined by subtracting the random expectation $\frac{d_i d_j}{n}$ from Equation 3.62:

$$\begin{aligned}
\sigma_{\text{significance}}(v_i, v_j) &= \sum_k A_{i,k} A_{j,k} - \frac{d_i d_j}{n} \\
&= \sum_k A_{i,k} A_{j,k} - n \frac{1}{n} \sum_k A_{i,k} \frac{1}{n} \sum_k A_{j,k} \\
&= \sum_k A_{i,k} A_{j,k} - n \bar{A}_i \bar{A}_j \\
&= \sum_k (A_{i,k} A_{j,k} - \bar{A}_i \bar{A}_j) \\
&= \sum_k (A_{i,k} A_{j,k} - \bar{A}_i \bar{A}_j - \bar{A}_i \bar{A}_j + \bar{A}_i \bar{A}_j) \\
&= \sum_k (A_{i,k} A_{j,k} - A_{i,k} \bar{A}_j - \bar{A}_i A_{j,k} + \bar{A}_i \bar{A}_j)
\end{aligned}$$

$$= \sum_k (A_{i,k} - \bar{A}_i)(A_{j,k} - \bar{A}_j), \qquad (3.63)$$

where $\bar{A}_i = \frac{1}{n}\sum_k A_{i,k}$. The term $\sum_k (A_{i,k} - \bar{A}_i)(A_{j,k} - \bar{A}_j)$ is basically the covariance between A_i and A_j. The covariance can be normalized by the multiplication of variances,

$$\begin{aligned} \sigma_{\text{pearson}}(v_i, v_j) &= \frac{\sigma_{\text{significance}}(v_i, v_j)}{\sqrt{\sum_k (A_{i,k} - \bar{A}_i)^2}\sqrt{\sum_k (A_{j,k} - \bar{A}_j)^2}} \\ &= \frac{\sum_k (A_{i,k} - \bar{A}_i)(A_{j,k} - \bar{A}_j),}{\sqrt{\sum_k (A_{i,k} - \bar{A}_i)^2}\sqrt{\sum_k (A_{j,k} - \bar{A}_j)^2}}, \qquad (3.64) \end{aligned}$$

Pearson Correlation

which is called the *Pearson correlation coefficient*. Its value, unlike the other two measures, is in the range $[-1, 1]$. A positive correlation value denotes that when v_i befriends an individual v_k, v_j is also likely to befriend v_k. A negative value denotes the opposite (i.e., when v_i befriends v_k, it is unlikely for v_j to befriend v_k). A zero value denotes that there is no linear relationship between the befriending behavior of v_i and v_j.

3.4.2 Regular Equivalence

In regular equivalence, unlike structural equivalence, we do not look at the neighborhoods shared between individuals, but at how neighborhoods themselves are similar. For instance, athletes are similar not because they know each other in person, but because they know similar individuals, such as coaches, trainers, and other players. The same argument holds for any other profession or industry in which individuals might not know each other in person, but are in contact with very similar individuals. Regular equivalence assesses similarity by comparing the similarity of neighbors and not by their overlap.

One way of formalizing this is to consider nodes v_i and v_j similar when they have many similar neighbors v_k and v_l. This concept is shown in Figure 3.15(a). Formally,

$$\sigma_{\text{regular}}(v_i, v_j) = \alpha \sum_{k,l} A_{i,k} A_{j,l} \sigma_{\text{regular}}(v_k, v_l). \qquad (3.65)$$

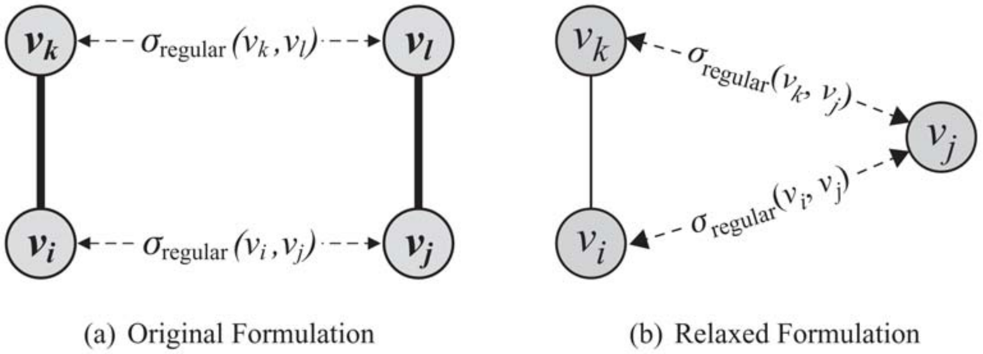

v_k
$\sigma_{\text{regular}}(v_k, v_l)$
v_l
v_i
$\sigma_{\text{regular}}(v_i, v_j)$
v_j
(a) Original Formulation
v_k
$\sigma_{\text{regular}}(v_k, v_j)$
v_j
v_i
$\sigma_{\text{regular}}(v_i, v_j)$
(b) Relaxed Formulation

Figure 3.15: Regular Equivalence. Solid lines denote edges, and dashed lines denote similarities between nodes. In regular equivalence, similarity between nodes v_i and v_j is replaced by similarity between (a) their neighbors v_k and v_l or between (b) neighbor v_k and node v_j.

Unfortunately, this formulation is self-referential because solving for i and j requires solving for k and l, solving for k and l requires solving for their neighbors, and so on. So, we relax this formulation and assume that node v_i is similar to node v_j when v_j is similar to v_i's neighbors v_k. This is shown in Figure 3.15(b). Formally,

$$\sigma_{\text{regular}}(v_i, v_j) = \alpha \sum_k A_{i,k} \sigma_{\text{regular}}(v_k, v_j). \tag{3.66}$$

In vector format, we have

$$\sigma_{\text{regular}} = \alpha A \sigma_{\text{regular}}. \tag{3.67}$$

A node is highly similar to itself. To make sure that our formulation guarantees this, we can add an identity matrix to this vector format. Adding an identity matrix will add 1 to all diagonal entries, which represent self-similarities $\sigma_{\text{regular}}(v_i, v_i)$:

$$\sigma_{\text{regular}} = \alpha A \sigma_{\text{regular}} + \mathbf{I}. \tag{3.68}$$

By rearranging, we get

$$\sigma_{\text{regular}} = (\mathbf{I} - \alpha A)^{-1}, \tag{3.69}$$

which we can use to find the regular equivalence similarity.

Note the similarity between Equation 3.69 and that of Katz centrality (Equation 3.21). As with Katz centrality, we must be careful how we choose α for convergence. A common practice is to select an α such that $\alpha < 1/\lambda$, where λ is the largest eigenvalue of A.

Example 3.15. *For the graph depicted in Figure 3.14, the adjacency matrix is*

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{3.70}$$

The largest eigenvalue of A *is 2.43. We set* $\alpha = 0.4 < 1/2.43$*, and we compute* $(I - 0.4A)^{-1}$*, which is the similarity matrix,*

$$\sigma_{\text{regular}} = (I - 0.4A)^{-1} = \begin{bmatrix} 1.43 & 0.73 & 0.73 & 0.26 & 0.26 & 0.16 \\ 0.73 & 1.63 & 0.80 & 0.56 & 0.32 & 0.26 \\ 0.73 & 0.80 & 1.63 & 0.32 & 0.56 & 0.26 \\ 0.26 & 0.56 & 0.32 & 1.31 & 0.23 & 0.46 \\ 0.26 & 0.32 & 0.56 & 0.23 & 1.31 & 0.46 \\ 0.16 & 0.26 & 0.26 & 0.46 & 0.46 & 1.27 \end{bmatrix}. \tag{3.71}$$

Any row or column of this matrix shows the similarity of a node to other nodes. We can see that node v_1 *is the most similar (other than itself) to nodes* v_2 *and* v_3*. Furthermore, nodes* v_2 *and* v_3 *have the highest similarity in this graph.*

3.5 Summary

In this chapter, we discussed measures for a social media network. Centrality measures attempt to find the most central node within a graph. Degree centrality assumes that the node with the maximum degree is the most central individual. In directed graphs, prestige and gregariousness are variants of degree centrality. Eigenvector centrality generalizes degree centrality and considers individuals who know many important nodes as central. Based on the Perron-Frobenius theorem, eigenvector centrality is determined by computing the eigenvector of the adjacency matrix. Katz centrality solves some of the problems with eigenvector centrality in directed graphs by adding a bias term. PageRank centrality defines a normalized version of Katz centrality. The Google search engine uses PageRank as a measure to rank webpages. Betweenness centrality assumes that central nodes act as hubs connecting other nodes, and closeness centrality implements the intuition that central nodes are close to all other nodes. Node centrality measures can be generalized to a group of nodes using group degree centrality, group betweenness centrality, and group closeness centrality.

Linking between nodes (e.g., befriending in social media) is the most commonly observed phenomenon in social media. Linking behavior is analyzed in terms of its transitivity and its reciprocity. Transitivity is “when a friend of my friend is my friend.” The transitivity of linking behavior is analyzed by means of the clustering coefficient. The global clustering coefficient analyzes transitivity within a network, and the local clustering coefficient performs that for a node. Transitivity is commonly considered for closed triads of edges. For loops of length 2, the problem is simplified and is called reciprocity. In other words, reciprocity is when “if you become my friend, I'll be yours.”

To analyze if relationships are consistent in social media, we used various social theories to validate outcomes. Social balance and social status are two such theories.

Finally, we analyzed node similarity measures. In structural equivalence, two nodes are considered similar when they share neighborhoods. We discussed cosine similarity and Jaccard similarity in structural equivalence. In regular equivalence, nodes are similar when their neighborhoods are similar.

3.6 Bibliographic Notes

General reviews of different measures in graphs, networks, the web, and social media can be found in [212, 304, 270, 120, 294].

A more detailed description of the PageRank algorithm can be found in [224, 174]. In practice, to compute the PageRank values, the *power iteration method* is used. Given a matrix A, this method produces an eigenvalue λ and an eigenvector v of A. In the case of PageRank, eigenvalue λ is set to 1. The iterative algorithm starts with an initial eigenvector v_0 and then, v_{k+1} is computed from v_k as follows,

$$v_{k+1} = Av_k. \tag{3.72}$$

The iterative process is continued until $v_k \approx v_{k+1}$ (i.e., convergence occurs). Other similar techniques to PageRank for computing influential nodes in a webgraph, such as the HITS [150] algorithm, can be found in [51, 153]. Unlike PageRank, the HITS algorithm[5] considers two types of nodes: authority nodes and hub nodes. An authority is a webpage that has many in-links. A hub is a page with many out-links. Authority pages have in-links from many hubs. In other words, hubs represent webpages that contain many useful links to authorities and authorities are influential nodes in the webgraph. HITS employs an iterative approach to compute authority and hub scores for all nodes in the graph. Nodes with high authority scores are classified as authorities and nodes with high hub scores as hubs. Webpage with high authority scores or hub scores can be recommended to users in a web search engine.

Betweenness algorithms can be improved using all-pair shortest paths algorithms [293] or algorithms optimized for computing betweenness, such as the Brandes' algorithm discussed in [45, 278].

A review of node similarity and normalization procedures is provided in [166]. Jaccard similarity was introduced in [133] and cosine similarity is introduced by Salton and McGill [244].

REGE [302, 303] and CATREGE [264] are well-known algorithms for computing regular equivalence.

[5] HITS stands for hypertext-induced topic search.

3.7 Exercises

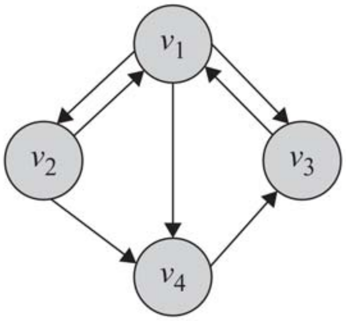
Centrality

1. Come up with an example of a directed connected graph in which eigenvector centrality becomes zero for some nodes. Describe when this happens.

2. Does β have any effect on the order of centralities? In other words, if for one value of β the centrality value of node v_i is greater than that of v_j, is it possible to change β in a way such that v_j's centrality becomes larger than that of v_i's?

3. In PageRank, what α values can we select to guarantee that centrality values are calculated correctly (i.e., values do not diverge)?

4. Calculate PageRank values for this graph when



   - $\alpha = 1, \beta = 0$
   - $\alpha = 0.85, \beta = 1$
   - $\alpha = 0, \beta = 1$

   Discuss the effects of different values of α and β for this particular problem.

5. Consider a full n-tree. This is a tree in which every node other than the leaves has n children. Calculate the betweenness centrality for the root node, internal nodes, and leaves.

6. Show an example where the eigenvector centrality of all nodes in the graph is the same while betweenness centrality gives different values for different nodes.

Transitivity and Reciprocity

7. In a directed graph $G(V, E)$,
    - Let p be the probability that any node v_i is connected to any node v_j. What is the expected reciprocity of this graph?
    - Let m and n be the number of edges and number of nodes, respectively. What is the maximum reciprocity? What is the minimum?
8. Given all graphs $\{G(V, E)|s.t., |E| = m, |V| = n\}$,
    (a) When $m = 15$ and $n = 10$, find a graph with a minimum average clustering coefficient (one is enough).
    (b) Can you come up with an algorithm to find such a graph for any m and n?

Balance and Status

9. Find all conflicting directed triad configurations for social balance and social status. A conflicting configuration is an assignment of positive/negative edge signs for which one theory considers the triad balanced and the other considers it unbalanced.

Similarity

10. In Figure 3.6,
    - Compute node similarity using Jaccard and cosine similarity for nodes v_5 and v_4.
    - Find the most similar node to v_7 using regular equivalence.