Lecture 3: Eigenvector Centrality Measures

Alexander Wolitzky

MIT

6.207/14.15: Networks, Spring 2022

Eigenvector-Based Centrality Measures

Last week, we introduced several different measures of the "centrality" of a node in a network.

- Degree centrality, closeness centrality, betweeness centrality.

Another, very important class of centrality measures are based on the idea that a node is important if it is connected to other important nodes.

This week's lectures introduce such **eigenvector-based centrality measures**, along with 3 important applications:

- How Google ranks webpages (PageRank).
- Which agents in a social network are influential in forming the group's long-run consensus opinion (DeGroot learning).
- Which firms in a production network are most systemically important (Leontief input-output analysis).

Setup

Recall that we have a network with a set of nodes $N = \{1, \ldots, n\}$ and adjacency matrix $g = [g_{ij}]_{i,j \in N}$, where $g_{ij} = 1$ indicates a link from i to j, and $g_{ij} = 0$ indicates no such link.

We want a measure of the importance of a node, whereby a node is important if other import nodes link to it.

- E.g. an important website is one that many other important websites link to.

Eigenvector Centrality

The simplest such measure is **eigenvector centrality**: a non-zero vector $c = (c_i)_{i \in N}$ such that, for some scalar $\lambda > 0$, we have

$$\lambda c_i = \sum_{j \neq i} g_{ji} c_j \quad \text{for all } i \in N.$$

That is, the centrality of each node i is proportional to the sum of the centrality of its neighbors.

- Note that in this definition we have g_{ji} rather than g_{ij}.
- This doesn't matter for undirected graphs. For directed graphs, it says that a node's centrality derives from the centrality of nodes that *point to it*.
- Interpretation: when "important" or "prestigous" nodes point to you, this makes you important/prestigious.
- Equations still hold if we multiply c by a scalar. We typically normalize c so that $\sum_{i \in N} c_i = 1$.

Example

Suppose $n = 3$ and

$$g = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Then eigenvector centrality (with the normalization $\sum_{i \in N} c_i = 1$) is defined as the solution to the system of equations

$$\begin{aligned} \lambda c_1 &= c_2 \\ \lambda c_2 &= c_1 \\ \lambda c_3 &= c_1 + c_2 \\ c_1 + c_2 + c_3 &= 1. \end{aligned}$$

Solving this system gives

$$\lambda = 1, \quad c_1 = c_2 = \frac{1}{4}, \quad c_3 = \frac{1}{2}.$$

Eigenvector Centrality (cntd.)

Eigenvector centrality $(c_i)_{i \in N}$ is defined by

$$\begin{aligned} \lambda c_i &= \sum_{j \neq i} g_{ji} c_j \quad \text{for all } i \in N, \\ c &\neq 0. \end{aligned}$$

It's not immediately obvious whether we can find such a vector c: that is, whether such a measure exists or is unique.

- n linear equations with n unknowns, so looks promising...

When is Eigenvector Centrality Well-Defined?

For strongly connected networks, it turns out that eigenvector centrality is always well-defined.

- Recall that a directed network is strongly connected if there exists a directed path between any two nodes.
- In particular, every connected undirected network is strongly connected.
- In general, a network is strongly connected if for every pair of nodes i, j, there exists a number ℓ such that $\left(g^{\ell}\right)_{ij} > 0$.
  - Matrices g with this property are called **irreducible**.
  - That is, a network is strongly connected if and only if its adjacency matrix is irreducible.

When is Eigenvector Centrality Well-Defined? (cntd.)

In matrix form, the equation for the c is

$$\lambda c = g' c,$$

where λ is a scalar, c is a $n \times 1$ vector, and g' is the transpose of the $n \times n$ adjacency matrix (transposed because we want $\sum_{j \neq i} g_{ji} c_j$ on the RHS: for directed graphs, we care about the nodes that link to you, not the nodes you link to).

- That is, c is an eigenvector of g', with λ the corresponding eigenvalue.
- The Perron-Frobenius theorem of linear algebra says that, for every irreducible non-negative matrix, its largest eigenvalue is positive, and the components of the corresponding eigenvector are also all positive.
- So, $\lambda c = g' c$ has a positive solution: λ is the largest eigenvalue of g', and c is the corresponding eigenvector.
- **Punchline:** for any strongly[8] connected network, the eigenvector centrality vector c is well-defined (and positive).

Interpretation as Long-Run Population Shares

Useful interpretation of eigenvector centrality as the long-run outcome of a reproduction process (which also explains why it's always well-defined for strongly connected networks):

- Suppose a "virus" starts at a random node in the graph.
- In each period, every virus sends one copy of itself along each link from the node where it is located. Then it dies.
- (So there's 1 virus in period 1, $|N_i|$ viruses in period 2, $\sum_{j \in N_i} |N_j|$ viruses in period 3, etc.)
- Letting this process run forever, the virus never dies out (because the network is strongly connected), and we can calculate the long-run fraction of viruses located at each node.
- The long-run fraction of viruses located at node i equals c_i.

(Why? Because the long-run fraction of viruses located at node i is proportional to the long-run fraction of viruses located at nodes that link to node i. This is the relationship that defines eigenvector centrality.)

A Caveat about this Interpretation

Since the network is strongly connected, the virus never dies out.

But, does the fraction of viruses located at each node have to converge, or can it perhaps cycle forever?

If we assume only that A is irreducible, the fraction of viruses at each node could cycle forver.

- E.g. the matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is irreducible, but the fraction of viruses at node 1 bounces back and forth between 0 and 1.

This is just a glitch in the virus interpetation, not in the definition of eigenvector centrality or the subsequent analysis.

- We'll return to this issue in Lecture 5, when we'll also see a simple additional condition on A that guarantees convergence rather than cycling.

Perron-Frobenius Theorem

Theorem

For every irreducible non-negative matrix A, its largest eigenvalue λ_1 is a positive real number, and the components of the corresponding eigenvector v_1 are also all positive.

The theorem says more than this, but this is what we need.

The proof is outside our scope, but we can give an informative informal argument.

Intuition for the Perron-Frobenius Theorem

- Fix any non-negative vector $x(0) \in \mathbb{R}^n$. Suppose that we can write it as a linear combination of the eigenvectors v_i of A:

$$x(0) = \sum_i \alpha_i v_i.$$

- Consider repeatedly multipying $x(0)$ by A.
(Matrix multiplication = copying viruses...)
After t steps, we get the vector

$$x(t) = A^t x(0) = A^t \sum_i \alpha_i v_i = \sum_i \alpha_i \lambda_i^t v_i = \lambda_1^t \sum_i \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^t v_i.$$

- Since λ_1 is the largest eigenvalue, $\left(\frac{\lambda_i}{\lambda_1}\right)^t \to 0$ as $t \to \infty$, for all $i \neq 1$. Therefore, $x(t)/\lambda_1^t \to \alpha_1 v_1$. That is, the limiting vector $x(\infty)$ is proportional to the largest eigenvector.
- Since $x(0)$ was non-negative and A is non-negative, each $x(t)$ is also non-negative. Therefore, λ_1 must be positive (else oscillates), and every component of v_1 must also be positive.

Other Insights from this Argument

Just like with the viruses, the limiting vector $x(\infty)$ is proportional to the largest eigenvector.

Normalizing this eigenvector so its components sum to 1 gives eigenvector centrality.

Aside: We might also ask *how fast* $x(t)$ converges to $x(\infty)$.

- This is determined by how fast $\left(\frac{\lambda_2}{\lambda_1}\right)^t$ goes to 0 as $t \to \infty$ (since $\left(\frac{\lambda_i}{\lambda_1}\right)^t$ goes to 0 faster than this for all $i \geq 3$).
- Bigger gap between first and second eigenvalue $\implies$ faster convergence.
- We won't study this question, but FYI this explains why the second eigenvalue often plays a role in analysis of networks / Markov chains.

Problems with Eigenvector Centrality in Directed Networks

Eigenvector centrality is well-defined for strongly connected directed networks, but for directed networks that are not strongly connected the only solution c to the required equations equals 0 for a large number of nodes (or even all of them).

- E.g. if the network is a directed line, eventually the virus reaches the end of the line, and then it dies without making any copies.

Many important directed networks are not strongly connected.

- Example: the Web. A large fraction of webpages are contained in a large strongly connected component of the Web, but many other webpages lie "upstream" or "downstream" of this strongly connected component (i.e., in its in-component or out-component).

Problems in Directed Networks (cntd.)

If a directed network is not strongly connected, only nodes that lie either in a strongly connected component or its out-component can have positive eigenvector centrality.

- All other nodes have only out-links, or have in-links only from nodes that have only out-links, or have in-links only from nodes that have in-links from nodes that have only out-links, etc..
- Such nodes have 0 eigenvector centrality. Why? (Think of the viruses.)

This can be a big problem with using eigenvector centrality to assign importance in directed networks.

- If Google used eigenvector centrality to determine webpages' importance/rank, a large fraction of webpages would have zero importance and thus would never appear on its search results!
- This problem motivates introducing our next measure (which also has many other uses)...

Katz-Bonacich Centrality

A way to "fix" eigenvector centrality to make it more useful for directed networks is to first give each node a certain amount of centrality β "for free".

We thus seek a non-negative vector c and scalar λ such that

$$c_i = \frac{1}{\lambda} \sum_{j \neq i} g_{ji} c_j + \beta \quad \text{for all } i \in N,$$

or in matrix form

$$c = \frac{1}{\lambda} g' c + \beta.$$

Solving this equation for c gives

$$c = \beta \left(I - \frac{1}{\lambda} g' \right)^{-1} \mathbf{1},$$

where I is the $n \times n$ identity matrix and $\mathbf{1}$ is the $n \times 1$ vector of 1's. This is called the vector of **Katz-Bonacich centralities with parameter** $1/\lambda$.

Katz-Bonacich Centrality (cntd.)

If we write α for $1/\lambda$, Katz-Bonacich centrality with parameter α is defined by

$$c = \beta \left(I - \alpha g'\right)^{-1} \mathbf{1},$$

Note:

1. The choice of β just scales the vector c. For this reason, we typically take $\beta = 1$ when discussing Katz-Bonacich centrality. However, note that this is a different normalization than taking $\sum_i c_i = 1$!
2. On the other hand, different choices for α yield different vectors c, which is why properly we speak of Katz-Bonacich centrality *with parameter* α.
   When discussing Katz-Bonacich centrality, the parameter α is typically called the **decay parameter**.

Katz-Bonacich Centrality and the Leontief Inverse

An (important) aside: What's the meaning of the matrix $(I - \alpha g')^{-1}$ in the formula for Katz-Bonacich centrality?

For any matrix A, the matrix $\Lambda = (I - \alpha A)^{-1}$ is called the **Leontief inverse of A with parameter** α.

- Thus, with the standard normalization $\beta = 1$, Katz-Bonacich centrality is defined as $\Lambda \mathbf{1}$, where Λ is the Leontief inverse of g'.

To understand the Leontief inverse, note that we can write

$$(I - \alpha A)^{-1} = I + \alpha A + \alpha^2 A^2 + \ldots.$$

(Why? It's much like a geometric series. Let $S_n = \sum_{k=0}^{n} (\alpha A)^k$. Then we have

$$(I - \alpha A) S_n = \sum_{k=0}^{n} (\alpha A)^k - \sum_{k=0}^{n} (\alpha A)^{k+1} = I - (\alpha A)^{n+1} \to_{n \to \infty} I.$$

Pre-multiply both sides by $(I - \alpha A)^{-1}$ to get $S_\infty = (I - \alpha A)^{-1}$.)

Leontief Inverse (cntd.)

We have

$$\Lambda = (I - \alpha A)^{-1} = I + \alpha A + \alpha^2 A^2 + \dots .$$

So, for any $i \neq j$,

$$\Lambda_{ij} = \alpha a_{ij} + \alpha^2 \sum_{k=1}^{n} a_{ik} a_{kj} + \dots .$$

That is, Λ_{ij} is the sum over all lengths ℓ of the value of all length-ℓ walks from i to j, where the value of a walk is the product of the weights on the links, and length-ℓ walks are "discounted" by α^ℓ.

- Longer walks get less weight. The decay parameter says how much less.

Intuitively, if a_{ij} is the "direct influence" of i on j, then Λ_{ij} is the "sum of the direct **and indirect** influence" of i on j, where "indirect influence" is via all arbitrarily long walks, with longer walks discounted according to the decay parameter.

Leontief Inverse (cntd.)

The idea that the Leontief inverse Λ captures the sum of direct and indirect influences is a little subtle at first, but it's the key to understanding what Katz-Bonacich centrality and similar measures are and why they're useful.

We'll an application of Katz-Bonacich centrality next class, but for now move on (finally!) to describing PageRank.

From Katz-Bonacich Centrality to PageRank

Katz-Bonacich centrality is similar to PageRank, the score that Google assigns to webpages to determine its search ranking.

- Brin and Page's key insight that allowed Google to take over the market in the early days of websearch: rather than appealing to some external authority to rank webpages, assign "importance" to webpages that are linked to by other "important" webpages.

However, if Google had simply used Katz-Bonacich centrality, they would have had a big problem: any webpage that is linked to by an important webpage would be assigned high importance.

- E.g. every obscure internet seller that sells on Amazon would be highly ranked.

PageRank (cntd.)

To avoid this, PageRank modifies Katz-Bonacich centrality by normalizing the adjacency matrix by nodes' out-degrees.

The vector of PageRank scores c (with parameter α) is given by

$$c_i = \alpha \sum_{j \neq i} \frac{g_{ji}}{d_j^{out}} c_j + 1 \quad \text{for all } i \in N,$$

where by convention d_j^{out} is set to 1 if node j has out-degree 0.

In matrix form, this gives

$$c = \alpha g' D^{-1} c + \mathbf{1},$$

where D is the diagonal matrix with entries $D_{ii} = \max\{d_i^{out}, 1\}$.

PageRank (cntd.)

Solving this matrix equation gives

$$c = \left(I - \alpha g' D^{-1}\right)^{-1} \mathbf{1}.$$

This is the vector of **PageRanks with parameter α.**

Note that α is a free parameter that can be used to tune the algorithm.

Because we understand what the Leontief inverse $\left(I - \alpha g' D^{-1}\right)^{-1}$ means, we can understand the tradeoff that Google faces in tuning the parameter α.

- α is the decay parameter in the Leontief inverse.
- The higher is α, the more weight PageRank puts on indirect links rather than direct links.
- In practice, Google sets $\alpha = 0.85$.

Let's see another way of looking at this...

Interpretation as Long-Run Frequencies

An interpretation of PageRank (similar but not the same as "viruses"):

- Suppose a "web surfer" starts at a random node in the graph.
- Each period, with probability α the surfer follows a random link from the node where she is currently located, and with probability $1 - \alpha$ she jumps to a random node in the graph.
- Letting this process run forever, we can calculate the long-run fraction of periods in which the surfer is located at each node.
- The long-run fraction of periods in which the surfer is located at node i equals $c_i \left(1 - \alpha\right) / n$ (=PageRank, normalized so the PageRanks of all nodes sum to 1).

Note: Like the "virus" interpretation of eigenvector centrality, but without reproduction (=normalizing by out-degree) and with jumps (="free" centrality). Also, the jumps rule out cycling, so now the process always converges.

Tuning the Free Parameter

This interpretation helps us understand how Google chooses a value for the free parameter α.

- Choosing α close to 1 introduces less noise (that is, puts more weight on the "eigenvector" term in the equation for c). The downside is that, as $\alpha \to 1$, PageRank converges to 0 for nodes that are upstream of all strongly connected components (just like eigenvector centrality).
- Choosing α close to 0 introduces more noise (that is, puts more weight on the constant term in the equation for c). The downside is that, as $\alpha \to 0$, PageRank converges to the vector of 1's, and thus loses all information about the network.

Google sets $\alpha = 0.85$.

So, PageRank=long-run frequency of a web surfer who follows links with probability .85 and randomly jumps with probability .15.

Computation

Finally, the web surfer interpretation also indicates how PageRank can be computed iteratively.

- At $t = 0$, initialize $c_i(0) = 1/n$ for each node i.
- For each t, given $(c_i(t))_{i \in N}$, compute $(c_i(t+1))_{i \in N}$ by

$$c_i(t+1) = \frac{1-\alpha}{n} + \alpha \sum_{j \neq i} \frac{g_{ji}}{d_j^{out}} c_j(t).$$

- Iterate until $|c_i(t+1) - c_i(t)| < \varepsilon$ for all $i \in N$, for some small $\varepsilon > 0$.

This is very similar to using **power iteration** to find the largest eigenvalue/eigenvector of the adjacency matrix.

Summary of Eigenvector Centrality-Type Measures

We have introduced three related (but different!) measures:
(For each measure, take g' or g depending on whether care about in-links or out-links. Here we write them with g', as in PageRank.)

1. **Eigenvector centrality:** $c = \frac{1}{\lambda} g' c$.
2. **Katz-Bonacich centrality:** $c = (I - \alpha g')^{-1} \mathbf{1}$.
3. **PageRank:** $c = \left(I - \alpha g' D^{-1}\right) \mathbf{1}$.

Warning: Jackson Ch. 2 uses slightly different terminology. He discusses two different measures from the same original paper by Katz. The first measure (P^K) is given by $c = g' D^{-1} c$. This is eigenvector centrality but with dividing by out-degree, or equivalently PageRank without the "free centrality"/constant term. The second measure (P^{K2}), as well as Jackson's version of Bonacich centrality, both correspond to what we call Katz-Bonacich centrality, but with the possibility of different initial weights on the nodes.

Summary (cntd.)

Our three measures together with P^K ("Katz's first measure") can be categorized according to whether we divide by out-degree and/or add a constant term ("free centrality"):

	With constant term	Without constant term
Divide by out-degree	$c = \left(I - \alpha g' D^{-1}\right)\mathbf{1}$ PageRank	$c = g' D^{-1} c$ Katz's first measure
No division	$c = \left(I - \alpha g'\right)^{-1}\mathbf{1}$ Katz-Bonacich centrality	$c = \frac{1}{\lambda} g' c$ Eigenvector centrality

Applications of the Measures

PageRank was developed with a particular application in mind: websearch.

Eigenvector centrality and Katz-Bonacich centrality are not as closely associated with one particular application. But they do have important applications. The next two lectures cover two important network models, which are closely related to eigenvector centrality and Katz-Bonacich centrality.

1. **Leontief input-output analysis** (closely related to Katz-Bonacich centrality but with some economics on top, actually came before their papers).
2. **The DeGroot learning model** (closely related to eigenvector centrality but with some dynamics on top).

14.15 / 6.207 Networks

Spring 2022