4. Eigenvector Centrality

Bookmark this page

Exercises due Oct 20, 2021 17:29 IST Completed

Eigenvector Centrality

Start of transcript. Skip to the end.

Prof Uhler: OK so those are the simpler ones, mathematically.

So now let's actually get to this cascading

effects, which is a bit more difficult, involved, in terms

of the mathematics.

So that's this eigenvector centrality.

So let's go through the math here.

0:00 / 0:00 2.0x

Video
Download video file

Transcripts
Download SubRip (.srt) file
Download Text (.txt) file

Eigenvalues and Eigenvectors

For a matrix of size $n \times n$ a value λ is called an **eigenvalue** that corresponds to an **eigenvector** $\mathbf{x}$ if

$$A\mathbf{x} = \lambda\mathbf{x}.$$

Technically, the vector $\mathbf{x}$ in the above is a right eigenvector of A. One can also define a left eigenvector as a row vector $\mathbf{y}^T$ that satisfies $\mathbf{y}^T A = \lambda' \mathbf{y}^T$ for a corresponding eigenvalue λ'. Note that all vectors are column vectors and hence $\mathbf{y}^T$ is a row vector.

Eigenvector Centrality

The **eigenvector centrality** of a node is the weighted importance of the nodes pointing to it (left eigenvector centrality) or the nodes that it points to (right eigenvector centrality). Let us consider, for sake of simplicity, the left eigenvectors in this discussion and the description for the case of right eigenvectors follows in a similar fashion with the interpretation that the importance of a particular node captured using right eigenvectors is an indication of how important the neighbors it points to are.

We define **eigenvector centrality** for a directed graph using the left eigenvector corresponding to the largest eigenvalue of all left eigenvectors. Formally, let $\mathbf{v}^T$ be the left eigenvector corresponding to the largest left eigenvalue $\lambda_{\max}$. Then, the eigenvector centrality of node i is the value at the i^{th} index of $\mathbf{v}$ and is denoted v_i.

The interpretation of eigenvector centrality is that the ranking of a particular node i satisfies

$$\sum_j v_j A_{ji} \quad = \lambda_{\max} v_i,$$

and this implies

$$v_i = \frac{1}{\lambda} \sum v_j A_{ji}.$$

$\lambda_{\max} \quad \overline{j}$

To understand the role of the eigenvector corresponding to the largest eigenvalue in defining centrality of a node based on the centrality of its neighbors, we turn to the Perron-Frobenius theorem. Let $\mathbf{y}^0 = \mathbf{1}$ denote the assginment of same centrality value to all the nodes. Let

$$\left(\mathbf{y}^k\right)^T = \left(\mathbf{y}^0\right)^T A^k$$

denote the updated (left) centrality vector after k iterations of updating the centrality of every node based on the centrality of its neighbors. We can show, under some conditions of the adjacency matrix A, that

$$\text{as } k \to \infty, \quad \left(\mathbf{y}^k\right)^T \to \alpha \lambda_{\max}^k \mathbf{v}^T.$$

The value $\lambda_{\max}$ is the largest eigenvalue of A and $\mathbf{v}^T$ is its corresponding left eigenvector. The constant α depends upon the choice of the initial centrality vector. Perron-Frobenius theorem ensures that the eigenvector $\mathbf{v}^T$, which corresponds to the largest eigenvalue, is a non-negative, non-zero vector. This satisfies a key requirement that our ranking of importance of every node be non-negative and that there is at least one node that has a non-zero importance.

For these exercises we will require that the left eigenvectors be normalized as follows:

$$\sqrt{\sum_i v_i^2} = 1,$$

as not all linear algebra libraries use the same normalization conventions.

Eigenvector Centrality - I

6/6 points (graded)

Consider the adjacency matrix

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

Raw matrix

Python:

```
    [[1,0,0,0],
[1,0,0,0],
[1,0,0,0],
[1,0,0,0]]
```

Mathematica:

```
    {{1, 0, 0, 0},
{1, 0, 0, 0},
{1, 0, 0, 0},
{1, 0, 0, 0}}
```

Hide

1. Assuming that the importance of a node is based on the importance of nodes pointing to it, which node is the most important in the graph? Answer by looking at the adjacency matrix or drawing the graph.

- [x] Node 0

- ○ Node 1
- ○ Node 2
- ○ Node 3

✔

2. Find the left eigenvector centrality of all the nodes using a computational software. For **networkx** in Python, make sure to build a **digraph**; the command to obtain the eigenvector centrality is `networkx.eigenvector_centrality`.

(Please round your answers to the nearest integer for this question).

Node 0: 1 ✔ **Answer:** 1

Node 1: 0 ✔ **Answer:** 0

Node 2: 0 ✔ **Answer:** 0

Node 3: 0 ✔ **Answer:** 0

3. Does the left eigenvector centrality computed using the tool match our intuition?

- ◉ Yes
- ○ No

✔

Solution:

1. **Node 0.** Every node points to Node 0 and there are no other edges in the graph.
2. **1, 0, 0, 0.**

   Python:

   ```
   graph = networkx.from_numpy_matrix(np.array(A), create_using=networkx.DiGraph)
   networkx.eigenvector_centrality(graph)
   ```

   or use `numpy.linalg.eig(np.array(A).T)` to find the left eigenvectors (the right eigenvectors of the transpose matrix).

   Mathematica:

   ```
   Eigensystem[Transpose[A]]
   ```

   Note that Mathematica does not used the same eigenvector normalization as numpy, so make sure to normalize any result you get.
3. **Yes.** The computational answer matches our intuition from the first part.

Submit You have used 1 of 5 attempts

ℹ Answers are displayed within the problem

Eigenvector Centrality - II

5/5 points (graded)

Consider the adjacency matrix

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

Raw matrix

Python:

```
    [[1,1,1,1],
[0,0,0,0],
[0,0,0,0],
[0,0,0,0]]
```

Mathematica:

```
    {{1, 1, 1, 1},
{0, 0, 0, 0},
{0, 0, 0, 0},
{0, 0, 0, 0}}
```

Hide

Assuming that the importance of a node is based on the importance of nodes pointing to it, which node is the most important in the graph? Answer by looking at the adjacency matrix or drawing the graph.

- [x] Node 0
- [x] Node 1
- [x] Node 2
- [x] Node 3

✔

2. Find the left eigenvector centrality of all the nodes using a computational software. For **networkx** in Python, make sure to build a **digraph**; the command to obtain the eigenvector centrality is `networkx.eigenvector_centrality`. Provide an answer accurate to one **significant figure** (graded to 10% tolerance).

Node 0: 0.5 ✔ **Answer:** 0.5

Node 1: 0.5 ✔ **Answer:** 0.5

Node 2: 0.5 ✔ **Answer:** 0.5

Node 3: 0.5 ✔ **Answer:** 0.5

Solution:

1. Node 0 points to itself and to every other node and there are no other edges in the graph. If the importance of node $i \neq 0$ is based on the importance of node 0, then the importance of every node $i \neq 0$ must be the same as the importance of node 0. Therefore, every node in this graph is equally important.

must be the same as the importance of node 0. Therefore, every node in this graph is equally important.

2. **0.5, 0.5, 0.5, 0.5.** The code from the previous question can also be used to answer this question (remember to normalize any eigenvectors).

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Eigenvector Centrality - III

4/4 points (graded)

Consider the adjacency matrix

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

Raw matrix

Python:

```
    [[1,1,1,1],
[1,0,0,0],
[1,0,0,0],
[1,0,0,0]]
```

Mathematica:

```
    {{1, 1, 1, 1},
{1, 0, 0, 0},
{1, 0, 0, 0},
{1, 0, 0, 0}}
```

Hide

Find the left eigenvector centrality of all the nodes using a computational software. Provide an answer accurate to at least three **significant figures** (graded to 1% tolerance).

Node 0: 0.7991714905460789 **Answer:** 0.7991

Node 1: 0.34704703269074766 **Answer:** 0.347

Node 2: 0.34704703269074766 **Answer:** 0.347

Node 3: 0.34704703269074766 **Answer:** 0.347

Solution:

0.799, 0.347, 0.347, 0.347. The code for the previous problem can be used for this question.

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Module 3: Network Analysis:Graph Centrality Measures / 4. Eigenvector Centrality

Add a Post

< All Posts

(STAFF) Left and right eigenvectors

discussion posted 2 months ago by **tfreixanet**

Maybe you can add that the left eigenvectors are the right eigenvectors of A^T. In Python linalg.eig will do the job to compute the centrality.

This post is visible to everyone.

Add a Response

1 response

MarkvdV
2 months ago

brilliant

Add a comment

Showing all responses

Add a response:

Preview

Submit

< Previous Next >

Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap

Connect

Blog
Contact Us
Help Center
Media Kit
Donate