This session will be taught in Hybrid format

When joining on Zoom

- When your microphone or camera are switched on you will be able to be heard in the teaching room and may be seen on the screen in the room.
- If the screen is recorded, you may be captured on the recording if you choose to contribute and have your camera switched on.
- Chat messages to everyone and privately to the host may be visible on screen in the room.

When joining in the Teaching room

- Academic teaching staff should let you know when Zoom is connected and you may see the Zoom meeting on the screen.
- Once connected to Zoom, microphones may capture sound from the whole room. Your voice may be shared to Zoom when speaking at a normal talking volume.
- Once connected to Zoom, cameras will capture video images from the room. These may be shared to Zoom. The camera capture will be a wide shot of the whole room or block of seating. It will not capture close-up images of individual students.
- If the session is recorded, video from the room will be captured on the recording. Your voice will be captured on the recording if you choose to contribute.
- If you connect to the Zoom meeting from within the room (for example to use the chat), please keep your audio on mute or use headphones to avoid feedback issues with the room audio.

The recordings may be stored in the cloud ad any personal information within the recording will be processed in accordance to the General Data Protection Regulations 2018.
A substantial contribution is considered to be anything more than merely answering questions or participating in a group discussion. Where you make a substantial contribution to the delivery of the recorded events, a signed consent form will be obtained prior to the recording being made available for viewing. The Consent Form will address your personal informationand any copyright or other intellectual property in the recording.

(Complex) Networks Analysis

Centrality Analysis (Part II)

Felipe Orihuela-Espina

November 12, 2024

Table of Contents

Section 1

Centrality analysis (Part II): Global measures

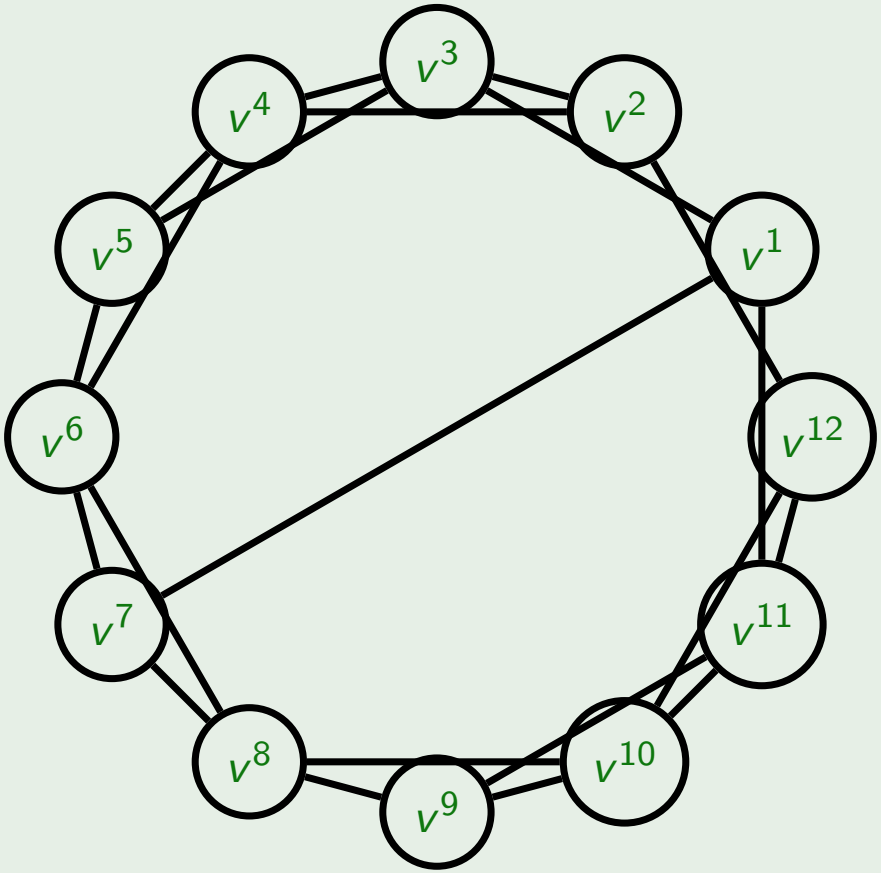
Subsection 1

Closeness centrality

Closeness centrality

Example

Who's the real influencer?



Closeness centrality

Sometimes a node's importance is not about the amount of connection (degree centrality) or the flow it supports (betweeness centrality), but about how close it is to all other nodes.
Although physical location of nodes is irrelevant, however this is not the same as the node's *logical* location.

- A node is as close to another as the shortest path between them
- A node is more central the shorter the average distance to all other nodes. This is referred to as closeness centrality.

In contrast to degree centrality and betweeness centrality which reflects *local* importance, Closeness centrality captures *global* importance.

Closeness centrality

Closeness centrality was defined as early as 1950 [1] as the reciprocal of the farness:

$$C^C(v^i) = \frac{1}{\sum_{v^j \in V} d(v^i, v^j)}$$

This is unnormalised by the order (number of nodes) of the network. Hence, more often, you will see its normalised version:

$$C^C(v^i) = \frac{|V| - 1}{\sum_{v^j \in V} d(v^i, v^j)}$$

Closeness centrality

Remark

Thus far, you have only learn how to calculate distances in unweighted undirected graphs by counting hops. In weighted graphs or directed graphs, the weight and direction are critical to establish the distance. We shall learn about how to calculate distances in weighted graphs later in the course e.g. using Dijkstra's algorithm.

From the point of view closeness centrality however, it is irrelevant how you did calculate the distance but the value of the distance itself. Hence, the definition given before is still valid.

Closeness centrality

Remark

If the graph is not connected, some distances would either be not defined. Hence, a number of corrections have been suggested for this degeneration. For instance, Beauchamp's correction changes not defined distances as infinite, and then simply takes the convention that $1/\infty = 0$. We shall learn about connectivity later in the course.

Again, from the point of view closeness centrality however, it is irrelevant how you did calculate the distance but the value of the distance itself. Hence, the definition given before is still valid.

Closeness centrality

We won't get into details but it can be shown that in many types of networks, closeness and degree are often related.

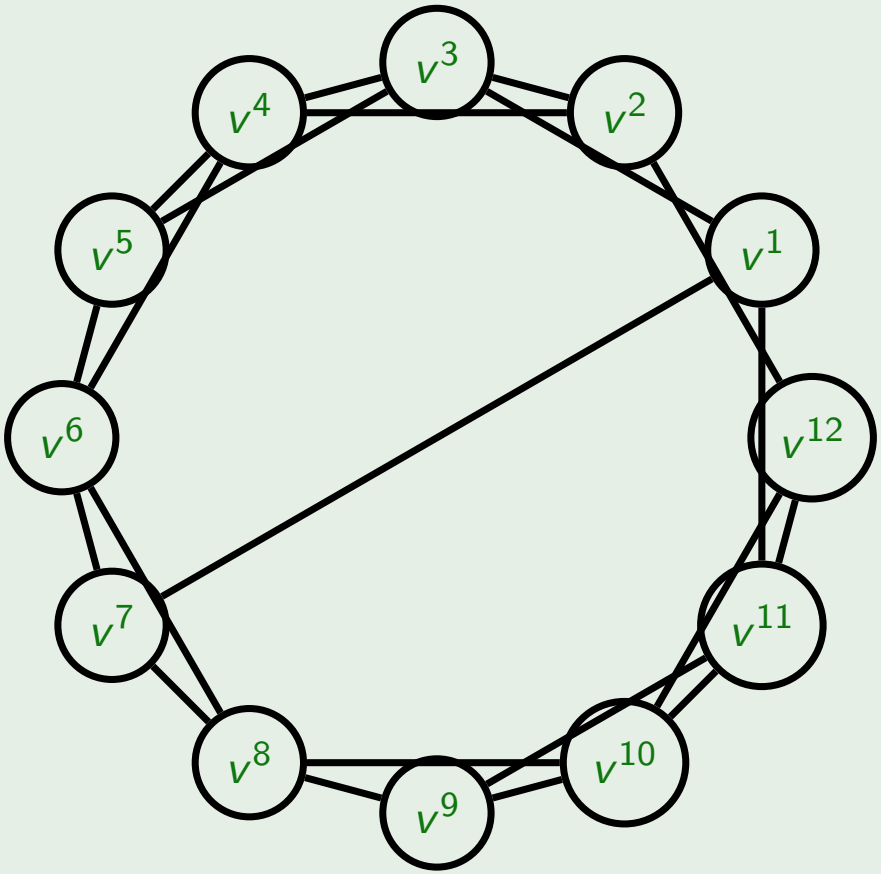
- The more connections a node has, the most likely its average distance is smaller.
- Moreover, in certain cases, the average shortest path length is the average of the inverse closeness.

Centrality analysis

Going back to the motivating scenario;

Example

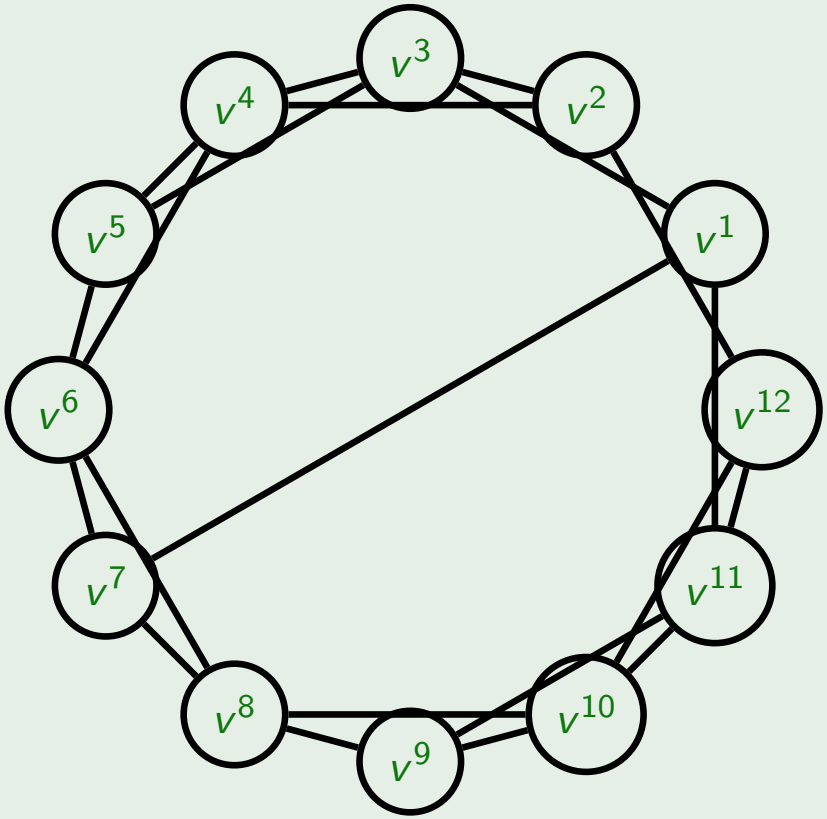
Who's the real influencer?



Closeness centrality

Example

Let's start by calculating all pairwise distances (symmetric for an unweighted undirected graph)



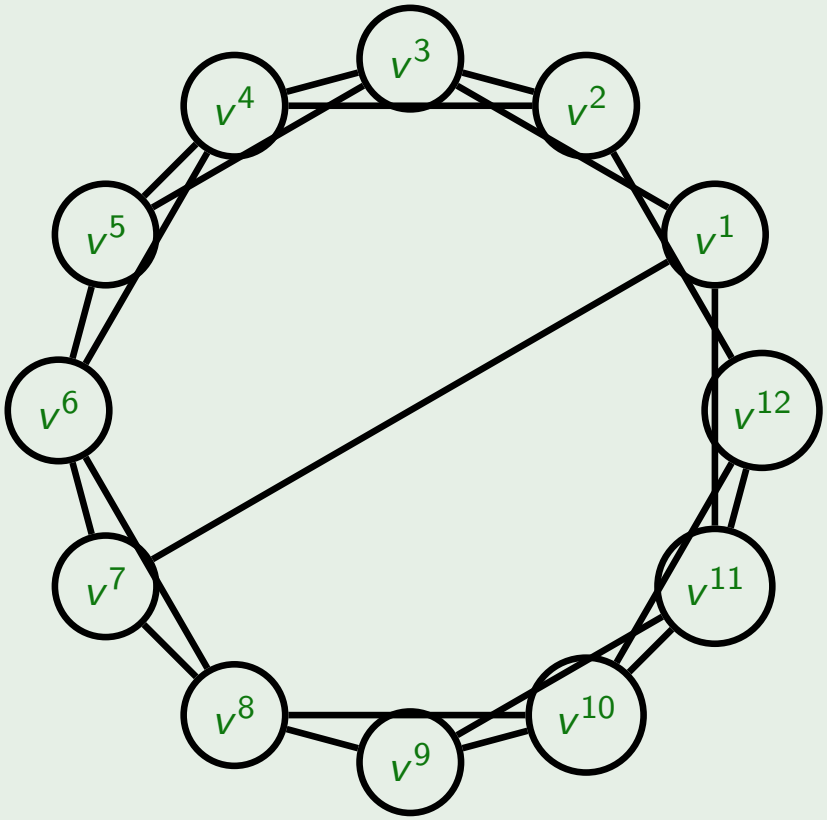
$dist =$

	v^1	v^2	v^3	v^4	v^5	v^6	v^7	v^8	v^9	v^{10}	v^{11}	v^{12}
v^1	0	2	1	2	2	2	1	2	2	2	1	2
v^2	2	0	1	1	2	2	3	3	3	2	2	1
v^3	1	1	0	1	1	2	2	3	3	3	2	2
v^4	2	1	1	0	1	1	3	2	3	3	3	2
v^5	2	2	1	1	0	1	2	2	3	3	3	3
v^6	2	2	2	1	1	0	1	1	2	2	3	3
v^7	1	3	2	3	2	1	0	1	2	2	2	3
v^8	2	3	3	2	2	1	1	0	1	1	2	2
v^9	2	3	3	3	3	2	2	1	0	1	1	2
v^{10}	2	2	3	3	3	2	2	1	1	0	1	1
v^{11}	1	2	2	3	3	3	2	2	1	1	0	1
v^{12}	2	1	2	2	3	3	3	2	2	1	1	0

Closeness centrality

Example

Now we can calculate (using either rows or columns) the total sum of distances from each to node to all other nodes.

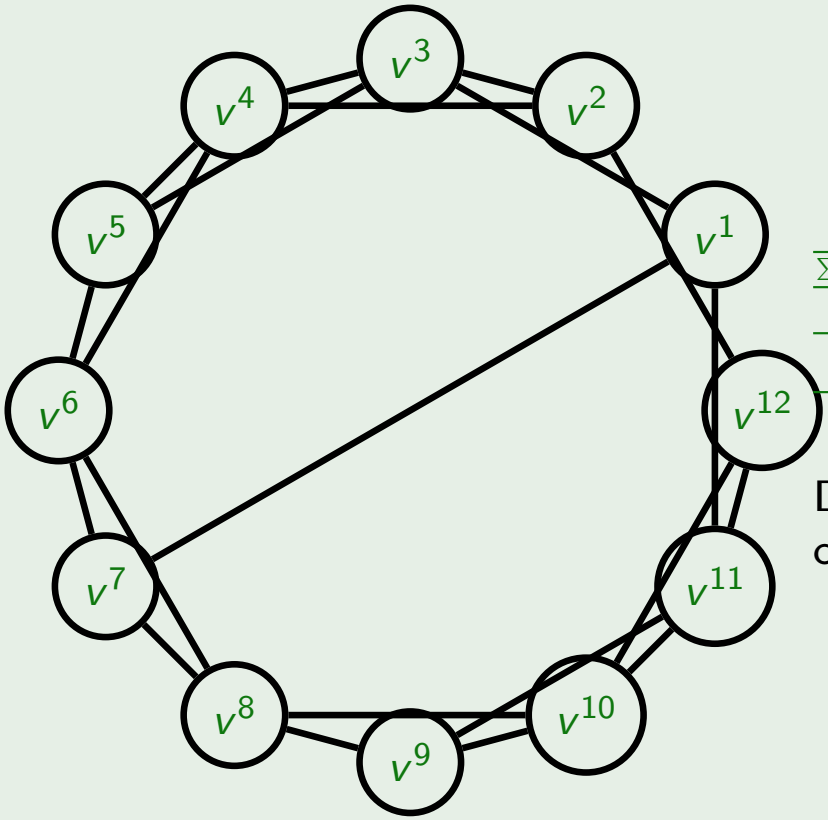


	v^1	v^2	v^3	v^4	v^5	v^6	v^7	v^8	v^9	v^{10}	v^{11}	v^{12}	$\sum_{v^j} d(v^i, v^j)$
v^1	0	2	1	2	2	2	1	2	2	2	1	2	19
v^2	2	0	1	1	2	2	3	3	3	2	2	1	22
v^3	1	1	0	1	1	2	2	3	3	3	2	2	21
v^4	2	1	1	0	1	1	3	2	3	3	3	2	22
v^5	2	2	1	1	0	1	2	2	3	3	3	3	23
v^6	2	2	2	1	1	0	1	1	2	2	3	3	20
v^7	1	3	2	3	2	1	0	1	2	2	2	3	20
v^8	2	3	3	2	2	1	1	0	1	1	2	2	20
v^9	2	3	3	3	3	2	2	1	0	1	1	2	23
v^{10}	2	2	3	3	3	2	2	1	1	0	1	1	21
v^{11}	1	2	2	3	3	3	2	2	1	1	0	1	21
v^{12}	2	1	2	2	3	3	3	2	2	1	1	0	22
$\sum_{v^j} d(v^i, v^j)$	19	22	21	22	23	20	20	20	23	21	21	22	

Closeness centrality

Example

Finally, we simply calculate the closeness centrality by calculating inverse. We can also normalize.



	v^1	v^2	v^3	v^4	v^5	v^6	v^7	v^8	v^9	v^{10}	v^{11}	v^{12}
$\sum_{v^j} d(v^i, v^j)$	19	22	21	22	23	20	20	20	23	21	21	22
$\frac{1}{\sum_{v^j} d(v^i, v^j)}$	1/19	1/22	1/21	1/22	1/23	1/20	1/20	1/20	1/23	1/21	1/21	1/22
$\frac{\lvert V \rvert - 1}{\sum_{v^j} d(v^i, v^j)}$	11/19	11/22	11/21	11/22	11/23	11/20	11/20	11/20	11/23	11/21	11/21	11/22

Despite having a smaller degree, than most of the other nodes, v^1 has the highest closeness centrality.

Subsection 2

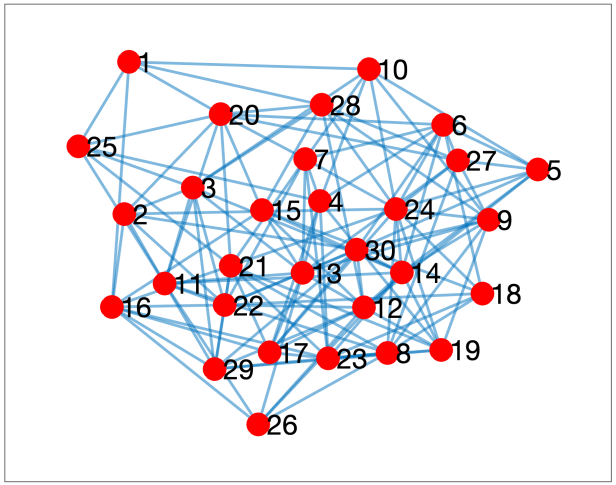
Eigencentrality

Eigencentrality

Example

What about now?

Say that you want to inject a virus to spread over the network. Which computer will you attack? and more importantly, why?



Eigencentrality

Yet another measure of node importance is Sometimes a node's importance is not about the amount of connection (degree centrality) or the flow it supports (betweeness centrality), but about how easy you can reach all other nodes. This is referred to as **Eigencentrality**.

- In a social analogy, you are as important as your friends are. Hence, it is not about how much information you handle (betweeness centrality), how many people you know (degree centrality) or how close you are on average to other locations (closeness centrality) but rather how well connected you are to the rest of the network. You may not have many connections but they are the *right* connections (the people you know matters). You may not pass much information but your information can reach all corners of the network fastest than anyone else.

In contrast to degree centrality and betweeness centrality which reflects *local* importance, eigencentrality captures *global* importance.

Eigencentrality

You should have learn the egiendecomposition while you were learning PCA in the first part of the module, but nevertheless here is a brief reminder.

Reminder: Eigendecomposition

Let be a vector space with vectors $\mathbf{x}$. A square matrix A represents a transformation in the vector space such that, when scaled by A, vector $\mathbf{x}$ is transformed into vector $\hat{\mathbf{x}}$.

$$A\mathbf{x} = \hat{\mathbf{x}}$$

The new vector $\hat{\mathbf{x}}$ may (or may not) have:

- a different *direction* from the original $\mathbf{x}$ i.e. it can be rotated,
- a different *length* (norm) from the original $\mathbf{x}$ i.e. it can be scaled,

Eigencentrality

Reminder: Eigendecomposition

For instance; Let

$$A\mathbf{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Then;

$$\hat{\mathbf{x}} = A\mathbf{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \end{bmatrix}$$

Eigencentrality

Reminder: Eigendecomposition

However, regardless of A some non-zero vectors will *only* be scaled but not rotated, e.g.[a] ;

$$A\mathbf{x} = \begin{bmatrix} 4 & 3 \\ 2 & -1 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Then;

$$\hat{\mathbf{x}} = A\mathbf{x} = \begin{bmatrix} 4 & 3 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 15 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} 5 = 5\mathbf{x}$$

We call these vectors the eigenvectors, and the companion scaling factor $\lambda_{\mathbf{x}} = 5$ their eigenvalue.

[a] The rest of the example is modified from https://online.stat.psu.edu/statprogram/reviews/matrix-algebra/eigendecomposition.

Eigencentrality

Reminder: Eigendecomposition

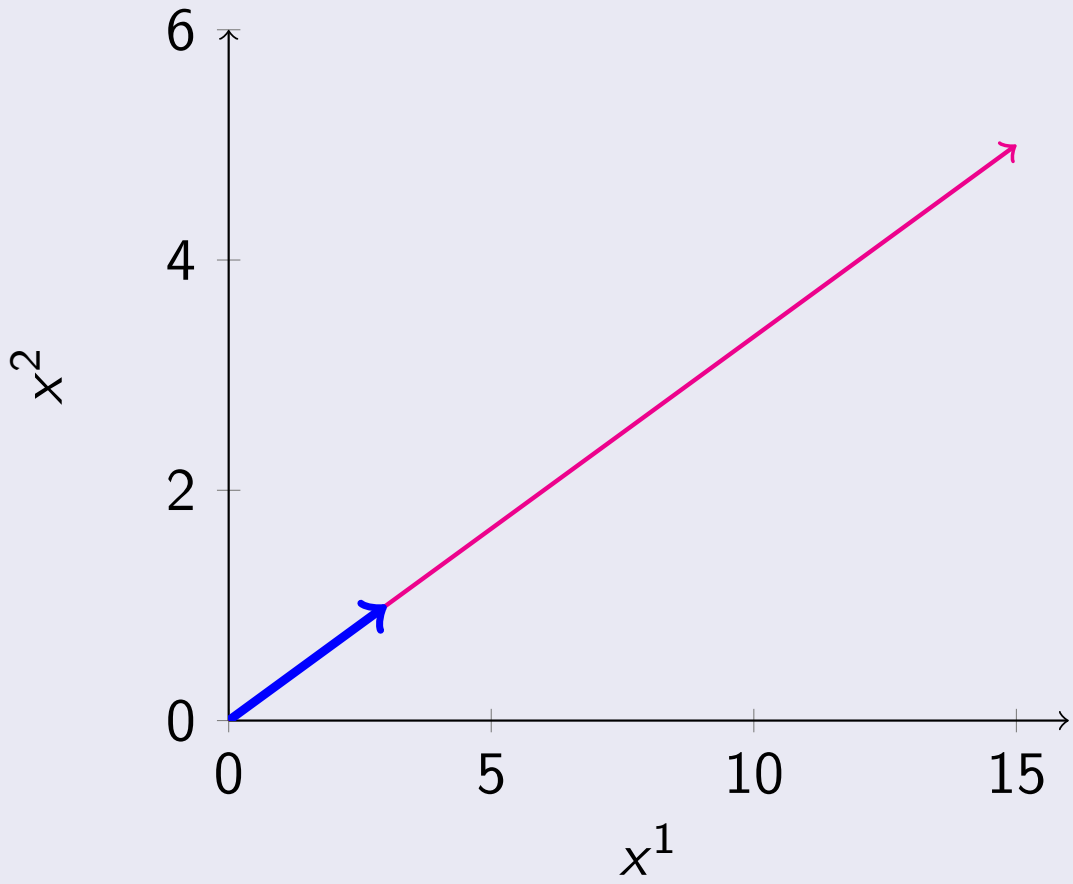


Figure: Example of eigenvector $\mathbf{p^i} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ and its image $\hat{\mathbf{p}}^{\mathbf{i}} = \begin{bmatrix} 15 \\ 5 \end{bmatrix}$. The eigenvector image is scaled but not rotated.

Eigencentrality

Reminder: Eigendecomposition

We can see, that for a given eigenvector p^i, it holds that:

$$A\mathbf{p^i} = \mathbf{p^i}\lambda_{p^i}$$

. . . or taking all eigenvectors at once;

$$AP = P\Lambda \qquad \Rightarrow \qquad A = P\Lambda P^{-1}$$

which is the eigendecomposition.

Eigencentrality

Reminder: Eigendecomposition

The next question is, how do we solve the eigendecomposition?

From the first part of the module, we already know a way to address this. A very simple algorithm to calculate the eigendecomposition relies on the so called characteristic equation of a matrix for the eigenvalues;

$$AP = P\Lambda \quad \Rightarrow \quad AP - AP = P\Lambda - AP \quad \Rightarrow \quad (\Lambda I - A)P = 0 \quad \Rightarrow \quad det(\Lambda I - A) = 0$$

Solving the characteristic equation $det(\lambda I - A) = 0$ is equivalent to finding the *roots* of the characteristic polynomial;

$$\begin{aligned} det(\lambda I - A) =& (-1)^n(\lambda_1 - \lambda)(\lambda_2 - \lambda)\dots(\lambda_n - \lambda) \\ =& (-1)(\lambda - \lambda_1)(-1)(\lambda - \lambda_2)\dots(-1)(\lambda - \lambda_n) \\ =& (\lambda - \lambda_1)(\lambda - \lambda_2)\dots(\lambda - \lambda_n) \end{aligned}$$

Eigencentrality

Reminder: Eigendecomposition

For instance; let:

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

Then;

$$det(\lambda I - A) = det\left(\lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 4 & 3 \\ 2 & -1 \end{bmatrix}\right) = det\left(\begin{bmatrix} \lambda - 4 & -3 \\ -2 & \lambda - (-1) \end{bmatrix}\right) = 0$$

Then;

$$\begin{aligned} \Rightarrow & det(\lambda I - A) = (\lambda - 4)(\lambda + 1) - (-3) \cdot (-2) = \lambda^2 - 3\lambda - 10 \\ & = (\lambda - \lambda_1)(\lambda - \lambda_2) = 0 \\ \Rightarrow & \lambda_1 = 5, \lambda_2 = -2 \quad \Rightarrow \quad (\lambda - 5)(\lambda + 2) = 0 \end{aligned}$$

That is, the eigenvalues of A are -2 and 5.

Eigencentrality

Reminder: Eigendecomposition

Once the eigenvalues are known, one can find the eigenvectors using the eigenvector equations.

$$A\mathbf{p^i} = \mathbf{p^i}\lambda_{p^i}$$

For $\lambda_1 = 5$

$$\begin{bmatrix} 4 & 3 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{p^1}_1 \\ \mathbf{p^1}_2 \end{bmatrix} = 5 \begin{bmatrix} \mathbf{p^1}_1 \\ \mathbf{p^1}_2 \end{bmatrix}$$

which is a simple system of equations:

$$\begin{bmatrix} 4\mathbf{p^1}_1 & 3\mathbf{p^1}_2 \\ 2\mathbf{p^1}_1 & -1\mathbf{p^1}_2 \end{bmatrix} = \begin{bmatrix} 5\mathbf{p^1}_1 \\ 5\mathbf{p^1}_2 \end{bmatrix} \quad \Rightarrow \quad \mathbf{p^1} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Reminder: Eigendecomposition

And analogously For $\lambda_2 = -2$

$$\begin{bmatrix} 4 & 3 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{p^2}_1 \\ \mathbf{p^2}_2 \end{bmatrix} = -2 \begin{bmatrix} \mathbf{p^2}_1 \\ \mathbf{p^2}_2 \end{bmatrix}$$

which is a simple system of equations:

$$\begin{bmatrix} 4\mathbf{p^2}_1 & 3\mathbf{p^2}_2 \\ 2\mathbf{p^2}_1 & -1\mathbf{p^2}_2 \end{bmatrix} = \begin{bmatrix} -2\mathbf{p^2}_1 \\ -2\mathbf{p^2}_2 \end{bmatrix} \Rightarrow \mathbf{p^2} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

Eigencentrality

Let $A = P\Lambda P^{-1}$ be the eigendecomposition of the adjacency matrix, leading to the different eigenvectors equations $A\mathbf{p^k} = \lambda^k \mathbf{p^k}$ with $\mathbf{p^k}$ the different eigenvectors, and λ^k the k-th eigenvalues.

Let's take the eigenvector $\mathbf{p}$ with the largest associated λ.

The eigencentrality score, $\mathbf{p}(v^i)$ of vertex v^i can be defined as:

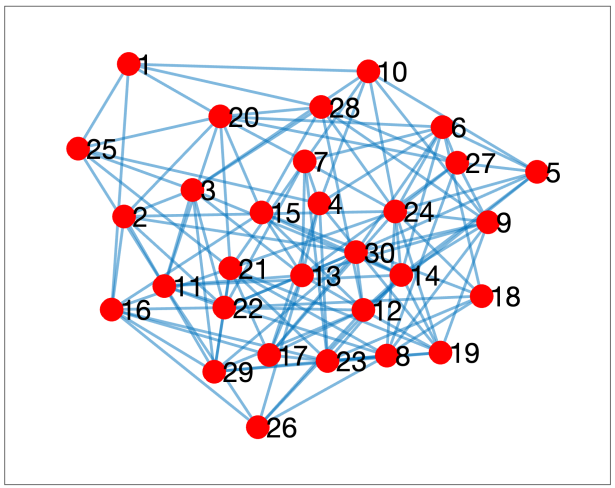
$$x(v^i) = \frac{1}{\lambda} \sum_{j \in M(v^i)} \mathbf{p}(v^j) = \frac{1}{\lambda} \sum_{j \in V} a_{i,j} \mathbf{p}(v^j)$$

with $M(v^i)$ the set of neighbors of v^i.
That is, the i-th component of the related eigenvector gives the relative centrality score of the vertex v^i in the network.

Eigencentrality

Example

We start by extracting the adjacency matrix of the graph



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

Eigencentrality

Example

. . . and then simply, calculate the eigendecomposition (you already know how to do this from the first part of the module) and apply the formula [a]. The eigenvector centrality with largest centrality is node 24

$$c = \begin{pmatrix} 0.0148 \\ 0.0289 \\ 0.0335 \\ 0.0282 \\ 0.0290 \\ 0.0343 \\ 0.0359 \\ 0.0341 \\ 0.0361 \\ 0.0246 \\ 0.0372 \\ 0.0419 \\ 0.0447 \\ 0.0456 \\ 0.0269 \\ 0.0307 \\ 0.0336 \\ 0.0239 \\ 0.0338 \\ 0.0325 \\ 0.0415 \\ 0.0427 \\ 0.0333 \\ 0.0499 \\ 0.0203 \\ 0.0253 \\ 0.0313 \\ 0.0289 \\ 0.0363 \\ 0.0404 \end{pmatrix}$$

[a] Calculating this by hand for a network of 30 nodes like this one takes a bit of time, hence I have used matlab here, but you can check, that for the above adjacency matrix the results are the ones shown

Eigencentrality

In Matlab, the above example is only a few lines of code:

```
%Create a simple (more or less) random network.
nNodes=30;
epsDensity=0.3;
tmp = ((nNodes)*(nNodes-1)/2); %Number of elements in the matrix
    lower half (exc. main diagonal).
A = squareform(rand(tmp,1))<epsDensity; %Adjacency matrix.
G = graph(A,'omitselfloops');
% Centrality analysis
C = centrality(G,'eigenvector')
```

Eigencentrality

Remark

Although I could have exemplified eigencentrality with a much small network so that calculations can still be done by hand, but I wanted to show how even for a relatively small network;

- Visual inspection becomes remarkably difficult
- Hand calculations become either incredible time consuming (or even unfeasible)
- Concepts that you already know (like the eigendecomposition) that you have learn whilst learning PCA, can be reused *intelligently* to retrieve knowledge from medium and large network datasets.

Eigencentrality

Exercise

What does it mean if a node has high betweenness centrality and low eigencentrality?

Eigencentrality

Exercise

What does it mean if a node has high betweenness centrality and low eigencentrality?

Answer:
High betweenness indicates that the node connects disparate parts of the network whereas the low eigencentrality indicates that the node is distant from the centers of power in the network.

Think of the Panama's channel; it supports heavy traffic but most of that traffic connects routes in Asia, North America or Europe where the big economic hubs are. Almost none of that economy happens in Central America.

Subsection 3

Page Rank

Page Rank

Example

How does Google rank pages after your search?

- PageRank centrality was developed by Larry Page and Sergei Brin while doing their PhD at Stanford University under supervisors, Terry Winograd (for Page) and Héctor García-Molina (for Brin).
- PageRank will later became the algorithm for ranking web pages in Google.
- Google itself uses several factors to rank the search results; inc. proximity, anchor text (text of links), and of course PageRank.

Remark

In Feb, 2020, John Mueller (Google Search Relations) declared; "Yes, we do use PageRank internally, among many, many other signals. It's not quite the same as the original paper, there are lots of quirks (eg, disavowed links, ignored links, etc.), and, again, we use a lot of other signals that can be much stronger."

Page Rank

Remark

Google search engine was introduced in 1998 in [2]. This paper is about the search engine itself, but it mentions the underpinning PageRank algorithm and further mentions that a paper in progress regarding this algorithm was in preparation. The draft is referenced in [2] as:

[Page 98] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web
http://google.stanford.edu/~backrub/pageranksub.ps

The indicated URL is no longer available and I have not been able to find such draft. However, a presentation from Page at Stanford's servers is still available:
http://infolab.stanford.edu/~page/papers/pagerank/ppframe.htm
Google scholar suggest a lot of links and citations to the original draft, but all that I've tried to followed are now broken links. I've been able to trace this one tough:
https://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/pagerank.pdf
. . . but note how it has no authors and it is not in Stanford's servers but in UPenn so I can't tell how close or far this draft is from the original.

Page Rank

The motivation:

- When someone makes a search in a search engine, searches are heavily under-determined;
  - There may be millions of pages matching a search.

Example

If you search "PageRank algorithm" in Google, one gets 2,610,000 results (search executed 12-Nov-2024 from UK).

- Hence, the relevance of a web page is not so much about its content (as there are millions matches) but about how many other web pages are pointing to it as well as the relevance of those pages themselves.
  - The idea of your importance being based on your contacts is, in principle, the concept of eigencentrality.

Page Rank

Classical eigencentrality expresses centrality in terms of your adjacent connections, but . . .

- It does only take into account the immediate adjacent connections; i.e. it does not propagate.
  - Perhaps you aren't that well connected on your own, but you are the son of a tycoon
  - You are only a research assistant, but you are in the lab of an important professor
- It does not take into account the direction of the connections.
  - In the (naive) influencer example above, it does not matter whether you follow or you are being followed.

Page Rank

PageRank centrality is a variant of eigencentrality whereby a node is deemed to be central if other important nodes link to it [2].

In contrast to basic eigencentrality, PageRank:

- Takes into account the direction of the connections; in-degree vs out-degree, and
  - A node gains centrality by being linked to from other nodes with higher PageRank score (page authority).
- Consider connections beyond the immediate adjacency; normalizing eigenvector centrality combined with a random jump assumption
  - The downward influence spread is divided equally among all outbound links.
  - PageRank works on a logarithmic scale; selective citations from authoritative pages are more "valuable" than uniform citations of a large number of pages.

Page Rank

The rationale (Cont.):

PageRank imitates a random surfer

- Given some random URL, a random person navigating the web, will click randomly on some of the links.
- Eventually, the random surfer will become bored and will move to a new random URL.
- PageRank is proportional to the number of visits that each page will receive given this random surfer.

http://infolab.stanford.edu/~page/papers/pagerank/

Page Rank

The rationale (Cont.):

Because the chances on landing on another page depends on how many outward link the current page has, the propagation of the PageRank score is divided proportionally.

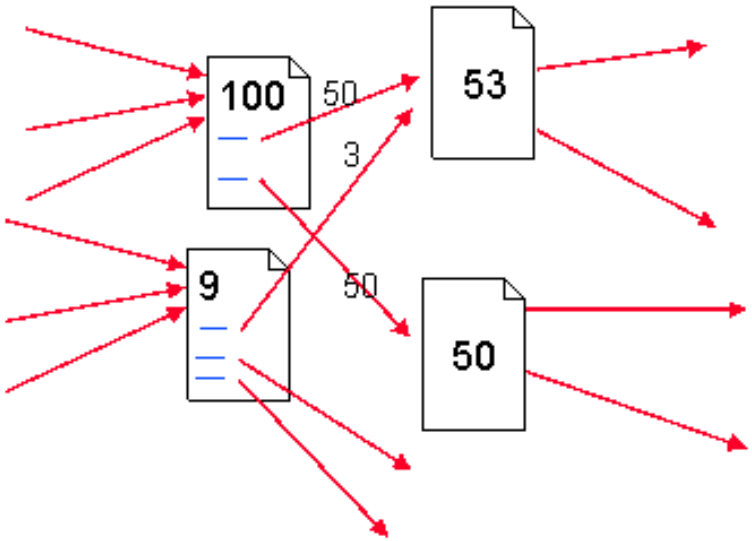


Figure: Idealized PageRank calculation. Figure from: [http://infolab.stanford.edu/~page/papers/pagerank/].

Page Rank

The rationale (Cont.):

Let N be the number of indexed pages.

The number of outgoing links from a page i to another page j (including potential multiple links the the same page) is $l_{i,j}$.

Example

Let be a web with only 2 pages indexed;

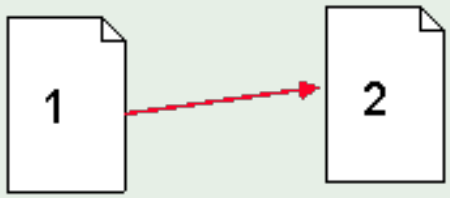


Figure: Calculation of $l_{i,j}$. Figure from: [http://infolab.stanford.edu/~page/papers/pagerank/].

Then:

$$l_{1,2} = 1$$

$$l_{2,1} = 0$$

Page Rank

The rationale (Cont.):

The total number of outgoing links from a page i to any *other* page (other than itself) is simply the sum of $l_{i,j}$ across all pages j;

$$n_i = \sum_{j=1, i \neq j}^{N} l_{i,j}$$

Ideally, in PageRank, the score of a page j *received* from source pages i is the summation of the number incoming links weighted by the proportion of links in each source;

$$W_j = \sum_{i=1, i \neq j}^{N} \left(l_{i,j} \frac{W_i}{n_i} \right)$$

Page Rank

The rationale (Cont.):

In PageRank, the scores W are sometimes referred to as **page authority**, **prestige** or simply **PageRank**.

Since the score of a page depends in turn on the score of other pages, this suggests that to compute W we need some either recursive or iterative algorithm.

$$W_j(t) = \sum_{i=1, i \neq j}^{N} \left(l_{i,j} \frac{W_i(t-1)}{n_i} \right)$$

with t the iteration.

Computing PageRank

Let's start by defining a normalized adjacency matrix $A^{Norm} = \{a_{ij}^{norm}\}$ such that:

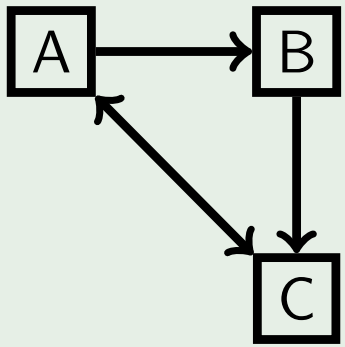
$$a_{ij}^{norm} = \begin{cases} \frac{1}{out-degree(v^i)|_{vj}} & \text{if } (v^i, v^j) \in E \\ 0 & \text{if } (v^i, v^j) \notin E \end{cases}$$

In the context of PageRank, this adjacency matrix is sometimes called the *transition matrix*.

Computing PageRank

Example

Normalized adjacency calculation for PageRank



$$A^{Norm} = \begin{pmatrix} 0 & 0 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \end{pmatrix}$$

Example from: http://infolab.stanford.edu/~page/papers/pagerank/ppframe.htm.

Page Rank

Computing PageRank

The normalized eigenvector prestige score [3] is defined as:

$$p(v^i) = \sum_{v^j} (A^{Norm})^T \cdot p(v^j)$$

with p the dominant eigenvector of A^{Norm}[1].

Remark

Originally developed by Seeley in 1949 [3], the prestige score precedes PageRank by almost 50 years! This is however, the seed for the final PageRank score developed by Brin and Page. Also, the original version was calculated over the un-normalized version of the adjacency matrix.

[1] Remember! The eigenvector equations are of the form $\lambda \mathbf{p} = \sum a_{ij} \mathbf{p}^j$ [4, pg. 306]which is Eq. scaled by λ as we only pick the highest eigenvector and non scaling will progressively shrink the scores.

Page Rank

Computing PageRank

A slightly more elegant vector form;

$$A^{Norm} = diag(A\mathbf{e})^{-1}A$$
$$\mathbf{p} = (A^{Norm})^T\mathbf{p}$$

with $\mathbf{e}$ a vector of rank sources (e.g. initially uniform vector of 1s). This vector is sometimes referred to as *personalization vector*.

This can be implemented recursively or iteratively[2].

$$\mathbf{p}(t) = (A^{Norm})^T\mathbf{p}(t-1)$$

[2]The equation is recursive, but it can be computer iteratively by starting with any set of ranks and iterating until convergence.

Page Rank

Computing PageRank

Unfortunately, things are not that straight forward; Imagine a collection of pages that link to themselves in a loop. In the context of PageRank, this is referred to as the Cyclotron.

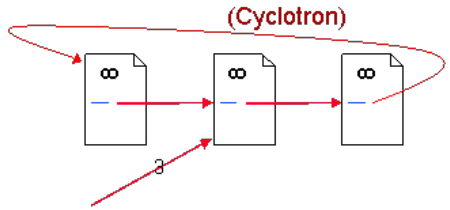


Figure: Example of Cyclotron. Figure from: [http://infolab.stanford.edu/~page/papers/pagerank/].

The cyclotron will artificially boost the PageRank of the pages in the loop.

Computing PageRank

In order to counteract the cyclotron, computation of the PageRank includes a *dumping factor* d $(0 < d < 1)$

$$\mathbf{p}(t) = d(A^{Norm})^T \mathbf{p}(t-1) + (1-d)\mathbf{e}$$

where $\mathbf{e}$ is the vector of rank sources (a.k.a. personalization vector).

Watch out!

Different authors may formalize the dumping factor differently. For instance; I've seen $\mathbf{p}(t) = (1-d)(A^{Norm})^T \mathbf{p}(t-1) + d\mathbf{e}$. Regardless, the idea, is simply that you "discount" the contribution over successive iterations thus simulating the random surfer behaviour.

Computing PageRank

Cyclotrons are not the only problem. What if you have a dangling page?

A dangling page is a page without hyperlinks. If v^i is a dangling page, then i-the column of transition matrix A^{norm} is null.

Computing PageRank

To circumvent the problem of dangling pages, several solutions have been proposed (see [Ng, 2001]).
Opt 1) Perhaps the simplest solution is to simply divide the rank of the dangling pages across all other pages uniformly.

- Indeed, when $\mathbf{e}$ is the uniform vector, this is the one implemented above; $A^{Norm} = diag(A\mathbf{e})^{-1}A$
- In practice, this is a simple substitution of rows in A^{Norm}

Opt 2) Another common one, introduces a dummy page v^r such that:

- Page v^r has a link to itself, and
- Every dangling page is made to point to v^r

The result is an *extended transition matrix* $\hat{A}^{norm}$.

References I


Alex Bavelas.
Communication patterns in task-oriented groups.
The journal of the acoustical society of America, 22(6):725–730, 1950.


Sergey Brin and Lawrence Page.
The anatomy of a large-scale hypertextual web search engine.
Computer networks and ISDN systems, 30(1-7):107–117, 1998.


Soumen Chakrabarti.
Mining the Web: Discovering knowledge from hypertext data.
Morgan Kaufmann, 2002.


Shuguang Cui, Alfred O Hero III, Zhi-Quan Luo, and José MF Moura.
Big data over networks.
Cambridge University Press, 2016.

Questions

Thank you! Questions?