Social Network Analysis: Ideas and Algorithms for Web Search

S. Durga Bhavani
School of Computer and Information Sciences,
University of Hyderabad, Hyderabad.

December 23, 2025

Networks Everywhere!

Friends network is modeled as a graph!

Networks Everywhere!

Friends network is modeled as a graph!

Figure: Friendship network: nodes: people and edges: friendship

Networks Everywhere!

Protein-Protein interaction network!

Networks Everywhere!

Protein-Protein interaction network!

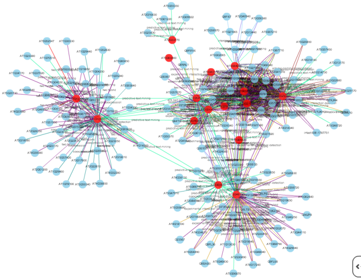


Figure: nodes: proteins edges: interactions

Networks Everywhere!

Co-authorship network!

Networks Everywhere!

Co-authorship network!

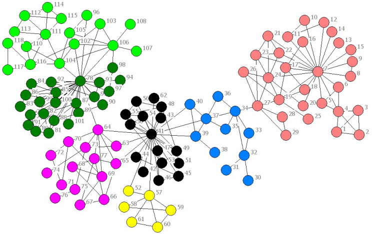


Figure: nodes: authors, edges: joint publication

Network Science

Networks are modeled as graphs $G = (V, E)$ where nodes in the network represent genes/ proteins/ authors and edges denote interactions between pairs of nodes.

- Networks are typically large directed/undirected weighted graphs
- Traditional graph theoretical analysis may not help since these are dynamic graphs and many a time probabilistic graphs.
- Combination of ideas from Physics for growth models; dynamical systems, probabilistic distributions and statistical analysis are applied.
- Hence Network Science is a highly interdisciplinary science.

Contents Outline

- Real-world Networks
- Preferential attachment
- Geometric meaning of matrices
- Importance of eigen vectors
- Webgraphs
- Page rank algorithm
- HITS algorithm

Real-world Networks

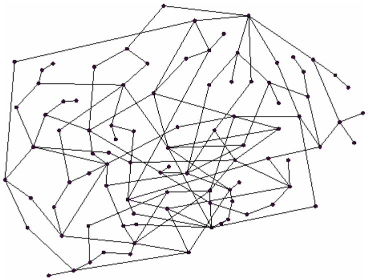
The networks that occur in nature/real world are NOT Random networks.

- Erdos-Renyi(ER) (Random Network)
- Barabasi-Albert Model (Real-world Network)

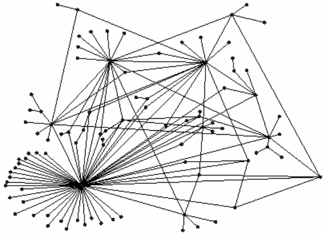
Real-world Networks

The networks that occur in nature/real world are NOT Random networks.

- Erdos-Renyi(ER) (Random Network)



- Barabasi-Albert Model (Real-world Network)



Pioneers of Network Models

Figure: Random Networks: Paul Erdos and Alfred Renyi

Pioneers of Network Models

Figure: Random Networks: Paul Erdos and Alfred Renyi

Figure: Preferential Attachment: A-L.Barabasi and Reka Albert

Idea: Preferential Attachment

Preferential attachment: A process in which a new node entering the graph does NOT connect randomly but shows a preference to attach to a node of a certain type (Eg. higher degree).

Idea: Preferential Attachment

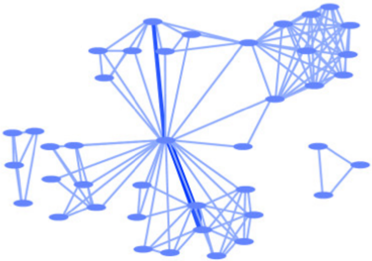
Preferential attachment: A process in which a new node entering the graph does NOT connect randomly but shows a preference to attach to a node of a certain type (Eg. higher degree).

- These kind of connections lead to the phenomenon 'Rich-gets-Richer'
- They show 'preferential attachment' creating 'hub' nodes and forming communities.

Idea: Preferential Attachment

Preferential attachment: A process in which a new node entering the graph does NOT connect randomly but shows a preference to attach to a node of a certain type (Eg. higher degree).

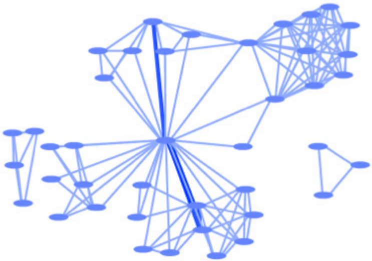
- These kind of connections lead to the phenomenon 'Rich-gets-Richer'
- They show 'preferential attachment' creating 'hub' nodes and forming communities.

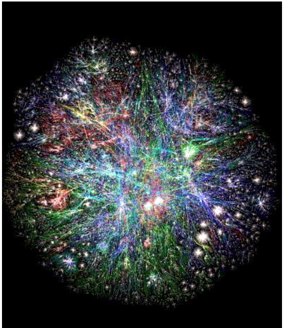


Idea: Preferential Attachment

Preferential attachment: A process in which a new node entering the graph does NOT connect randomly but shows a preference to attach to a node of a certain type (Eg. higher degree).

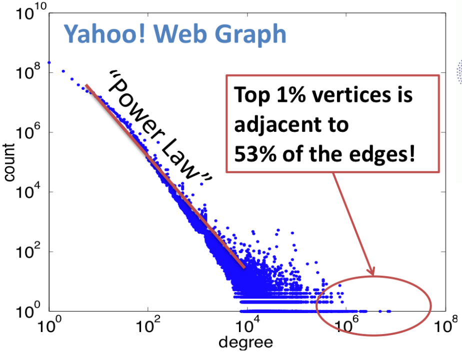
- These kind of connections lead to the phenomenon 'Rich-gets-Richer'
- They show ‘preferential attachment’ creating ‘hub’ nodes and forming communities.





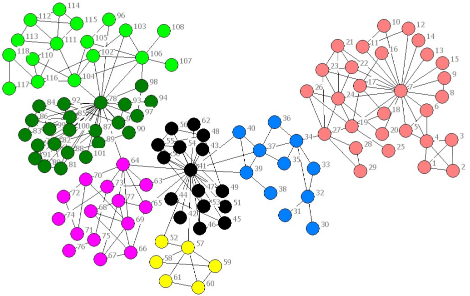
The Power-law

- Natural Graphs:



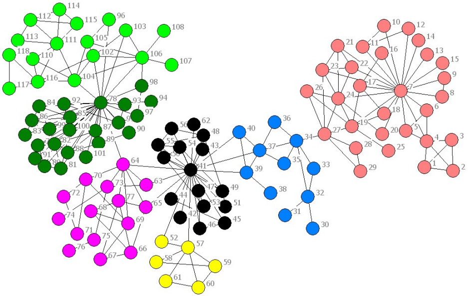
Courtesy: Wikipedia

Important properties of the real-world networks



- Degree Centrality: d_i is the number of neighbours of node i

Important properties of the real-world networks

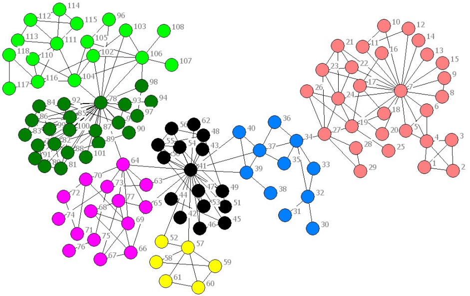


- Degree Centrality: d_i is the number of neighbours of node i
- Clustering Coefficient: What is the probability that two of my friends are also friends?

$$CC(i) = \frac{2T_i}{d_i(d_i - 1)}$$

Where T_i is the number of triangles at the given node i.

Important properties of the real-world networks



- Degree Centrality: d_i is the number of neighbours of node i
- Clustering Coefficient: What is the probability that two of my friends are also friends?

$$CC(i) = \frac{2T_i}{d_i(d_i - 1)}$$

Where T_i is the number of triangles at the given node i.

- Community structure: Clusters of nodes are seen.

Summary Comparison

Type of Network	Degree Distribution	Average Path Length	Clustering Coefficient
Random Network	Poisson	Short	Low
Real-world	Power-law	Short	High
Regular	Uniform	Long	High

Web graph

WWW can be modeled as a graph in which each page on the internet is considered as a node and a page 1 is connected with a directed link to page 2 if there is a hyperlink in page 1 pointing to page 2.

Web graph

WWW can be modeled as a graph in which each page on the internet is considered as a node and a page 1 is connected with a directed link to page 2 if there is a hyperlink in page 1 pointing to page 2.

Web graph

Stanford Graph Learning Workshop 2024

Stanford Data Science Affiliates Program

Register Now! Submit talk/poster!

The workshop will bring together leaders from academia and industry to showcase recent advances in Machine Learning and AI in Relational domains, Foundation Models, and Agents. The workshop will discuss methodological advancements, a wide range of applications to different domains, machine learning frameworks and practical challenges for large-scale training and deployment of AI models.

We are excited to announce a call for contributed talks and posters/demos at the upcoming Stanford Graph Learning Workshop. Submit your contribution!

Registration

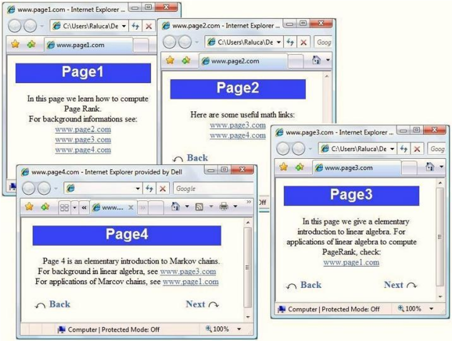
The Stanford Graph Learning Workshop will be held on Tuesday, Nov 5 2024, 09:00 - 18:00 Pacific Time.

The event will take place at Stanford University and will be live-streamed online. Free registrations are available. Register here.

Schedule

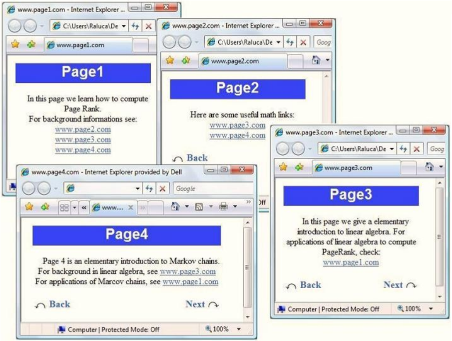
08:00 - 09:00		Registration & Breakfast
09:00 - 09:10	Jure Leskovec, Stanford University	Welcome and Overview
09:10 - 09:30	Joshua Robinson, Isomorphic Labs & Stanford	Relational Deep Learning - Graph Representation Learning on Relational Databases
09:30 - 09:50	Matthias Fey & Akhiro Nitta, PyG & Kumo.AI	What's New in PyG
09:50 - 10:10	Rishabh Ranjan, Stanford University	RelBench: A Benchmark for Deep Learning on Relational

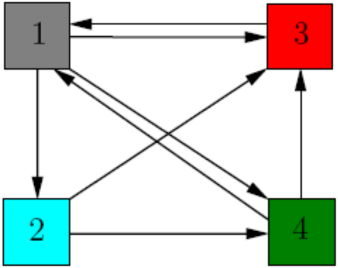
Web graph



www.page1.com - Internet Explorer ...
Page1
In this page we learn how to compute Page Rank.
For background informations see:
www.page2.com
www.page3.com
www.page4.com
www.page2.com - Internet Explorer ...
Page2
Here are some useful math links:
www.page3.com
www.page4.com
Back
www.page3.com - Internet Explorer ...
Page3
In this page we give a elementary introduction to linear algebra. For applications of linear algebra to compute PageRank, check:
www.page1.com
Back
Next
Computer | Protected Mode: Off
100%
www.page4.com - Internet Explorer provided by Dell
Page4
Page 4 is an elementary introduction to Markov chains.
For background in linear algebra, see www.page3.com
For applications of Marcov chains, see www.page1.com
Back
Next
Computer | Protected Mode: Off
100%

Web graph

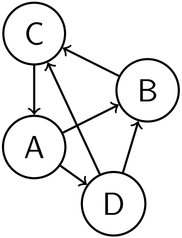




Courtesy: Slides from Prof. Remus, Cornell University

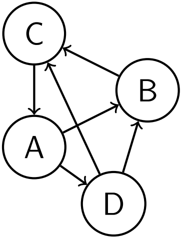
Graphs/ Networks as Matrices

A network is modeled as a



graph.

Graphs/ Networks as Matrices

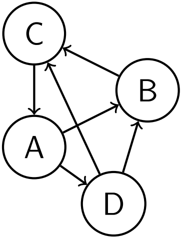
A network is modeled as a 

graph.

Form the adjacency matrix A of a graph.

Graphs/ Networks as Matrices

A network is modeled as a



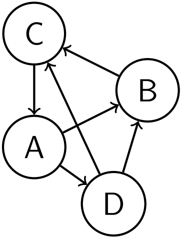
graph.

Form the adjacency matrix A of a graph.

	A	B	C	D
A	0	1	0	1
B	0	0	1	0
C	1	0	0	0
D	0	1	1	0

Graphs/ Networks as Matrices

A network is modeled as a



graph.

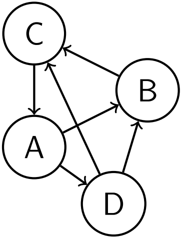
Form the adjacency matrix A of a graph.

	A	B	C	D
A	0	1	0	1
B	0	0	1	0
C	1	0	0	0
D	0	1	1	0

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

Graphs/ Networks as Matrices

A network is modeled as a



graph.

Form the adjacency matrix A of a graph.

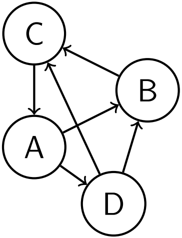
	A	B	C	D
A	0	1	0	1
B	0	0	1	0
C	1	0	0	0
D	0	1	1	0

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

What does this matrix denote?

Graphs/ Networks as Matrices

A network is modeled as a



graph.

Form the adjacency matrix A of a graph.

	A	B	C	D
A	0	1	0	1
B	0	0	1	0
C	1	0	0	0
D	0	1	1	0

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

What does this matrix denote? Path matrix of length 2!

Matrix as Linear Transformation

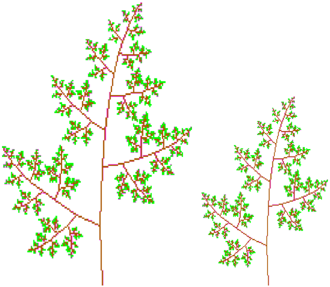
Scaling Transformations

Shrinking

$$\begin{pmatrix} 2/3 & 0 \\ 0 & 2/3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Zooming

$$\begin{pmatrix} 3/2 & 0 \\ 0 & 3/2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



Geometric meaning of matrices

Rotation Transformation

Rotation

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Geometric meaning of matrices

Rotation Transformation

Rotation

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$\theta = ?$

Geometric meaning of matrices

Rotation Transformation

Rotation

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$\theta = ?$

180^o

Eigen Vectors!

Importance of eigen values and eigen vectors of these matrices!!
Recall: The eigen vector x of a square matrix A satisfies the equation:

$$Ax = \lambda x$$

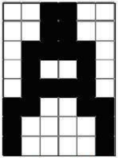
Whenever a vector x satisfies this equation, x is called the eigen vector corresponding to the eigen value λ.

Application: Face Recognition

Every image for a computer is just a matrix

Application: Face Recognition

Every image for a computer is just a matrix



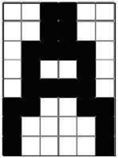
Digitized "A"

$$\begin{array}{cccccc}
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1
\end{array}$$

Matrix Form of "A"

Application: Face Recognition

Every image for a computer is just a matrix



Digitized "A"

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

Matrix Form of "A"

Image from PARI: People's Archive of Rural India

Problem: How do you match a given face against a million face database efficiently?

Principal Component Analysis

Answer: Dimensionality reduction using Principal Component Analysis

Principal Component Analysis

Answer: Dimensionality reduction using Principal Component Analysis

- The idea is to eliminate 'features' which are not contributing to the image in a significant way.
- Compute Principal Components: Find the dominant eigen vectors of the Covariance matrix $\Sigma_X = E[(X - \mu_X)(X - \mu_X)^T]$

Principal Component Analysis

Answer: Dimensionality reduction using Principal Component Analysis

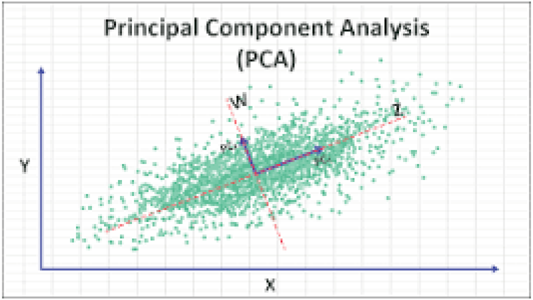
- The idea is to eliminate 'features' which are not contributing to the image in a significant way.
- Compute Principal Components: Find the dominant eigen vectors of the Covariance matrix $\Sigma_X = E[(X - \mu_X)(X - \mu_X)^T]$
- The eigen vector corresponding to the largest eigen value is called the principal component.

Principal Component Analysis

Answer: Dimensionality reduction using Principal Component Analysis

- The idea is to eliminate 'features' which are not contributing to the image in a significant way.
- Compute Principal Components: Find the dominant eigen vectors of the Covariance matrix $\Sigma_X = E[(X - \mu_X)(X - \mu_X)^T]$
- The eigen vector corresponding to the largest eigen value is called the principal component.
- The top eigen vectors derive the major 'directions' in which the image is concentrated.

Idea of PCA



Google Search Engine

How does google rank the results for our query?

- All the webpages are in a hyper-linked environment with each webpage pointing to other webpages.

Google Search Engine

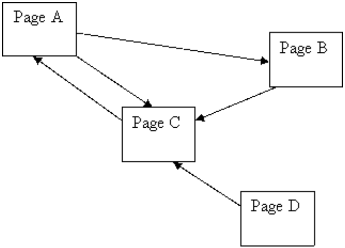
How does google rank the results for our query?

- All the webpages are in a hyper-linked environment with each webpage pointing to other webpages.
- This can be modeled as a network, with each webpage considered as a node. There is a directed edge between nodes A and B if A contains a hyperlink to page B.

Google Search Engine

How does google rank the results for our query?

- All the webpages are in a hyper-linked environment with each webpage pointing to other webpages.
- This can be modeled as a network, with each webpage considered as a node. There is a directed edge between nodes A and B if A contains a hyperlink to page B.



Google Search Engine

How does google rank the results for our query?

Figure: Larry Page and Sergei Brin

Page Rank

Idea!

Start a Random walker traversing the network following the links!
The more a node gets visited in the random walk, its importance grows more!

Start a Random walker traversing the network following the links! The more a node gets visited in the random walk, its importance grows more!

Wikipedia: size of each face proportional to the total size of faces pointing to it

Power iteration method

- Idea : Eigenvector Centrality:
- PageRank or r(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web.
- PageRank extends this idea by not counting links from all pages equally, and by normalizing by the number of links on a page.

Recursive Updation

Given a web graph with N nodes, where the nodes are pages and edges are hyperlinks.

1. Assign each node an initial page rank $\frac{1}{N}$
2. Repeat until convergence $\sum_i |r_j(t+1) - r_j(t)| < \epsilon$
   - Calculate the page rank of each node as sum of the page ranks of the nodes that point to it

$$r_j(t+1) = \sum_{i \to j} \frac{r_i(t)}{d_i}$$

d_i is the outdegree of node i.

Recursive Updation

Given a web graph with N nodes, where the nodes are pages and edges are hyperlinks.

1. Assign each node an initial page rank $\frac{1}{N}$
2. Repeat until convergence $\sum_i |r_j(t+1) - r_j(t)| < \epsilon$
   - Calculate the page rank of each node as sum of the page ranks of the nodes that point to it

$$r_j(t+1) = \sum_{i \to j} \frac{r_i(t)}{d_i}$$

d_i is the outdegree of node i.

The vector r that satisfies the matrix equation:

$$r = Mr$$

Recursive Updation

Given a web graph with N nodes, where the nodes are pages and edges are hyperlinks.

1. Assign each node an initial page rank $\frac{1}{N}$
2. Repeat until convergence $\sum_i |r_j(t+1) - r_j(t)| < \epsilon$
   - Calculate the page rank of each node as sum of the page ranks of the nodes that point to it

$$r_j(t+1) = \sum_{i \to j} \frac{r_i(t)}{d_i}$$

d_i is the outdegree of node i.

The vector r that satisfies the matrix equation:

$$r = Mr$$

An eigenvector equation!

This does not work if there is a 'dead-end' node!

Google Page Rank Algorithm

- The surfer randomly chooses one of its neighbours (hyperlinks) to visit with probability β

Google Page Rank Algorithm

- The surfer randomly chooses one of its neighbours (hyperlinks) to visit with probability β
- And with probability $(1 - \beta)$, jump to any node in the network.
- Idea : Recursive Updation of Ranks

Google Page Rank Algorithm

- The surfer randomly chooses one of its neighbours (hyperlinks) to visit with probability β
- And with probability $(1 - \beta)$, jump to any node in the network.
- Idea : Recursive Updation of Ranks
- Initialization: $r(j) = 1/N$ for all the nodes j in the network of size N.

Google Page Rank Algorithm

- The surfer randomly chooses one of its neighbours (hyperlinks) to visit with probability β
- And with probability $(1-\beta)$, jump to any node in the network.
- Idea : Recursive Updation of Ranks
- Initialization: $r(j) = 1/N$ for all the nodes j in the network of size N.

$$r(j) = \frac{1-\beta}{N} + \beta \left(\sum_{i \to j} \frac{r(i)}{d_i} \right)$$

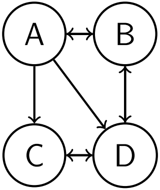
Google Page Rank Algorithm

- The surfer randomly chooses one of its neighbours (hyperlinks) to visit with probability β
- And with probability $(1-\beta)$, jump to any node in the network.
- Idea : Recursive Updation of Ranks
- Initialization: $r(j) = 1/N$ for all the nodes j in the network of size N.

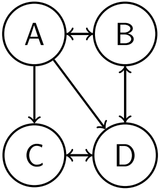
$$r(j) = \frac{1-\beta}{N} + \beta \left(\sum_{i \to j} \frac{r(i)}{d_i} \right)$$

- Solution of the equation corresponds to the principal eigenvector of the normalized link matrix of the web.

Example



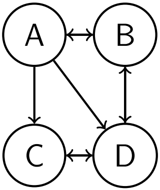
Example



Link Matrix L

A	0	1/3	1/3	1/3
B	1/2	0	0	1/2
C	0	0	0	1
D	0	1/2	1/2	0

Example

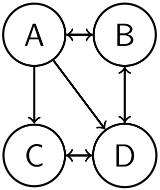


Link Matrix L

A	0	1/3	1/3	1/3
B	1/2	0	0	1/2
C	0	0	0	1
D	0	1/2	1/2	0

Using the update equation, the stochastic matrix is built.

Example

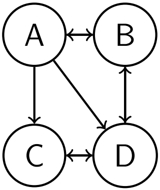


Link Matrix L

A	0	1/3	1/3	1/3
B	1/2	0	0	1/2
C	0	0	0	1
D	0	1/2	1/2	0

Using the update equation, the stochastic matrix is built. Find the eigen vector solution using an iterative method which converges.

Example



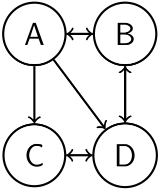
Link Matrix L

A	0	1/3	1/3	1/3
B	1/2	0	0	1/2
C	0	0	0	1
D	0	1/2	1/2	0

Using the update equation, the stochastic matrix is built. Find the eigen vector solution using an iterative method which converges.

r(A)	r(B)	r(C)	r(D)
0.125	0.2083	0.2083	0.4583

Example



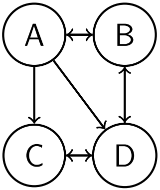
Link Matrix L

A	0	1/3	1/3	1/3
B	1/2	0	0	1/2
C	0	0	0	1
D	0	1/2	1/2	0

Using the update equation, the stochastic matrix is built. Find the eigen vector solution using an iterative method which converges.

r(A)	r(B)	r(C)	r(D)
0.125	0.2083	0.2083	0.4583
0.1354	0.2118	0.2118	0.4410

Example



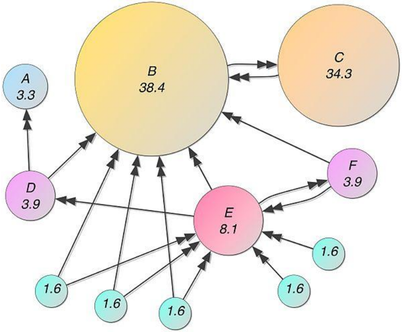
Link Matrix L

A	0	1/3	1/3	1/3
B	1/2	0	0	1/2
C	0	0	0	1
D	0	1/2	1/2	0

Using the update equation, the stochastic matrix is built. Find the eigen vector solution using an iterative method which converges.

r(A)	r(B)	r(C)	r(D)
0.125	0.2083	0.2083	0.4583
0.1354	0.2118	0.2118	0.4410
0.1084	0.2611	0.2611	0.4410
..			..
0.1200	0.2400	0.2400	0.4200

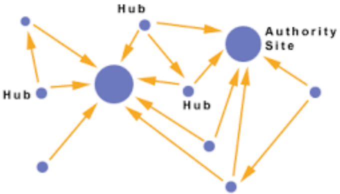
Example from Wikipedia



Kleinberg's HITS algorithm

Jon Kleinberg's proposed HITS (Hyperlink Induced Topic Search) algorithm which assigns two scores to nodes in a directed graph.

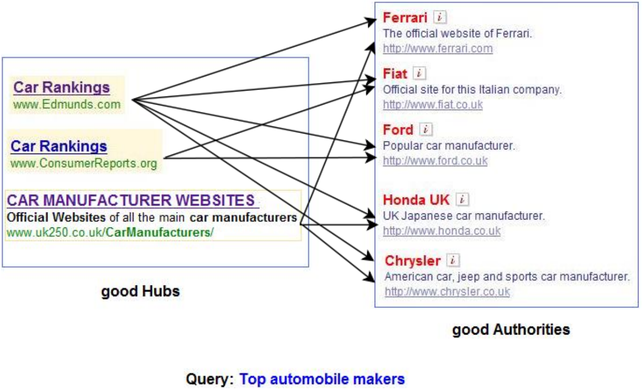
- Authority node: one that has many in-links
- Hub node: one that has many out-links



Hub and Authority nodes: Idea

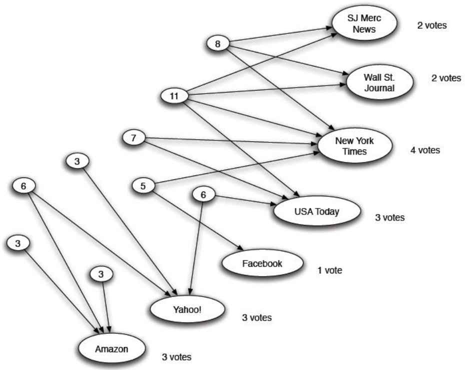
Acknowledgement: Most of these pictures are from Prof. Leskovec's slides.

- A good hub links to many good authorities
- A good authority is linked from many good hubs



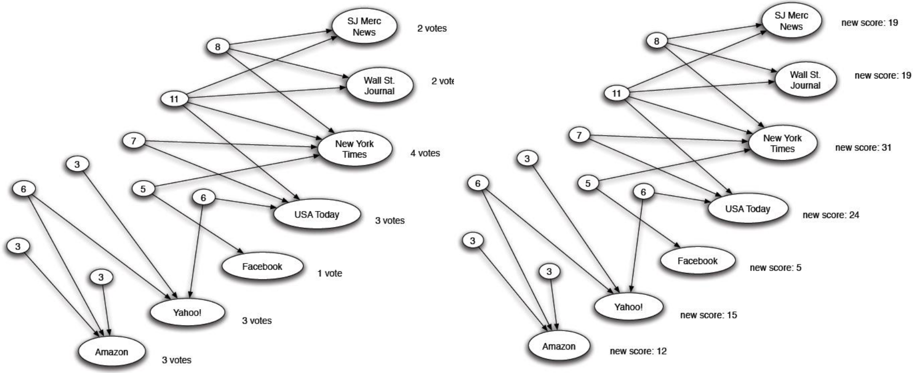
Hub and Authority Scores

- These weights are defined recursively.
- A higher authority(hub) weight occurs if the page is pointed to by pages with high hub (authority) weights and vice-versa.



Hub and Authority Scores

- These weights are defined recursively.
- A higher authority(hub) weight occurs if the page is pointed to by pages with high hub (authority) weights and vice-versa.



HITS Algorithm

Each page j has 2 scores: authority score a_j and hub score h_j.
Initialize: $a_j(0) = 1/\sqrt{N}, h_j(0) = 1/\sqrt{N}$.
Repeat the following updation steps for all the nodes i until convergence:

1. $$a_j(t+1) = \sum_{i \to j} h_i(t)$$
2. $$h_j(t+1) = \sum_{j \to i} a_i(t)$$
3. Normalize the scores.

Recursive updates

The algorithm basically recursively updates the two operations

$$a(t+1) = (A^T.A).a(t)$$

Recursive updates

The algorithm basically recursively updates the two operations

$$a(t+1) = (A^T.A).a(t)$$

$$h(t+1) = (A.A^T).h(t)$$

with $a(0)$ and $h(0)$ initialized as unit vectors.

- Once again ‘eigen vector’ equations!!

Recursive updates

The algorithm basically recursively updates the two operations

$$a(t+1) = (A^T.A).a(t)$$

$$h(t+1) = (A.A^T).h(t)$$

with $a(0)$ and $h(0)$ initialized as unit vectors.

- Once again 'eigen vector' equations!!
- The authority weight vector a is the probabilistic eigenvector corresponding to the largest eigenvalue of $A^T A$.

Recursive updates

The algorithm basically recursively updates the two operations

$$a(t+1) = (A^T.A).a(t)$$

$$h(t+1) = (A.A^T).h(t)$$

with $a(0)$ and $h(0)$ initialized as unit vectors.

- Once again 'eigen vector' equations!!
- The authority weight vector a is the probabilistic eigenvector corresponding to the largest eigenvalue of $A^T A$.
- The hub weights h of the nodes are given by the probabilistic eigenvector of the largest eigenvalue of AA^T.

Conclusions

- Page Rank and HITS algorithms are two of the most important algorithms for web search.

Conclusions

- Page Rank and HITS algorithms are two of the most important algorithms for web search.
- The WWW can be modeled as a graph and the search algorithms give rise to recursive update equations.

Conclusions

- Page Rank and HITS algorithms are two of the most important algorithms for web search.
- The WWW can be modeled as a graph and the search algorithms give rise to recursive update equations.
- The equations turn out to be eigen vector equations.

Conclusions

- Page Rank and HITS algorithms are two of the most important algorithms for web search.
- The WWW can be modeled as a graph and the search algorithms give rise to recursive update equations.
- The equations turn out to be eigen vector equations.
- Page rank scores correspond to the scores given by the eigen vector corresponding to the largest eigen value of the page rank matrix M.

Conclusions

- Page Rank and HITS algorithms are two of the most important algorithms for web search.
- The WWW can be modeled as a graph and the search algorithms give rise to recursive update equations.
- The equations turn out to be eigen vector equations.
- Page rank scores correspond to the scores given by the eigen vector corresponding to the largest eigen value of the page rank matrix M.
- Hubs and authority scores are given by the eigen vectors corresponding to the largest eigen values of $A^T A$ and AA^T respectively.

References

1. M.J.Newman, The structure and function of complex networks, SIAM Review, 45:167-256, 2003.
2. J.Lescovec, The power of social networks, Stanford Series: The Future of Everything, 2017.
3. D.Easley, J.Kleinberg, Networks, Crowds and Markets, Cambridge Univ Press, 2010
4. D.J.Watts, S.H.Strogatz, Collective dynamics of ‘small-world’ networks, Nature, 393:440-442,, 1998.
5. S.Wasserman, K.Faust, Social Network Analysis, 1994.
6. S.Fortunato, Community detection in graphs, Physics Reports, 2010.
7. Charu C. Aggarwal, Social Network Data Analytics, Springer, 2011.

Acknowledgment: To Google Images!!

Acknowledgment: To Google Images!!

Thank you for listening!