Analysis of large graphs with PageRank vectors

Dimosthenis Pasadakis

Università della Svizzera italiana
Faculty of Informatics
Institute of Computing
CI

Institute of Computing
Università della Svizzera italiana

September 26, 2023

Course calendar

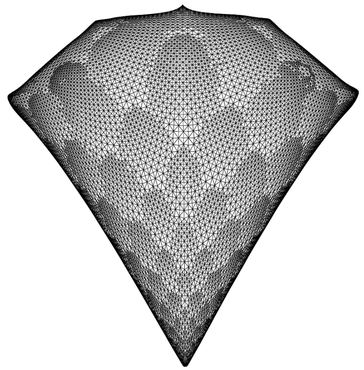
Project 1	Project 2	Project 3	Project bonus	Project 4
PagerRank and networks	Graph partitioning	Graph clustering	Linear least squares	Image deblurring with CG
26.09	04.10	18.10	08.11	15.11

Today: PageRank and networks

Elements: directed networks, columm stochastic matrices, teleportation

PageRank vector: Network structure + Iterative computation

Using: eigenvalue computations, linear system solution



Course calendar

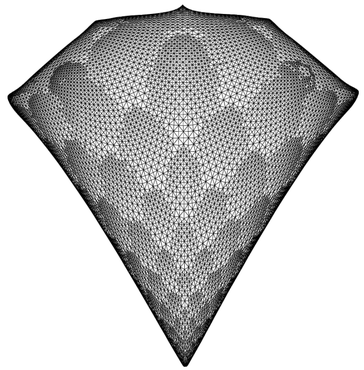
Project 1	Project 2	Project 3	Project bonus	Project 4
PagerRank and networks	Graph partitioning	Graph clustering	Linear least squares	Image deblurring with CG
26.09	04.10	18.10	08.11	15.11

Today: PageRank and networks

Elements: directed networks, columm stochastic matrices, teleportation

PageRank vector: Network structure + Iterative computation

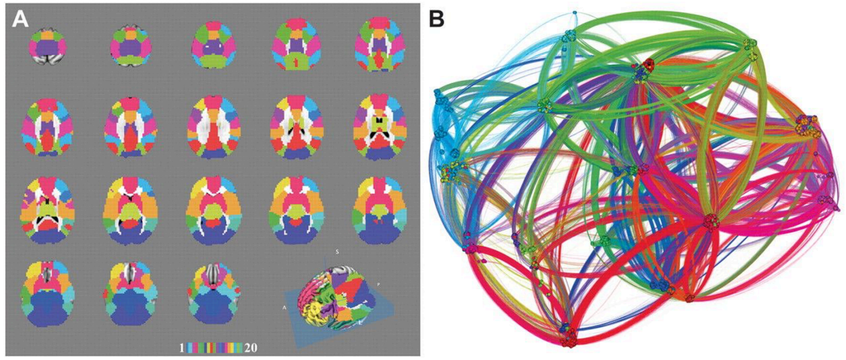
Using: eigenvalue computations, linear system solution



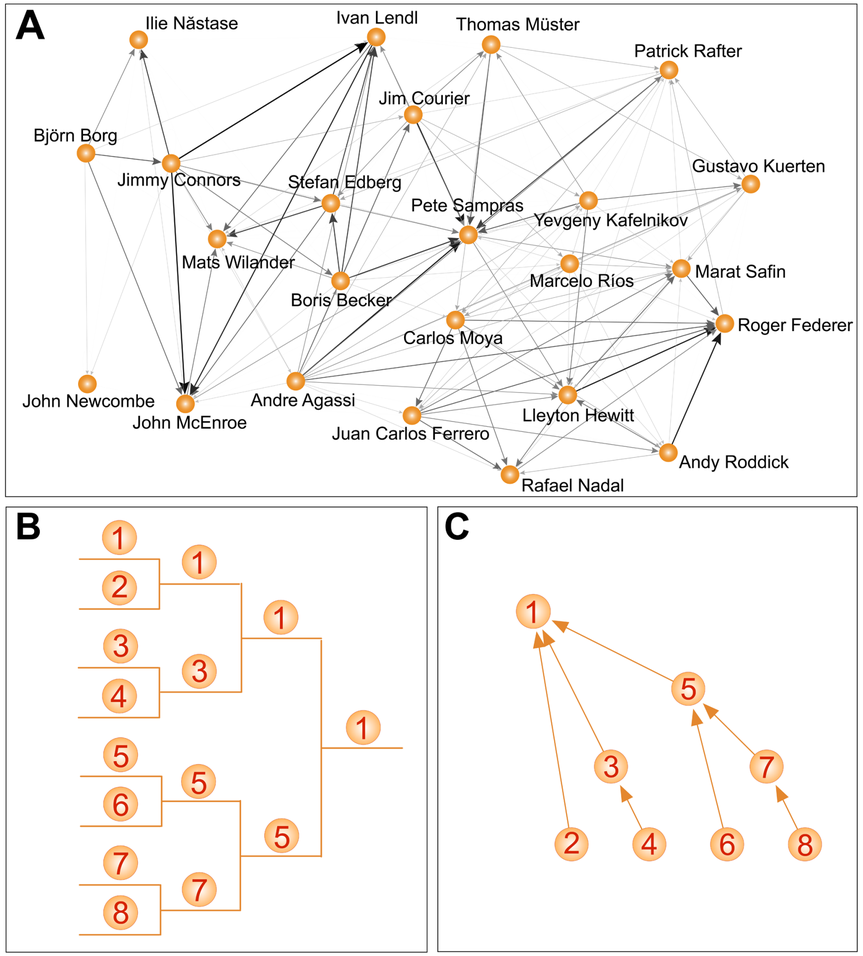
Applications

Rank the importance of graph nodes

- Biology
- Sports
- Local clustering
- ...
- **Web search engines**



- X.-N. Zuo, et al., Network centrality in the human functional connectome, Cerebral Cortex, 2012.

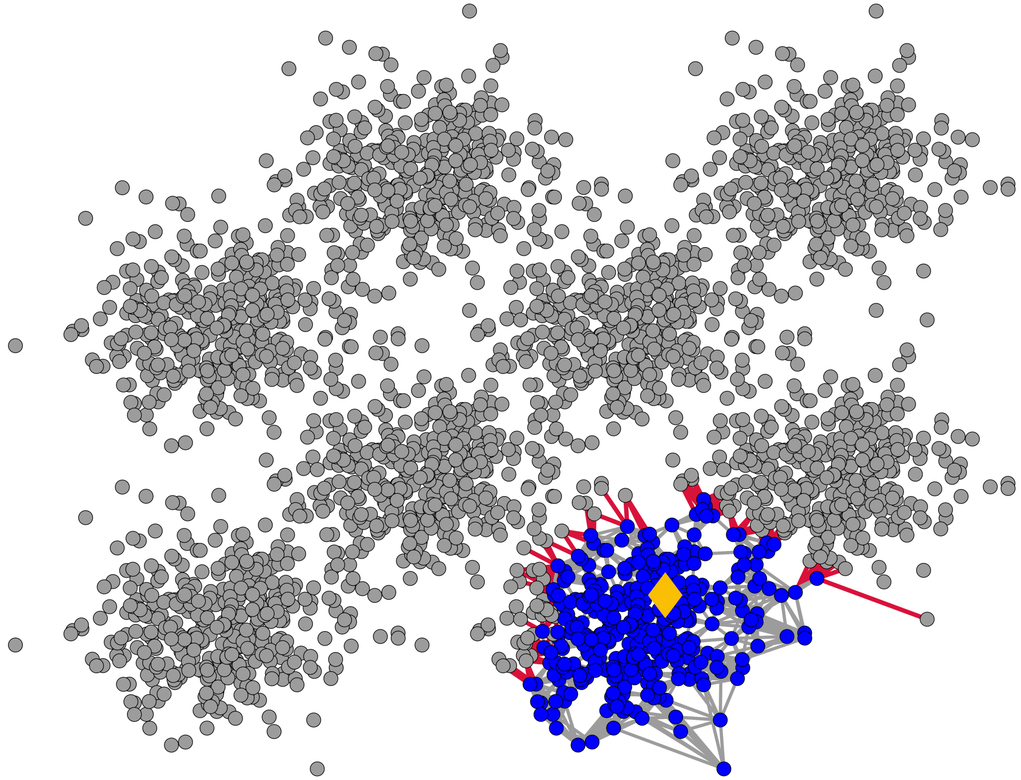


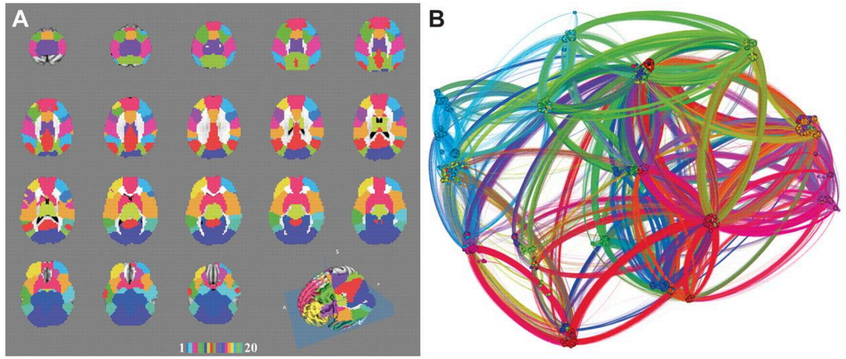
- F. Radicchi, Who is the best player ever? A complex network analysis of the history of professional tennis, PLoS ONE, 2011.

Applications

Rank the importance of graph nodes

- Biology
- Sports
- Local clustering
- ...
- **Web search engines**





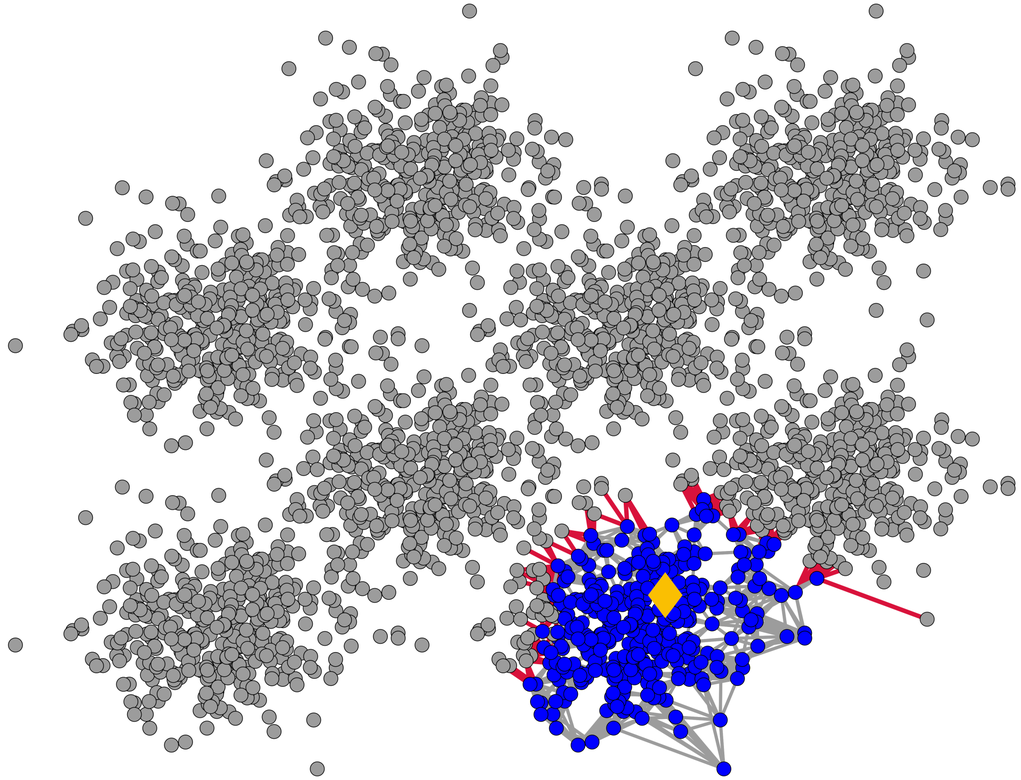
- C. Kodsi, and D. Pasadakis, Nonlinear PageRank for local graph partitioning, *in preparation*, 2023.

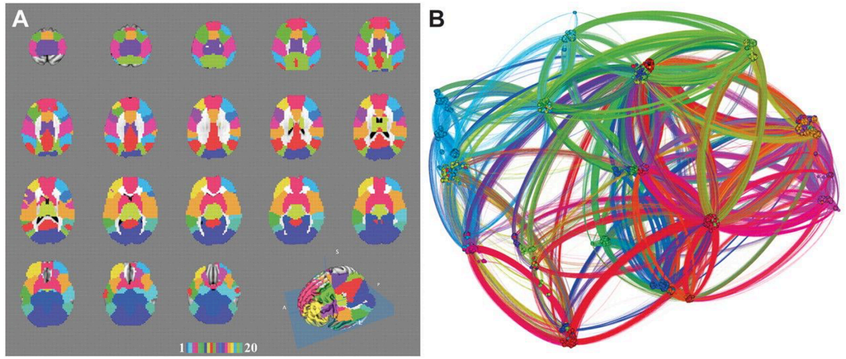
- X.-N. Zuo, et al., Network centrality in the human functional connectome, Cerebral Cortex, 2012.

Applications

Rank the importance of graph nodes

- Biology
- Sports
- Local clustering
- ...
- **Web search engines**





- C. Kodsi, and D. Pasadakis, Nonlinear PageRank for local graph partitioning, *in preparation*, 2023.

- X.-N. Zuo, et al., Network centrality in the human functional connectome, Cerebral Cortex, 2012.

Bringing order to the web

Search engines and users' queries

❶ Text processing to find all documents using the query terms.
- Look-up into an inverted file, with a vector space method, or
- query expander that uses a thesaurus.

✖ Massive size of the web, thousands of results.

➜ Rank this list based on some ranking criterion.

Martin Grandjean. A social network analysis of Twitter: Mapping the digital humanities community. Cogent Arts & Humanities, 3(1), apr 2016.

Bringing order to the web

Search engines and users' queries

❷ Exploit additional information to rank the results

- The web has a hyperlink structure ➞ link analysis.
- The **Google** search engine uses

➜ IR (Information Retrieval) score, combined with,

➜ PR (PageRank) score.

Martin Grandjean. A social network analysis of Twitter: Mapping the digital humanities community. Cogent Arts & Humanities, 3(1), apr 2016.

✔ **We will focus on the PR score in this class.**

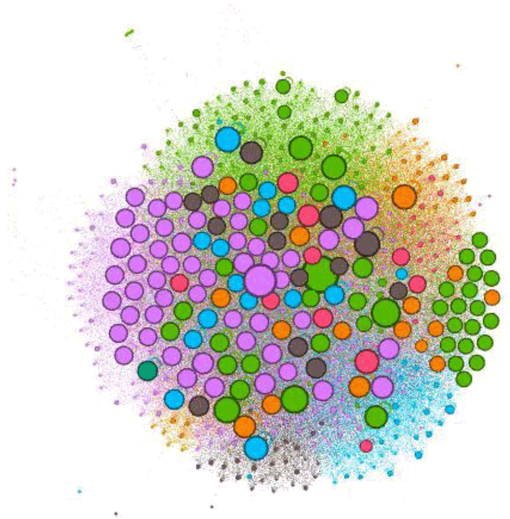
The web as a directed graph

Graphical structure

- Nodes: webpages ■
- Edges: hyperlinks →
- Not all web pages have the same importance (weight).
  www.usi.ch
  vs
  wikipedia.org/wiki/Empididae

Importance ∼ number of links

- In-coming links →■
- Out-going links ■→
- Links from important pages ➽ more gravity.



We will rank the pages based on the link structure.

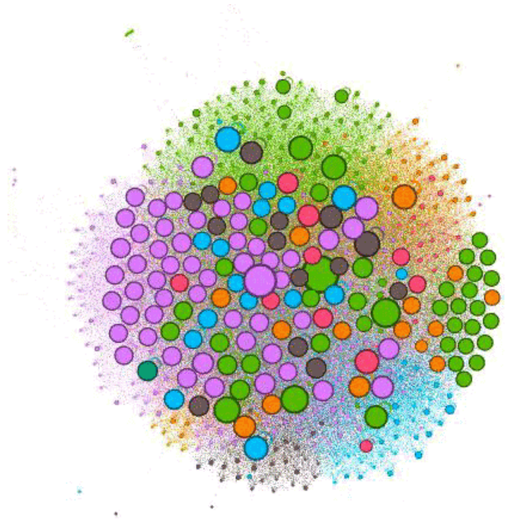
The web as a directed graph

Graphical structure

- Nodes: webpages ■
- Edges: hyperlinks →
- Not all web pages have the same importance (weight).
  www.usi.ch
  vs
  wikipedia.org/wiki/Empididae

Importance ∼ number of links

- In-coming links →■
- Out-going links ■→
- Links from important pages ➽ more gravity.

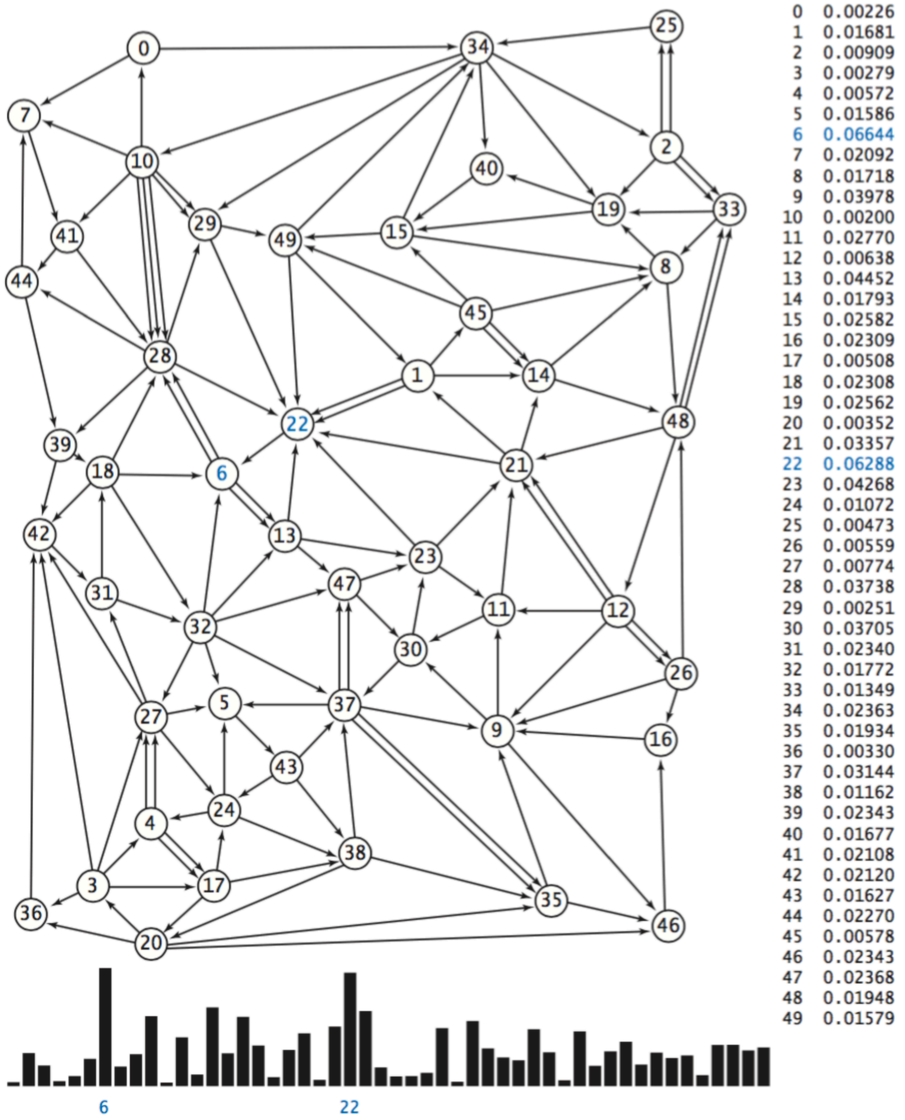


We will rank the pages based on the link structure.

Random surfing

Upon visiting a page on the web, our random surfer tosses a coin.

- ▲ **Heads:** randomly click a link on the current page and transition.
- ▼ **Tails:** teleport to a possibly random page.
  - Pages where more likely to be visited (based on the web's structure) are more important in a PR sense.
  - **Teleporting:** designed also to model external influence on the importance of nodes, can be more than a random choice.

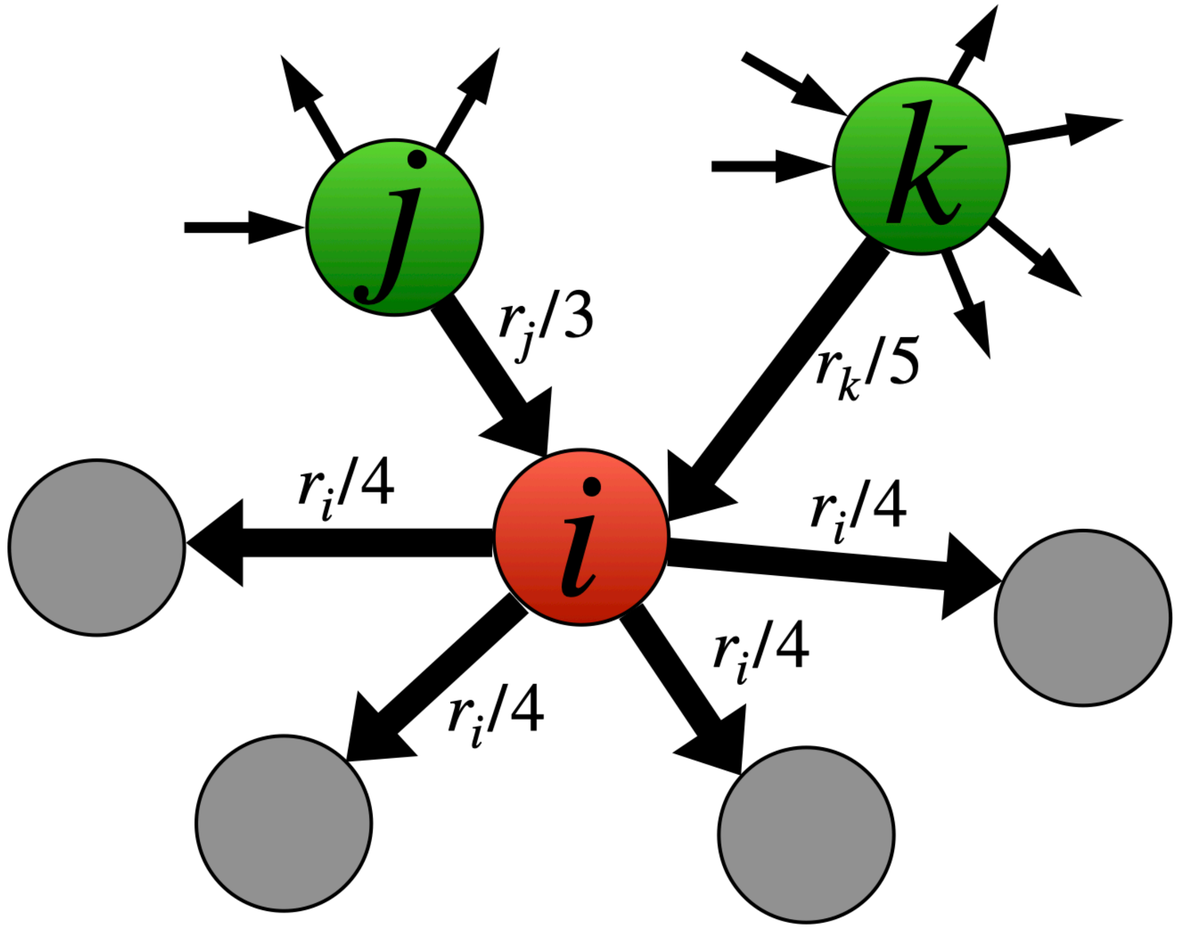


- https://pi.math.cornell.edu/, The Mathematics of Google Search.

The PageRank score

A recursive calculation

- Each link's gravity is proportional to the importance of its source page.
- A page i with importance r_i has n out-links. Each link gets r_i/n weight.
- The importance of page i is the sum of the weights on its in-links.



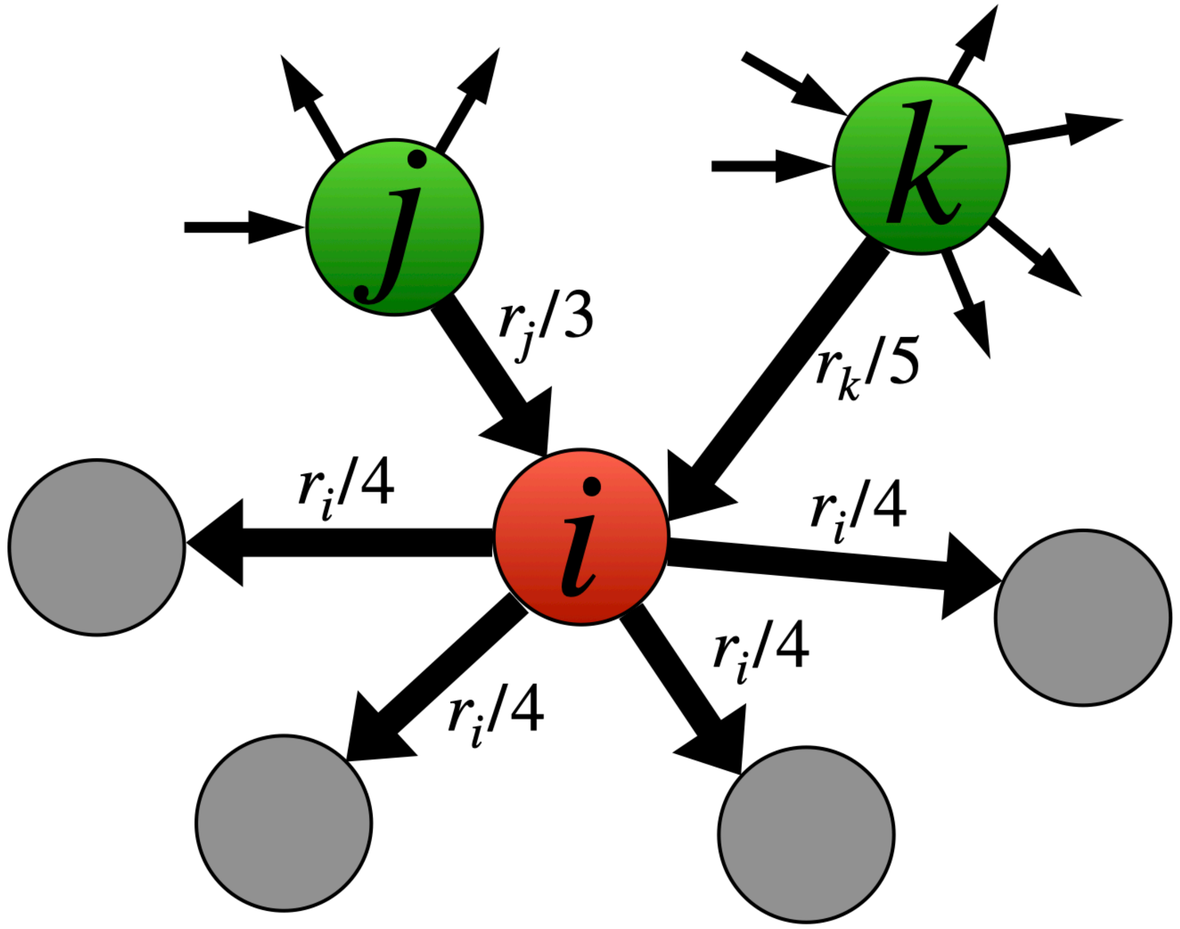
$$r_i = r_j/3 + r_k/5$$

PageRank as a flow model

- The weight of an important page is higher.
- A page is important if other important pages → to it.
- The rank r_i of a page i is defined as

$$r_j = \sum_{i \to j} \frac{r_i}{d_i},$$

where d_i is the out-degree of node i, and r_j the solutions to the flow equation.



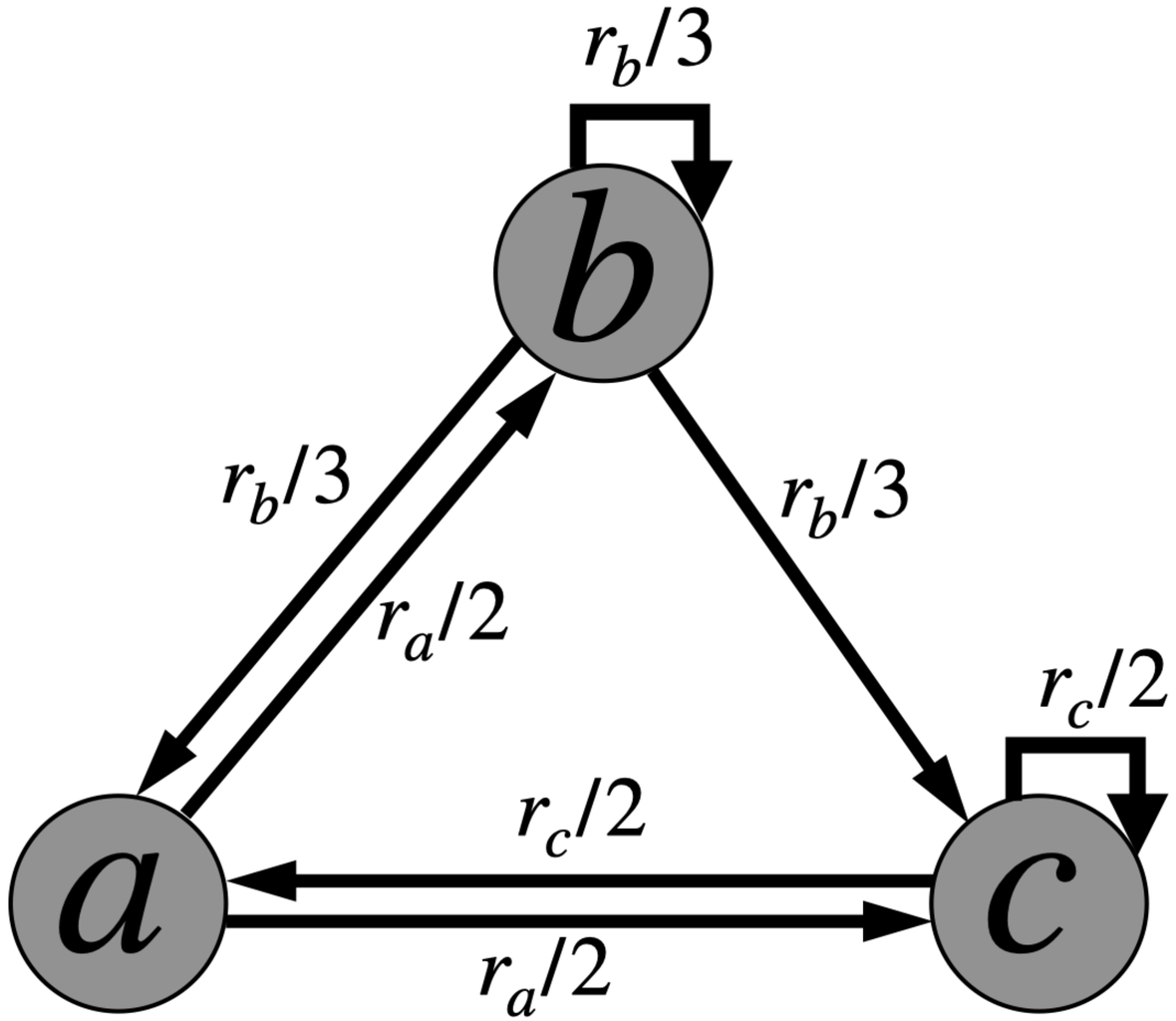
A small example

The flow equations

$$r_a = r_b/3 + r_c/2,$$
$$r_b = r_a/2 + r_b/3,$$
$$r_c = r_a/2 + r_b/3 + r_c/2.$$

- Enforce $r_a + r_b + r_c = 1$ to guarantee uniqueness.
- Gaussian elimination can solve this.

We need a better/faster method, and a new formulation for web-size graphs.



A new formulation

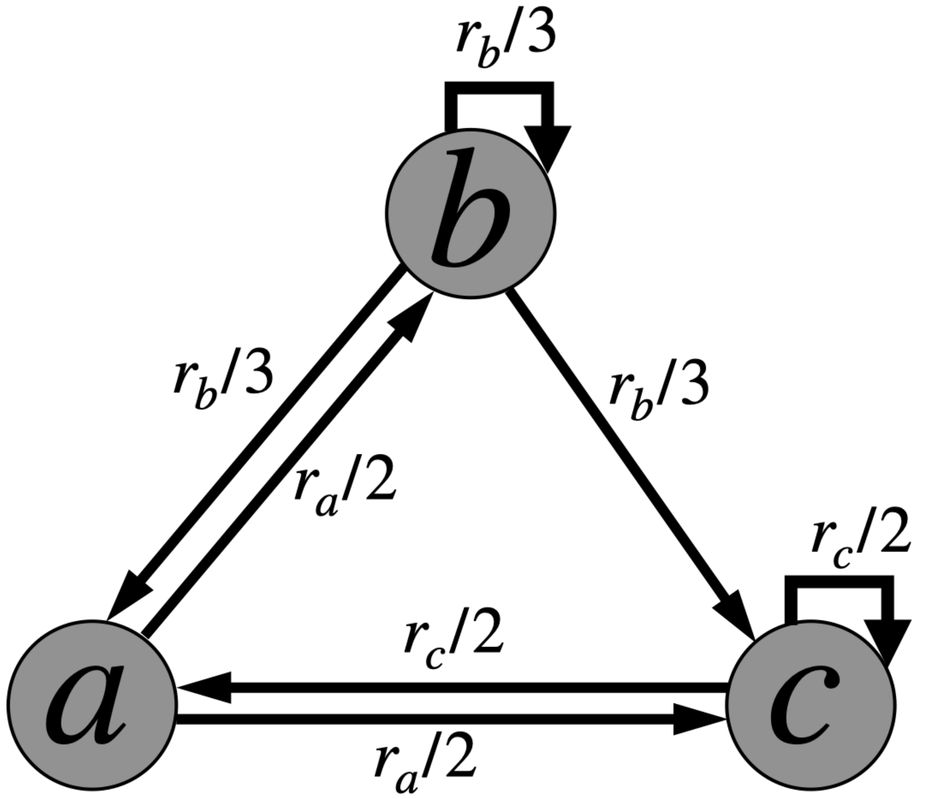
$$r_j = \sum_{i \to j} \frac{r_i}{d_i} \Rightarrow \mathbf{r} = \mathbf{M}\mathbf{r}.$$

Matrix formulation

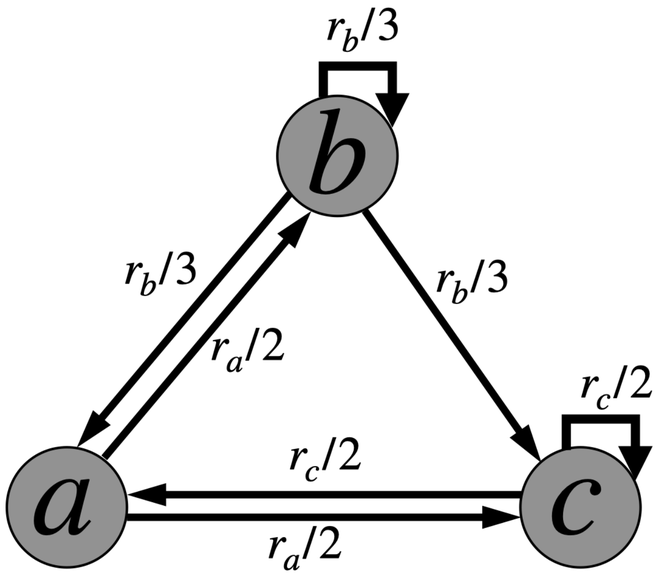
- Define the column stochastic adjacency matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$.
- $\sum_i \mathbf{M}_{ij} = 1$.
- Page i has d_i out-links.
- If $i \to j$, then $\mathbf{M}_{ji} = \frac{1}{d_i}$, else $\mathbf{M}_{ji} = 0$.
- The rank vector $\mathbf{r}$ has one entry per page ➜ importance of the page.

$$\mathbf{r}_i = \text{importance of page } i$$

$$\sum_i \mathbf{r}_i = 1$$



The same small example



The flow equations

$$r_a = r_b/3 + r_c/2,$$
$$r_b = r_a/2 + r_b/3,$$
$$r_c = r_a/2 + r_b/3 + r_c/2.$$

In matrix form

$$\mathbf{M} = \begin{matrix} \\ a \\ b \\ c \end{matrix} \begin{matrix} \begin{matrix} a & b & c \end{matrix} \\ \begin{pmatrix} 0 & 1/3 & 1/2 \\ 1/2 & 1/3 & 0 \\ 1/2 & 1/3 & 1/2 \end{pmatrix} \end{matrix}$$

$$\mathbf{r} = \mathbf{M} \cdot \mathbf{r}$$

$$\begin{bmatrix} r_a \\ r_b \\ r_c \end{bmatrix} = \begin{pmatrix} 0 & 1/3 & 1/2 \\ 1/2 & 1/3 & 0 \\ 1/2 & 1/3 & 1/2 \end{pmatrix} \cdot \begin{bmatrix} r_a \\ r_b \\ r_c \end{bmatrix}$$

Is this a familiar equation?

Eigenvector approximation

The power iteration

- An approximation method for computing the eigenvectors of a matrix. It uses the following recursive formula to find the principal eigenvector $\mathbf{r}$ of a matrix $\mathbf{M}$:

$$\mathbf{r}^{n+1} = \frac{\mathbf{M} \cdot \mathbf{r}^{\mathbf{n}}}{\|\mathbf{M} \cdot \mathbf{r}^{n}\|}$$

ALGORITHM: Power iteration method

1: **Input**: Graph $\mathbf{M}$ with n nodes and m links
2: **Initialize**: $\mathbf{r}_0 = [1/n, \ldots, 1/n]^\intercal$
3: **while** $|\mathbf{r}_k - \mathbf{r}_{k-1}|_1 \geq \epsilon$ **do**
4: $\quad \tilde{\mathbf{r}} = \mathbf{M} \cdot \mathbf{r}_{k-1}$
5: $\quad \mathbf{r}_k = \tilde{\mathbf{r}}/\|\tilde{\mathbf{r}}\|$
6: **end while**

A simplification

- In the case of PR, $\mathbf{M}$ is col.-stochastic, and $\mathbf{r}$ is a probability vector, therefore $\|\mathbf{M} \cdot \mathbf{r}\| = 1$, and we reduce the power iteration to

$$\mathbf{r}^{n+1} = \mathbf{M} \cdot \mathbf{r}^{\mathbf{n}}$$

Eigenvector formulation

Flow equations in matrix form $\mathbf{r} = \mathbf{M} \cdot \mathbf{r}$

- $\mathbf{x}$ is an eigenvector with corresponging eigenvalue λ if: $\mathbf{Ax} = \lambda\mathbf{x}$.
- $\mathbf{r}$ (the rank vector) is an eigenvector of the col. stochastic web adjacency matrix $\mathbf{M}$.
- Starting from any stochastic $\mathbf{u}$, the rank $\mathbf{r}$ is the limit of $\mathbf{M} \cdot \mathbf{M} \cdots \mathbf{Mu}$.
- The associated eigenvalue $\lambda = 1$.
- We can efficiently solve for $\mathbf{r}$.

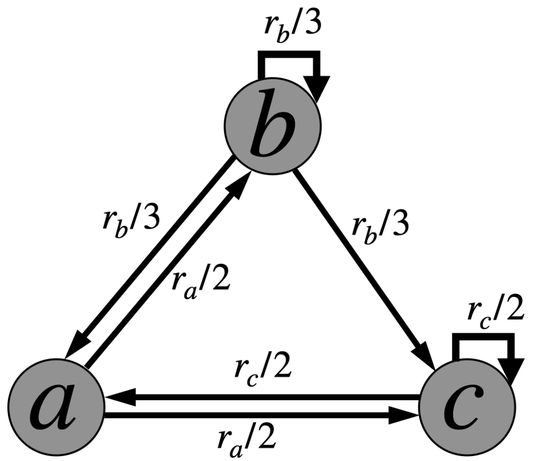
(Power method or Shift and inverse)

ALGORITHM: Power iteration method

1: **Input**: Graph $\mathbf{M}$ with n nodes and m links
2: **Initialize**: $\mathbf{r}_0 = [1/n, \ldots, 1/n]^\intercal$
3: **while** $|\mathbf{r}_k - \mathbf{r}_{k-1}|_1 \geq \epsilon$ **do**
4: $\quad \tilde{\mathbf{r}} = \mathbf{M} \cdot \mathbf{r}_{k-1}$
5: $\quad \mathbf{r}_k = \tilde{\mathbf{r}}/\|\tilde{\mathbf{r}}\|$
6: **end while**

- L1 norm (Manhattan distance/Taxicab norm)
  $|\mathbf{r}|_1 = \sum_{1 \leq i \leq n} |r_i|$.

The same small example



$$\mathbf{M} = \begin{array}{c} \begin{matrix} a & b & c \end{matrix} \\ \begin{pmatrix} 0 & 1/3 & 1/2 \\ 1/2 & 1/3 & 0 \\ 1/2 & 1/3 & 1/2 \end{pmatrix} \end{array} \begin{matrix} a \\ b \\ c \end{matrix}$$

❶ $\mathbf{M}$ is col. stoch., and $\mathbf{M}_{ij} \geq 0$.

➠ Perron-Frobenius: A unique $\lambda = 1$ exists, $|\lambda_1| = 1 > |\lambda_2| \geq |\dots| \geq |\lambda_n|$, and $\mathbf{v}_1 \in \mathbb{R}^n$.

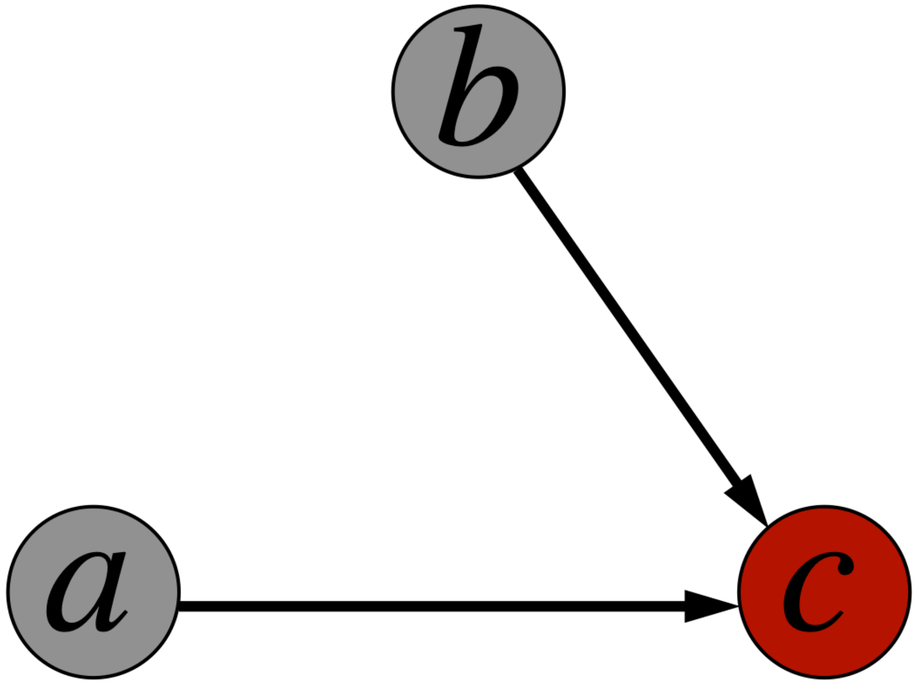
➠ Initial unbiased guess $\mathbf{r}_0 = [1/3, 1/3, 1/3]^\intercal$

❷ Converges to $\mathbf{r} = \mathbf{M}^8 * \mathbf{r}_0 = [0.3077, 0.2308, 0.4615]^\intercal$, so $\mathrm{rank} = [2, 3, 1]$.

$$\mathbf{r}_1 = \mathbf{M} \cdot \mathbf{r}_0,$$
$$\mathbf{r}_2 = \mathbf{M} \cdot \mathbf{r}_1 = \mathbf{M}(\mathbf{M}\mathbf{r}_0) = \mathbf{M}^2 \cdot \mathbf{r}_0,$$
$$\mathbf{r}_3 = \mathbf{M} \cdot \mathbf{r}_2 = \mathbf{M}(\mathbf{M}^2\mathbf{r}_0) = \mathbf{M}^3 \cdot \mathbf{r}_0,$$
$$\mathbf{r}_4 = \dots$$

Adapting to real-world (1)

Col stochastic assumption violated.



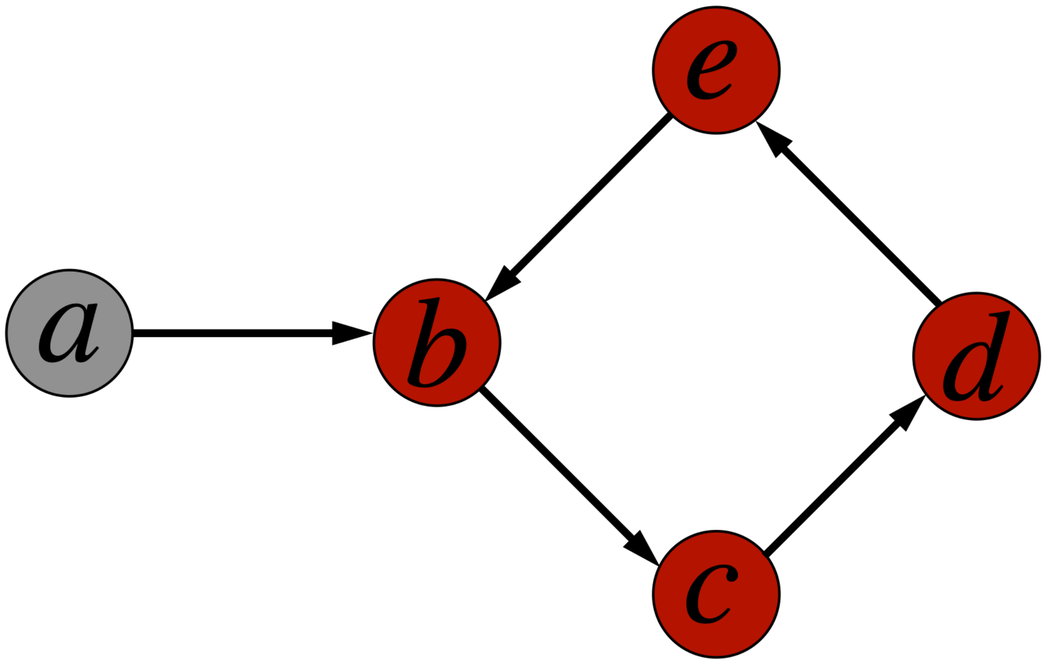
Dangling nodes

- No outlinks from a certain webpage.
- The corresponding column in A is zero.
- At node c, we are "stuck".
- Change the whole column from zeros to entries $1/n$.
- If surfers follow links and arrive at a webpage without outlinks, they can jump out to any webpage with equal probability.
- **M** is now col. stochastic.

Always teleport!

Adapting to real-world (2)

The trap absorbs all importance.



Probabilistically teleport!

Spider traps

- Cyclic path with no way out.
- Iteratively, the trap will absorb all importance.
- Propabilistically teleport
- With probability p follow a link at random (follow the edges structure).
- With probability $1 - p$ jump to a random page. $(1 - p)/n$ for a specific jump.
- Teleport out of the trap in a finite number of steps.
- Typically $p \in [0.8, 0.9]$, jump every 5 steps on avg.

The Google matrix

- Preprocess matrix $\mathbf{M}$ to remove dangling nodes.
- PageRank equation reads

$$r_j = \sum_{i \to j} p \frac{r_i}{d_i} + (1-p)\frac{1}{n},$$

d_i: the out-degree of node i.

- The Google matrix $\mathbf{A}$ reads

$$\begin{aligned} \mathbf{A} &= p\mathbf{M} + (1-p)\left[\frac{1}{N}\right]_{n\times n}, \\ &= p\mathbf{M} + (1-p)\mathbf{u}\mathbf{e}^\mathsf{T}, \end{aligned}$$

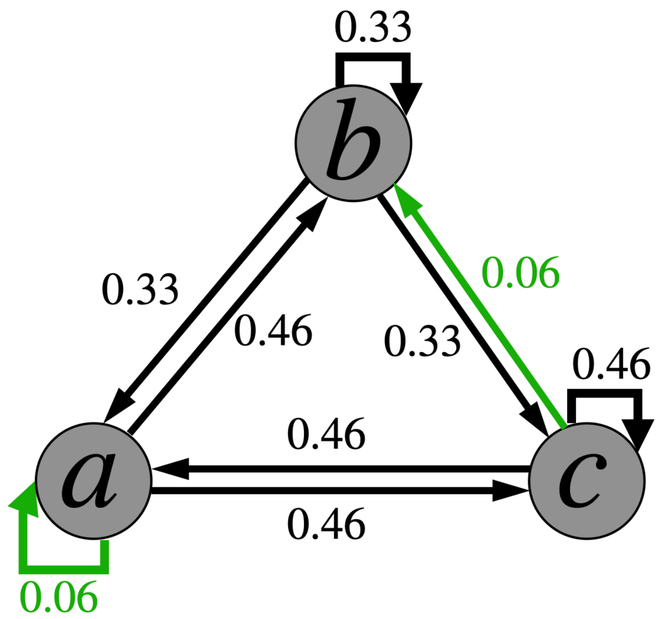
$\mathbf{u} = \frac{1}{n}\mathbf{e}$: unbiased personalization vector.

Dominant eigenvector computation

- Recursive problem $\mathbf{r} = \mathbf{A} \cdot \mathbf{r}$.
- Dominant eigenvalue $\lambda = 1$.

➜ We will employ the power iteration.

The Google matrix - computation



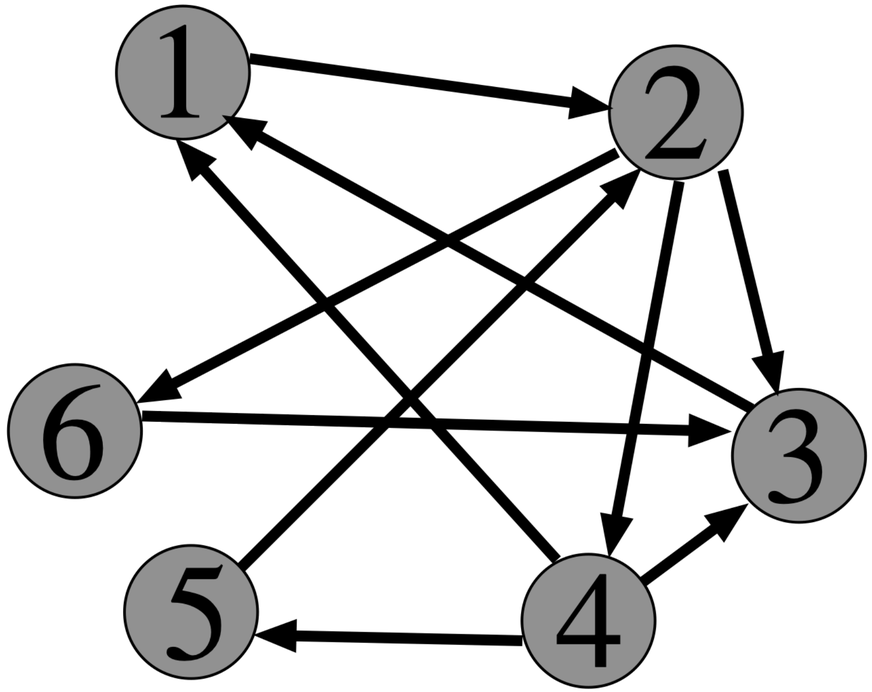
- If the personalization vector $\mathbf{u}$ contains zeros, then the random jump is done only to the nodes of the graph that has a corr. $u_i \neq 0, i \in \{1, n\}$.

$$\mathbf{r} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{array}{ccccc} \mathbf{A}\mathbf{r}_0 & \mathbf{A}^2\mathbf{r}_0 & \mathbf{A}^3\mathbf{r}_0 & \mathbf{A}^k\mathbf{r}_0 & \mathbf{A}^7\mathbf{r}_0 \\ 1/3 & 0.2889 & 0.3126 & \cdots & 0.3085 \\ 1/3 & 0.2889 & 0.2593 & \cdots & 0.2594 \\ 1/3 & 0.4222 & 0.4281 & \cdots & 0.4321 \end{array}$$

Let $p = 0.8, \mathbf{u} = \frac{1}{n}\mathbf{e}$.

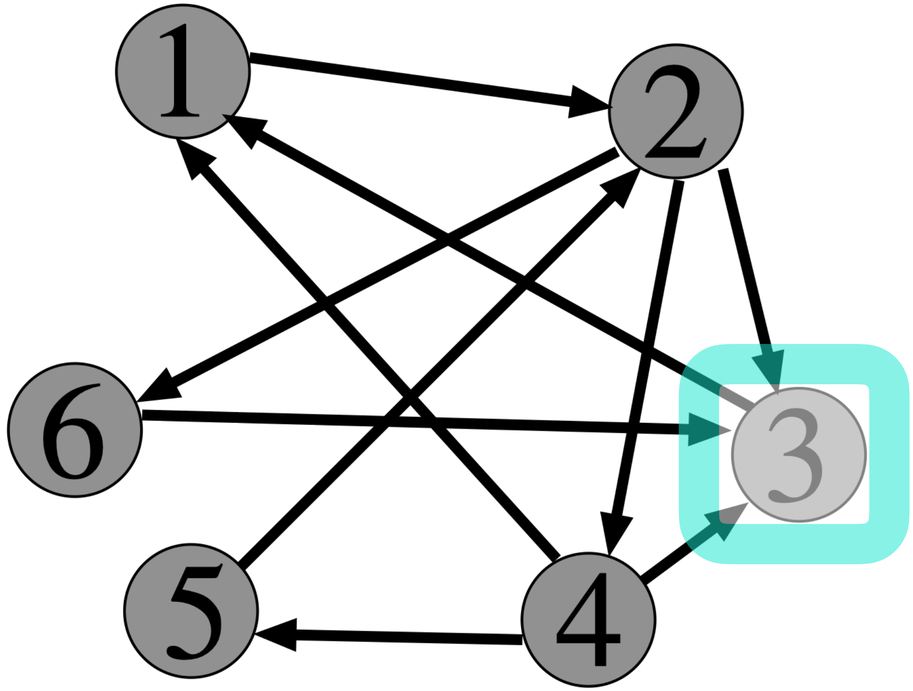
$$\mathbf{A} = p\mathbf{M} + (1-p)\mathbf{u}\mathbf{e}^\intercal = 0.8 \cdot \begin{pmatrix} 0 & 1/3 & 1/2 \\ 1/2 & 1/3 & 0 \\ 1/2 & 1/3 & 1/2 \end{pmatrix} + 0.2 \cdot \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} = \begin{pmatrix} 0.0667 & 0.3333 & 0.4667 \\ 0.4667 & 0.3333 & 0.0667 \\ 0.4667 & 0.3333 & 0.4667 \end{pmatrix}$$

Degree $\neq$ PageRank



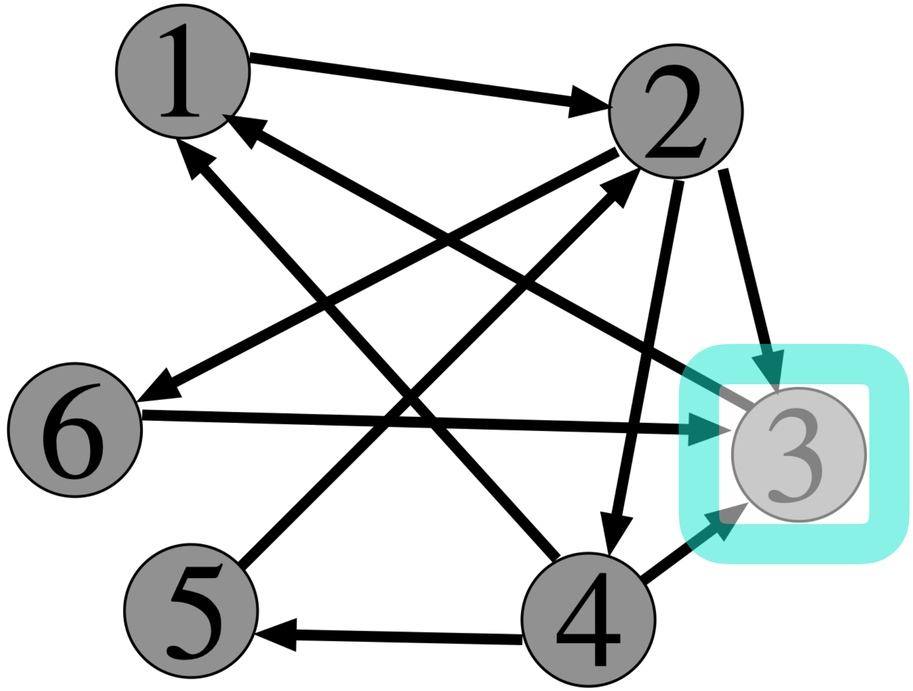
Can you identify the node with the highest in-degree?

Degree $\neq$ PageRank

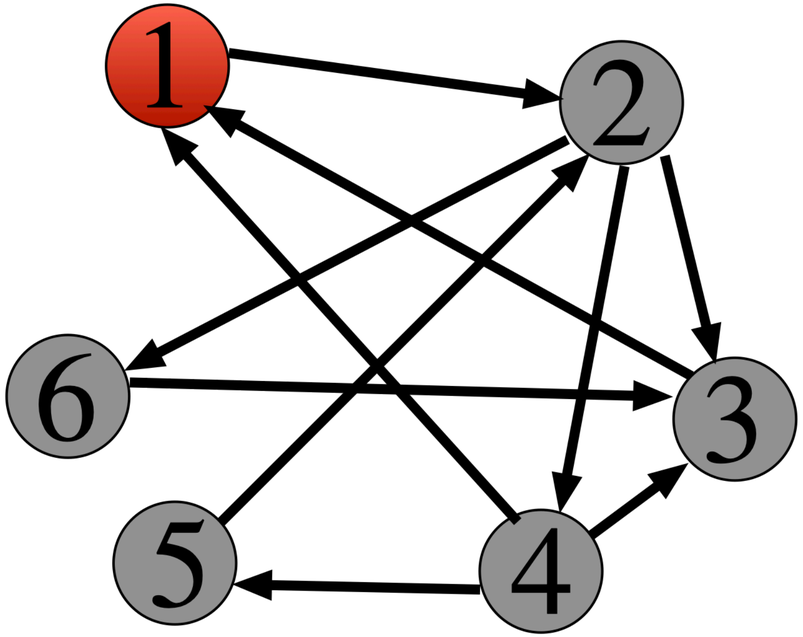


Can you identify the node with the highest in-degree?

Degree $\neq$ PageRank

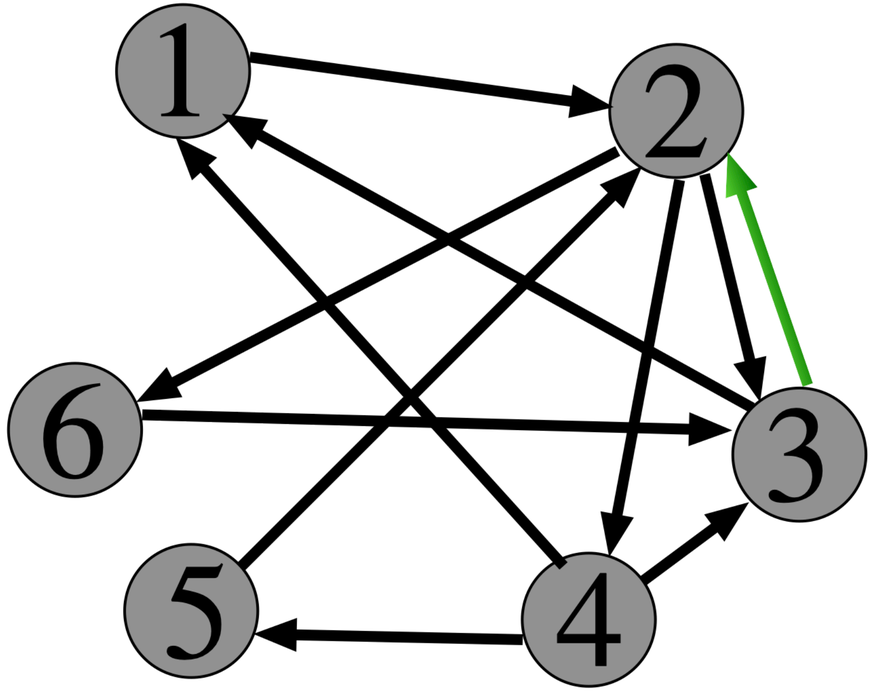


Can you identify the node with the highest in-degree?



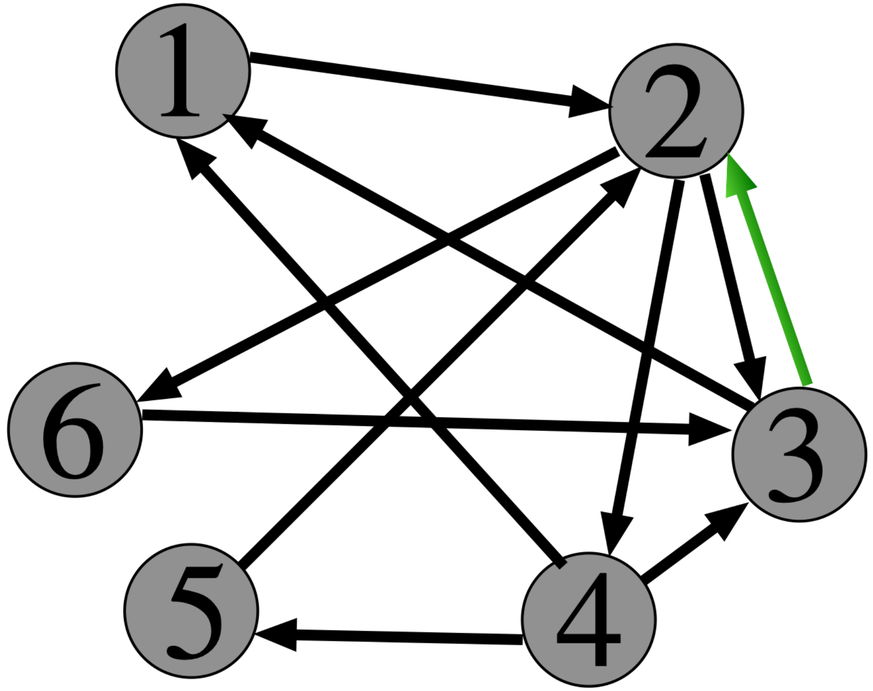
The PageRank ranking does not always correspond to the degree ranking.

Degree $\neq$ PageRank

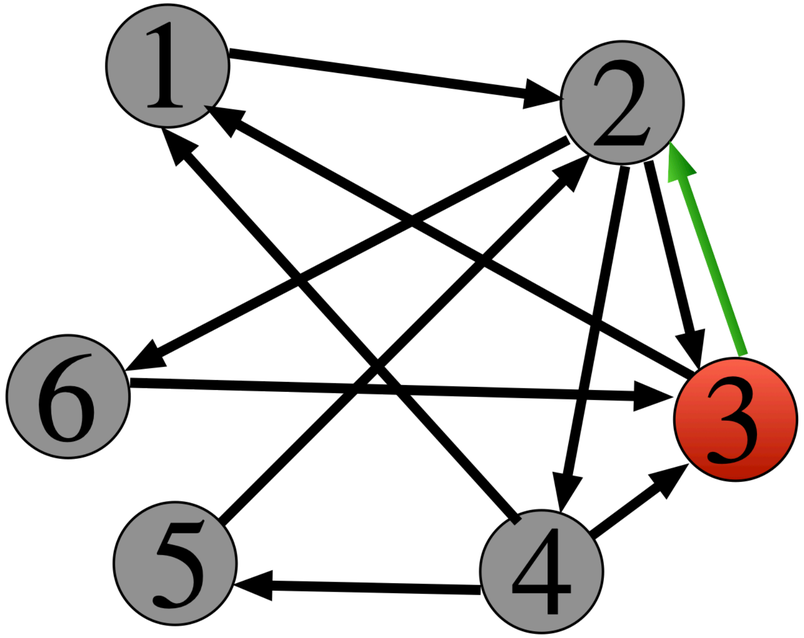


Let's perturb the graph slightly.

Degree $\neq$ PageRank



Let's perturb the graph slightly.



Small changes in the (small) graph structure change the PR drastically.

The picture today

Does Google still use it?

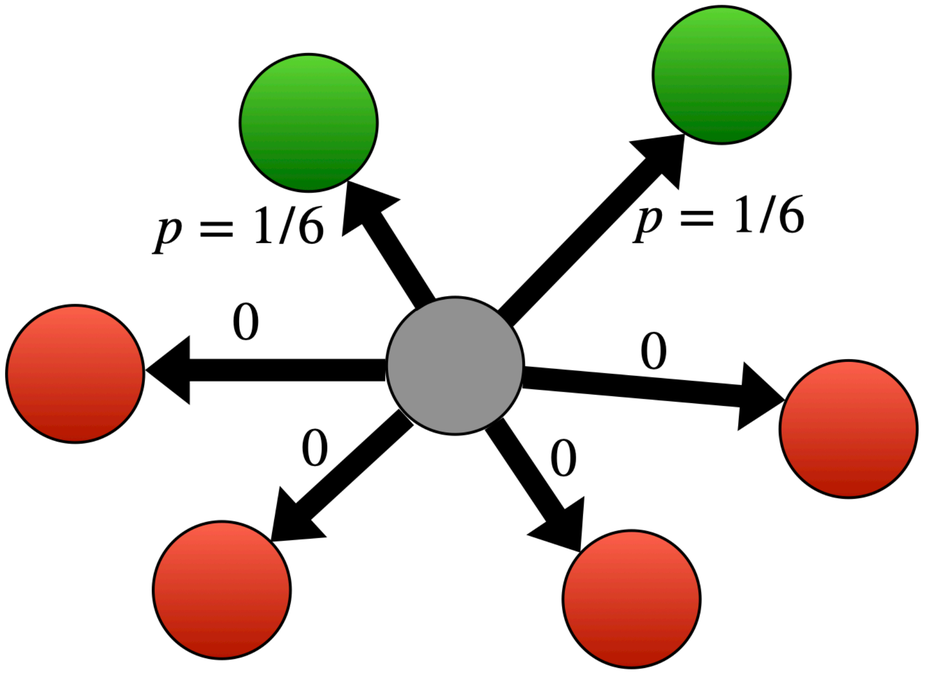
- ⇛ Over 200 types of ranking metrics to determine the final order.

  S. Levy (2010), How Google's algorithm rules the web, Wired Magazine, 17, www.wired.com.

- ⇛ Unclear to what extend PR is used.
- ⇛ Using a full substochastic matrix to handle spam url links.

  M. Cutts (2009), PageRank sculpting, Matt Cutts: Gadgets, Google, and SEO blog. www.mattcutts.com/blog/pagerank-sculpting.

Google's PageRank computation is a pseudo-PageRank problem now.



Equivalent formulations for PageRank (1)

Let $\mathbf{M}$ be a col. stochastic matrix with nonnegative entries, and the sum of entries in each column is 1. Let $\mathbf{v}$ be a col. stochastic vector ($\mathbf{e}^T\mathbf{v} = 1$), and let $0 < p < 1$ be the teleportation parameter. Then the PR problem is to find the solution to the linear system

$$(\mathbf{I} - p\mathbf{M})\,\mathbf{x} = (1 - p)\,\mathbf{v},$$

where the solution $\mathbf{x}$ is called the PageRank vector.

Equivalent formulations for PageRank (2)

The eigenvector and linear system formulations are equivalent if we seek an eigenvector $\mathbf{x}$ with $\mathbf{x} \geq 0$ and $\mathbf{e}^\mathsf{T}\mathbf{x} = 1$, in which case

$$\mathbf{x} = p\mathbf{M}\mathbf{x} + (1-p)\mathbf{v}\mathbf{e}^\mathsf{T}\mathbf{x} = p\mathbf{M}\mathbf{x} + (1-p)\mathbf{v}$$

$$\Longleftrightarrow$$

$$(\mathbf{I} - p\mathbf{M})\mathbf{x} = (1-p)\mathbf{v}.$$

- The matrix $\mathbf{I} - p\mathbf{M}$ is a diagonally dominant M-matrix.
  - ➠ Existence and uniqueness of $\mathbf{x}$.
  - ➠ $\mathbf{x}$ is nonnegative.
- Error after an iteration $\mathbf{x} - \mathbf{x}^{k+1} = \underbrace{p\mathbf{M}\mathbf{x} + (1-p)\mathbf{v}}_{\text{the true solution } \mathbf{x}} - \underbrace{p\mathbf{M}\mathbf{x}^k + (1-p)\mathbf{v}}_{\text{the updated iterate } \mathbf{x}^{k+1}} = p\mathbf{M}(\mathbf{x} - \mathbf{x}^k)$
  - ➠ Good convergence properties when p is not too close to 1.

Equivalent formulations for PageRank (2)

The eigenvector and linear system formulations are equivalent if we seek an eigenvector $\mathbf{x}$ with $\mathbf{x} \geq 0$ and $\mathbf{e}^\mathsf{T}\mathbf{x} = 1$, in which case

$$\mathbf{x} = p\mathbf{M}\mathbf{x} + (1-p)\mathbf{v}\mathbf{e}^\mathsf{T}\mathbf{x} = p\mathbf{M}\mathbf{x} + (1-p)\mathbf{v}$$

$$\Longleftrightarrow$$

$$(\mathbf{I} - p\mathbf{M})\mathbf{x} = (1-p)\mathbf{v}.$$

- The matrix $\mathbf{I} - p\mathbf{M}$ is a diagonally dominant M-matrix.
  - Existence and uniqueness of $\mathbf{x}$.
  - $\mathbf{x}$ is nonnegative.
- Error after an iteration $\mathbf{x} - \mathbf{x}^{k+1} = \underbrace{p\mathbf{M}\mathbf{x} + (1-p)\mathbf{v}}_{\text{the true solution } \mathbf{x}} - \underbrace{p\mathbf{M}\mathbf{x}^k + (1-p)\mathbf{v}}_{\text{the updated iterate } \mathbf{x}^{k+1}} = p\mathbf{M}(\mathbf{x} - \mathbf{x}^k)$
  - Small values of p means that we teleport too often!

Shift and inverse technique

- The power iteration has cost $O(n^2)$ (matrix-vector product) per iteration.
- Convergence can be very slow if for eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ of the matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ we have $|\frac{\lambda_2}{\lambda_1}| \approx 1$.

✍ Linear convergence with $|\frac{\lambda_2}{\lambda_1}|$ the asymptotic error constant (see proof in book).

- Shift and invert technique improves the convergence rate, but solves a linear system at each iteration with $O(n^3)$ cost.

✍ Attractive for smaller problems, or problems that have a special structure that enable fast direct methods.

Shift and inverse technique - main idea

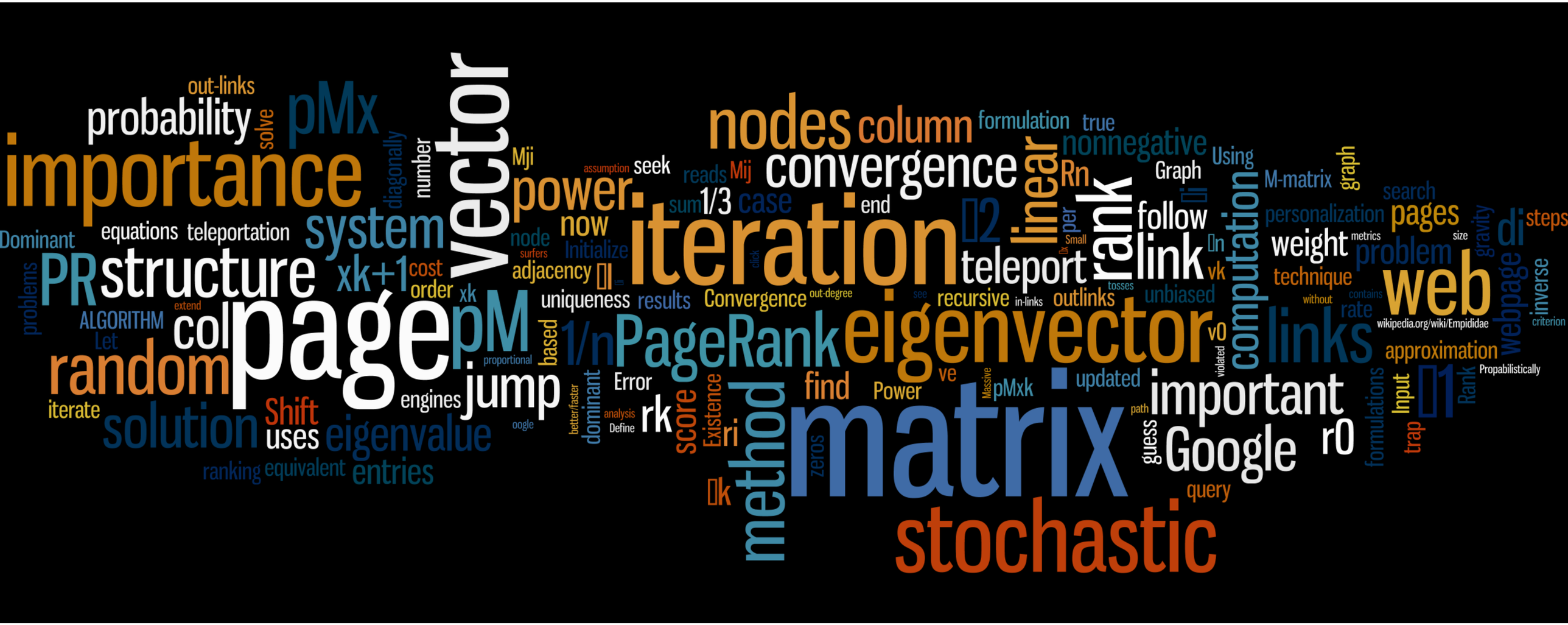
- λ_i: eigvals of $\mathbf{A}$, then $\lambda_i - \alpha$: of $\mathbf{A} - \alpha\mathbf{I}$, and $\mu_i = \frac{1}{\lambda_i - \alpha}$: of $\mathbf{B} = (\mathbf{A} - \alpha\mathbf{I})^{-1}$.
- ➠ Imagine applying the power method on $\mathbf{B}$
- Convergence rate now is $|\frac{\mu_2}{\mu_1}| = |\frac{\lambda_1 - \alpha}{\lambda_2 - \alpha}|$. If $\alpha \approx \lambda_1$ ➜ rapid convergence.
- ➠ Popular choices for $\alpha = |\mathbf{A}|_1$, or $\alpha = \lambda^{k-1}$.
- If α fixed, inversion of $\mathbf{B}$ takes place outside the for loop.

ALGORITHM: Inverse iteration

1: **Input**: Graph $\mathbf{A}$, initial guess $\mathbf{v}_0$, shift α
2: **Initialize**: $\mathbf{r}_0 = [1/n, \ldots, 1/n]^\intercal$
3: **for** $k = 1, 2, \ldots,$ until termination criteria **do**
4: Solve $(\mathbf{A} - \alpha\mathbf{I})\tilde{\mathbf{v}} = \mathbf{v}_{k-1}$
5: $\mathbf{v}_k = \tilde{\mathbf{v}}/\|\mathbf{v}\|$
6: $\lambda^k = \mathbf{v}_k^\intercal \mathbf{A} \mathbf{v}_k$
7: **end for**

- If $\mathbf{v}_0$ normalized, $\lambda^0 = \mathbf{v}_0^\intercal \mathbf{A} \mathbf{v}_0$, and $\alpha = \lambda^{k-1}$, the Rayleight quotient is approximated.
- $\lambda(\mathbf{v}) = \frac{\mathbf{v}^\intercal \mathbf{A} \mathbf{v}}{\mathbf{v}^\intercal \mathbf{v}}$
- Cubic convergence justifies increased cost/iteration.

Complementary reading



Complementary reading

1. Amy N. Langville and Carl D. Meyer, Deeper Inside PageRank, Internet Math. 1 (3) 335 - 380, 2003/2004.
   www.stat.uchicago.edu/~lekheng/meetings/mathofranking/ref/langville.pdf
2. Cornell University blogs, The Academic Paper That Started Google, 2019.
   https://blogs.cornell.edu/info2040/2019/10/28/the-academic-paper-that-started-google
3. David F. Gleich, PageRank Beyond the Web, SIAM Review, Vol. 57, Iss. 3, 2015.
   https://doi.org/10.1137/140976649
4. Uri M. Ascher, Chen Greif, A First Course in Numerical Methods, Chapter 8: Eigenvalues and Singular Values. Book available at icorsi