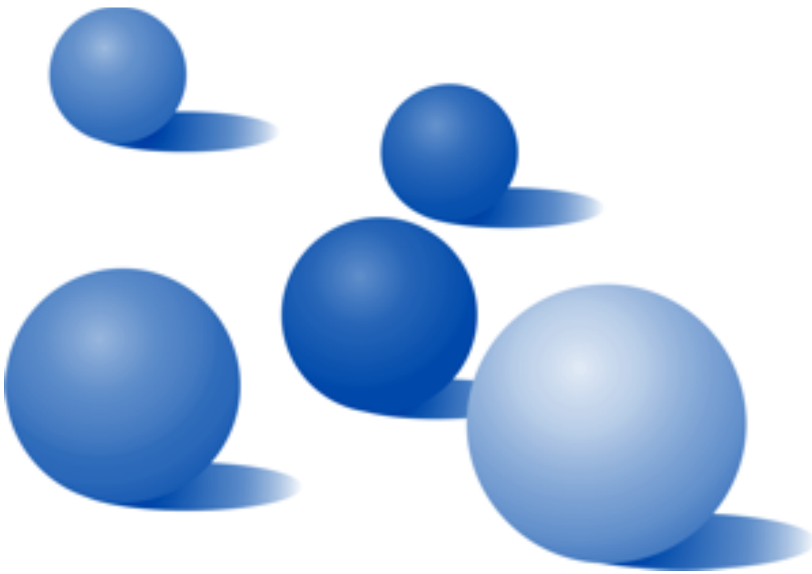
PageRank

INTRODUCTION TO DATA SCIENCE

teaching
datascience
.org

OUTLINE



- Introduction
- **The Basic Idea**
- The Initial PageRank Model
- The *Human Surfer* Model
- Advanced Aspects
- Alternative Model

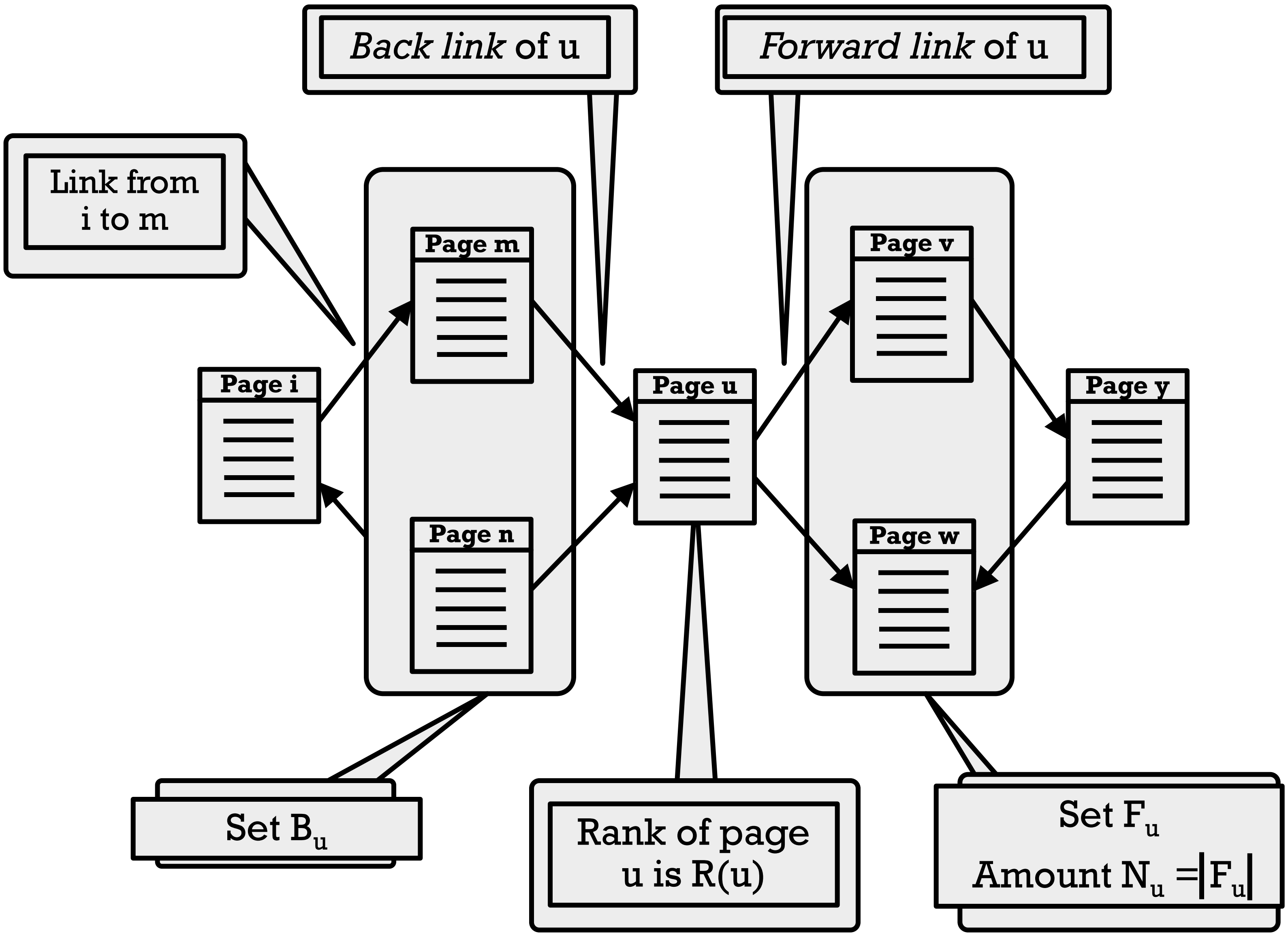
WHY PAGERANK?

- **The major challenge of web search engines is to rank the retrieved pages**
- **Most users don't go beyond the 1-2 first pages of search results.**
- **First generation search engine (AltaVista) ranked results based on keywords and relevance measures**
- **Easy to manipulate**
- **Google introduced "link analysis" as a tool for evaluating page "quality"**
- **Hyperlink-Induced Topic Search (HITS) - hubs and authorities**

PageRank is an example of unsupervised learning – it evaluates page quality without a training set.

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	PageRank
Continuous	regression	dimensionality reduction

THE WEB AS DIRECTED GRAPH



BACK LINKS AS INITIAL IDEA

- **Citation analysis as basis**
- **Idea: Pages with a lot of *back links* are more important**
- **Intuitive approach**

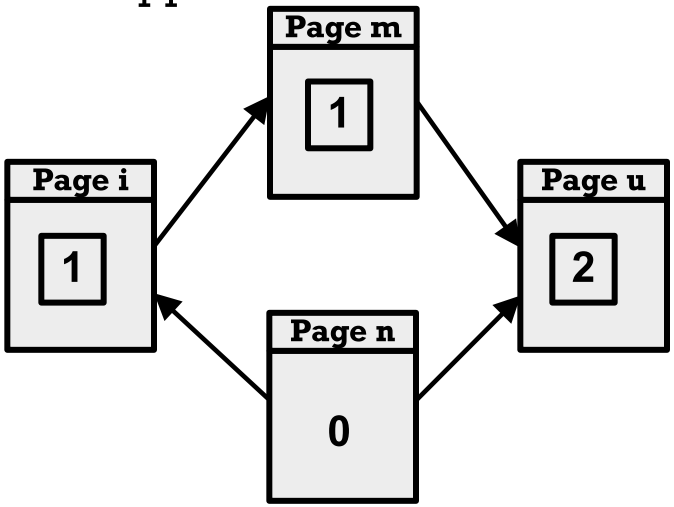
$$R(u) = \sum_{v \in B_u} 1$$

- **Extension: Each page has a "vote" of 1**

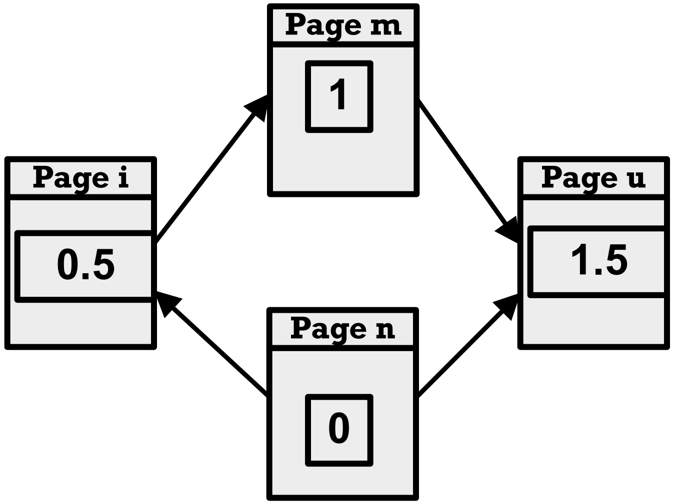
$$R(u) = c \sum_{v \in B_u} \frac{1}{N_v}$$

- **c normalizing factor (here c=1)**

Intuitive Approach



Extended Approach



FROM ANALYZING *BACK LINKS* TO PAGERANK

Back links

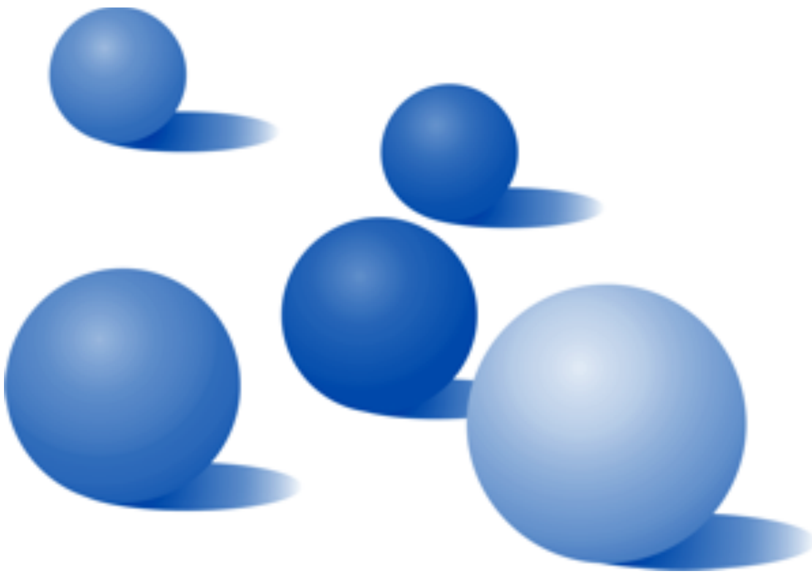
- Easy to calculate
- Suitable for well-controlled documents such as scientific articles
- For web pages: manipulation is easy
- Not in line with the common sense notion of "relevance"

PageRank

- Extension of the simple analysis of *back link*s
- Idea: Include the relevance of the referring (*back-link*) pages in the calculations of the ranks
- Manipulations are more difficult

OUTLINE

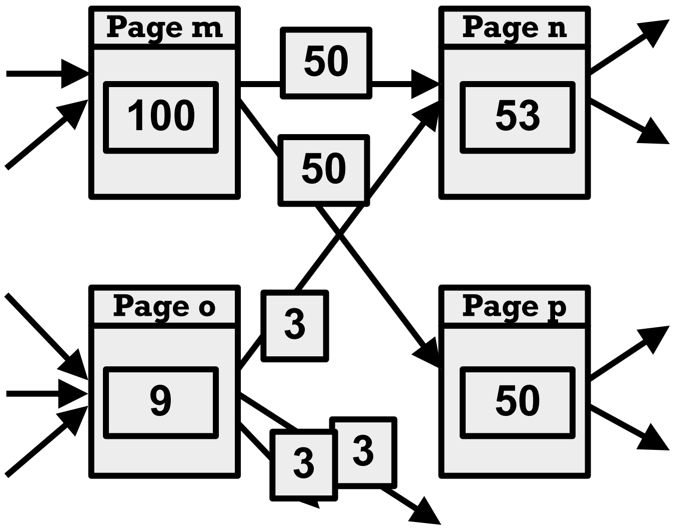
- Introduction
- The Basic Idea
- **The Initial PageRank Model**
- The *Human Surfer* Model
- Advanced Aspects
- Alternative Model

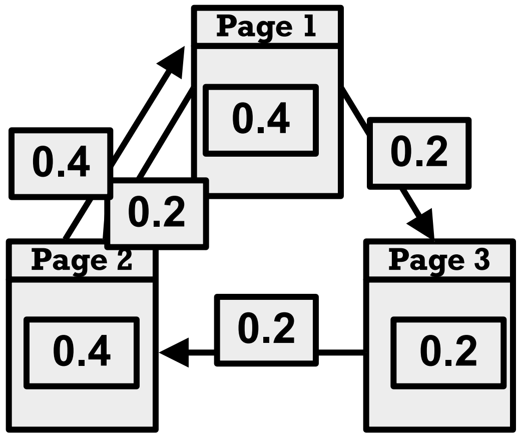


INTUITIVE DEFINITION OF PAGERANKS

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- **Rank spread evenly among the forward links**
- **Recursive calculation of R(u) until there is convergence**
- **Factor c**
  - **For normalization**
  - **usually c > 1, as there are pages without links**





MATHEMATICALLY, THIS IS AN EIGENVECTOR PROBLEM

- **Web as matrix A**
  - **if there is an edge between u and v (i.e., a link from u to v)**

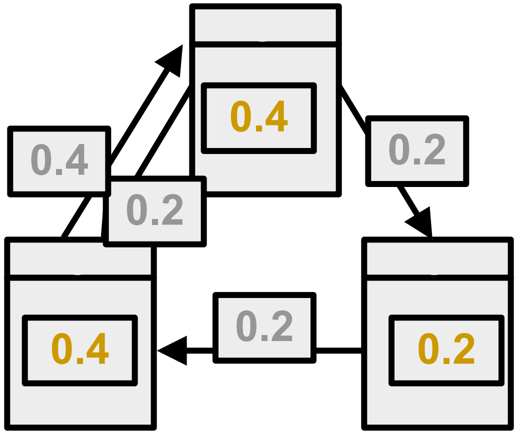
$$A_{u,v} = \frac{1}{N_u}$$

  - **else**

$$A_{u,v} = 0$$

$$A = \begin{pmatrix} 0{,}0 & 0{,}5 & 0{,}5 \\ 0{,}0 & 0{,}0 & 1{,}0 \\ 1{,}0 & 0{,}0 & 0{,}0 \end{pmatrix}$$

$$R = \begin{pmatrix} 0{,}4 & 0{,}2 & 0{,}4 \end{pmatrix}$$



- **R vector of page ranks**
- **This is the left eigenvector of A to the eigenvalue c**
- **R = RAc**

EIGENVECTORS AND EIGENVALUES

Definitions

Consider the square matrix *A*.

We say that c is an **eigenvalue** of *A* if there exists a non-zero vector *x* such that Ax = cx.

In this case, *x* is called a (right) **eigenvector** (corresponding to c), and the pair (c,*x*) is called an **eigenpair** for *A*.

Right eigenvectors satisfy the equation Ax = xc

c_1 is called the **dominant** eigenvalue if

$$|c_1| \geq |c_2| \geq |c_3| \geq .. \geq |c_n|$$

Example

The matrix; $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$

has two eigenvectors:

$v_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

and

$v_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

with eigenvalues 1 and 3 respectively.

SOLVING THE EIGENVALUE PROBLEM

Algebraic Approaches

- **Various Methods**
- **Example: calculate the determinant**

$\det(A-cI)=0$

I = Identity Matrix

$$A=\begin{pmatrix} 0,0 & 0,5 & 0,5 \\ 0,0 & 0,0 & 1,0 \\ 1,0 & 0,0 & 0,0 \end{pmatrix}$$

$\det(A-cI)$

$=-c^3+0,5+0+0,5c-0-0=0$

$c=1$

Power Iterations

- **Principle**

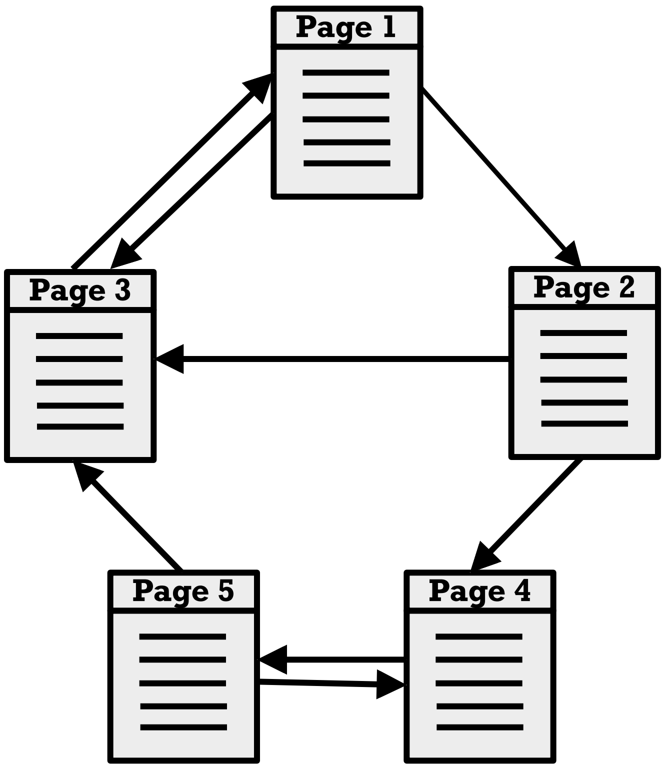
```
x = any vector with ||x||=1
eps = any value < 1
while(psi>eps)
  xTemp = x
  x = x * A             // multiply A
  x = x / ||x||₂        // normalise
  c = xᵀ * A * x        // eigenvalue
  psi = ||x – xTemp||₂
wend
```

where $||\bullet||_2$=Euclid norm

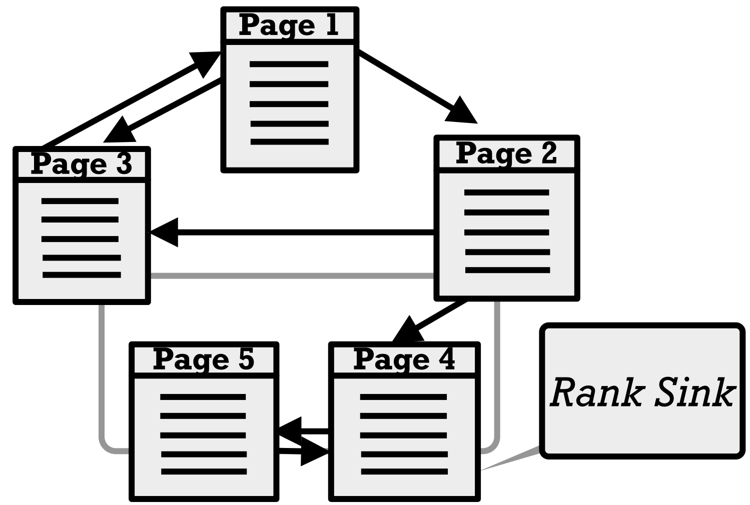
- In case of a stochastic matrix A

PROBLEMS WITH "IMPERFECT" GRAPHS

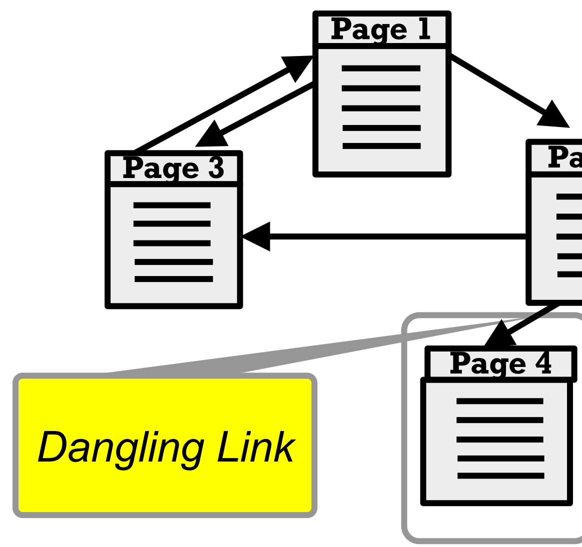
Perfect Graph



Graph with *Rank Sink*



Graph with *dangling link (Rank Leak)*



RANK SOURCE SOLVES *RANK SINKS*

Introduction to Rank Source

- **E(u): vector of web pages**

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u)$$

- **where c -> max**

$$\|R'\|_1 = \sum_i |x_i| = 1$$

- **As eigenvalue problem:**

$R' = c(A + E \otimes 1)R'$

where $1 = (1,1,\ldots,1)$

- Simplified Version:
  - Same *Rank Source* for all pages
  - Normalisation to 1
  - New formula:

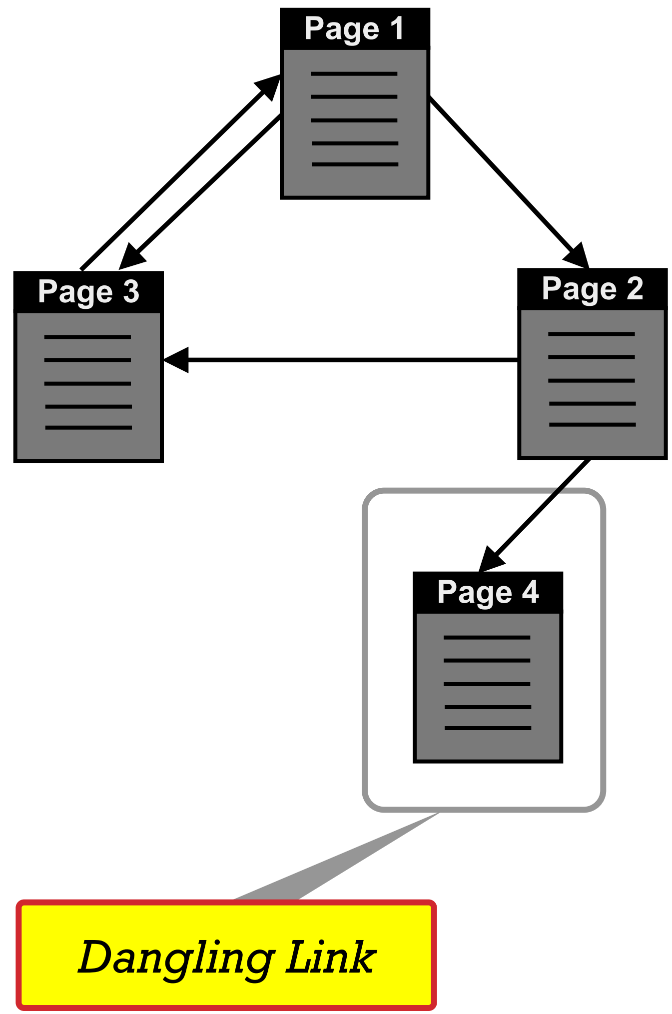
$$R''(u) = d \sum_{v \in B_u} \frac{R''(v)}{N_v} + \frac{(1-d)}{\#\ \text{Pages}}$$

DANGLING LINKS

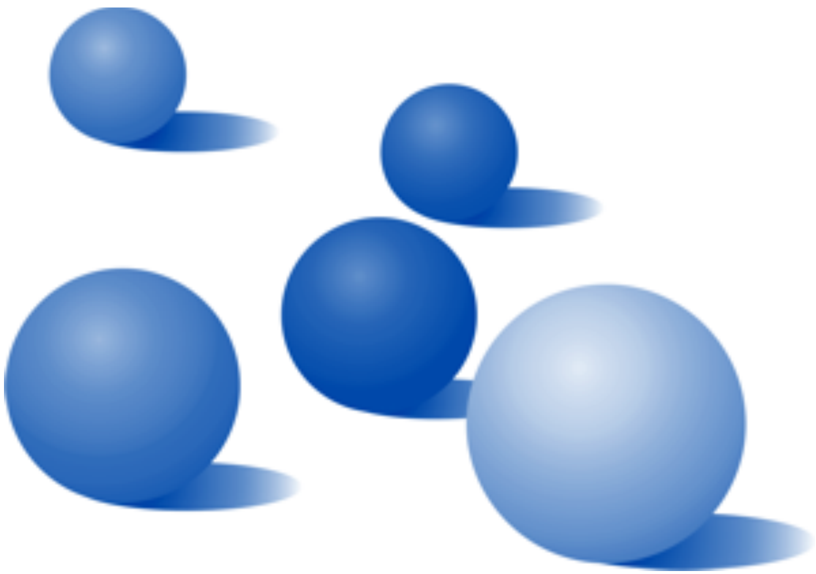
- Reduce “distributable” PageRank
- Rather frequent
  - Pages without links
  - Pages not yet indexed by Google
  - PDFs etc.

- Removed prior to calculation
- Added with the immediate page rank after the final iteration

- Result hardly affected



OUTLINE

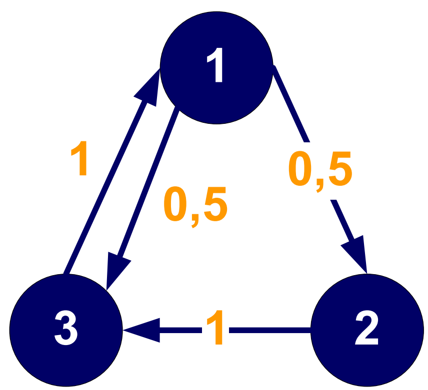


- Introduction
- The Basic Idea
- The Initial PageRank Model
- **The *Human Surfer* Model**
- Advanced Aspects
- Alternative Model

MARKOV CHAIN

- Homogeneous discrete stochastic process with transition matrix P
  - Transitions depend only on the current state (Markov property)
  - Transitions from node i to node k happen at discrete points of time t=1,2,…
  - Transition from node i to node k happens with probability P_{ik}
  - The transition probability is independent of the time t (homogeneous)
  - The initial node is selected arbitrarily based on a distribution q^0 over V
  - q^t : row vector, whose k-th entry gives the likelihood of being in state k after transition t
- It holds:

$$q^{t+1}=q^tP \Leftrightarrow q^{t+1}=qP^t$$

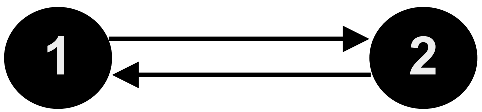


$$P=\begin{pmatrix} 0.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 \end{pmatrix}$$

$$q^0=\begin{pmatrix} 1.0 & 0.0 & 0.0 \end{pmatrix}$$

$$q^1=q^0P=\begin{pmatrix} 0.0 & 0.5 & 0.5 \end{pmatrix}$$

LIMIT BEHAVIOUR



- Intuition:

  Both states equally likely

- $q^0 = (1,0)$ leads to

$q^{2n} = (0,1)$

$q^{2n+1} = (1,0)$

Limit Distribution

q^∞

$= \lim_{n\to\inf} q^0P^n$

$= \lim_{n\to\inf} q^nP$

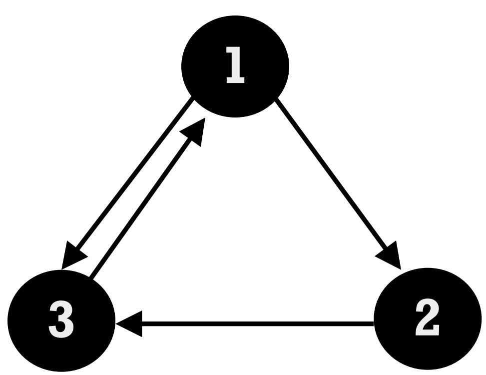
- does not always exist
- can depend on the initial distribution
- is not necessarily unique

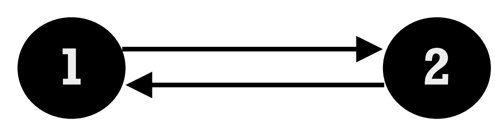
PROPERTIES OF MARKOV CHAINS

- **Irreducibility:**
  - Any node of a Markov Chain can be reached from any node (in a finite number of steps).

- **Aperiodicity:**
  - The greatest common divisor of the length of all „round-trips“ is 1.

Irreducible **y** Aperiodic **y**





Irreducible **y** Aperiodic **NO**

STEADY STATE OR STATIONARY DISTRIBUTION

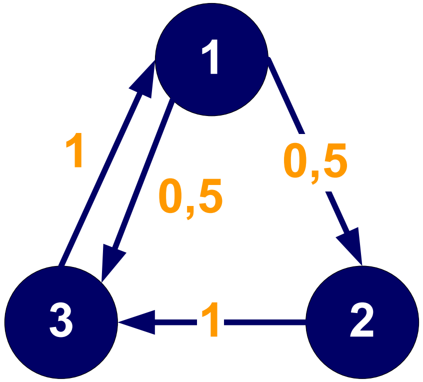
Wanted

- Stationary distribution such that: $q^{\infty} = q^{\infty} P$
- i.e., the eigenvector to the eigenvalue 1

Theorem

- Assume that P is
  - irreducible
  - aperiodic
  - finite
- Then there is a unique stationary distribution q^{∞}
- Let N(i,t) be the number of visits that a random surfer pays to page i until the point in time t. Then

$$\lim_{t\to\infty} \frac{N(i,t)}{t} = q_i^{\infty}$$



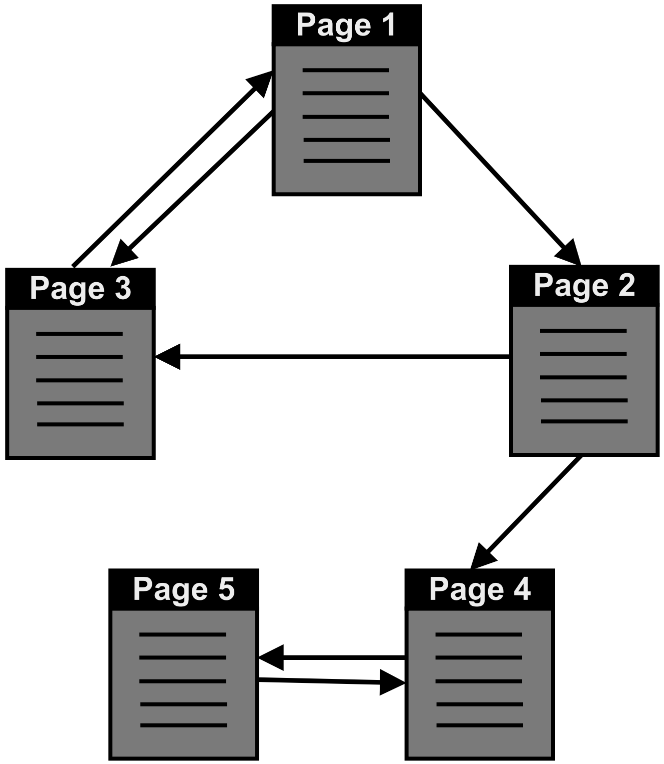
$$P = \begin{pmatrix} 0.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 \end{pmatrix}$$

$$q^{\infty} = \begin{pmatrix} 0.4 & 0.2 & 0.4 \end{pmatrix}$$

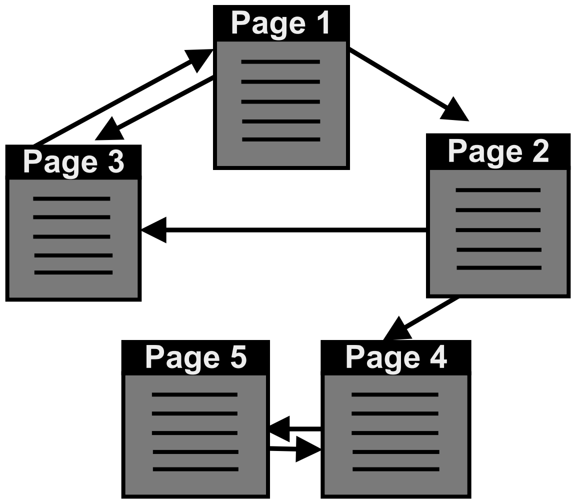
HUMAN SURFER AS A MARKOV CHAIN

- The web surfer starts at a randomly selected page
- At each period the surfer chooses between the following alternatives:
  - Follow a ramdomly selected link on the current page (probability d)
  - Jump to another page of the web without following a link (probability (1-d))

$$A' = dA + (1-d)\frac{1}{Pages}1\times1$$



TRANSITION MATRIX



$$A=\begin{pmatrix} 0 & 0{,}5 & 0{,}5 & 0 & 0 \\ 0 & 0 & 0{,}5 & 0{,}5 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$d=0{,}8$

$$A'=\underbrace{\begin{pmatrix} 0 & 0.4 & 0.4 & 0 & 0 \\ 0 & 0 & 0.4 & 0.4 & 0 \\ 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 0.8 & 0 \end{pmatrix}}_{dA}+\underbrace{\begin{pmatrix} 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \end{pmatrix}}_{(1-d)\frac{1}{\#\ \text{Pages}}1\times 1}=\begin{pmatrix} 0.04 & 0.44 & 0.44 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.44 & 0.44 & 0.04 \\ 0.84 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.84 \\ 0.04 & 0.04 & 0.04 & 0.84 & 0.04 \end{pmatrix}$$

STEADY STATE AND THE *HUMAN SURFER*

- *Steady State* is a distribution vector satisfying

$$R = RA'$$

- Can be regarded as a special form of

$$R' = cR'(A + E \otimes 1)$$

  - Normalised to 1
  - *Rank of Source* same for all pages

- *Dangling Links*
  - Can either be removed
  - Or be treated as a page linking to all other pages

THE ROLE OF D

- d=0.85
- E equally distributed
- *Dangling Links* added for final iteration

- d=0.85 reportedly used by Google (at least initially)
- Probably what Google does
- Additional adaptations are applied, algorithm is optimized

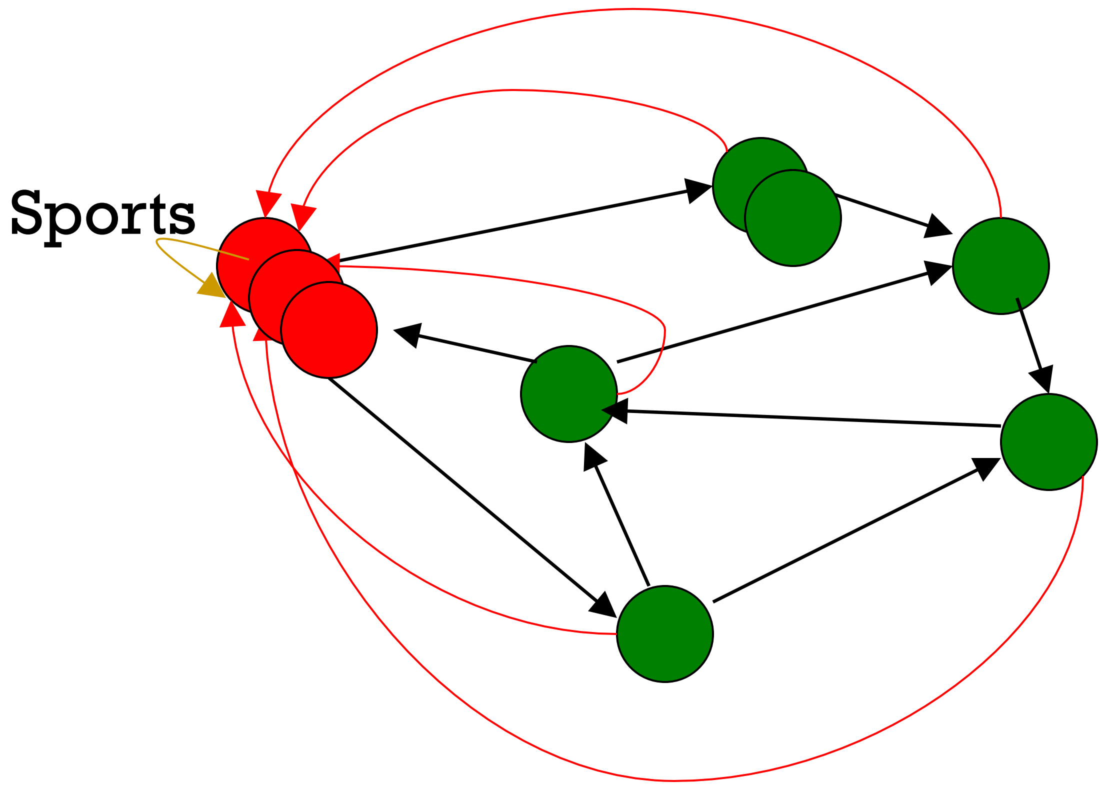
- d=0
- E equally distributed
- *Dangling Links* added for final iteration

- Extreme case: All pages are equally likely
- Assumes that all pages are equally important
- Comparable to the simple search engines

- d<=1
- E only for one page, e.g. private home page
- *Dangling Links* added for final iteration

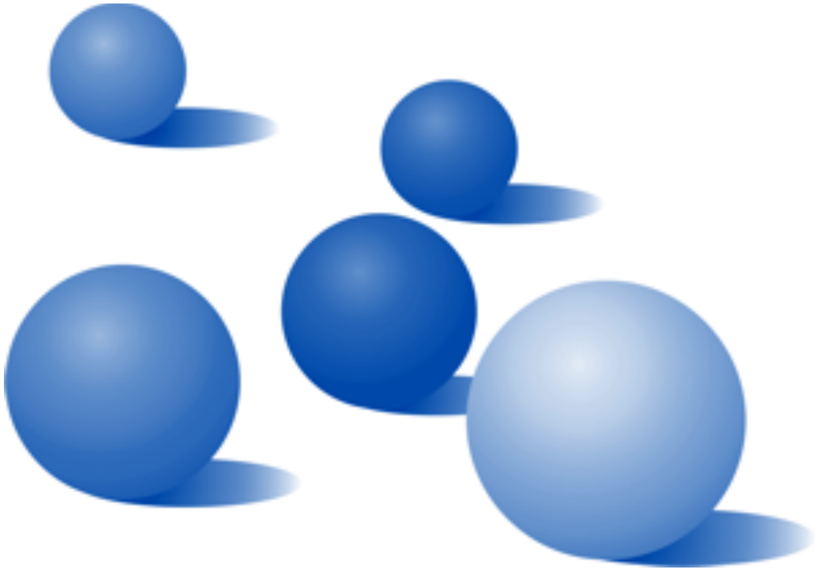
- Mirrors user preferences
- Assumes that the page is representative
- Alternatively one could derive E from historic user behaviour (e.g., using web logs)

NON-UNIFORM TELEPORTATION



Teleport with 10% probability to a Sports page

OUTLINE

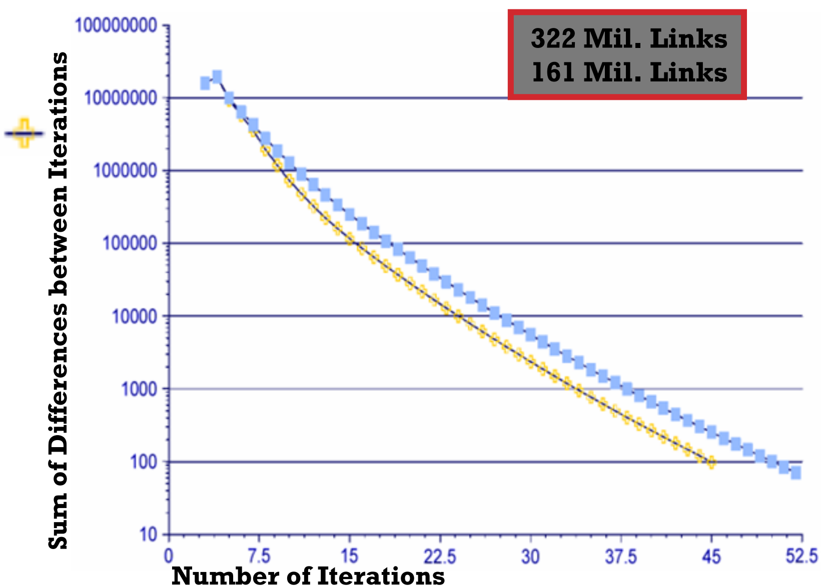


- Introduction
- The Basic Idea
- The Initial PageRank Model
- The *Human Surfer* Model
- **Advanced Aspects**
- Alternative Model

CONVERGENCE & RUNTIME OF POWER ITERATION

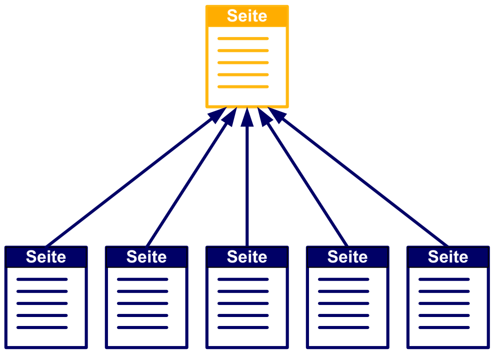
- Convergence ensured by adapting the transition matrix
- The number of required iterations
  - Depends on the distance to second eigenvalue and thus the value d
  - Is less affected by the number of links
- Google calculates PageRank regularly, updates are released appr. every day

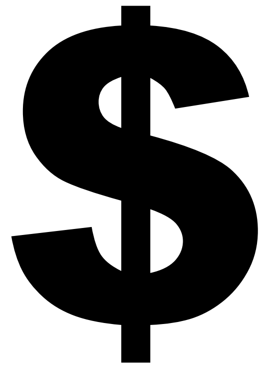
Convergence (d=0.85)

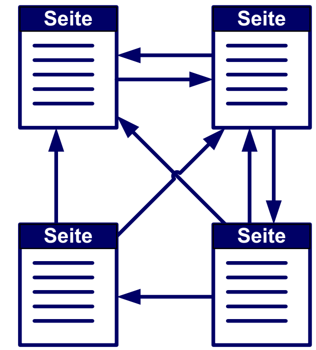


Source: Brin+:The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford University, 1999

MANIPULATING PAGERANK







Artificial creation of back links

- Across domains
- Linked

- Theoretically possible
- *Anti-Spamming* mechanisms exist
  - – PageRank 0
  - – BadRank

Purchasing links

- E.g., Banner on a page with high Page Rank

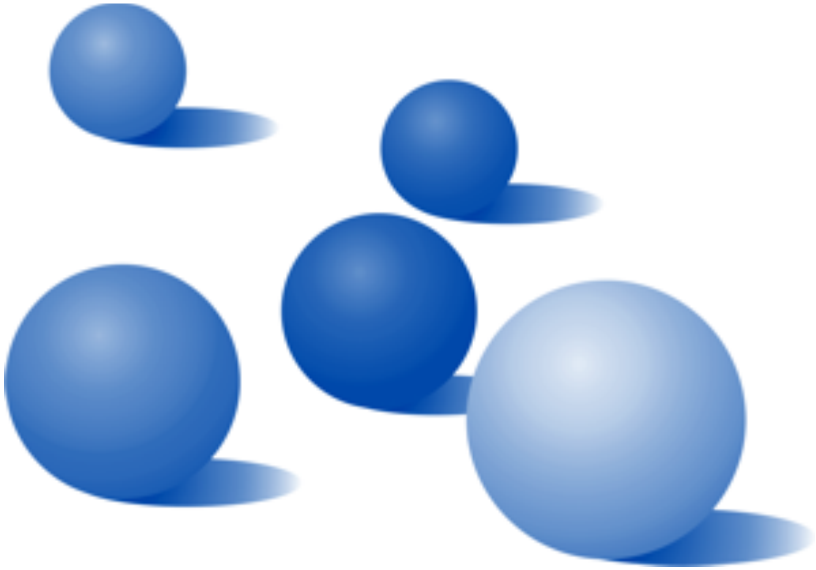
- Possible
- Costs money, so a bit controlled

Create Google-tailored pages

- Multiple linked pages
- Links to bad pages using JavaScript

- To a certain extend feasible
- Too much might lead to exclusion from page rank calculation

AGENDA

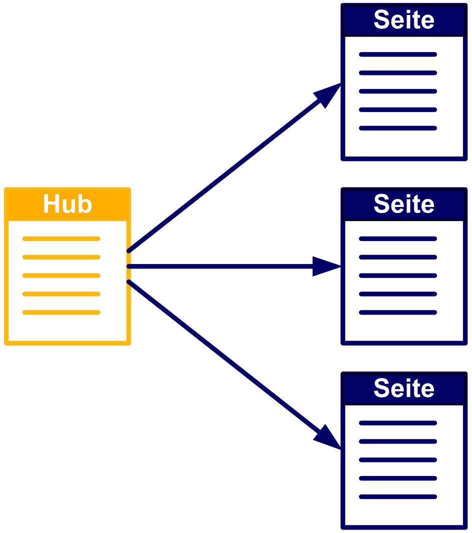


- Introduction
- The Basic Idea
- The Initial PageRank Model
- The *Human Surfer* Model
- Advanced Aspects
- **Alternative Model**

ALTERNATIVE *RANKING* METHODOLOGIES

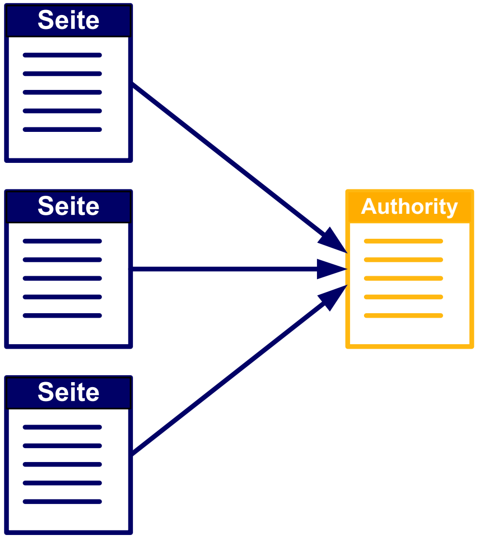
Hypertext Included Topic Selection

- Web as directed graph
- Algorithm operates on a part of the graph
- Algorithm runs subject-specific and distinguishes
  - "expert" pages (*Authorities*) for a topic
  - pages linking to *Authorities* (*Hubs*)
- HITS is based on balance of *Hubs* and *Authorities*



Salsa

- Extends HITS for probabilities
- undirected graph
- *Hub Walk* and *Authority Walk*



HIGH-LEVEL SCHEME

Extract from the web a base set of pages that *could* be good hubs or authorities.

From these, identify a small set of top hub and authority pages;

iterative algorithm.

BASE SET

Given text query (say *soccer*), use a text index to get all pages containing *soccer.*

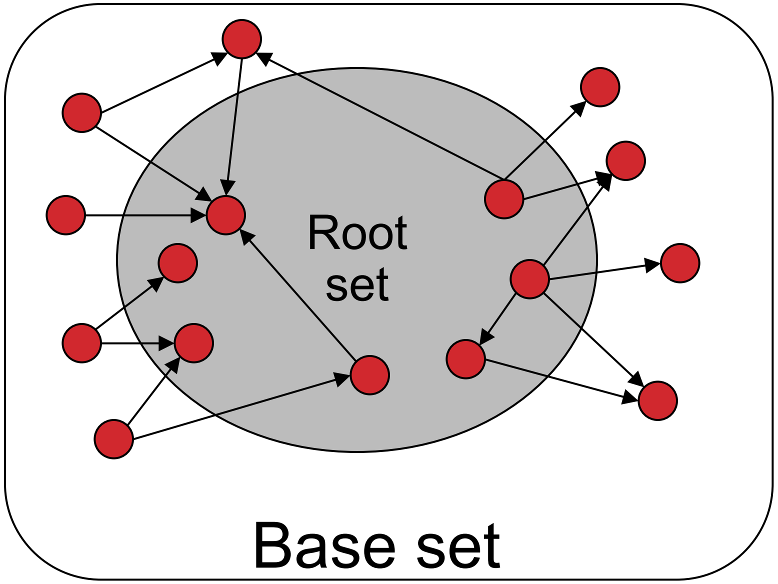
- Call this the <u>root set</u> of pages.

Add in any page that either

- points to a page in the root set, or
- is pointed to by a page in the root set.

Call this the <u>base set</u>.

VISUALIZATION



ASSEMBLING THE BASE SET

- **Root set typically 200-1000 nodes.**
- **Base set may have up to 5000 nodes.**
- **How do you find the base set nodes?**
  - Follow out-links by parsing root set pages.
  - Get in-links (and out-links) from a *connectivity server.*
  - (Actually, suffices to text-index strings of the form ***href=*** ***"URL"*** to get in-links to *URL*.)

DISTILLING HUBS AND AUTHORITIES

Compute, for each page *x* in the base set, a <u>hub score</u> *h(x)* and an <u>authority score</u> *a(x)*.

1. Initialize: for all *x*, *h(x)←1; a(x) ←1*;

2. Iteratively update all *h(x), a(x)*;

Key

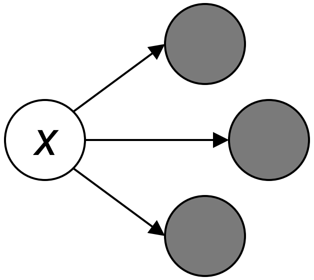
3. After iterations

1. output pages with highest *h()* scores as top hubs
2. highest *a()* scores as top authorities.

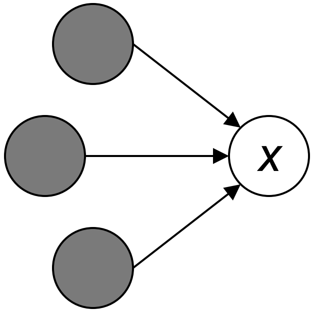
ITERATIVE UPDATE

Repeat the following updates, for all *x*:

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$



SCALING

To prevent the *h()* and *a()* values from getting too big, can scale down after each iteration.

Scaling factor doesn't really matter:

- we only care about the *relative* values of the scores.

HOW MANY ITERATIONS?

- **Claim: relative values of scores will converge after a few iterations:**
  - in fact, suitably scaled, *h()* and *a()* scores settle into a steady state!
- **We only require the <u>relative orders</u> of the *h()* and *a()* scores - not their absolute values.**
- **In practice, ~5 iterations get you close to stability.**

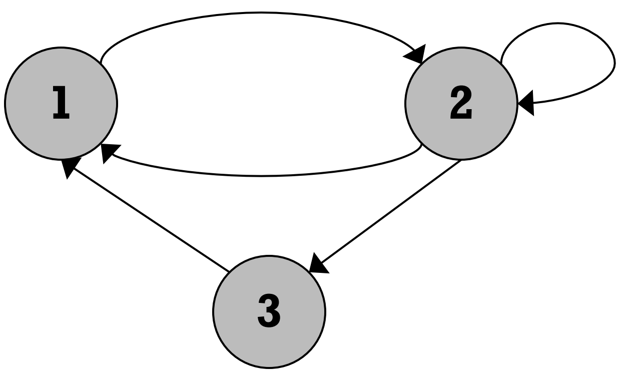
THINGS TO NOTE

- **Pulled together good pages regardless of language of page content.**
- **Use *only* link analysis <u>after</u> base set assembled**
  - iterative scoring is query-independent.
- **Iterative computation <u>after</u> text index retrieval - significant overhead.**

PROOF OF CONVERGENCE

***n*×*n* adjacency matrix A:**

- **each of the *n* pages in the base set has a row and column in the matrix.**
- **Entry $A_{ij} = 1$ if page *i* links to page *j*, else = 0.**



$$\begin{array}{c|ccc|} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 1 \\ 3 & 1 & 0 & 0 \\ \hline \end{array}$$

HUB/AUTHORITY VECTORS

View the hub scores *h()* and the authority scores *a()* as vectors with *n* components.

Recall the iterative updates

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$

$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$

REWRITE IN MATRIX FORM

- $\mathbf{h}=\mathbf{A}\mathbf{a}$.
- $\mathbf{a}=\mathbf{A}^t\mathbf{h}$.

- Substituting, $\mathbf{h}=\mathbf{A}\mathbf{A}^t\mathbf{h}$ and $\mathbf{a}=\mathbf{A}^t\mathbf{A}\mathbf{a}$.

- Thus, **h** is an eigenvector of $\mathbf{A}\mathbf{A}^t$ and **a** is an eigenvector of $\mathbf{A}^t\mathbf{A}$.

- Further, our algorithm is a particular, known algorithm for computing eigenvectors: the *power iteration* method.

Guaranteed to converge.

ISSUES

Topic Drift

- **Off-topic pages can cause off-topic "authorities" to be returned**
  - E.g., the neighborhood graph can be about a "super topic"
- **Mutually Reinforcing Affiliates**
  - Affiliated pages/sites can boost each others' scores
  - Linkage between affiliated pages is not a useful signal

LITERATURE

- Sergey Brin, Rajeev Motwani, Lawrence Page, Terry Winograd:
  The PageRank Citation Ranking: Bringing Order to the Web.
  Technical Report 1999-66, Stanford, Berkeley, 1999.
- Sergey Brin, Lawrence Page:
  The Anatomy of a Large-Scale Hypertextual Web Search Engine.
  Computer Networks and ISDN Systems 1998-30, Seite 107-117.
- Sergey Brin, Rajeev Motwani, Lawrence Page, Terry Winograd:
  What can you do with a Web in your Pocket?
  Data Engineering Bulletin 1998-21, Seite 37-47.
- Amy N. Langville, Carl D. Meyer:
  Deeper Inside PageRank.
  Technical Report 2003, North Carolina State University, Raleigh, 2003.
- Tara Calishain, Rael Dornfest:
  Google Hacks: 100 Industrial-Strength Tips & Tools.
  Volume 1, Beihing, 2003.
- Taher H. Haveliwala, Sepandar D. Kamvar:
  The Second Eigenvalue of the Google Matrix.
  Technical Report 2003, Stanford University, Berkeley, CA, 2003.
- Taher H. Haveliwala:
  Topic-sensitive PageRank.
  In: Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, May 2002.

SLIDES CAN BE FOUND AT:

TEACHINGDATASCIENCE.ORG