Link Analysis

Stony Brook University
CSE545, Fall 2016

The Web , circa 1998

The Web , circa 1998

The Web , circa 1998

Match keywords, language (*information retrieval*)

Explore directory

The Web , circa 1998

Easy to game with "term spam"

Match keywords, language (*information retrieval*)

Explore directory

Time-consuming; Not open-ended

Enter PageRank

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we presen
use of the structure pres
and produce much more
text and hyperlink datab

The PageRank Citation Ranking: Bringing Order to the Web

January 29, 1998

...

Abstract

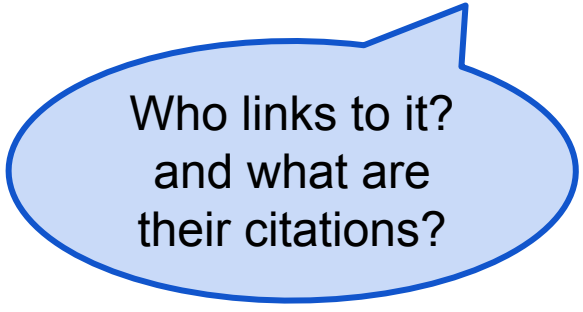
The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively

PageRank

Key Idea: Consider the citations of the website in addition to keywords.

PageRank

Key Idea: Consider the citations of the website in addition to keywords.

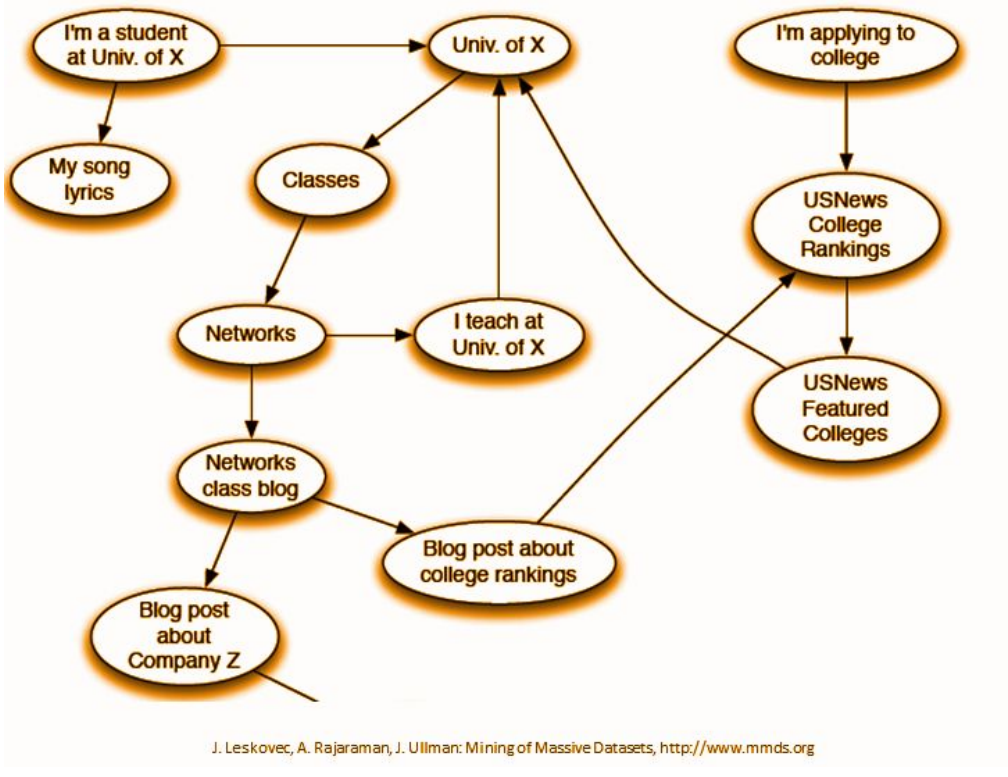


PageRank

Key Idea: Consider the citations of the website in addition to keywords.

Who links to it?
and what are
their citations?

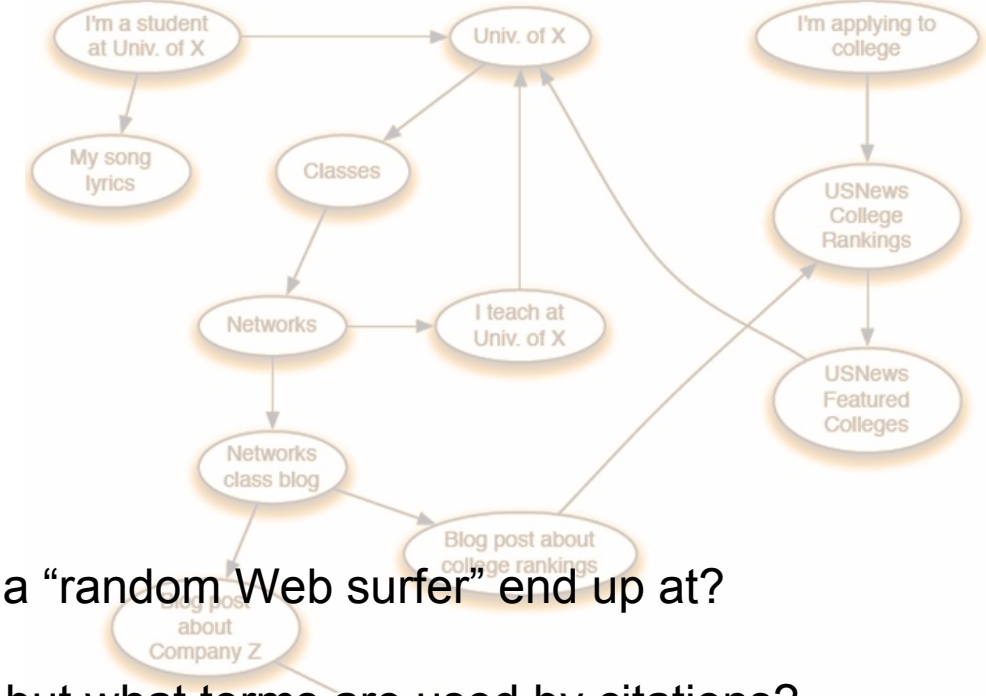
The Web as a directed graph:



PageRank

Key Idea: Consider the citations of the website in addition to keywords.

Who links to it?
and what are
their citations?



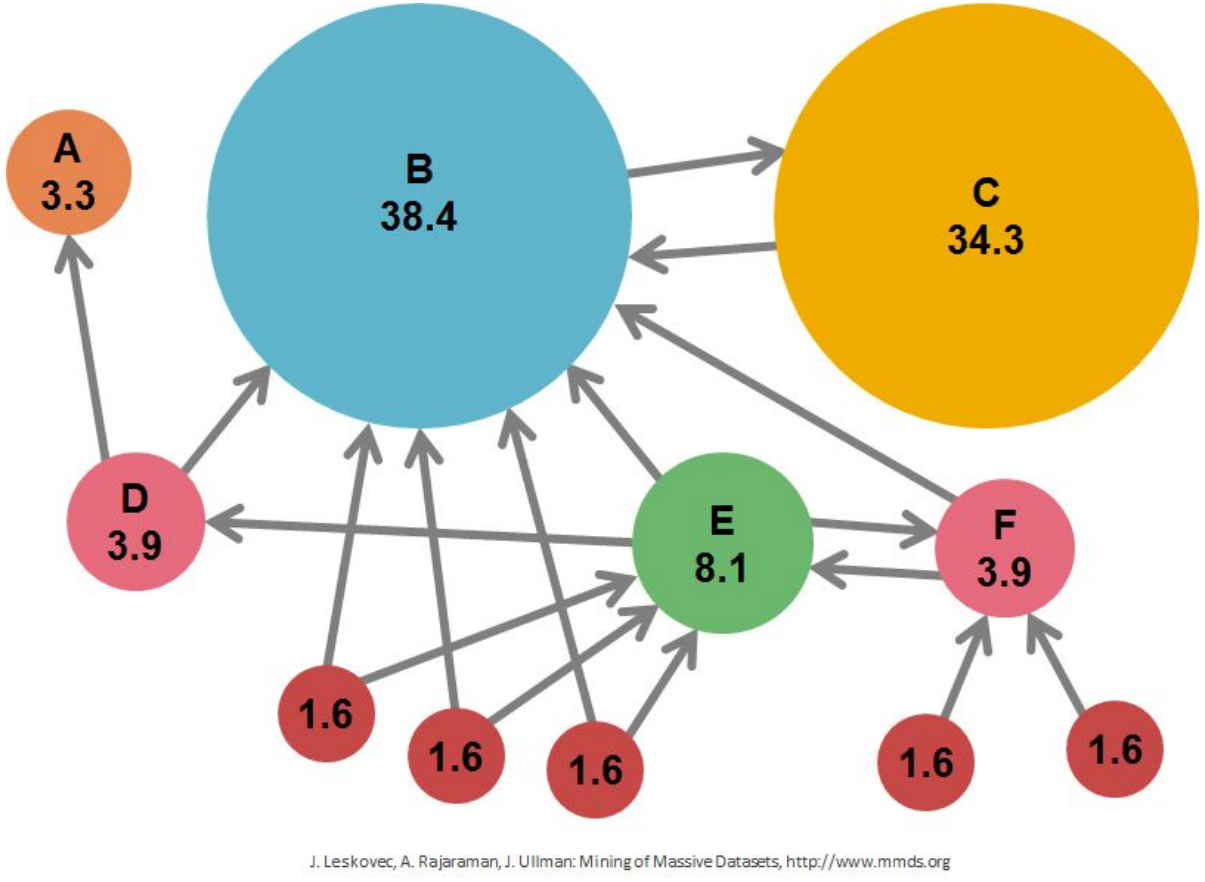
Innovation 1: What pages would a “random Web surfer” end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank

Key Idea: Consider the

Flow Model:
in-links as votes



Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank

Key Idea: Consider the citations of the website in addition to keywords.

Flow Model:
in-links (citations) as votes

But citations from important pages should count more.

Use recursion to figure out if each page is important.

Innovation 1: What pages would a “random Web surfer” end up at?

Innovation 2: Not just own terms but what terms are used by citations?

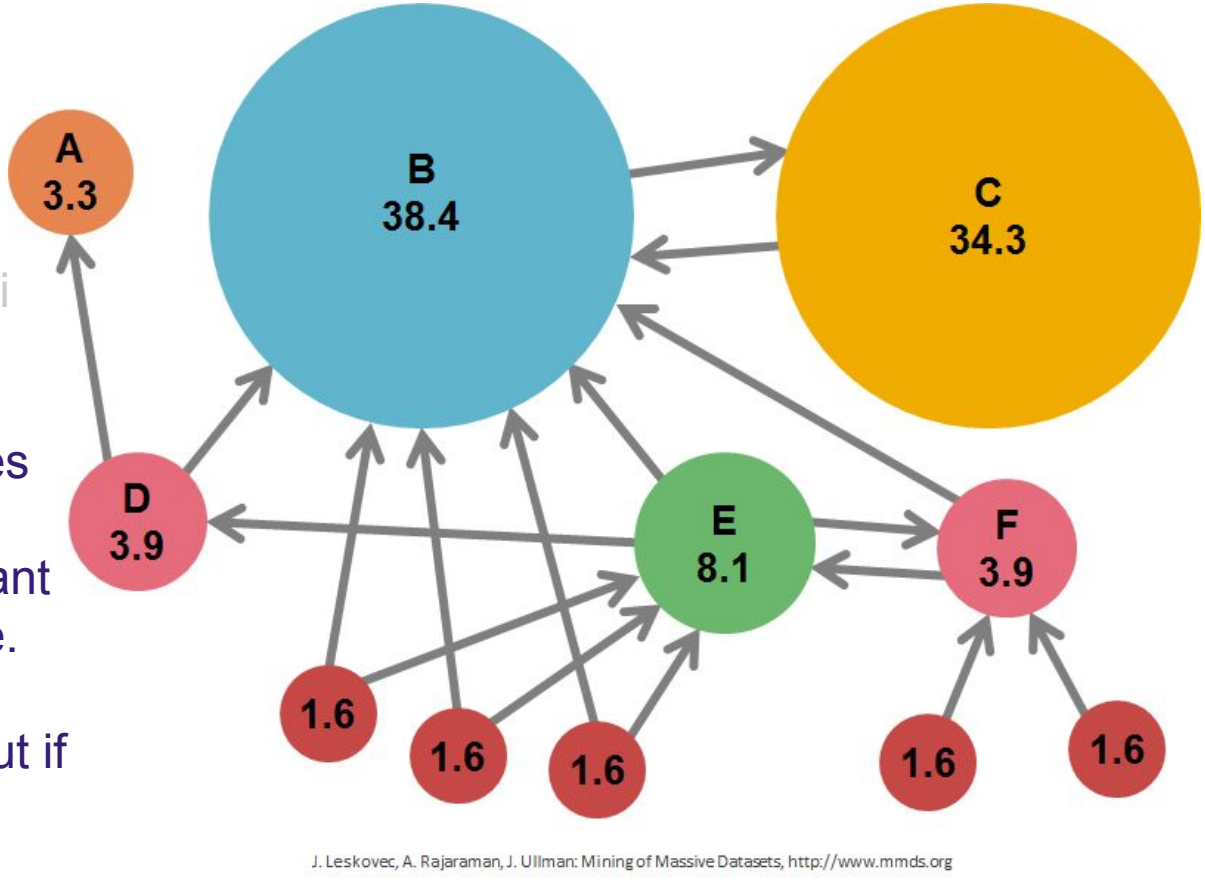
PageRank

Key Idea: Consider the citati

Flow Model:
in-links (citations) as votes

But citations from important pages should count more.

Use recursion to figure out if each page is important.



J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank

Key Idea: Consider the citations of the website in addition to keywords.

Flow Model:
in-links (citations) as votes

But citations from important pages should count more.

Use recursion to figure out if each page is important.

How to compute?

Each page (*j*) has an importance (i.e. rank, r_j)

$$vote_j = \frac{r_j}{n_j}$$

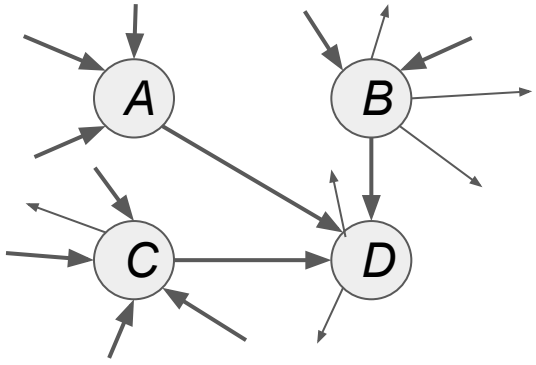
(n_j is |out-links|)

$$r_j = \sum_{i \in inLinks(j)} vote_i$$

Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank



How to compute?

Each page (*j*) has an importance (i.e. rank, r_j)

$$vote_j = \frac{r_j}{n_j}$$

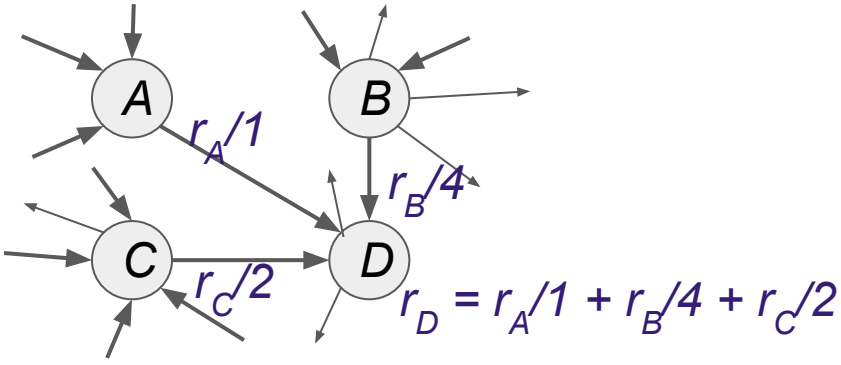
(n_j is |out-links|)

$$r_j = \sum_{i \in inLinks(j)} vote_i$$

Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank



How to compute?

Each page (*j*) has an importance (i.e. rank, r_j)

$$vote_j = \frac{r_j}{n_j}$$

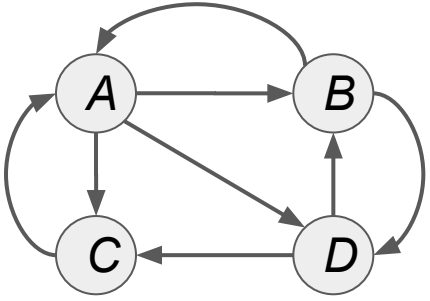
(n_j is |out-links|)

$$r_j = \sum_{i \in inLinks(j)} vote_i$$

Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank



How to compute?

Each page (*j*) has an importance (i.e. rank, r_j)

$$vote_j = \frac{r_j}{n_j}$$

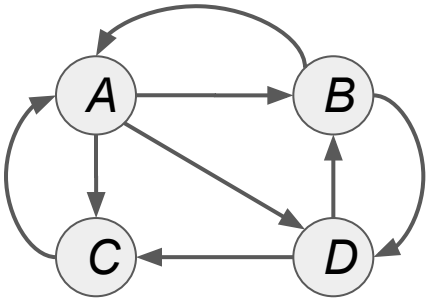
(n_j is |out-links|)

$$r_j = \sum_{i \in inLinks(j)} vote_i$$

Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank



$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

A system of equations?

How to compute?

Each page (*j*) has an importance (i.e. rank, r_j)

$$vote_j = \frac{r_j}{n_j}$$

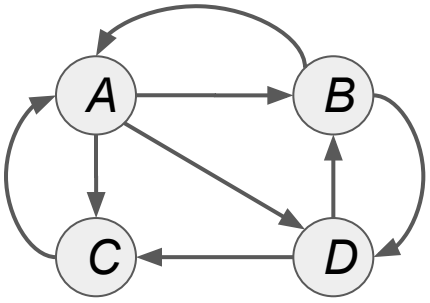
(n_j is |out-links|)

$$r_j = \sum_{i \in inLinks(j)} vote_i$$

Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank



$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

A system of equations?

$$1 = r_A + r_B + r_C + r_D$$

How to compute?

Each page (*j*) has an importance (i.e. rank, r_j)

$$vote_j = \frac{r_j}{n_j}$$

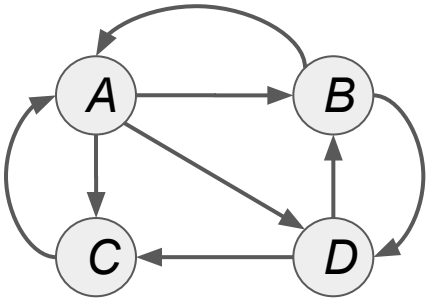
(n_j is |out-links|)

$$r_j = \sum_{i \in inLinks(j)} vote_i$$

Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank



$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

A system of equations?

Provides intuition, but impractical to solve at scale.

$$1 = r_A + r_B + r_C + r_D$$

How to compute?

Each page (*j*) has an importance (i.e. rank, r_j)

$$vote_j = \frac{r_j}{n_j}$$

(n_j is |out-links|)

$$r_j = \sum_{i \in inLinks(j)} vote_i$$

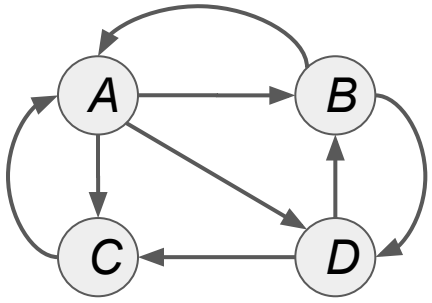
Innovation 1: What pages would a "random Web surfer" end up at?

Innovation 2: Not just own terms but what terms are used by citations?

PageRank

$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

$$1 = r_A + r_B + r_C + r_D$$



to \ from	***A***	***B***	***C***	***D***
A	0	1/2	1	0
B	1/3	0	0	1/2
C	1/3	0	0	1/2
D	1/3	1/2	0	0

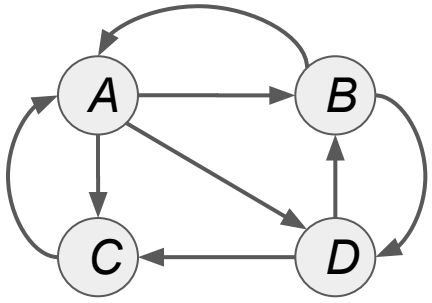
"Transition Matrix", *M*

Innovation 1: What pages would a "random Web surfer" end up at?

PageRank

$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

$$1 = r_A + r_B + r_C + r_D$$



to \ from	***A***	***B***	***C***	***D***
A	0	1/2	1	0
B	1/3	0	0	1/2
C	1/3	0	0	1/2
D	1/3	1/2	0	0

"Transition Matrix", *M*

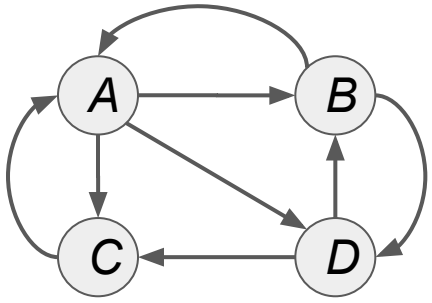
Innovation 1: What pages would a "random Web surfer" end up at?

To start: N=4 nodes, so *r* = [¼, ¼, ¼, ¼,]

PageRank

$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

$$1 = r_A + r_B + r_C + r_D$$



to \ from	***A***	***B***	***C***	***D***
A	0	1/2	1	0
B	1/3	0	0	1/2
C	1/3	0	0	1/2
D	1/3	1/2	0	0

"Transition Matrix", *M*

Innovation 1: What pages would a "random Web surfer" end up at?

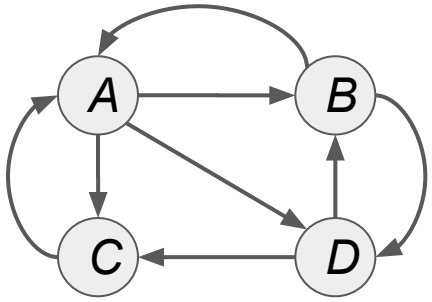
To start: N=4 nodes, so r = [¼, ¼, ¼, ¼,]

after first iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]

PageRank

$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

$$1 = r_A + r_B + r_C + r_D$$



to \ from	***A***	***B***	***C***	***D***
A	0	1/2	1	0
B	1/3	0	0	1/2
C	1/3	0	0	1/2
D	1/3	1/2	0	0

"Transition Matrix", *M*

Innovation 1: What pages would a "random Web surfer" end up at?

To start: N=4 nodes, so r = [¼, ¼, ¼, ¼,]
after first iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]
after second iteration: $M(M \cdot r) = M^2 \cdot r$ = [15/48, 11/48, …]

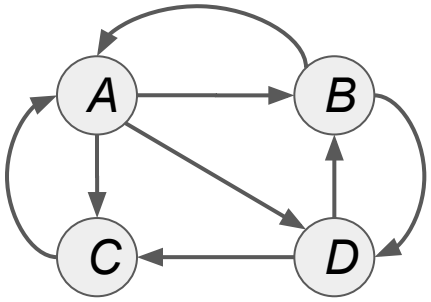
PageRank

Power iteration algorithm

```
Initialize:    r[0] = [1/N, …, 1/N],
               r[-1]=[0,...,0]
while (err_norm(r[t],r[t-1])>min_err):




err_norm(v1, v2) = |v1 - v2| #L1 norm
```



to \ from	***A***	***B***	***C***	***D***
A	0	1/2	1	0
B	1/3	0	0	1/2
C	1/3	0	0	1/2
D	1/3	1/2	0	0

"Transition Matrix", *M*

Innovation 1: What pages would a "random Web surfer" end up at?

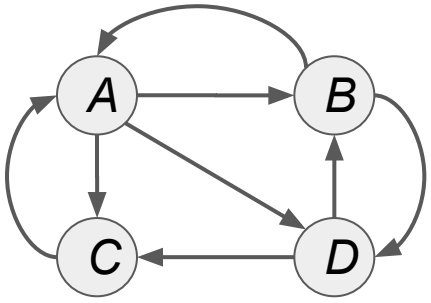
To start: N=4 nodes, so r = [¼, ¼, ¼, ¼,]
after first iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]
after second iteration: $M(M \cdot r) = M^2 \cdot r$ = *[15/48, 11/48, …]*

PageRank

Power iteration algorithm

```
Initialize:    r[0] = [1/N, …, 1/N],
               r[-1]=[0,...,0]
while (err_norm(r[t],r[t-1])>min_err):
    r[t+1] = M·r[t]
    t+=1
solution = r[t]

err_norm(v1, v2) = |v1 - v2| #L1 norm
```



to \ from	***A***	***B***	***C***	***D***
A	0	1/2	1	0
B	1/3	0	0	1/2
C	1/3	0	0	1/2
D	1/3	1/2	0	0

"Transition Matrix", *M*

Innovation 1: What pages would a "random Web surfer" end up at?

To start: N=4 nodes, so r = [¼, ¼, ¼, ¼,]
after first iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]
after second iteration: $M(M \cdot r) = M^2 \cdot r$ = [15/48, 11/48, …]

PageRank

Power iteration algorithm

```
Initialize:    r[0] = [1/N, …, 1/N],
               r[-1]=[0,...,0]
while (err_norm(r[t],r[t-1])>min_err):
    r[t+1] = M·r[t]
    t+=1
solution = r[t]

err_norm(v1, v2) = |v1 - v2| #L1 norm
```

As err_norm gets smaller we are moving toward:

$r = M \cdot r$

We are actually just finding the ***eigenvector*** of M.

Innovation 1: What pages would a "random Web surfer" end up at?

To start: N=4 nodes, so r = [¼, ¼, ¼, ¼,]

after first iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]

after second iteration: $M(M \cdot r) = M^2 \cdot r$ = [15/48, 11/48, …]

PageRank

Power iteration algorithm

```
Initialize:    r[0] = [1/N, …, 1/N],
               r[-1]=[0,...,0]
while (err_norm(r[t],r[t-1])>min_err):
    r[t+1] = M·r[t]
    t+=1
solution = r[t]

err_norm(v1, v2) = |v1 - v2| #L1 norm
```

As err_norm gets smaller we are moving toward:

$r = M \cdot r$

We are actually just finding the ***eigenvector*** of M.

x is an *eigenvector* of **λ** if:
$A \cdot \mathbf{x} = \lambda \cdot \mathbf{x}$

Innovation 1: What pages would a "random Web surfer" end up at?

To start: N=4 nodes, so r = [¼, ¼, ¼, ¼,]
after first iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]
after second iteration: $M(M \cdot r) = M^2 \cdot r$ = [15/48, 11/48, …]

PageRank

Power iteration algorithm

finds the...

```
Initialize:    r[0] = [1/N, …, 1/N],
               r[-1]=[0,...,0]
while (err_norm(r[t],r[t-1])>min_err):
    r[t+1] = M·r[t]
    t+=1
solution = r[t]

err_norm(v1, v2) = |v1 - v2| #L1 norm
```

As err_norm gets smaller we are moving toward:

$r = M \cdot r$

We are actually just finding the ***eigenvector*** of M.

x is an *eigenvector* of λ if:

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

A = 1

since columns of M sum to 1.

thus, *1r=Mr*

Innovation 1: What pages would a "random Web surfer" end up at?

To start: N=4 nodes, so r = [¼, ¼, ¼, ¼,]

after first iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]

after second iteration: $M(M \cdot r) = M^2 \cdot r$ = *[15/48, 11/48, …]*

PageRank

Power iteration algorithm

finds the...

```
Initialize:    r[0] = [1/N, …, 1/N],
               r[-1]=[0,...,0]
while (err_norm(r[t],r[t-1])>min_err):
    r[t+1] = M·r[t]
    t+=1
solution = r[t]

err_norm(v1, v2) = |v1 - v2| #L1 norm
```

As err_norm gets smaller we are moving toward:

$r = M \cdot r$

We are actually just finding the ***eigenvector*** of M.

x is an *eigenvector* of λ if:

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

A = 1

since columns of M sum to 1.

thus, *1r=Mr*

Innovation 1: What pages would a "random Web surfer" end up at?

To start: N=4 nodes, so r = [¼, ¼, ¼, ¼,]

after first iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]

after second iteration: $M(M \cdot r) = M^2 \cdot r$ = *[15/48, 11/48, …]*

PageRank

Power iteration algorithm

```
Initialize:    r[0] = [1/N, …, 1/N],
               r[-1]=[0,...,0]
while (err_norm(r[t],r[t-1])>min_err):
    r[t+1] = M·r[t]
    t+=1
solution = r[t]

err_norm(v1, v2) = |v1 - v2| #L1 norm
```

Where is surfer at time t+1?
p(t+1) = $M \cdot$ p(t)

Suppose: p(t+1) = p(t),
then p(t) is a
stationary distribution
of a random walk.

Thus, r is a stationary distribution. Probability of being at given node.

Innovation 1: What pages would a "random Web surfer" end up at?

PageRank

aka 1st order Markov Process

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:

Where is surfer at time t+1?
$p(t+1) = M \cdot p(t)$

Suppose: $p(t+1) = p(t)$,
then $p(t)$ is a *stationary distribution* of a ***random walk***.

Thus, r is a stationary distribution. Probability of being at given node.

Innovation 1: What pages would a "random Web surfer" end up at?

PageRank

aka 1st order Markov Process

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - No “dead-ends”
      a node can’t propagate its rank
    - No “spider traps”
      set of nodes with no way out.

Where is surfer at time t+1?
$p(t+1) = M \cdot p(t)$

Suppose: $p(t+1) = p(t)$,
then $p(t)$ is a *stationary distribution* of a ***random walk***.

Thus, r is a stationary distribution. Probability of being at given node.

Innovation 1: What pages would a “random Web surfer” end up at?

PageRank

aka 1st order Markov Process

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - No “dead-ends”
      a node can’t propagate its rank
    - No “spider traps”
      set of nodes with no way out.

(technically: it needs to be:
stochastic, *irreducible*, and *aperiodic*)

Where is surfer at time t+1?
$p(t+1) = M \cdot p(t)$

Suppose: p(t+1) = p(t),
then p(t) is a *stationary distribution* of a ***random walk***.

Thus, r is a stationary distribution. Probability of being at given node.

Innovation 1: What pages would a “random Web surfer” end up at?

PageRank

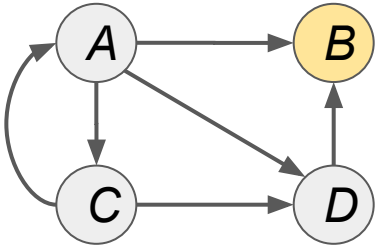
aka 1st order Markov Process

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - No “dead-ends”
      a node can’t propagate its rank
    - No “spider traps”
      set of nodes with no way out.

Where is surfer at time t+1?
$p(t+1) = M \cdot p(t)$

Suppose: $p(t+1) = p(t)$,
then $p(t)$ is a *stationary distribution* of a ***random walk***.

Thus, r is a stationary distribution. Probability of being at given node.



to \ from	***A***	***B***	***C***	***D***
A	0	0	1	0
B	1/3	0	0	1
C	1/3	0	0	0

r would eventually converge to [0, 0, …]

PageRank

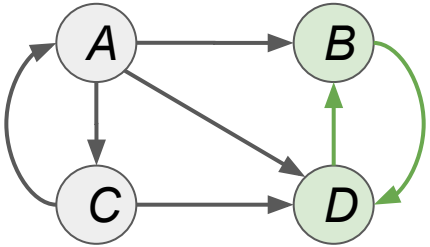
aka 1st order Markov Process

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - No “dead-ends”
      a node can’t propagate its rank
    - No “spider traps”
      set of nodes with no way out.

Where is surfer at time t+1?
$p(t+1) = M \cdot p(t)$

Suppose: $p(t+1) = p(t)$,
then $p(t)$ is a *stationary distribution* of a ***random walk***.

Thus, r is a stationary distribution. Probability of being at given node.



PageRank

aka 1st order Markov Process

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - No "dead-ends"
      a node can't propagate its rank
    - No "spider traps"
      set of nodes with no way out.

(technically: it needs to be:
stochastic, *irreducible*, and *aperiodic*)

Where is surfer at time t+1?
$p(t+1) = M \cdot p(t)$

Suppose: $p(t+1) = p(t)$,
then $p(t)$ is a *stationary distribution* of a ***random walk***.

Thus, r is a stationary distribution. Probability of being at given node.

Innovation 1: What pages would a "random Web surfer" end up at?

columns sum to 1

same node doesn't repeat at a regular interval

non-zero chance of going to any another node

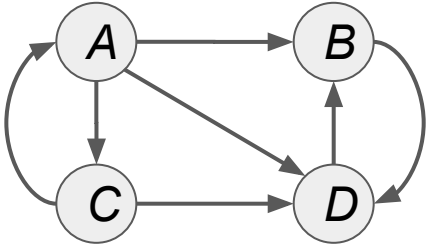
PageRank

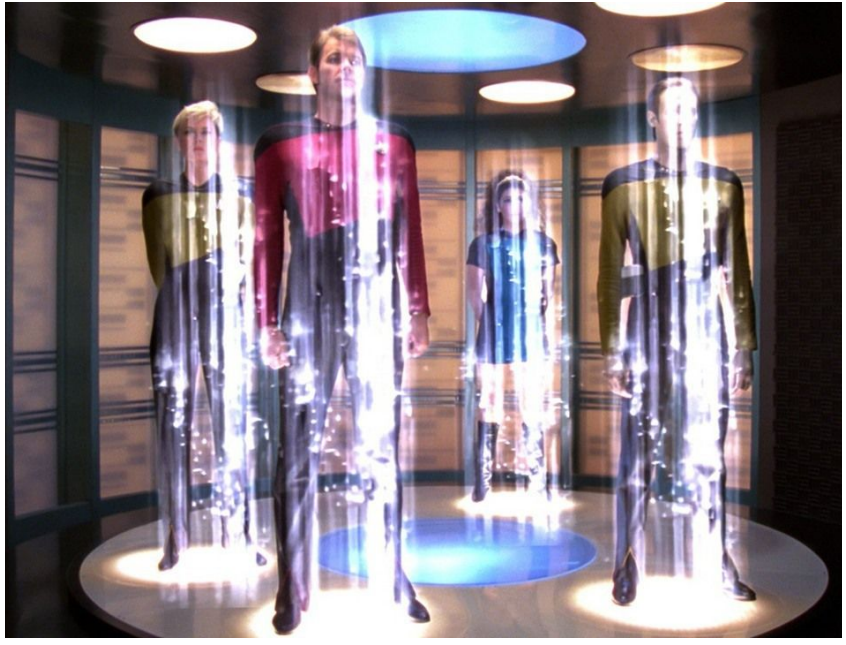
No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation

Add teleportation:At each step, two choices

1. Follow a random link (probability, β = ~.85)
2. Teleport to a random node (probability, 1-β)





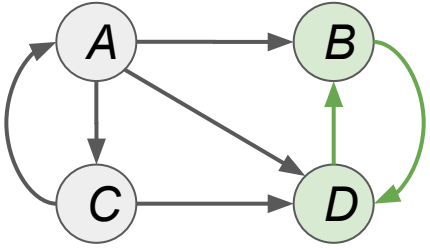
PageRank

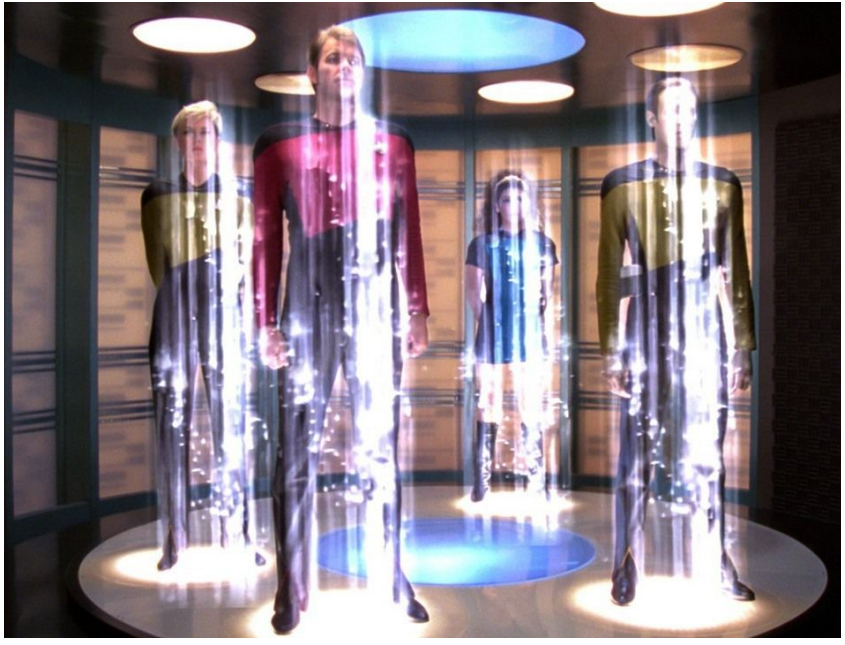
No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation

Add teleportation:At each step, two choices

1. Follow a random link (probability, β = ~.85)
2. Teleport to a random node (probability, 1-β)





PageRank

No “dead-ends”
No “spider traps”

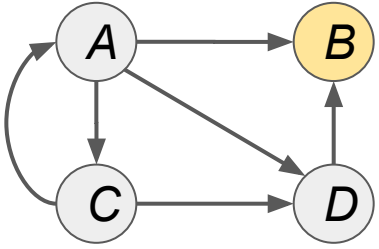
The “Google” PageRank Formulation

Add teleportation chance at all nodes.

i.e. at each step, two choices

1. Follow a random link (probability, β = ~.85)
2. Teleport to a random node (probability, 1-β)

Add teleportation from dead end with probability 1



to \ from	***A***	***B***	***C***	***D***
A	0	0	1	0
B	1/3	0	0	1
C	1/3	0	0	0
D	1/3	0	0	0

PageRank

No “dead-ends”
No “spider traps”

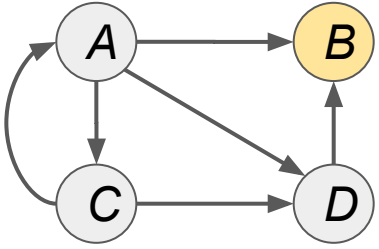
The “Google” PageRank Formulation

Add teleportation chance at all nodes.

i.e. at each step, two choices

1. Follow a random link (probability, β = ~.85)
2. Teleport to a random node (probability, 1-β)

Add teleportation from dead end with probability 1



to \ from	***A***	***B***	***C***	***D***
A	0	¼	1	0
B	1/3	¼	0	1
C	1/3	¼	0	0
D	1/3	¼	0	0

PageRank

No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation

Add teleportation chance at all nodes.

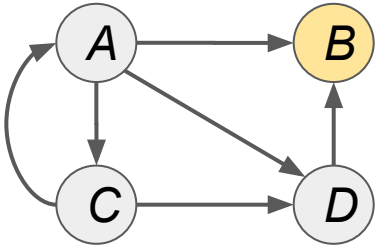
i.e. at each step, two choices

1. Follow a random link (probability, β = ~.85)
2. Teleport to a random node (probability, 1-β)

Add teleportation from dead end with probability 1

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

(Brin and Page, 1998)



to \ from	***A***	***B***	***C***	***D***
A	0	0	1	0
B	1/3	0	0	1
C	1/3	0	0	0
D	1/3	0	0	0

PageRank

No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation

Add teleportation chance at all nodes.

i.e. at each step, two choices

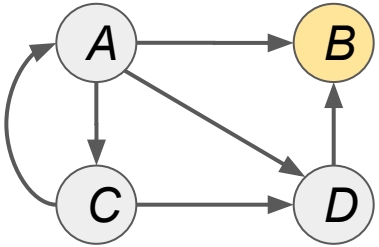
1. Follow a random link (probability, β = ~.85)
2. Teleport to a random node (probability, *1*-β)

Add teleportation from dead end with probability 1

Flow Model

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

(Brin and Page, 1998)



to \ from	***A***	***B***	***C***	***D***
A	0	0	1	0
B	1/3	0	0	1
C	1/3	0	0	0
D	1/3	0	0	0

Matrix Model

$$M' = \beta M + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}$$

	A	***B*** ...
A	⅘*0 + ⅕*¼	¼
B	⅘*⅓ + ⅕*¼	¼
C	⅘*⅓ + ⅕*¼	¼
D	⅘*⅓ + ⅕*¼	¼

assume
β = ⅘

PageRank

No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation

Add teleportation chance at all nodes.

i.e. at each step, two choices

1. Follow a random link (probability, β = ~.85)
2. Teleport to a random node (probability, *1*-β)

Add teleportation from dead end with probability 1

Matrix Model

$$M' = \beta M + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}$$

To apply:
run power iterations over M’
instead of M.

	A	B ...
A	⅘*0 + ⅕*¼	¼
B	⅘*⅓ + ⅕*¼	¼
C	⅘*⅓ + ⅕*¼	¼
D	⅘*⅓ + ⅕*¼	¼

assume β = ⅘