Link Analysis

Stony Brook University
CSE545, Spring 2019

The Web , circa 1998

The Web , circa 1998

Match keywords, language (*information retrieval*)

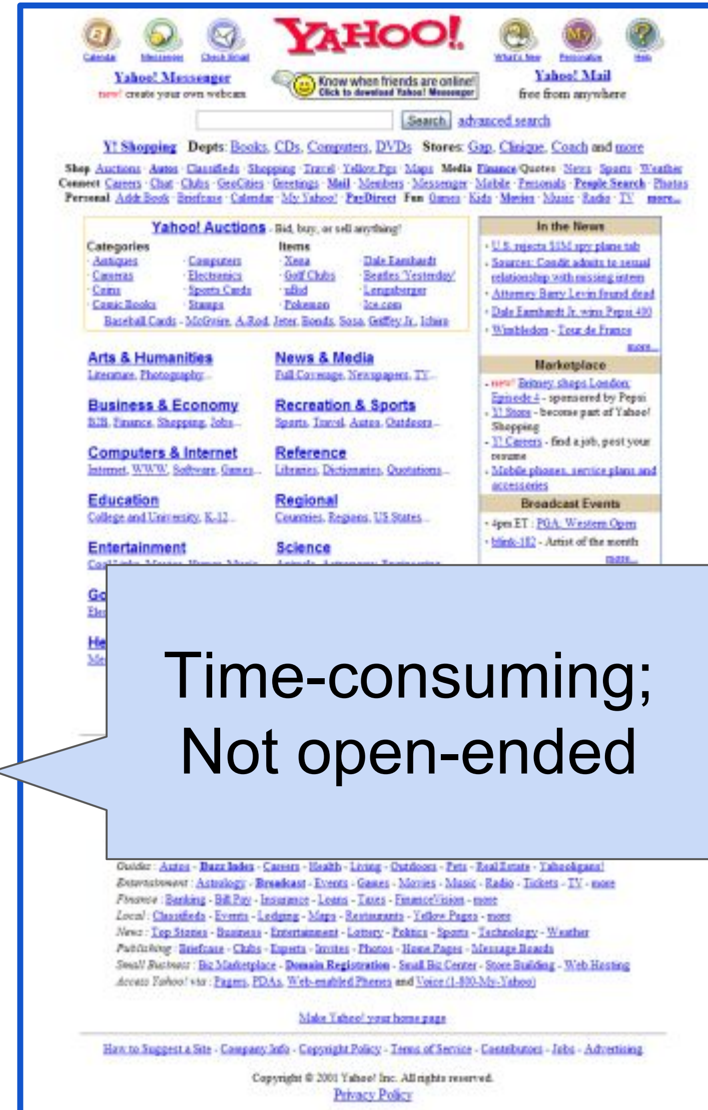
Explore directory

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 present Google, a prototype of a large-scale search engine which makes heavy
use of the structure
and produce much
text and hyperlink

...

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.

PageRank

Key Idea: Consider the citations of the website.

Who links to it?
and what are
their citations?

PageRank

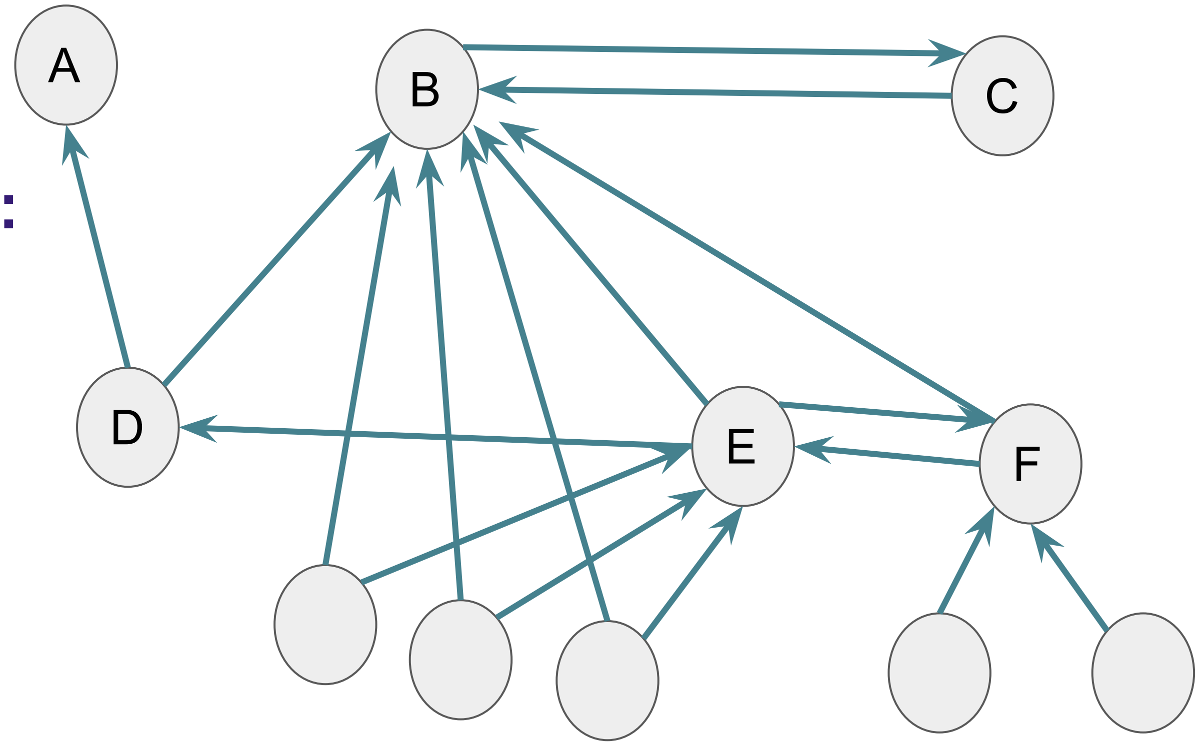
Key Idea: Consider the citations of the website.

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



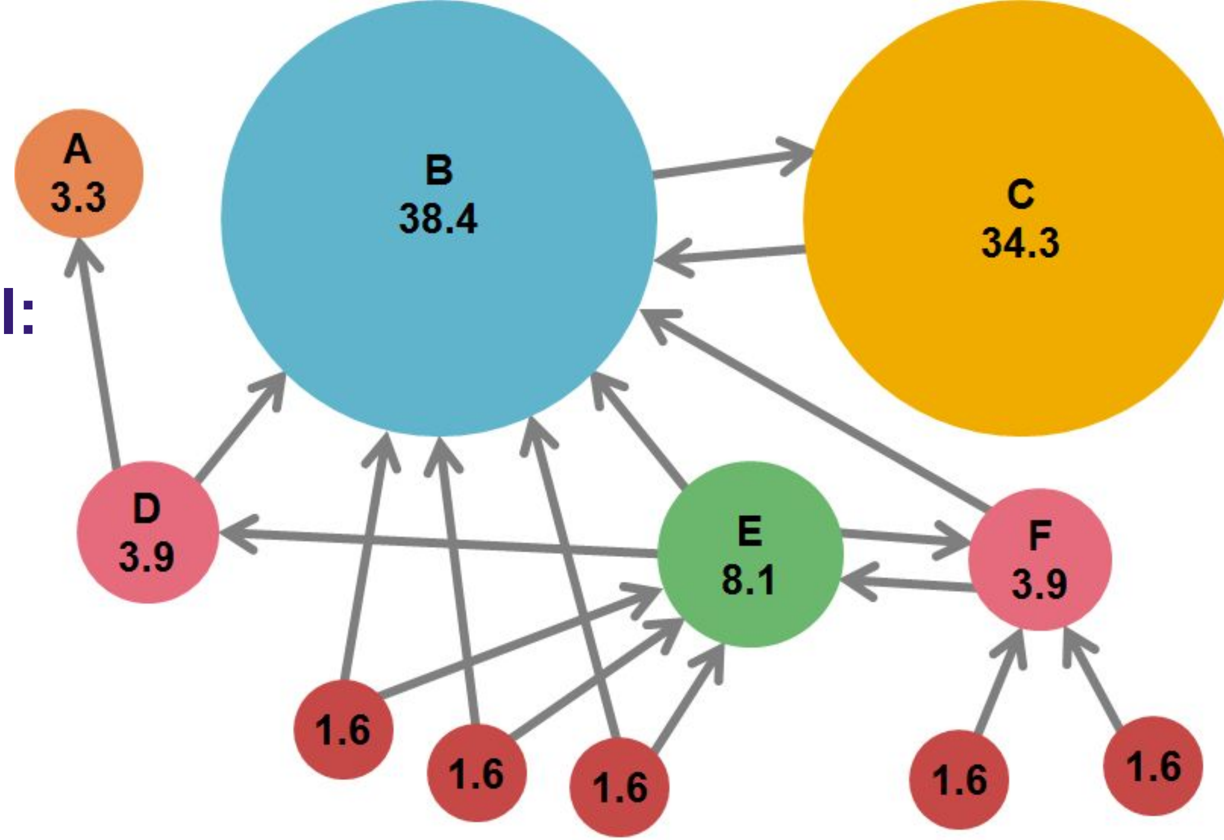
View 1: Flow Model:
in-links as votes

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

View 1: Flow Model:
in-links as votes



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

View 1: Flow Model:

in-links (citations) as votes

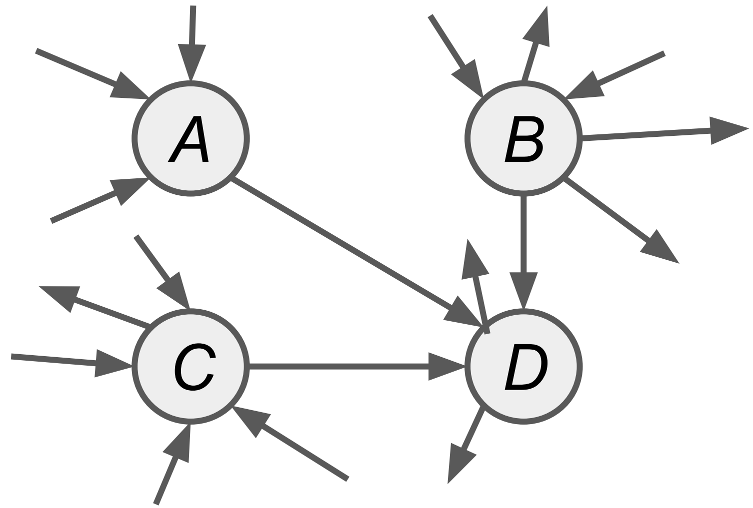
but, citations from important pages should count more.

=> 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



View 1: Flow Model:

How to compute?

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

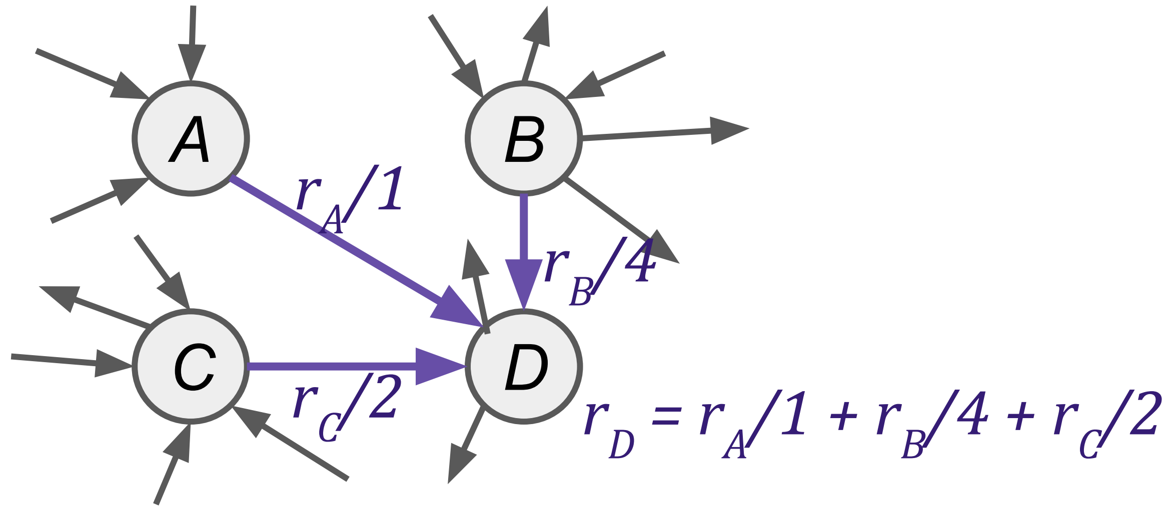
$$vote_j = \frac{r_j}{n_j}$$

(n_j is |out-links|)

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

PageRank

View 1: Flow Model:



How to compute?

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

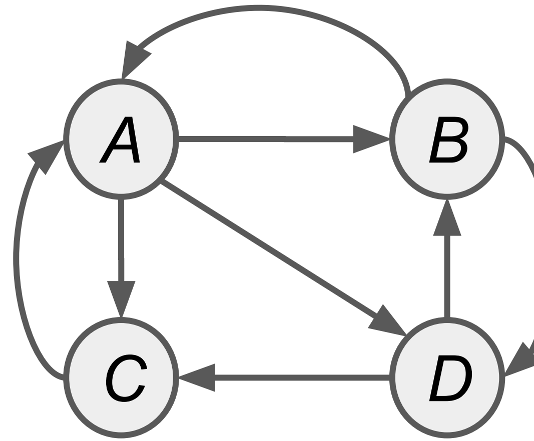
$$vote_j = \frac{r_j}{n_j}$$

(n_j is |out-links|)

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

PageRank

View 1: Flow Model:



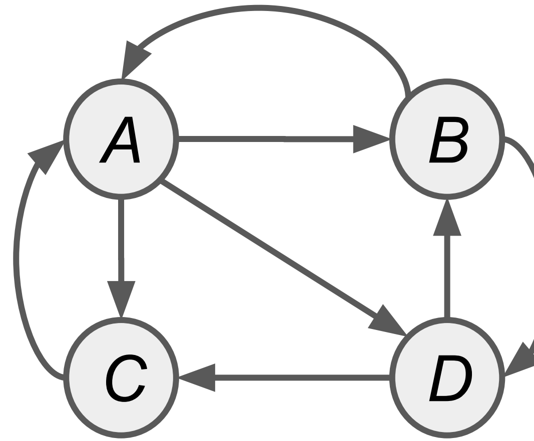
How to compute?

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

$$vote_j = \frac{r_j}{n_j}$$ (n_j is |out-links|)

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

PageRank



View 1: Flow Model:

A System of Equations:

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

How to compute?

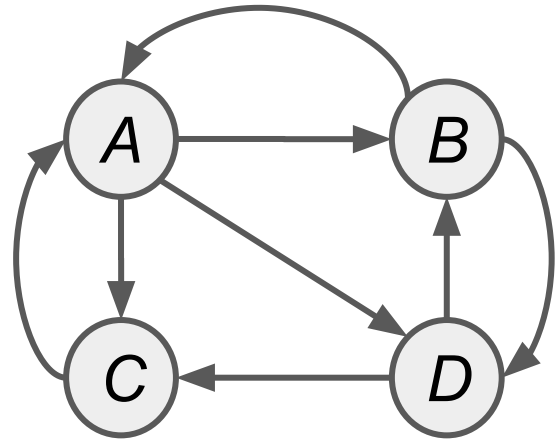
Each page (j) has an importance (i.e. rank, r_j)

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

(n_j is |out-links|)

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

PageRank



View 1: Flow Model:

A System of Equations:

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

How to compute?

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

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

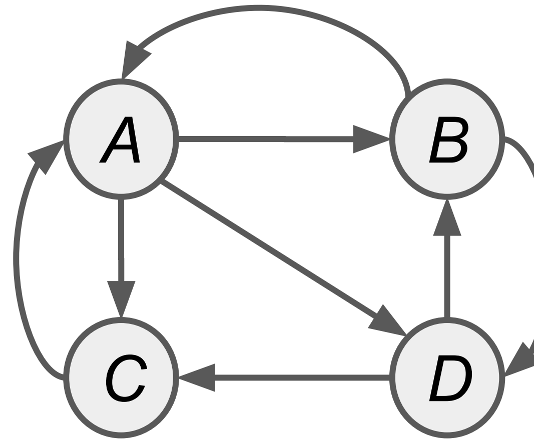
(n_j is |out-links|)

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

PageRank

View 1: Flow Model: Solve

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



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

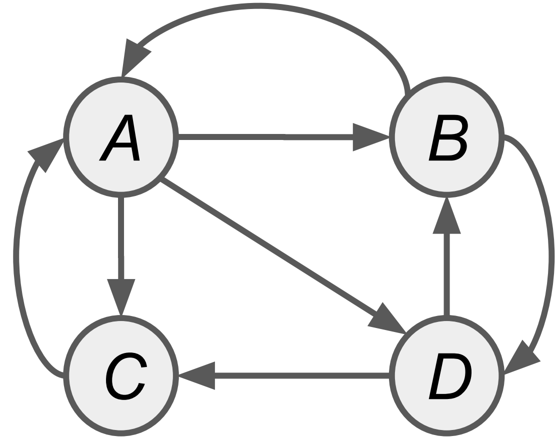
How to compute?

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

$$vote_j = \frac{r_j}{n_j}$$ (n_j is |out-links|)

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

PageRank



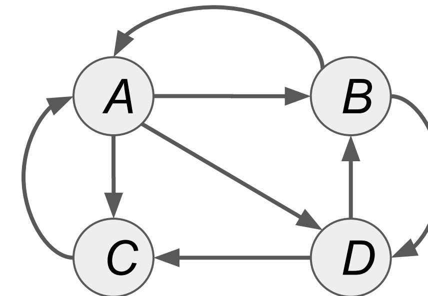
$$1 = r_A + r_B + r_C + r_D$$

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

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

View 2: Matrix Formulation



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

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

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

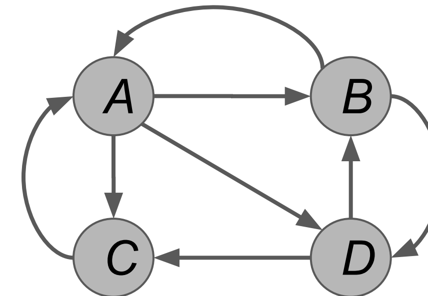
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$

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

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



View 2: Matrix Formulation

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

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

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

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

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

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

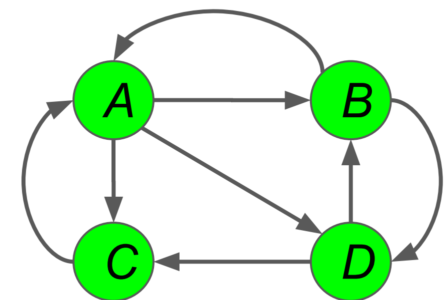
Transition Matrix, M

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

To Start, all are equally likely at ¼

View 2: Matrix Formulation

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



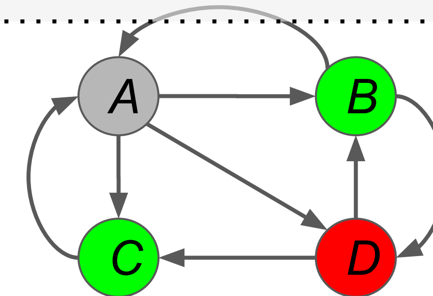
$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

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

To Start, all are equally likely at ¼: ends up at D



View 2: Matrix Formulation

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

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

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

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

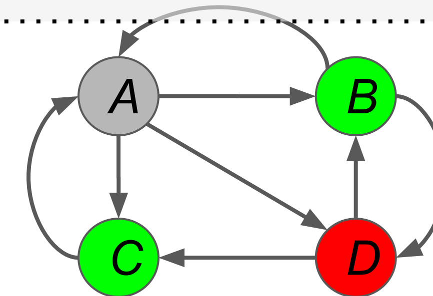
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

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

To Start, all are equally likely at ¼: ends up at D
C and B are then equally likely: ->D->B=¼*½; ->D->C=¼*½



View 2: Matrix Formulation

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

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

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

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

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

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

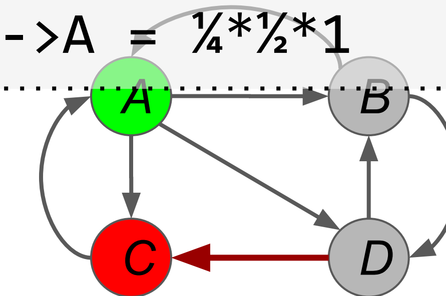
Transition Matrix, M

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

To Start, all are equally likely at ¼: ends up at D

C and B are then equally likely: ->D->B=¼*½; ->D->C=¼*½

Ends up at C: then A is only option: ->D->C->A = ¼*½*1



View 2: Matrix Formulation

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

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

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

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

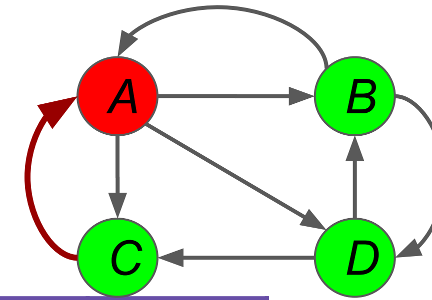
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

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

. . .



View 2: Matrix Formulation

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

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

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

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

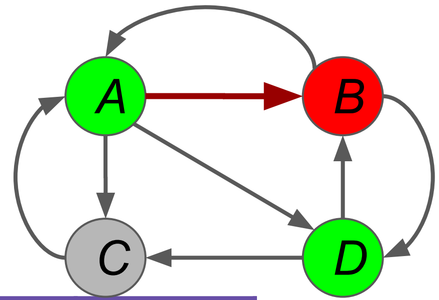
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

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

. . .



View 2: Matrix Formulation

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

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

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

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

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

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

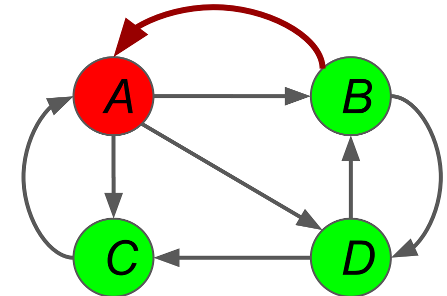
Transition Matrix, M

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

. . .

View 2: Matrix Formulation

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



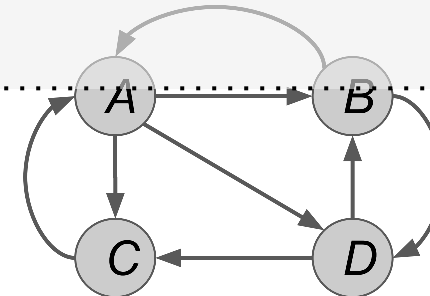
$$r_A = \frac{r_B}{2} + \frac{r_C}{1}$$
$$r_B = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_C = \frac{r_A}{3} + \frac{r_D}{2}$$
$$r_D = \frac{r_A}{3} + \frac{r_B}{2}$$

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

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



View 2: Matrix Formulation

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

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

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

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

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

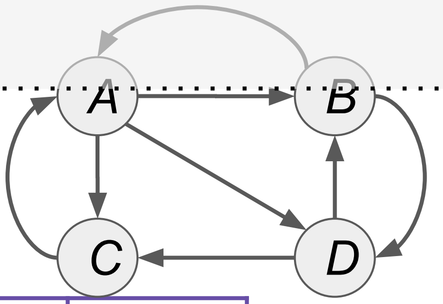
to \ from	***A***	***B***	***C***	***D***
A	0	1/2	1	0
B	1/3	0	0	1/2
C	1/3	0	0	1/2
D	1/3	1/2	0	0

Transition Matrix, M

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

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

after 1st iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]



View 2: Matrix Formulation

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

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

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

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

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

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

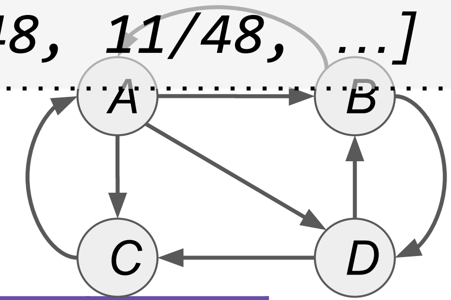
Innovation: What pages would a "random Web surfer" end up at?

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

after 1st iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]

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

View 2: Matrix Formulation



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

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

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

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

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

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

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

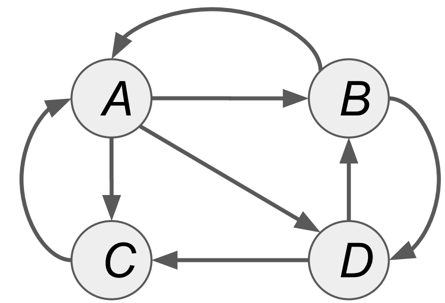
after 1st iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]

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

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

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

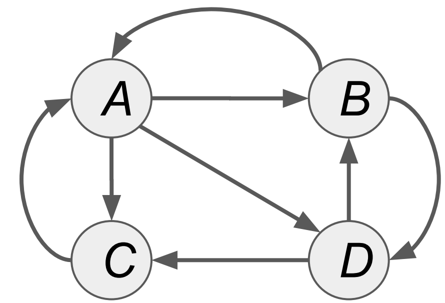
after 1st iteration: $M \cdot r$ = [3/8, 5/24, 5/24, 5/24]

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

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*

As err_norm gets smaller we are moving toward: $r = M \cdot r$

View 3: Eigenvectors:

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: $r = M \cdot r$

View 3: Eigenvectors:

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

finds the...

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
```

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

(Leskovec at al., 2014; http://www.mmds.org/)

As err_norm gets smaller we are moving toward: $r = M \cdot r$

View 3: Eigenvectors:

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

finds the...

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) = sum(|v1 - v2|)
                   #L1 norm
```

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

$\lambda = 1$ (eigenvalue for 1st principal eigenvector)
since columns of M sum to 1.
Thus, *if* r is **x**, then $Mr=1r$

View 4: Markov Process

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.

View 4: Markov Process

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***.

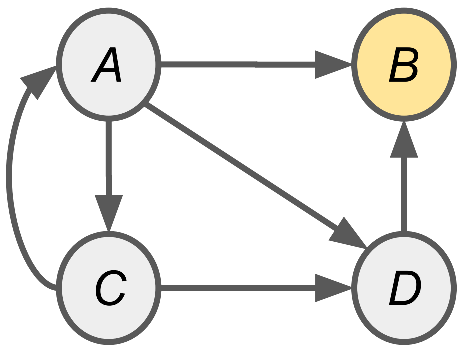
Thus, r is a stationary distribution. Probability of being at given node.

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.

Also known as being *stochastic*, *irreducible*, and *aperiodic.*

View 4: Markov Process - Problems for vanilla PI



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

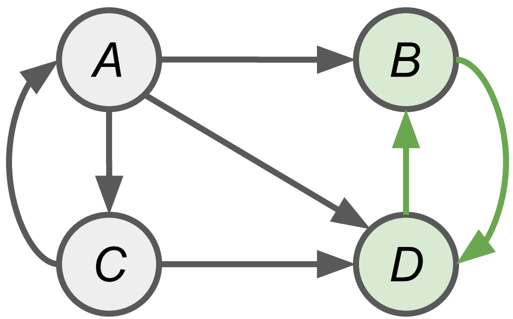
What would *r* converge to?

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.

Also known as being *stochastic*, *irreducible*, and *aperiodic.*

View 4: Markov Process - Problems for vanilla PI



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	1	0	0

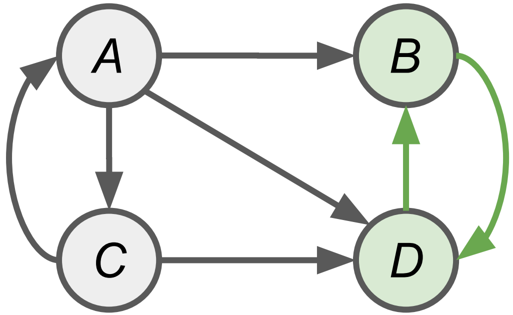
What would *r* converge to?

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.

Also known as being *stochastic*, *irreducible*, and *aperiodic.*

View 4: Markov Process - Problems for vanilla PI



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	1	0	0

What would *r* converge to?

aka 1st order Markov Process

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

same node doesn't repeat at regular intervals

columns sum to 1 non-zero chance of going to any other node

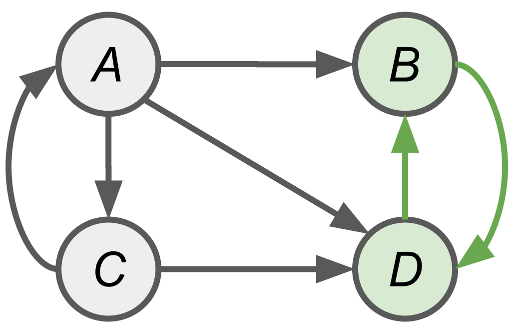
Also known as being *stochastic*, *irreducible*, and *aperiodic*.

Goals:
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)

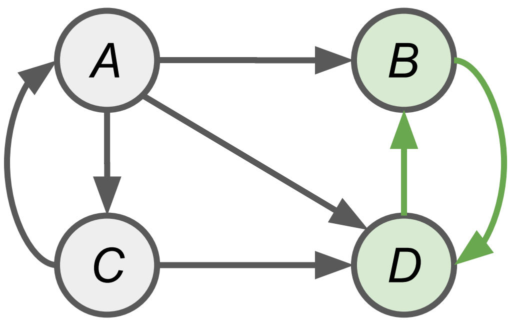


Goals:
No "dead-ends"
No "spider traps"

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)



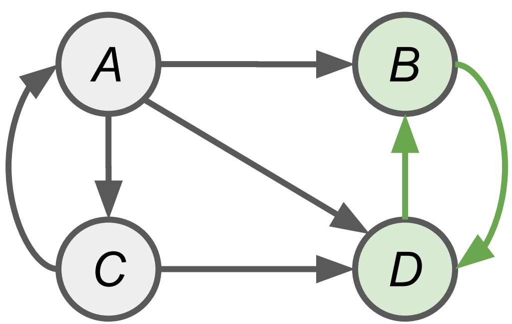
to \ from	***A***	***B***	***C***	***D***
A	0	0	1	0
B	⅓	0	0	1
C	⅓	0	0	0
D	⅓	1	0	0

Goals:
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)



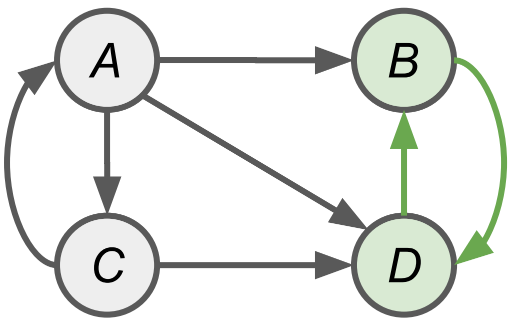
to \ from	***A***	***B***	***C***	***D***
A	0	0+.15*¼	1	0+.15*¼
B	⅓	0+.15*¼	0	.85*1+.15*¼
C	⅓	0+.15*¼	0	0+.15*¼
D	⅓	.85*1 +.15*¼	0	0+.15*¼

Goals:
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)



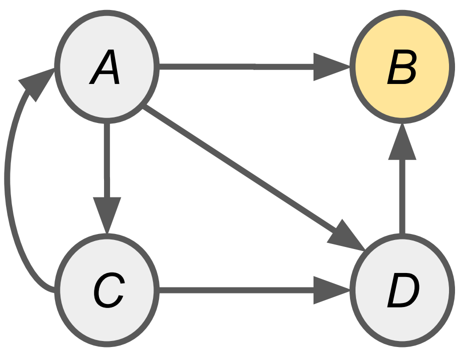
to \ from	***A***	***B***	***C***	***D***
A	0+.15*¼	0+.15*¼	85*1+.15*¼	0+.15*¼
B	.85*⅓+.15*¼	0+.15*¼	0+.15*¼	.85*1+.15*¼
C	.85*⅓+.15*¼	0+.15*¼	0+.15*¼	0+.15*¼
D	.85*⅓+.15*¼	.85*1+.15*¼	0+.15*¼	0+.15*¼

Goals:
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)



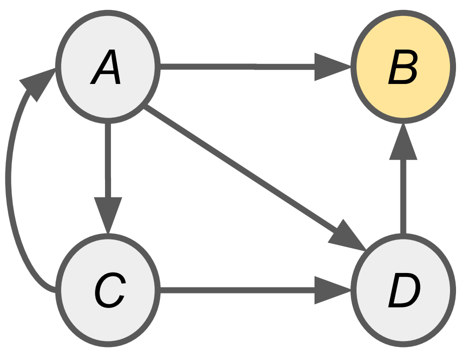
to \ from	***A***	***B***	***C***	***D***
A	0	0	1	0
B	⅓	0	0	1
C	⅓	0	0	0
D	⅓	0	0	0

Goals:
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)



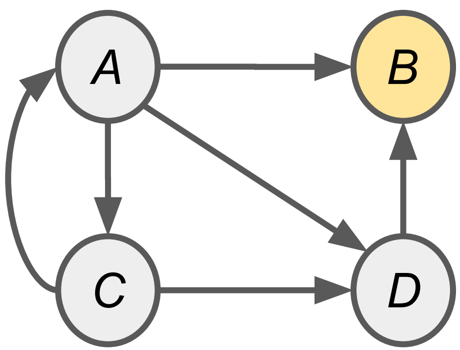
to \ from	***A***	***B***	***C***	***D***
A	0	¼	1	0
B	⅓	¼	0	1
C	⅓	¼	0	0
D	⅓	¼	0	0

Goals:
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)



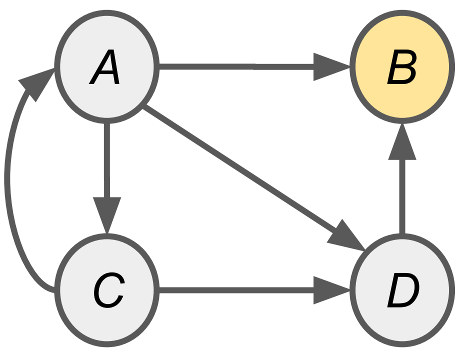
to \ from	***A***	***B***	***C***	***D***
A	0	.85*¼+.15*¼	1	0
B	⅓	.85*¼+.15*¼	0	1
C	⅓	.85*¼+.15*¼	0	0
D	⅓	.85*¼+.15*¼	0	0

Goals:
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)
   (Teleport from a dead-end has probability 1)



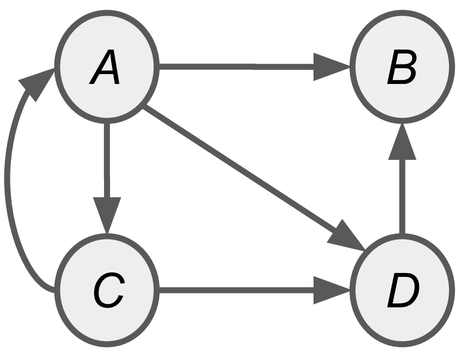
to \ from	***A***	***B***	***C***	***D***
A	0+.15*¼	1*¼	85*1+.15*¼	0+.15*¼
B	.85*⅓+.15*¼	1*¼	0+.15*¼	.85*1+.15*¼
C	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼
D	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼

Goals:
No “dead-ends”
No “spider traps”

Teleportation, as 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+.15*¼	1*¼	85*1+.15*¼	0+.15*¼
B	.85*⅓+.15*¼	1*¼	0+.15*¼	.85*1+.15*¼
C	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼
D	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼

Goals:
No “dead-ends”
No “spider traps”

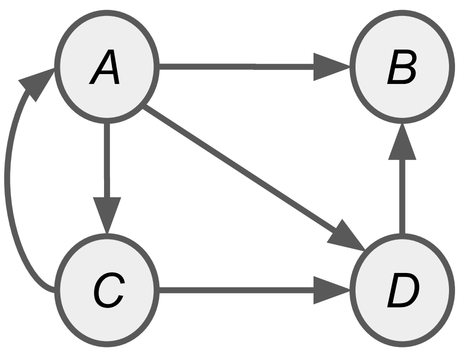
Teleportation, as Flow Model:

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

(Brin and Page, 1998)

Teleportation, as Matrix Model:

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



to \ from	***A***	***B***	***C***	***D***
A	0+.15*¼	1*¼	85*1+.15*¼	0+.15*¼
B	.85*⅓+.15*¼	1*¼	0+.15*¼	.85*1+.15*¼
C	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼
D	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼

Goals:
No “dead-ends”
No “spider traps”

Teleportation, as Flow Model:

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

(Brin and Page, 1998)

Teleportation, as Matrix Model:

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

to \ from	***A***	***B***	***C***	***D***
A	0+.15*¼	.85*¼+.15*¼	85*1+.15*¼	0+.15*¼
B	.85*⅓+.15*¼	.85*¼+.15*¼	0+.15*¼	.85*1+.15*¼
C	.85*⅓+.15*¼	.85*¼+.15*¼	0+.15*¼	0+.15*¼
D	.85*⅓+.15*¼	.85*¼+.15*¼	0+.15*¼	0+.15*¼

Goals:
No “dead-ends”
No “spider traps”

Teleportation, as Flow Model:

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

(Brin and Page, 1998)

Teleportation, as 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.

to \ from	***A***	***B***	***C***	***D***
A	0+.15*¼	1*¼	85*1+.15*¼	0+.15*¼
B	.85*⅓+.15*¼	1*¼	0+.15*¼	.85*1+.15*¼
C	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼
D	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼

Goals:
No “dead-ends”
No “spider traps”

Teleportation, as Flow Model:

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

(Brin and Page, 1998)

Teleportation, as Matrix Model:

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

Steps:

1. Compute M
2. Add 1/N to all dead-ends.
3. Convert *M* to *M’*
4. Run Power Iterations.

to \ from	***A***	***B***	***C***	***D***
A	0+.15*¼	1*¼	85*1+.15*¼	0+.15*¼
B	.85*⅓+.15*¼	1*¼	0+.15*¼	.85*1+.15*¼
C	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼
D	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼

Goals:
No “dead-ends”
No “spider traps”

Teleportation, as Flow Model:

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

(Brin and Page, 1998)

Teleportation,
as Matrix Model: $M' = \beta\, M + (1-\beta)\left[\frac{1}{N}\right]_{N \times N}$

Steps:

1. Compute M
2. Add 1/N to all dead-ends.
3. Convert *M* to *M’*
4. Run Power Iterations.

to \ fr				D
A				+.15*¼
B				5*1+.15*¼
C				+.15*¼
D	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼

In Practice, Just store β *M* as sparse matrix and distribute r acoording to above.

In other words, you only need to store M (as a sparse matrix) and r (as a vector), but never store M'. Use this function within the inner loop of power iterations to achieve the same result as if using M'.

Teleportation, as Flow Model:

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

(Brin and Page, 1998)

Teleportation, as Matrix Model:

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

Steps:

1. Compute M
2. Add 1/N to all dead-ends.
3. Convert *M* to *M'*
4. Run Power Iterations.

to \ fr				D
A				+.15*¼
B				5*1+.15*¼
C				+.15*¼
D	.85*⅓+.15*¼	1*¼	0+.15*¼	0+.15*¼

In Practice, Just store β *M* as sparse matrix and distribute r acoording to above.

Summary

- Flow View: Link Voting
- Matrix View: Linear Algebra
  - Eigenvectors View
- Markov Process View
- How to remove:
  - Dead Ends
  - Spider Traps

In practice, sparse matrix, implement teleportation functionally rather than update M’