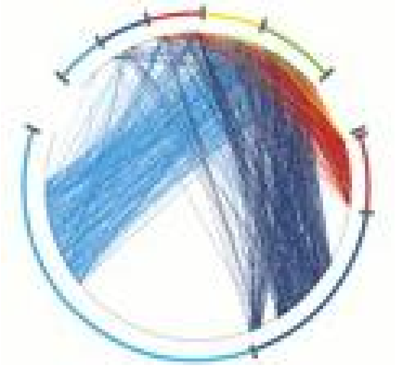
Models and Algorithms for Complex Networks

Link Analysis Ranking



Why Link Analysis?

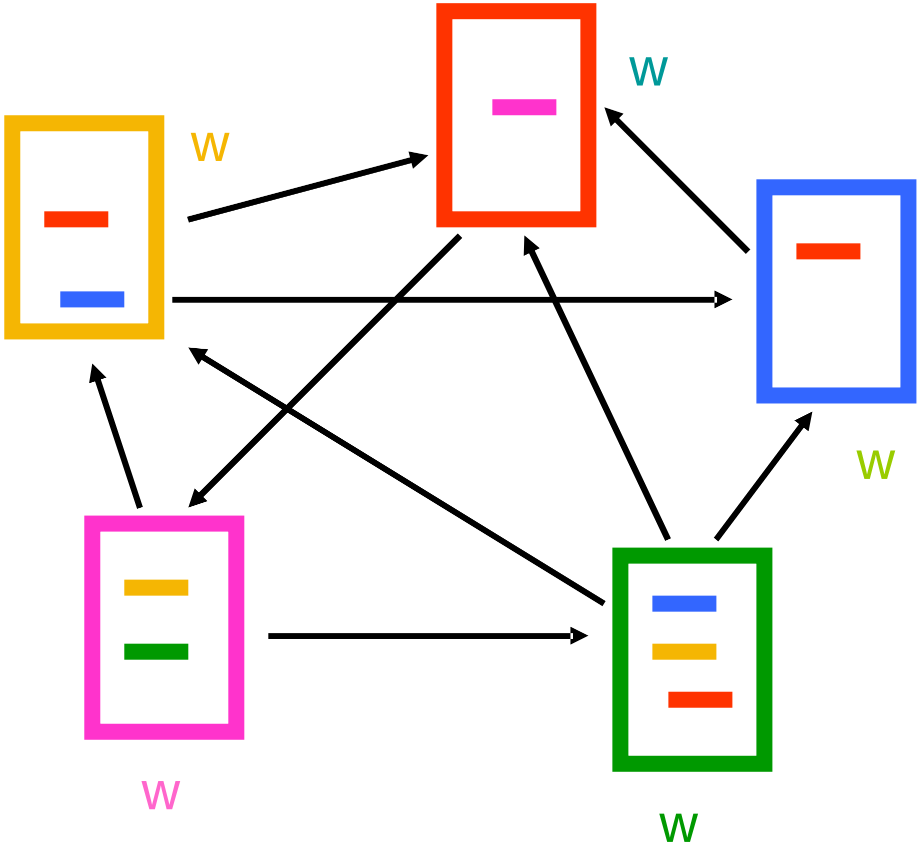
- First generation search engines
  - view documents as flat text files
  - could not cope with size, spamming, user needs
- Second generation search engines
  - Ranking becomes critical
  - use of Web specific data: Link Analysis
  - shift from relevance to authoritativeness
  - a success story for the network analysis

Link Analysis: Intuition

- A link from page p to page q denotes endorsement
  - page p considers page q an authority on a subject
  - mine the web graph of recommendations
  - assign an authority value to every page

Link Analysis Ranking Algorithms

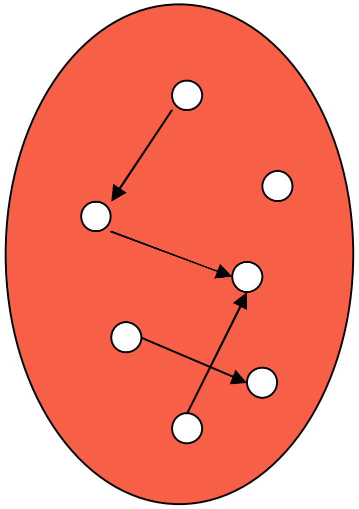
- Start with a collection of web pages
- Extract the underlying hyperlink graph
- Run the LAR algorithm on the graph
- Output: an authority weight for each node



Algorithm input

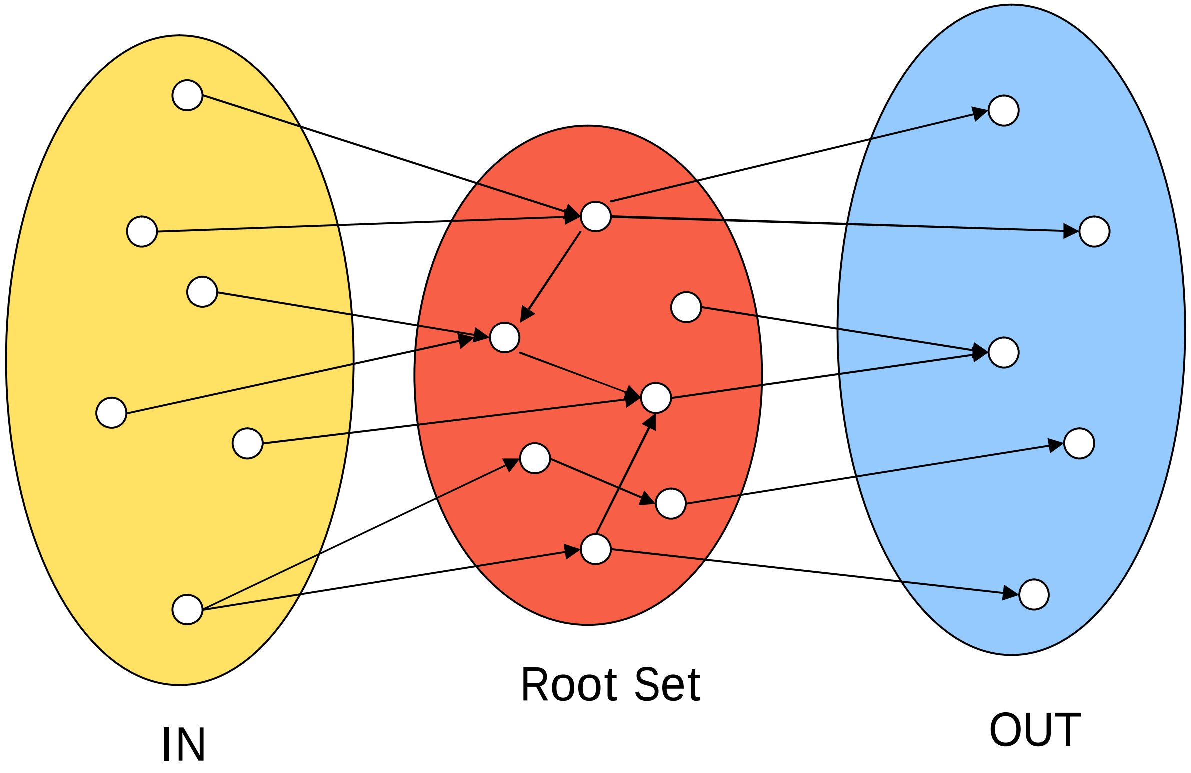
- Query independent: rank the whole Web
  - PageRank (Brin and Page 98) was proposed as query independent
- Query dependent: rank a small subset of pages related to a specific query
  - HITS (Kleinberg 98) was proposed as query dependent

Query dependent input

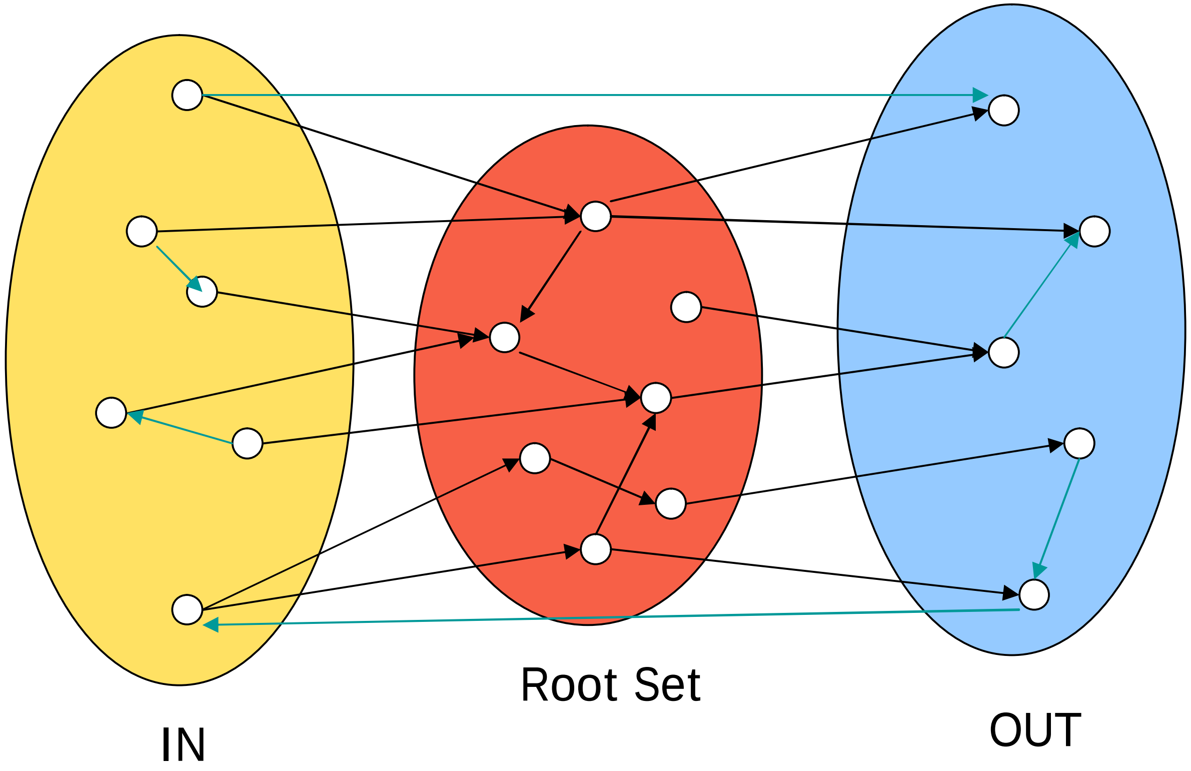


Root Set

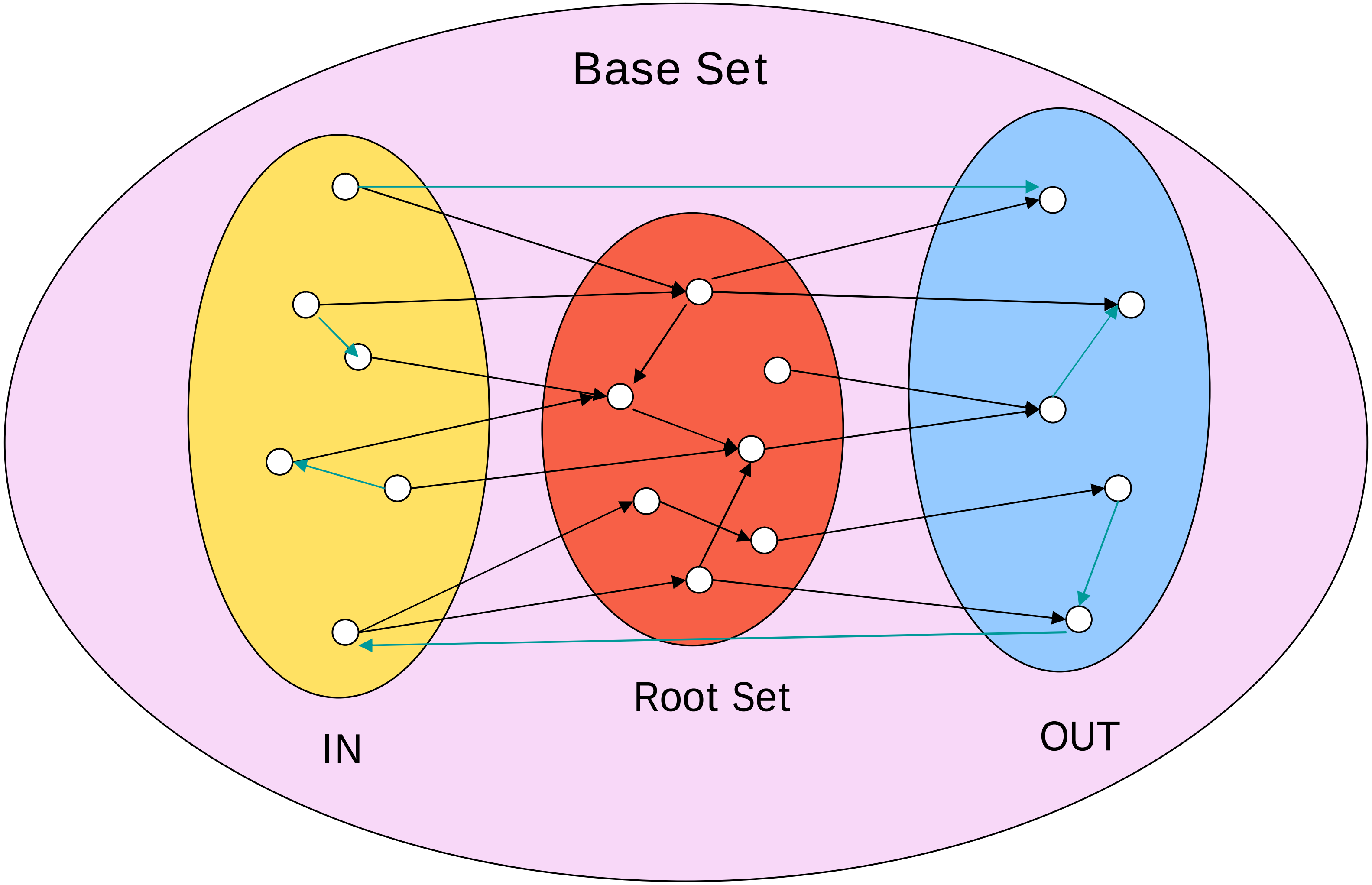
Query dependent input



Query dependent input



Query dependent input



Link Filtering

- Navigational links: serve the purpose of moving within a site (or to related sites)
    - `www.espn.com` → `www.espn.com/nba`
    - `www.yahoo.com` → `www.yahoo.it`
    - `www.espn.com` → `www.msn.com`
- Filter out navigational links
    - same domain name
        - `www.yahoo.com` vs `yahoo.com`
    - same IP address
    - other way?

Outline

- previous work
- ...in the beginning...
- some more algorithms
- some experimental data
- a theoretical framework

Previous work

- The problem of identifying the most important nodes in a network has been studied before in social networks and bibliometrics
- The idea is similar
  - A link from node p to node q denotes endorsement
  - mine the network at hand
  - assign an centrality/importance/standing value to every node

Social network analysis

- Evaluate the centrality of individuals in social networks
  - degree centrality
    - the (weighted) degree of a node
  - distance centrality
    - the average (weighted) distance of a node to the rest in the graph

$$D_c(v) = \frac{1}{\sum_{u \neq v} d(v,u)}$$

  - betweenness centrality
    - the average number of (weighted) shortest paths that use node v

$$B_c(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Random walks on undirected graphs

- In the stationary distribution of a random walk on an undirected graph, the probability of being at node i is proportional to the (weighted) degree of the vertex

- Random walks on undirected graphs are not “interesting”

Counting paths – Katz 53

- The importance of a node is measured by the weighted sum of paths that lead to this node
- $A^m[i,j]$ = number of paths of length m from i to j
- Compute

$$P = bA + b^2A^2 + \cdots + b^mA^m + \cdots = (I - bA)^{-1} - I$$

- converges when $b < \lambda_1(A)$
- Rank nodes according to the column sums of the matrix P

Bibliometrics

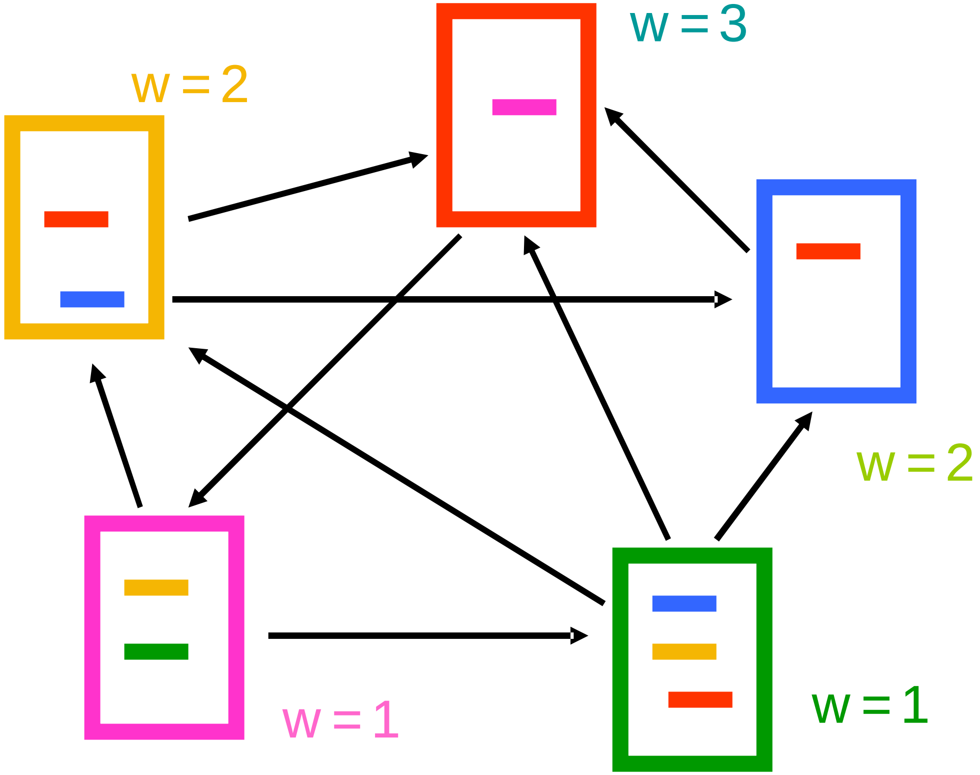
- Impact factor (E. Garfield 72)
  - counts the number of citations received for papers of the journal in the previous two years
- Pinsky-Narin 76
  - perform a random walk on the set of journals
  - P_{ij} = the fraction of citations from journal i that are directed to journal j

Outline

- previous work
- ...in the beginning...
- some more algorithms
- some experimental data
- a theoretical framework

InDegree algorithm

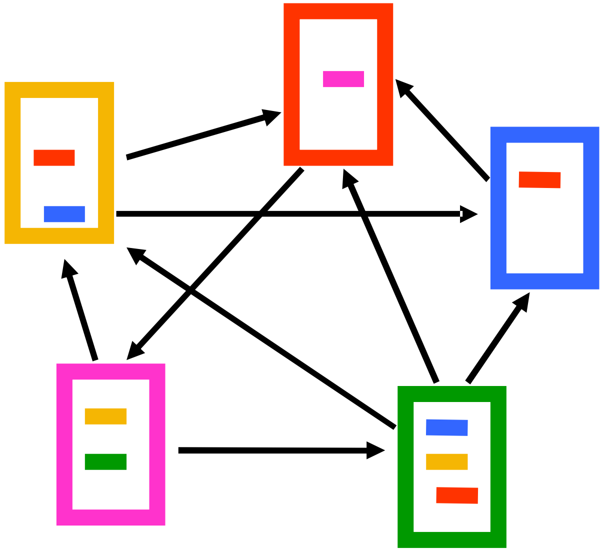
- Rank pages according to in-degree
  - $w_i = |B(i)|$



1. Red Page
2. Yellow Page
3. Blue Page
4. Purple Page
5. Green Page

PageRank algorithm [BP98]

- Good authorities should be pointed by good authorities
- Random walk on the web graph
  - pick a page at random
  - with probability $1-\alpha$ jump to a random page
  - with probability α follow a random outgoing link
- Rank according to the stationary distribution
- $$PR(p) = \alpha \sum_{q \to p} \frac{PR(q)}{|F(q)|} + (1-\alpha)\frac{1}{n}$$



1. Red Page
2. Purple Page
3. Yellow Page
4. Blue Page
5. Green Page

Markov chains

- A Markov chain describes a discrete time stochastic process over a set of states

$$S = \{s_1, s_2, \dots s_n\}$$

according to a transition probability matrix

$$P = \{P_{ij}\}$$

  - P_{ij} = probability of moving to state j when at state i
    - $\sum_j P_{ij} = 1$ (stochastic matrix)

- Memorylessness property: The next state of the chain depends only at the current state and not on the past of the process (first order MC)
  - higher order MCs are also possible

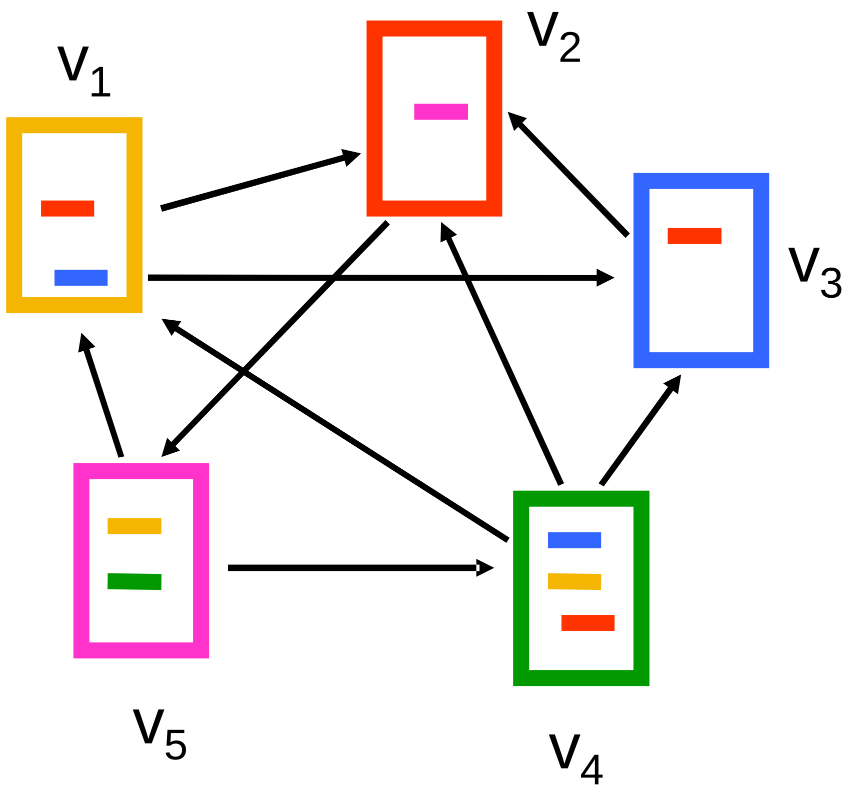
Random walks

- Random walks on graphs correspond to Markov Chains
  - The set of states S is the set of nodes of the graph G
  - The transition probability matrix is the probability that we follow an edge from one node to another

An example

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

$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \end{bmatrix}$$



State probability vector

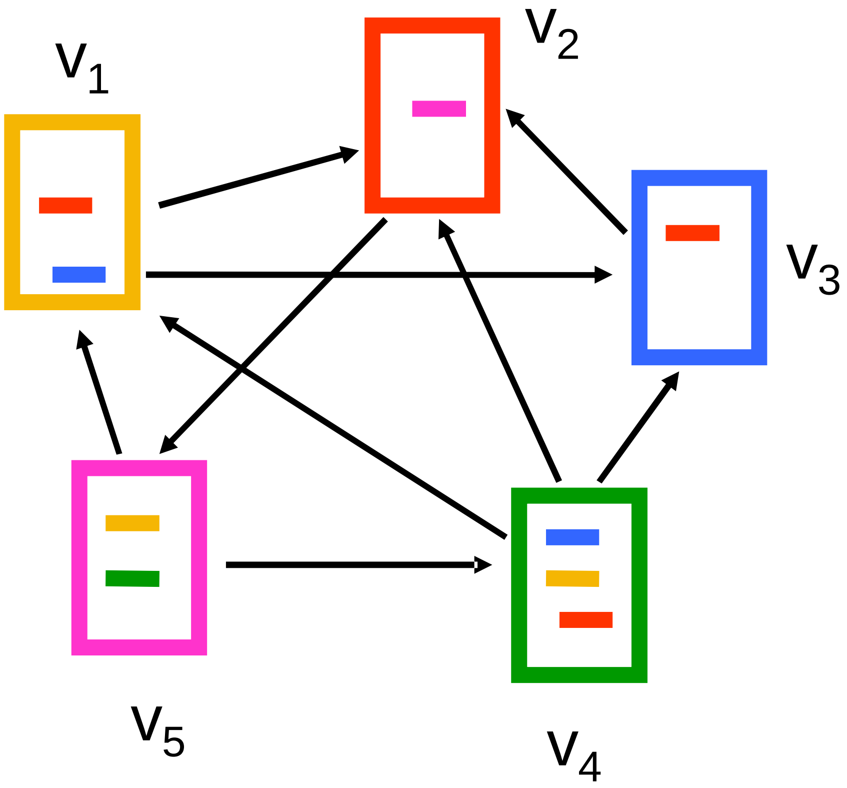
§ The vector $q^t = (q^t_1, q^t_2, \dots, q^t_n)$ that stores the probability of being at state i at time t

  § q^0_i = the probability of starting from state i

$$q^t = q^{t-1} P$$

An example

$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



$q^{t+1}_1 = 1/3\ q^t_4 + 1/2\ q^t_5$

$q^{t+1}_2 = 1/2\ q^t_1 + q^t_3 + 1/3\ q^t_4$

$q^{t+1}_3 = 1/2\ q^t_1 + 1/3\ q^t_4$

$q^{t+1}_4 = 1/2\ q^t_5$

$q^{t+1}_5 = q^t_2$

Stationary distribution

- A stationary distribution for a MC with transition matrix P, is a probability distribution π, such that $\pi = \pi P$
- A MC has a unique stationary distribution if
  - it is irreducible
    - the underlying graph is strongly connected
  - it is aperiodic
    - for random walks, the underlying graph is not bipartite
- The probability π_i is the fraction of times that we visited state i as $t \longrightarrow \infty$
- The stationary distribution is an eigenvector of matrix P
  - the principal left eigenvector of P – stochastic matrices have maximum eigenvalue 1

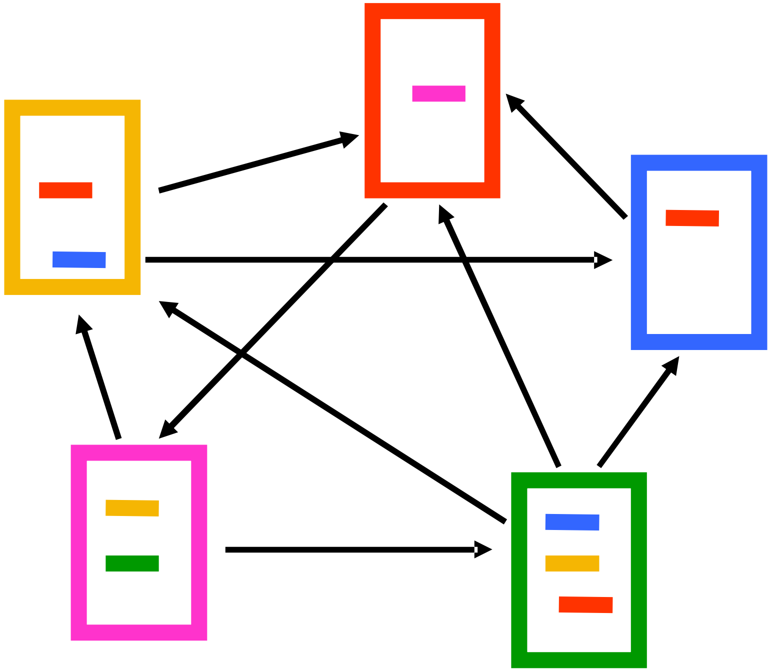
Computing the stationary distribution

- The Power Method
  - Initialize to some distribution q^0
  - Iteratively compute $q^t = q^{t-1}P$
  - After enough iterations $q^t \approx \pi$
  - Power method because it computes $q^t = q^0P^t$
- Why does it converge?
  - follows from the fact that any vector can be written as a linear combination of the eigenvectors
    - $q^0 = v_1 + c_2v_2 + \ldots c_nv_n$
- Rate of convergence
  - determined by λ_2^t

The PageRank random walk

- Vanilla random walk
  - make the adjacency matrix stochastic and run a random walk

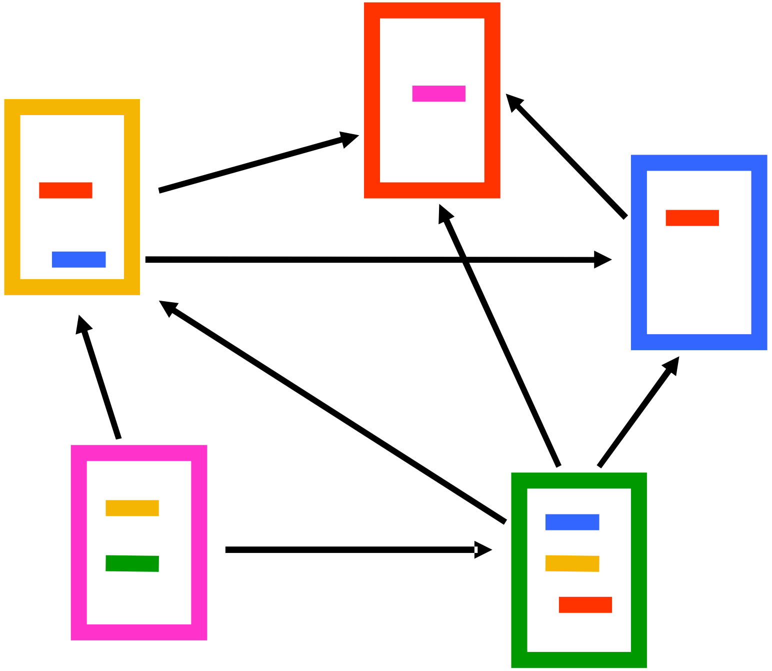
$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



The PageRank random walk

- What about sink nodes?
  - what happens when the random walk moves to a node without any outgoing inks?

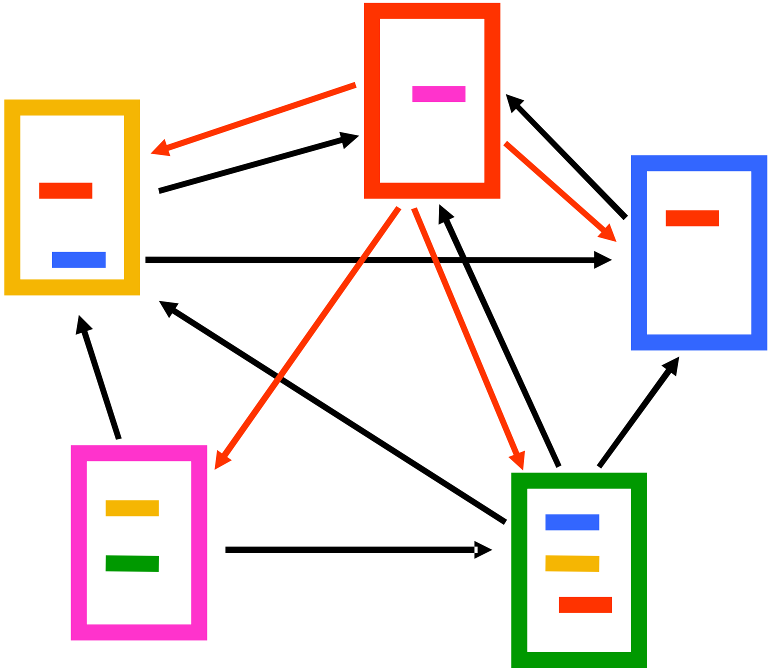
$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



The PageRank random walk

- Replace these row vectors with a vector v
  - typically, the uniform vector

$$P' = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



$P' = P + dv^T$

$$d = \begin{cases} 1 & \text{if i is sink} \\ 0 & \text{otherwise} \end{cases}$$

The PageRank random walk

- How do we guarantee irreducibility?
  - add a random jump to vector v with prob α
    - typically, to a uniform vector

$$P'' = \alpha \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \end{bmatrix} + (1-\alpha) \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix}$$

$P'' = \alpha P' + (1-\alpha)uv^T$, where u is the vector of all 1s

Effects of random jump

- Guarantees irreducibility
- Motivated by the concept of random surfer
- Offers additional flexibility
  - personalization
  - anti-spam
- Controls the rate of convergence
  - the second eigenvalue of matrix P'' is α

A PageRank algorithm

- Performing vanilla power method is now too expensive – the matrix is not sparse

```
$q^0 = v$
$t = 1$
repeat
    $q^t = (P'')^T q^{t-1}$
    $\delta = \|q^t - q^{t-1}\|$
    $t = t + 1$
until $\delta < \varepsilon$
```

Efficient computation of $y = (P'')^T x$

```
$y = \alpha P^T x$
$\beta = \|x\|_1 - \|y\|_1$
$y = y + \beta v$
```

P = normalized adjacency matrix

$P' = P + dv^T$, where d_i is 1 if i is sink and 0 o.w.

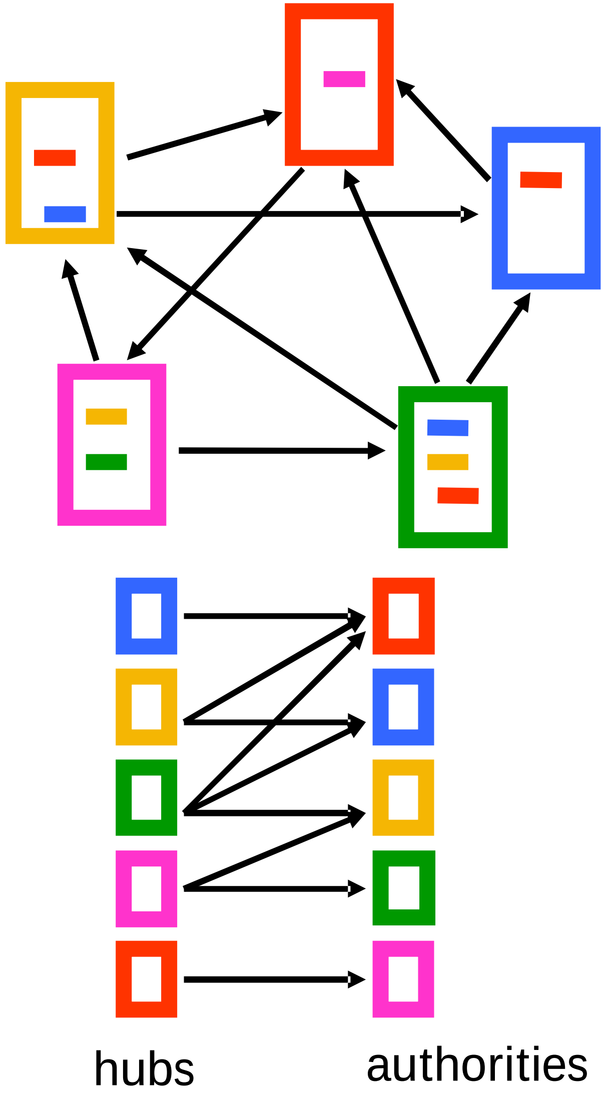
$P'' = \alpha P' + (1-\alpha)uv^T$, where u is the vector of all 1s

Research on PageRank

- Specialized PageRank
  - personalization [BP98]
    - instead of picking a node uniformly at random favor specific nodes that are related to the user
  - topic sensitive PageRank [H02]
    - compute many PageRank vectors, one for each topic
    - estimate relevance of query with each topic
    - produce final PageRank as a weighted combination
- Updating PageRank [Chien et al 2002]
- Fast computation of PageRank
  - numerical analysis tricks
  - node aggregation techniques
  - dealing with the “Web frontier”

Hubs and Authorities [K98]

- Authority is not necessarily transferred directly between authorities
- Pages have double identity
  - hub identity
  - authority identity
- Good hubs point to good authorities
- Good authorities are pointed by good hubs



HITS Algorithm

- Initialize all weights to 1.
- Repeat until convergence
  - *O* operation : hubs collect the weight of the authorities

$$h_i = \sum_{j: i \to j} a_j$$

  - *I* operation: authorities collect the weight of the hubs

$$a_i = \sum_{j: j \to i} h_j$$

  - Normalize weights under some norm

HITS and eigenvectors

- The HITS algorithm is a power-method eigenvector computation
  - in vector terms $a^t = A^T h^{t-1}$ and $h^t = A a^{t-1}$
  - so $a = A^T A a^{t-1}$ and $h^t = A A^T h^{t-1}$
  - The authority weight vector a is the eigenvector of $A^T A$ and the hub weight vector h is the eigenvector of $A A^T$
  - Why do we need normalization?
- The vectors a and h are singular vectors of the matrix A

Singular Value Decomposition

$$A = U \quad \Sigma \quad V^{T} = \begin{bmatrix} \overset{\circ}{u_1} & \overset{\circ}{u_2} & \mathrm{i} & \overset{\circ}{u_r} \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \mathrm{l} & \\ & & & \sigma_r \end{bmatrix} \begin{bmatrix} \underset{\circ}{v_1} \\ v_2 \\ \underset{\circ}{\mathrm{j}} \\ v_r \end{bmatrix}$$

$$[n\times r]\,[r\times r]\,[r\times n]$$

§ r : rank of matrix A

§ $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$: singular values (square roots of eig-vals AA^T, A^TA)

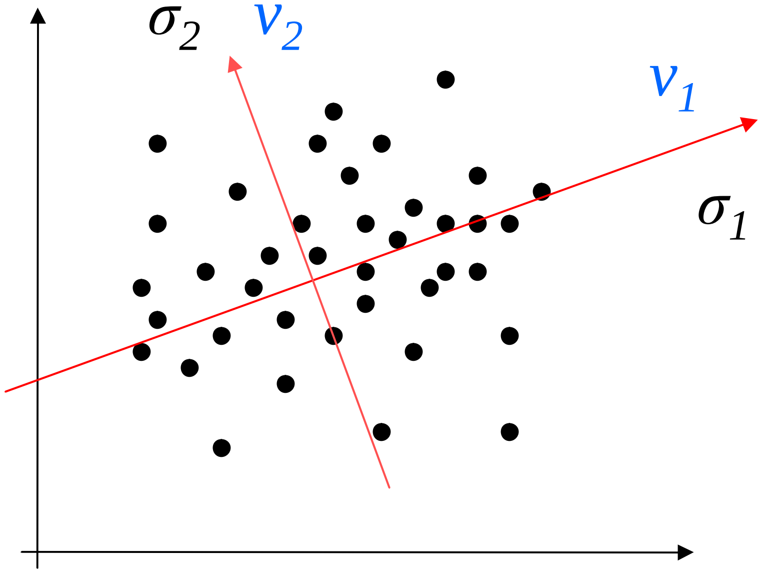
§ $u_1, u_2, \mathrm{i}\ , u_r$: left singular vectors (eig-vectors of AA^T)

§ $v_1, v_2, \mathrm{i}\ , v_r$: right singular vectors (eig-vectors of A^TA)

§ $$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \mathrm{i} + \sigma_r u_r v_r^T$$

Singular Value Decomposition

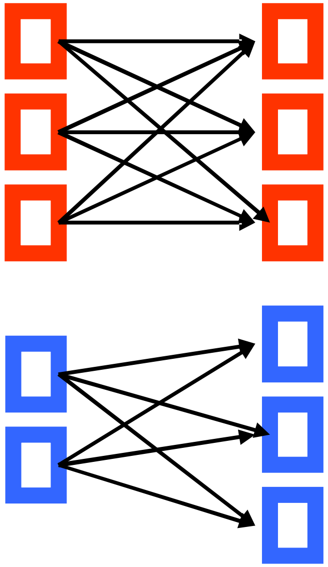
- Linear trend v in matrix A:
  - the tendency of the row vectors of A to align with vector v
  - strength of the linear trend: Av
- SVD discovers the linear trends in the data
- u_i , v_i : the i-th strongest linear trends
- σ_i : the strength of the i-th strongest linear trend



- HITS discovers the strongest linear trend in the authority space

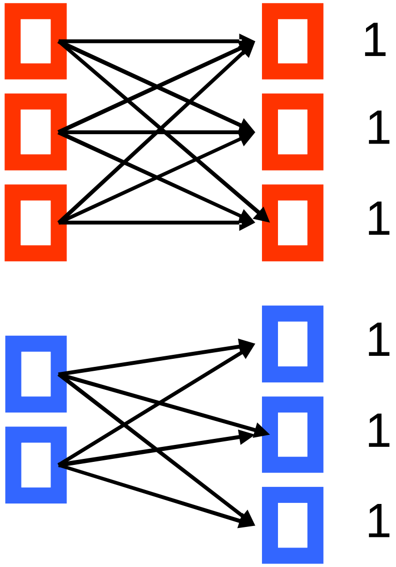
HITS and the TKC effect

- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect



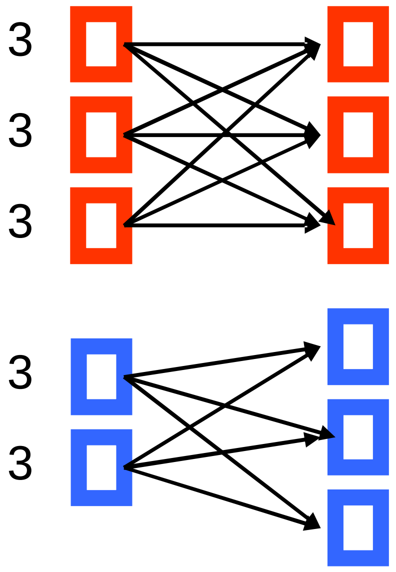
HITS and the TKC effect

- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect



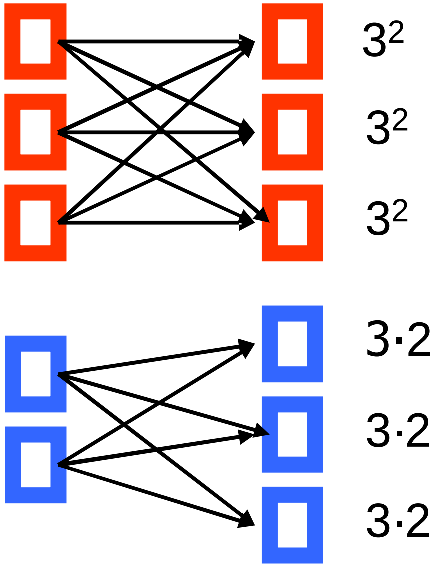
HITS and the TKC effect

- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect



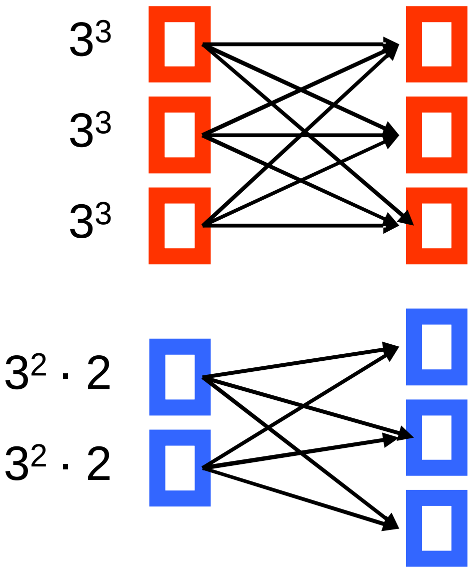
HITS and the TKC effect

- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect



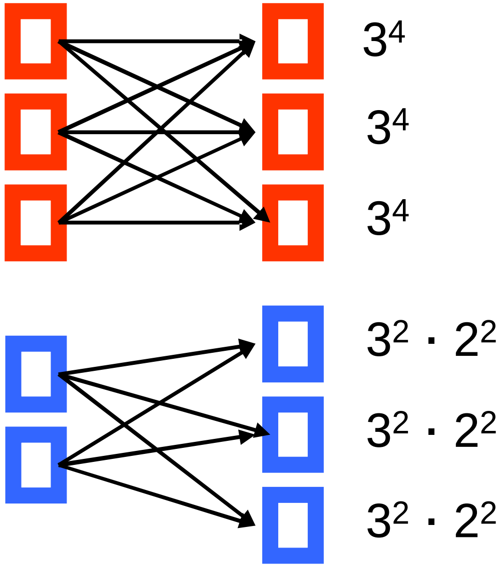
HITS and the TKC effect

- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect



HITS and the TKC effect

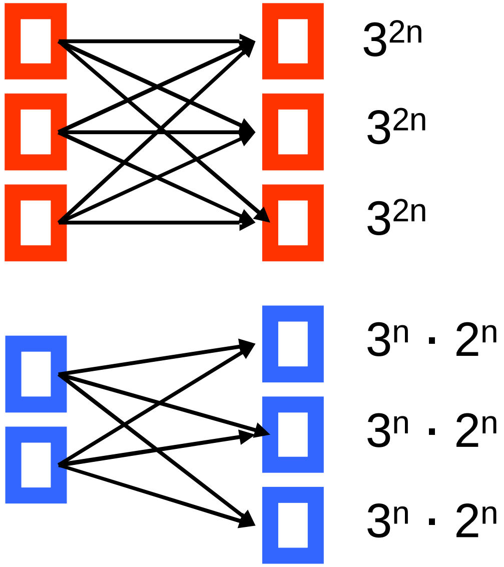
- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect



HITS and the TKC effect

- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect

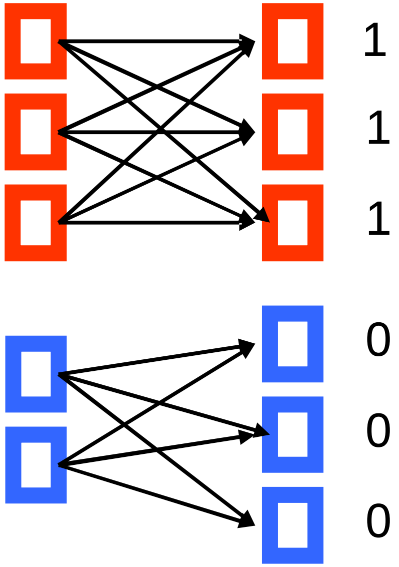
weight of node p is proportional to the number of $(BF)^n$ paths that leave node p



after n iterations

HITS and the TKC effect

- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect



after normalization with the max element as n → ∞

Outline

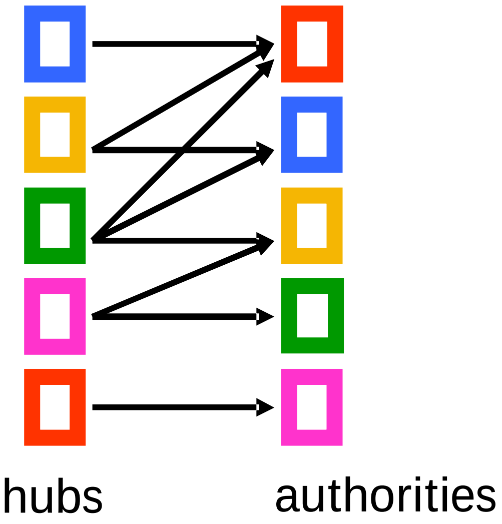
- previous work
- ...in the beginning...
- some more algorithms
- some experimental data
- a theoretical framework

Combining link and text analysis [BH98]

- Problems with HITS
  - multiple links from or to a single host
    - view them as one node and normalize the weight of edges to sum to 1
  - topic drift: many unrelated pages
    - prune pages that are not related to the topic
    - weight the edges of the graph according the relevance of the source and destination
- Other approaches?

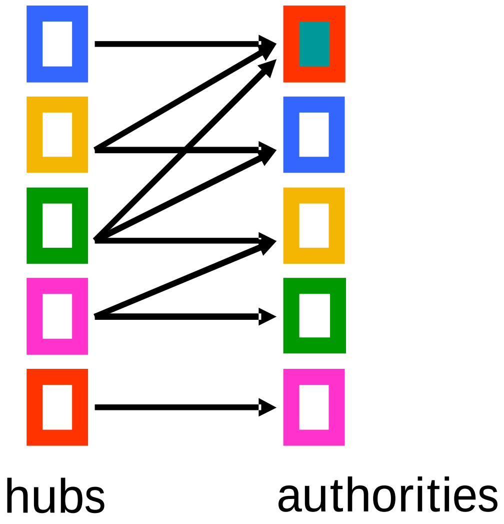
The SALSA algorithm [LM00]

- Perform a random walk alternating between hubs and authorities



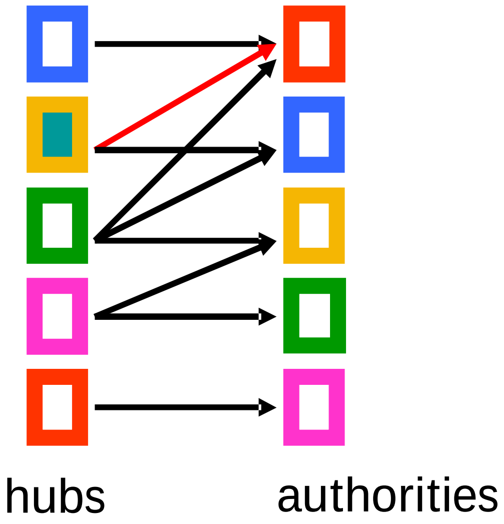
The SALSA algorithm [LM00]

- Start from an authority chosen uniformly at random
  - e.g. the red authority



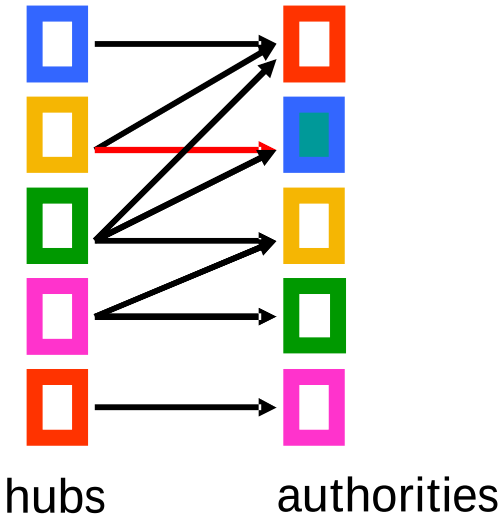
The SALSA algorithm [LM00]

- Start from an authority chosen uniformly at random
  - e.g. the red authority
- Choose one of the in-coming links uniformly at random and move to a hub
  - e.g. move to the yellow authority with probability 1/3



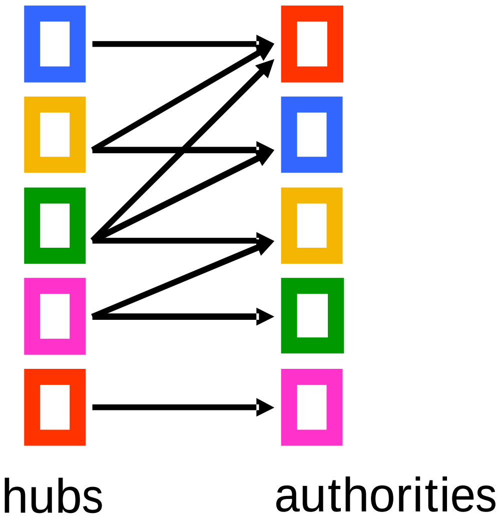
The SALSA algorithm [LM00]

- Start from an authority chosen uniformly at random
  - e.g. the red authority
- Choose one of the in-coming links uniformly at random and move to a hub
  - e.g. move to the yellow authority with probability 1/3
- Choose one of the out-going links uniformly at random and move to an authority
  - e.g. move to the blue authority with probability 1/2



The SALSA algorithm [LM00]

- In matrix terms
  - A_c = the matrix A where columns are normalized to sum to 1
  - A_r = the matrix A where rows are normalized to sum to 1
  - p = the probability state vector
- The first step computes
  - $y = A_c p$
- The second step computes
  - $p = A_r^T y = A_r^T A_c p$
- In MC terms the transition matrix
  - $P = A_r A_c^T$

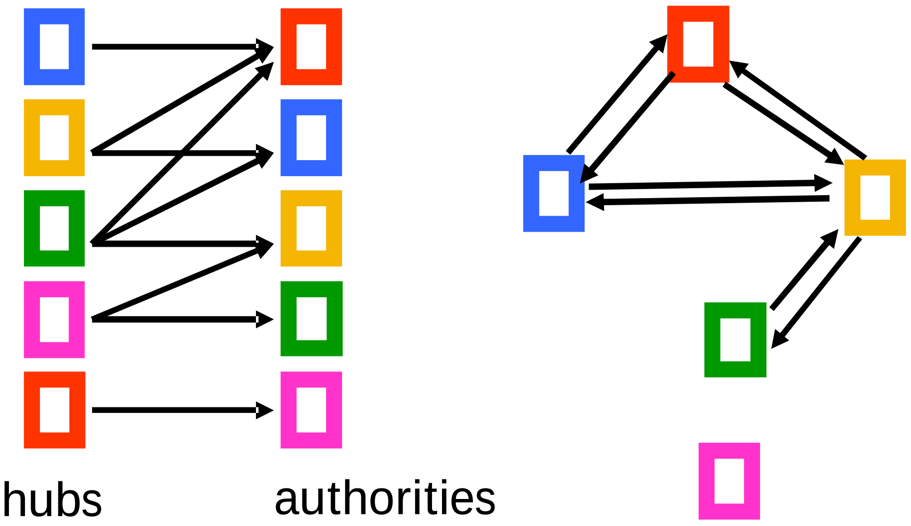


$$y_2 = 1/3\ p_1 + 1/2\ p_2$$

$$p_1 = y_1 + 1/2\ y_2 + 1/3\ y_3$$

The SALSA algorithm [LM00]

- The SALSA performs a random walk on the authority (right) part of the bipartite graph
  - There is a transition between two authorities if there is a BF path between them



$$P(i, j) = \sum_{\substack{k:k \to j \\ i \to k}} \frac{1}{in(i)} \frac{1}{out(k)}$$

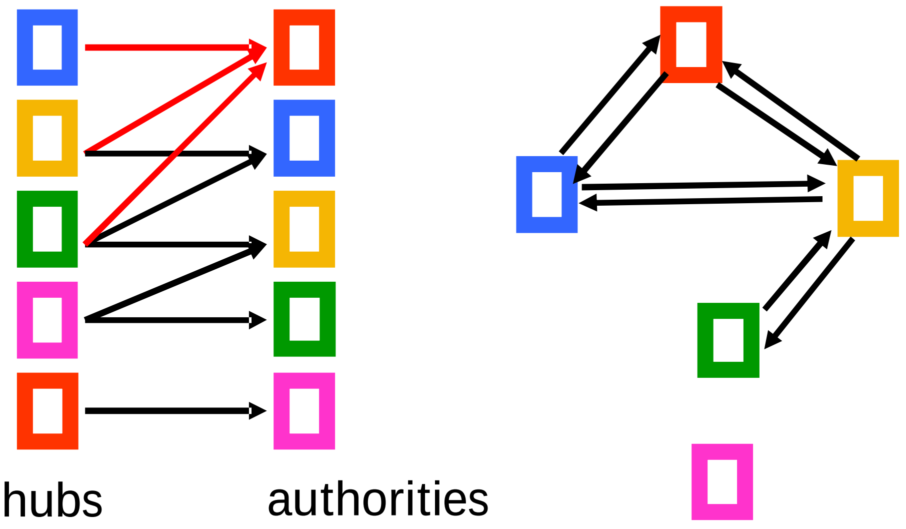
The SALSA algorithm [LM00]

- Stationary distribution of SALSA
  - authority weight of node i =

    fraction of authorities in the hub-authority community of i

    ×

    fraction of links in the community that point to node i
  - Reduces to InDegree for single community graphs

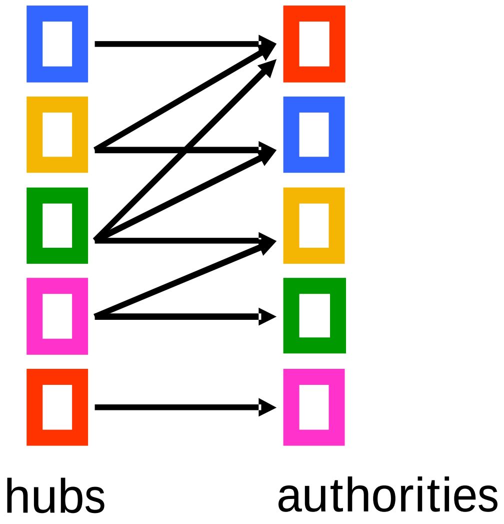


w = 4/5 × 3/8

w = 1/5 × 1

The BFS algorithm [BRRT01]

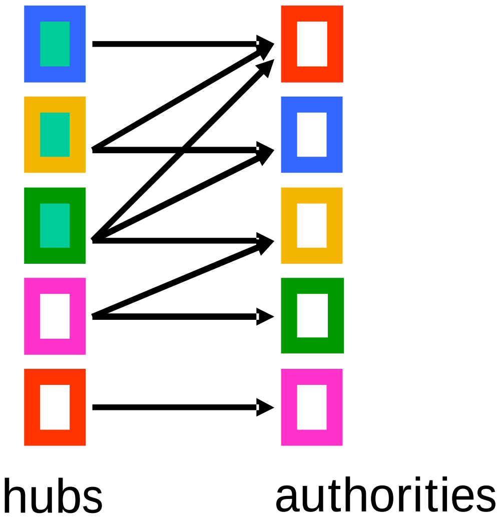
- Rank a node according to the reachability of the node
- Create the neighborhood by alternating between Back and Forward steps
- Apply exponentially decreasing weight as you move further away



w =

The BFS algorithm [BRRT01]

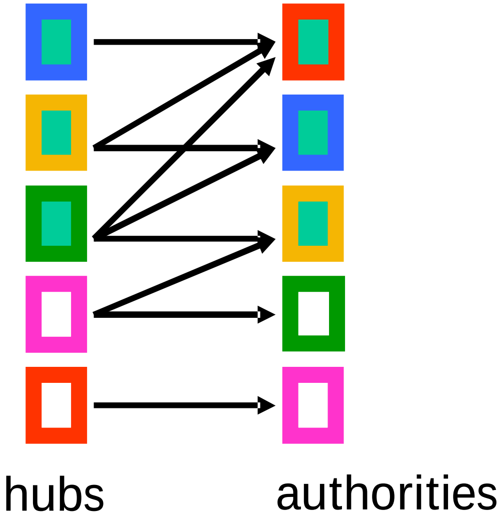
- Rank a node according to the reachability of the node
- Create the neighborhood by alternating between Back and Forward steps
- Apply exponentially decreasing weight as you move further away



w = 3*1

The BFS algorithm [BRRT01]

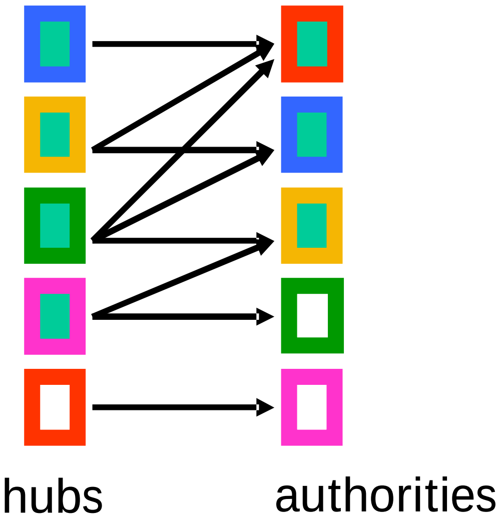
- Rank a node according to the reachability of the node
- Create the neighborhood by alternating between Back and Forward steps
- Apply exponentially decreasing weight as you move further away



w = 3+(1/2)*0

The BFS algorithm [BRRT01]

- Rank a node according to the reachability of the node
- Create the neighborhood by alternating between Back and Forward steps
- Apply exponentially decreasing weight as you move further away

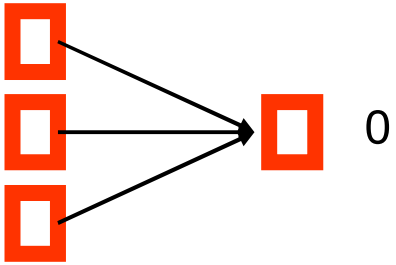


w = 3 +(1/4)*1

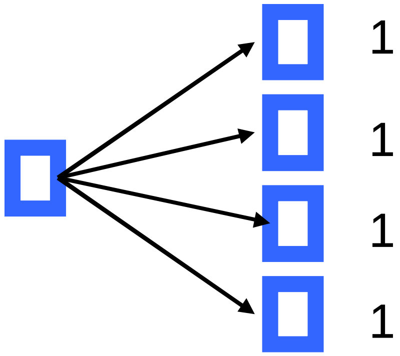
Implicit properties of the HITS algorithm

- Symmetry
  - both hub and authority weights are defined in the same way (through the sum operator)
  - reversing the links, swaps values
- Equality
  - the sum operator assumes that all weights are equally important

A bad example



- The red authority seems better than the blue authorities.
  - quantity becomes quality



- Is the hub quality the same as the authority quality?
  - asymmetric definitions
  - preferential treatment

Authority Threshold AT(k) algorithm

- Small authority weights should not contribute to the computation of the hub weights

- Repeat until convergence
  - *O* operation : hubs collect the k highest authority weights

    $$h_i = \sum_{j:i\to j} a_j : a_j \in F_k(i)$$

  - *I* operation: authorities collect the weight of hubs

    $$a_i = \sum_{j:j\to i} h_j$$

  - Normalize weights under some norm

Norm(p) algorithm

- Small authority weights should contribute less to the computation of the hub weights

- Repeat until convergence
  - *O* operation : hubs compute the p-norm of the authority weight vector

$$h_i = \left(\sum_{j:i\to j} a_j^{\;p} \right)^{1/p} = \left\| \overrightarrow{F(i)} \right\|_p$$

  - *I* operation: authorities collect the weight of hubs

$$a_i = \sum_{j:j\to i} h_j$$

  - Normalize weights under some norm

The MAX algorithm

- A hub is as good as the best authority it points to

- Repeat until convergence
  - *O* operation : hubs collect the highest authority weight

$$h_i = \max_{j:i\to j} a_j$$

  - *I* operation: authorities collect the weight of hubs

$$a_i = \sum_{j:j\to i} h_j$$

  - Normalize weights under some norm

- Special case of AT(k) (for k=1) and Norm(p) (p=∞)

Dynamical Systems

- Discrete Dynamical System: The repeated application of a function g on a set of weights

```
Initialize weights to $w^0$
For $t=1,2,\ldots$
    $w^t=g(w^{t-1})$
```

- LAR algorithms: the function g propagates the weight on the graph G
- Linear vs Non-Linear dynamical systems
  - eigenvector analysis algorithms (PageRank, HITS) are linear dynamical systems
  - AT(k), Norm(p) and MAX are non-linear

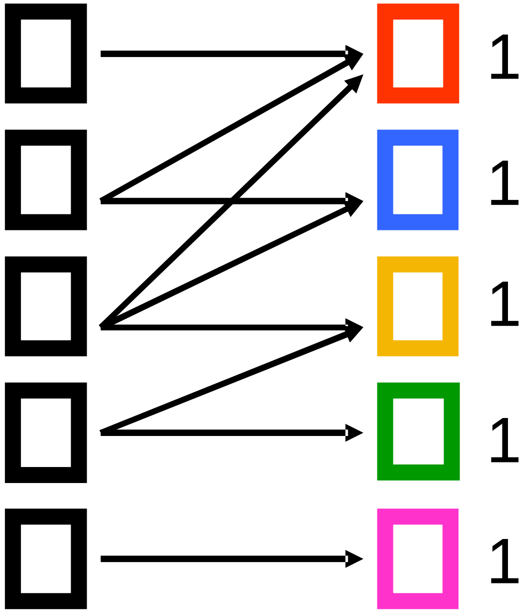
Non-Linear dynamical systems

- Notoriously hard to analyze not well understood
  - we cannot easily prove convergence
  - we do not know much about stationary weights
- Convergence is important for an LAR algorithm to be well defined.

- The MAX algorithm converges for any initial configuration

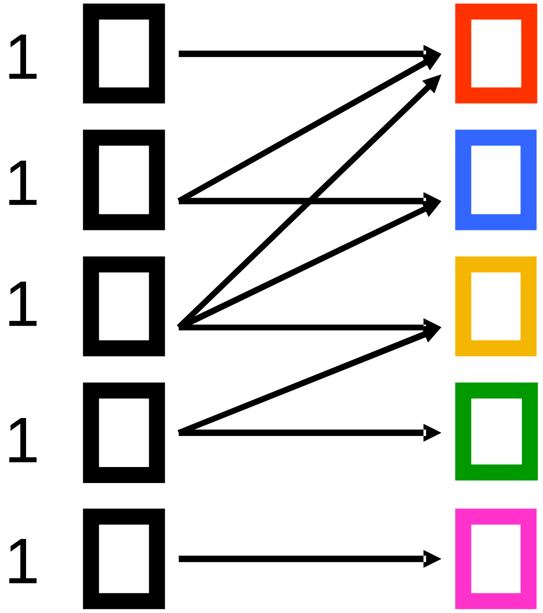
The stationary weights of MAX

- The node with the highest in-degree (seed node) receives maximum weight



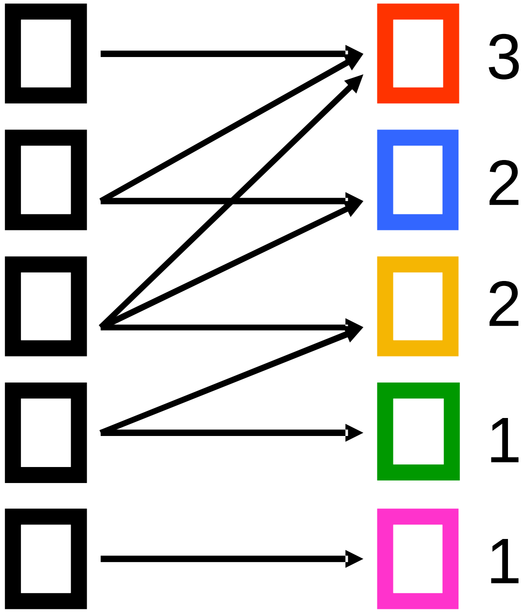
The stationary weights of MAX

- The node with the highest in-degree (seed node) receives maximum weight



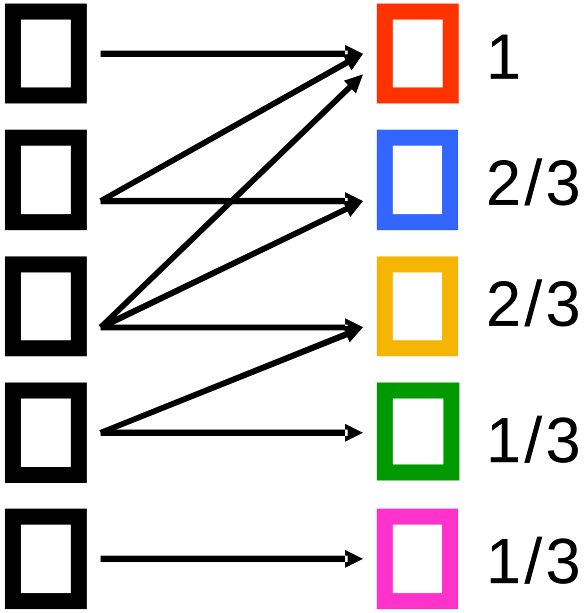
The stationary weights of MAX

- The node with the highest in-degree (seed node) receives maximum weight



The stationary weights of MAX

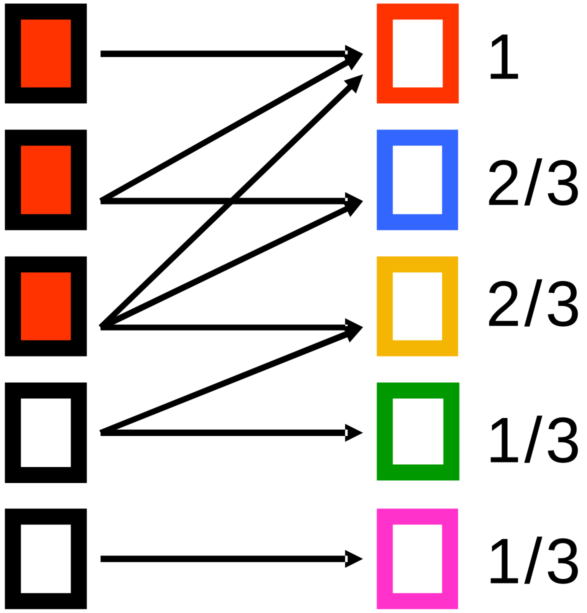
- The node with the highest in-degree (seed node) receives maximum weight



after normalization
with the max weight

The stationary weights of MAX

- The node with the highest in-degree (seed node) receives maximum weight



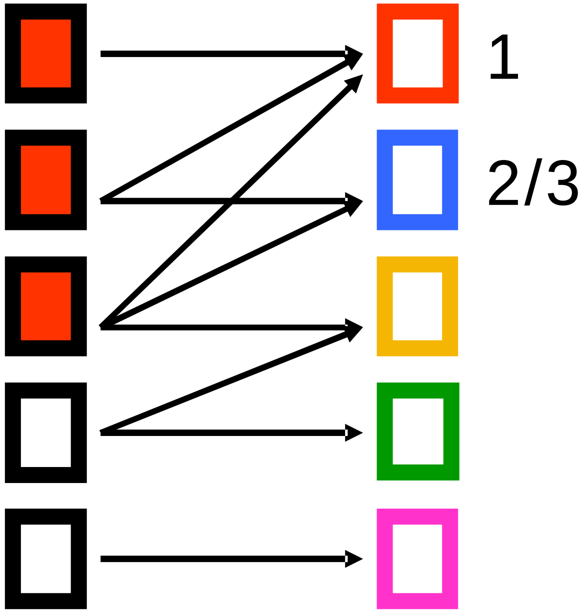
The hubs are mapped to the seed node

before normalization w=3
after normalization with the max weight w=1

normalization factor = 3

The stationary weights of MAX

- The weights of the non-seed nodes depend on their relation with the seed node

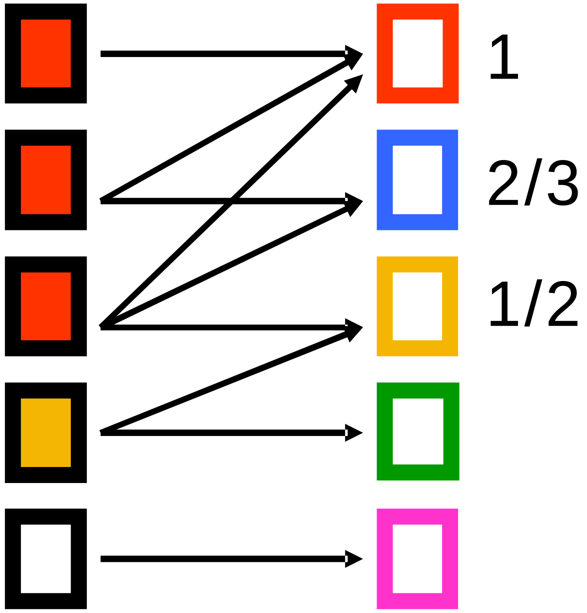


weight of blue node

$w = 2w/3 = 2/3$

The stationary weights of MAX

- The weights of the non-seed nodes depend on their relation with the seed node



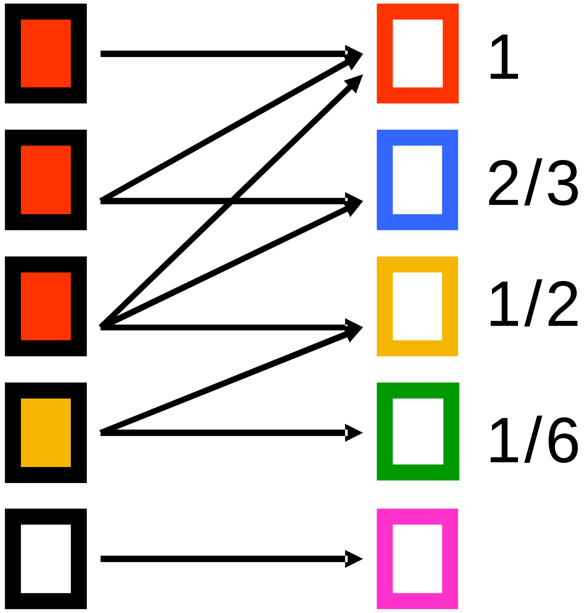
weight of yellow node

$w = (1+ w)/3$

$w = 1/2$

The stationary weights of MAX

- The weights of the non-seed nodes depend on their relation with the seed node



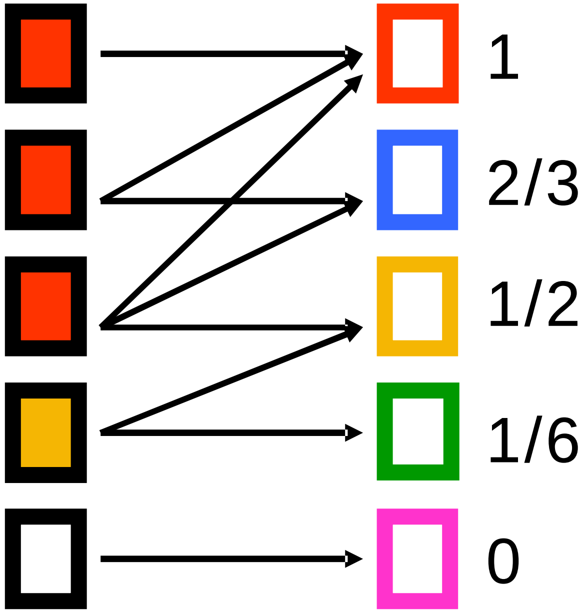
weight of green node

$w = w/3$

$w = 1/6$

The stationary weights of MAX

- The weights of the non-seed nodes depend on their relation with the seed node



weight of purple node

w = 0

Outline

§ ...in the beginning...

§ previous work

§ some more algorithms

§ some experimental data

§ a theoretical framework

Some experimental results

- 34 different queries
- user relevance feedback
  - high relevant/relevant/non-relevant
- measures of interest
  - “high relevance ratio”
  - “relevance ratio”
- Data (and code?) available at

http://www.cs.toronto.edu/~tsap/experiments/journal (or /thesis)

Aggregate Statistics

	AVG HR	STDEV HR	AVG R	STDEV R
HITS	22%	24%	45%	39%
PageRank	24%	14%	46%	20%
In-Degree	35%	22%	58%	29%
SALSA	35%	21%	59%	28%
MAX	38%	25%	64%	32%
BFS	43%	18%	73%	19%

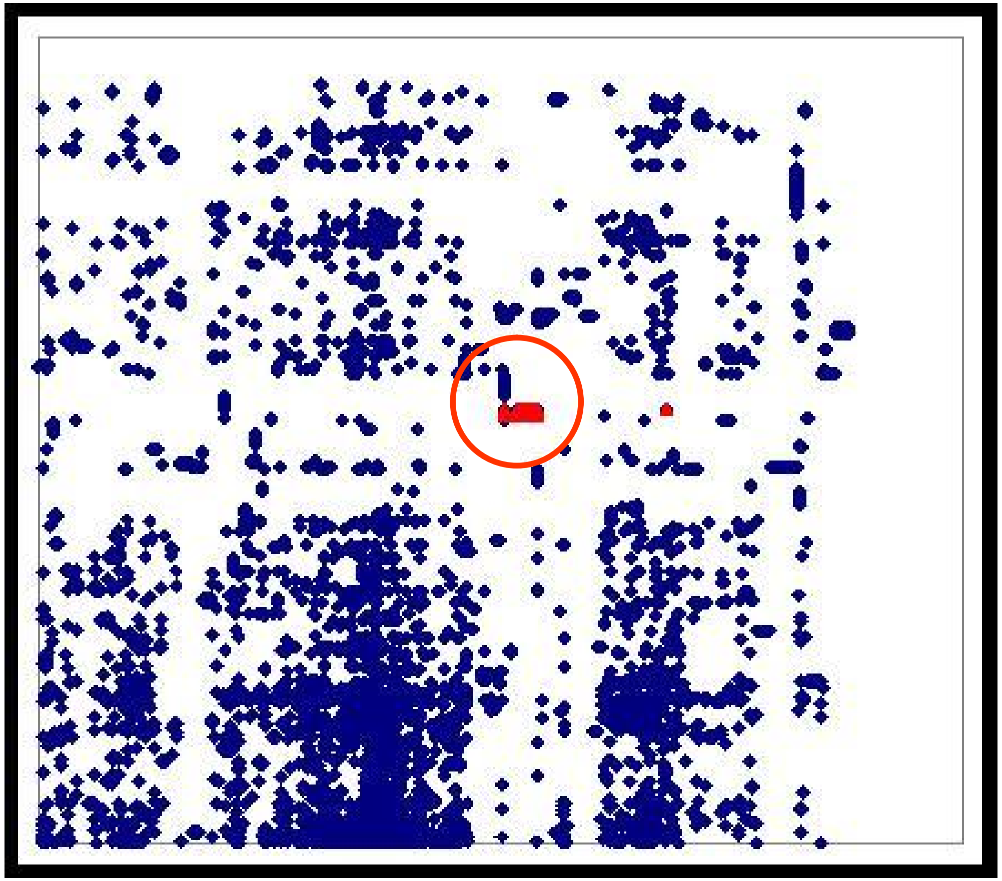
Aggregate Statistics

	AVG HR	STDEV HR	AVG R	STDEV R
HITS	22%	24%	45%	39%
PageRank	24%	14%	46%	20%
In-Degree	35%	22%	58%	29%
SALSA	35%	21%	59%	28%
MAX	38%	25%	64%	32%
BFS	43%	18%	73%	19%

Aggregate Statistics

	AVG HR	STDEV HR	AVG R	STDEV R
HITS	22%	24%	45%	39%
PageRank	24%	14%	46%	20%
In-Degree	35%	22%	58%	29%
SALSA	35%	21%	59%	28%
MAX	38%	25%	64%	32%
BFS	43%	18%	73%	19%

HITS and the TKC effect



“recipes”

- 1. (1.000) HonoluluAdvertiser.com
  URL: http://www.hawaiisclassifieds.com
- 2. (0.999) Gannett Company, Inc.
  URL: http://www.gannett.com
- 3. (0.998) AP MoneyWire
  URL: http://apmoneywire.mm.ap.org
- 4. (0.990) e.thePeople : Honolulu Advertiser
  URL: http://www.e-thepeople.com/
- 5. (0.989) News From The Associated Press
  URL: http://customwire.ap.org/
- 6. (0.987) Honolulu Traffic
  URL: http://www.co.honolulu.hi.us/
- 7. (0.987) News From The Associated Press
  URL: http://customwire.ap.org/
- 8. (0.987) News From The Associated Press
  URL: http://customwire.ap.org/
- 9. (0.987) News From The Associated Press
  URL: http://customwire.ap.org/

10. (0.987) News From The Associated Press
URL: http://customwire.ap.org/

MAX – “net censorship”

- 1. (1.000) EFF: Homepage
  URL: http://www.eff.org
- 2. (0.541) Internet Free Expression Alliance
  URL: http://www.ifea.net
- 3. (0.517) The Center for Democracy and Technology
  URL: http://www.cdt.org
- 4. (0.517) American Civil Liberties Union
  URL: http://www.aclu.org
- 5. (0.386) Vtw Directory Page
  URL: http://www.vtw.org
- 6. (0.357) PEACEFIRE
  URL: http://www.peacefire.org
- 7. (0.277) Global Internet Liberty Campaign Home Page
  URL: http://www.gilc.org
- 8. (0.254) libertus.net: about censorship and free speech
  URL: http://libertus.net
- 9. (0.196) EFF Blue Ribbon Campaign Home Page
  URL: http://www.eff.org/blueribbon.html
- 10. (0.144) The Freedom Forum
  URL: http://www.freedomforum.org

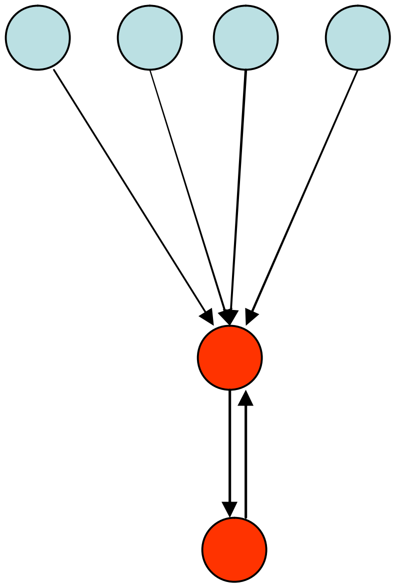
MAX – “affirmative action”

- 1. (1.000) Copyright Information
  URL: http://www.psu.edu/copyright.html
- 2. (0.447) PSU Affirmative Action
  URL: http://www.psu.edu/dept/aaoffice
- 3. (0.314) Welcome to Penn State's Home on the Web
  URL: http://www.psu.edu
- 4. (0.010) University of Illinois
  URL: http://www.uiuc.edu
- 5. (0.009) Purdue University-West Lafayette, Indiana
  URL: http://www.purdue.edu
- 6. (0.008) UC Berkeley home page
  URL: http://www.berkeley.edu
- 7. (0.008) University of Michigan
  URL: http://www.umich.edu
- 8. (0.008) The University of Arizona
  URL: http://www.arizona.edu
- 9. (0.008) The University of Iowa Homepage
  URL: http://www.uiowa.edu
- 10. (0.008) Penn: University of Pennsylvania
  URL: http://www.upenn.edu

PageRank

- 1. (1.000) WCLA Feedback
  URL: http://www.janeylee.com/wcla
- 2. (0.911) Planned Parenthood Action Network
  URL: http://www.ppaction.org/ppaction/
- 3. (0.837) Westchester Coalition for Legal Abortion
  URL: http://www.wcla.org
- 4. (0.714) Planned Parenthood Federation
  URL: http://www.plannedparenthood.org
- 5. (0.633) GeneTree.com Page Not Found
  URL: http://www.qksrv.net/click
- 6. (0.630) Bible.com Prayer Room
  URL: http://www.bibleprayerroom.com
- 7. (0.609) United States Department of Health
  URL: http://www.dhhs.gov

  8. (0.538) Pregnancy Centers Online
  URL: http://www.pregnancycenters.org
- 9. (0.517) Bible.com Online World
  URL: http://bible.com
- 10. (0.516) National Organization for Women
  URL: http://www.now.org



link-spam structure

Outline

- ...in the beginning...
- previous work
- some more algorithms
- some experimental data
- a theoretical framework

Theoretical Analysis of LAR algorithms [BRRT05]

- Why bother?
  - Plethora of LAR algorithms: we need a formal way to compare and analyze them
  - Need to define properties that are useful
    - sensitivity to spam
  - Need to discover the properties that characterize each LAR algorithm

A Theoretical Framework

- A Link Analysis Ranking Algorithm is a function that maps a graph to a real vector

$$A:G_n \rightarrow R^n$$

- G_n : class of graphs of size n
- LAR vector the output $A(G)$ of an algorithm A on a graph G
- G_{n} : the class of all possible graphs of size n

Comparing LAR vectors

🟥 🟦 🟧 🟩 🟪

$$w_1 = [\ 1 \quad 0.8 \quad 0.5 \quad 0.3 \quad 0\ ]$$

$$w_2 = [\ 0.9 \quad 1 \quad 0.7 \quad 0.6 \quad 0.8\ ]$$

- How close are the LAR vectors w_1, w_2?

Distance between LAR vectors

- Geometric distance: how close are the numerical weights of vectors w_1, w_2?

$$d_1(w_1, w_2) = \sum |w_1[i] - w_2[i]|$$

🟥 🟦 🟨 🟩 🟪

w_1 = [1.0 0.8 0.5 0.3 0.0]

w_2 = [0.9 1.0 0.7 0.6 0.8]

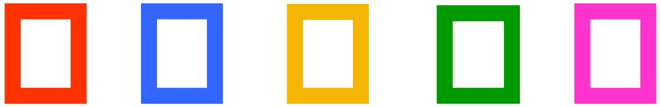
$d_1(w_1,w_2)$ = 0.1+0.2+0.2+0.3+0.8 = 1.6

Distance between LAR vectors

- Rank distance: how close are the ordinal rankings induced by the vectors w_1, w_2?
  - Kendal's τ distance

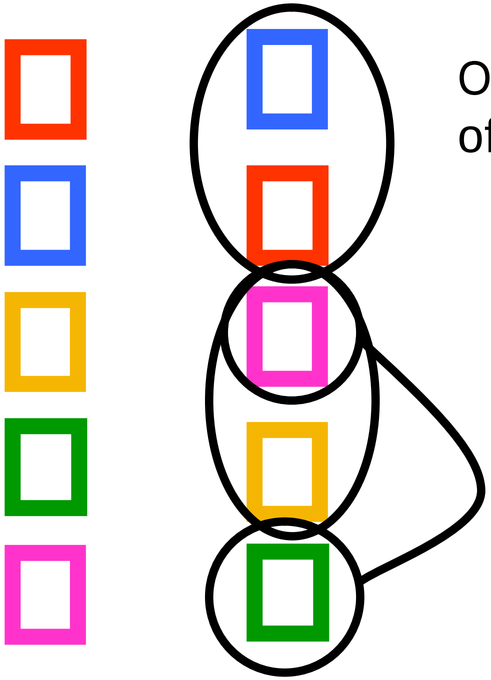
$$d_r(w_1, w_2) = \frac{\text{pairs ranked in a different order}}{\text{total number of distinct pairs}}$$

Rank distance



$$w_1 = [\ 1 \quad 0.8 \quad 0.5 \quad 0.3 \quad 0\ ]$$

$$w_2 = [\ 0.9 \quad 1 \quad 0.7 \quad 0.6 \quad 0.8\ ]$$

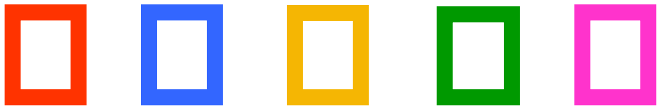


Ordinal Ranking of vector w_1

Ordinal Ranking of vector w_2

$$d_r(w_1, w_2) = \frac{3}{5 * 4/2} = 0.3$$

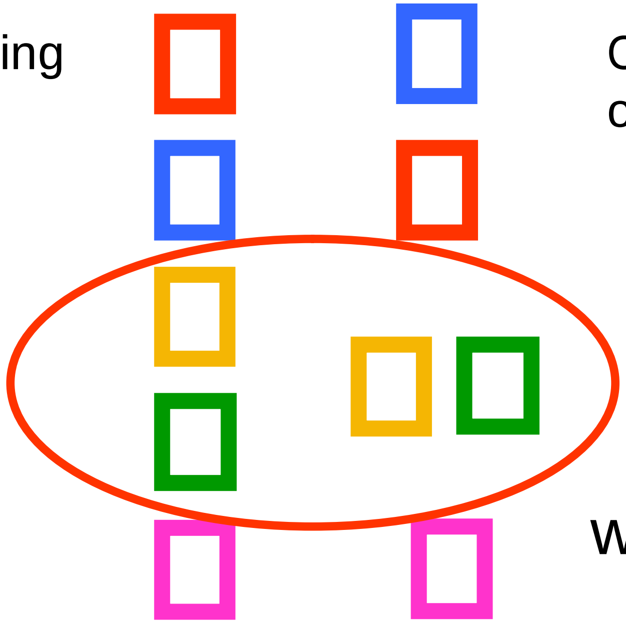
Rank distance of partial rankings



$$w_1 = [\ 1 \quad 0.8 \quad 0.5 \quad 0.3 \quad 0\ ]$$

$$w_2 = [\ 0.9 \quad 1 \quad 0.7 \quad 0.7 \quad 0.3\ ]$$

Ordinal Ranking of vector w_1

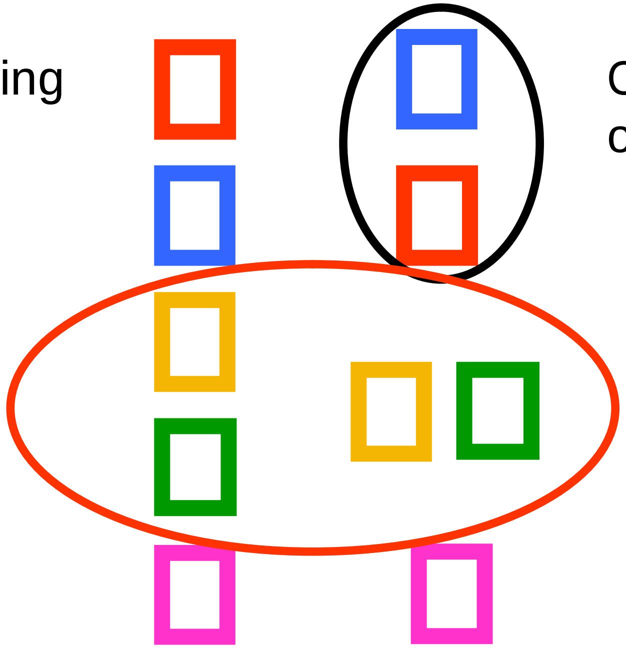


Ordinal Ranking of vector w_2

what do we do with such pairs?

Rank distance of partial rankings

- Charge penalty p for each pair (i,j) of nodes such that $w_1[i] \neq w_1[j]$ and $w_2[i] = w_2[j]$



Ordinal Ranking of vector w_1

Ordinal Ranking of vector w_2

$$d_r(w_1, w_2) = \frac{1+p}{10}$$

Rank distance of partial rankings

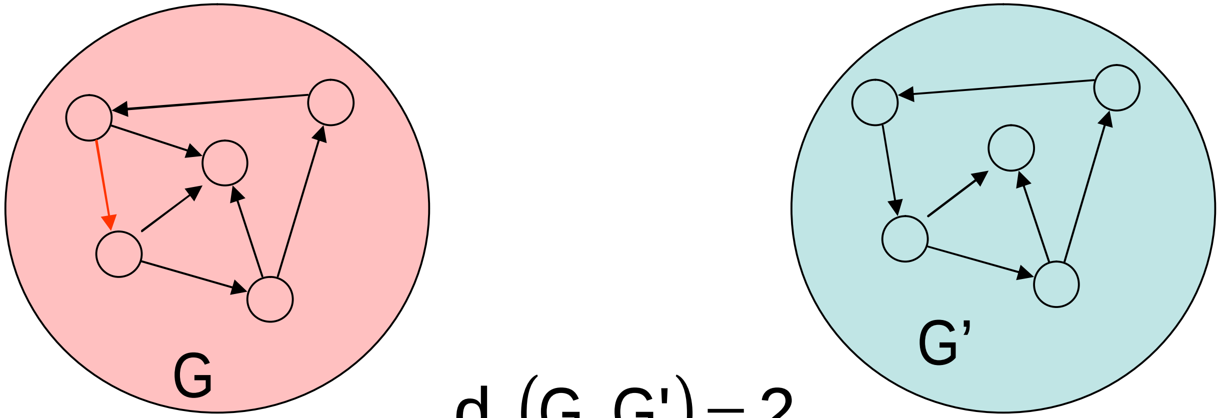
- Extreme value $p = 1$
  - charge for every potential conflict
- Extreme value $p = 0$
  - charge only for inconsistencies
  - problem: not a metric
- Intermediate values $0 < p < 1$
  - Details [FMNKS04] [T04]
  - Interesting case $p = 1/2$

- We will use whatever gives a stronger result

Stability: graph distance

- Intuition: a small change on a graph should cause a small change on the output of the algorithm.
- Definition: Link distance between graphs G=(P,E) and G'=(P,E')

$$d_a(G,G') = |E \cup E'| - |E \cap E'|$$



$$d_a(G,G') = 2$$

Stability

- $C_k(G)$: set of graphs G’ such that $d_\ell(G,G')\le k$

- Definition: Algorithm A is stable if

$$\lim_{n\to\infty}\max_{G}\max_{G'\in C_k(G)} d_1(A(G),A(G'))=0$$

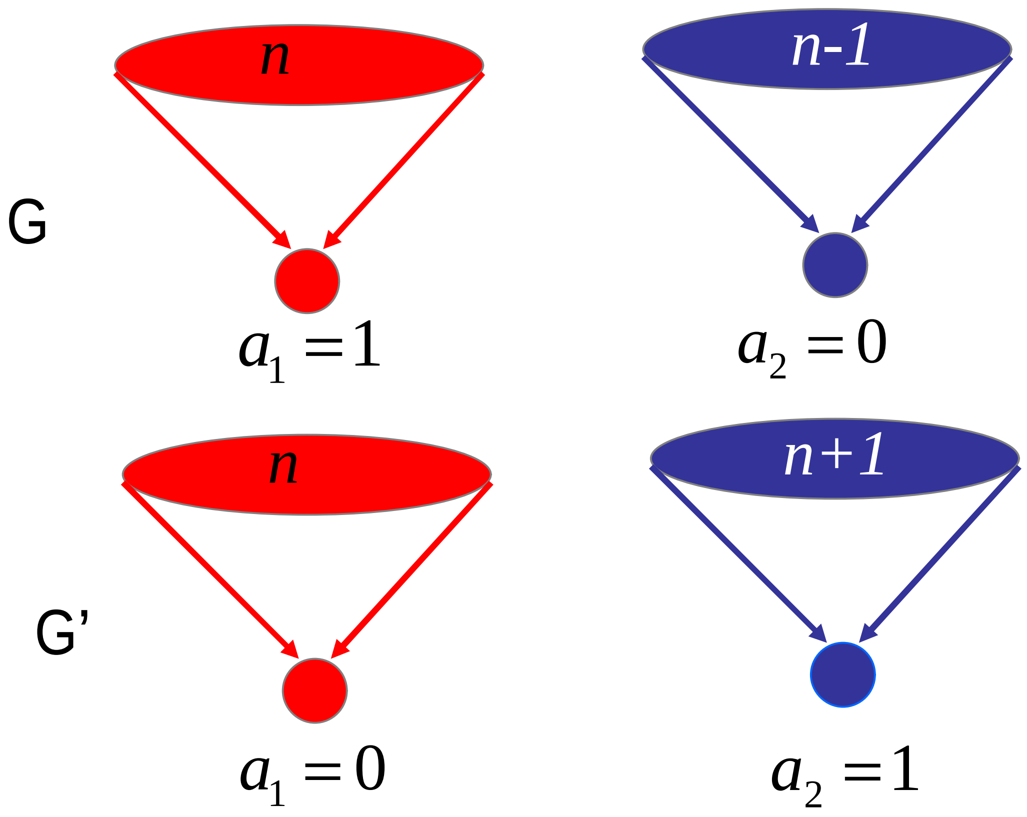
- Definition: Algorithm A is rank stable if

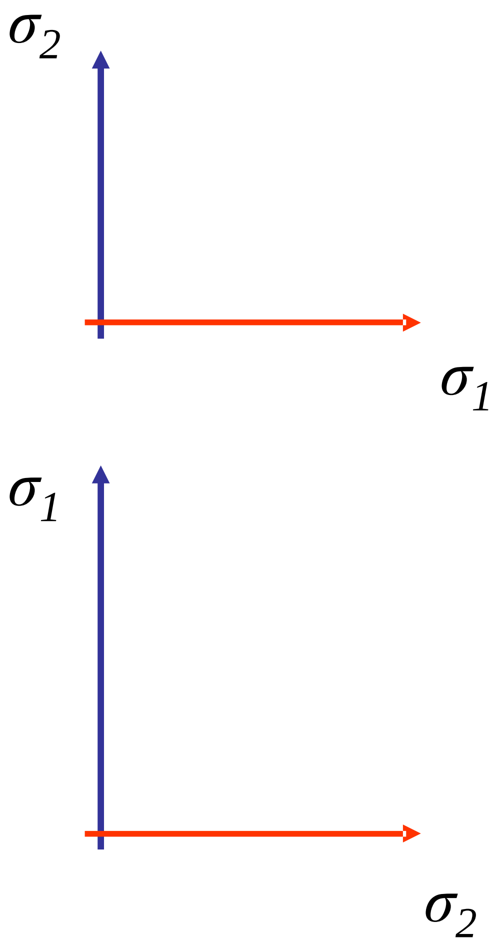
$$\lim_{n\to\infty}\max_{G}\max_{G'\in C_k(G)} d_r\left(A(G),A(G')\right)=0$$

Stability: Results

- InDegree algorithm is stable and rank stable on the class G_n
- HITS, Max are neither stable nor rank stable on the class G_n

Instability of HITS





Eigengap $\sigma_1 - \sigma_2 = 1$

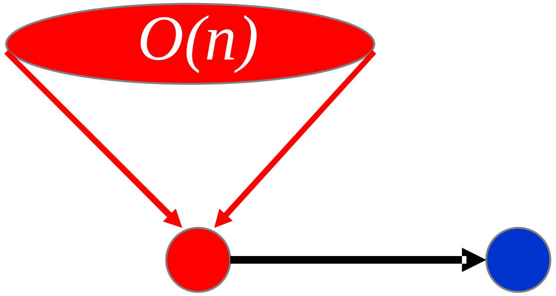
Stability of HITS

- HITS is stable if $\sigma_1-\sigma_2 \to \infty$ [NZJ01]
  - The two strongest linear trends are well separated

- What about the converse?

Instability of PageRank

- PageRank is unstable



- PageRank is rank unstable [Lempel Moran 2003]

Stability of PageRank

- Perturbations to unimportant nodes have small effect on the PageRank values [NZJ01][BGS03]

$$d_1\left(A(G), A(G')\right) \leq \frac{2\alpha}{1-2\alpha} \sum_{i \in P} A(G)[i]$$

Stability of PageRank

- Lee Borodin model [LB03]
    - upper bounds depend on authority and hub values
    - PageRank, Randomized SALSA are stable
    - HITS, SALSA are unstable

- Open question: Can we derive conditions for the stability of PageRank in the general case?

Similarity

- Definition: Two algorithms A_1, A_2 are similar if

$$\lim_{n\to\infty} \frac{\max\limits_{G\in G_n} d_1\left(A_1(G), A_2(G)\right)}{\max\limits_{w_1, w_2} d_1\left(w_1, w_2\right)} = 0$$

- Definition: Two algorithms A_1, A_2 are rank similar if

$$\lim_{n\to\infty} \max_{G\in G_n} d_r\left(A_1(G), A_2(G)\right) = 0$$

- Definition: Two algorithms A_1, A_2 are rank equivalent if

$$\max_{G\in G_n} d_r\left(A_1(G), A_2(G)\right) = 0$$

Similarity: Results

- No pairwise combination of InDegree, SALSA, HITS and MAX algorithms is similar, or rank similar on the class of all possible graphs G_n

Product Graphs

- Latent authority and hub vectors a, h
  - h_i = probability of node i being a good hub
  - a_j = probability of node j being a good authority

- Generate a link i→j with probability $h_i a_j$

$$W[i,j] = \begin{cases} 1 & \text{with probability } h_i a_j \\ 0 & \text{with probability } 1 - h_i a_j \end{cases}$$

  - Azar, Fiat, Karlin, McSherry Saia 2001, Michail, Papadimitriou 2002,Chung, Lu, Vu 2002

- The class of product graphs G_n^p

Similarity on Product Graphs

- Theorem: HITS and InDegree are similar with high probability on the class of product graphs, G_n^p (subject to some assumptions)

References

- [BP98] S. Brin, L. Page, The anatomy of a large scale search engine, WWW 1998
- [K98] J. Kleinberg. Authoritative sources in a hyperlinked environment. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.
- G. Pinski, F. Narin. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. Information Processing and Management, 12(1976), pp. 297--312.
- L. Katz. A new status index derived from sociometric analysis. Psychometrika 18(1953).
- R. Motwani, P. Raghavan, Randomized Algorithms
- S. Kamvar, T. Haveliwala, C. Manning, G. Golub, Extrapolation methods for Accelerating PageRank Computation, WWW2003
- A. Langville, C. Meyer, Deeper Inside PageRank, Internet Mathematics

References

- [BP98] S. Brin, L. Page, The anatomy of a large scale search engine, WWW 1998
- [K98] J. Kleinberg. Authoritative sources in a hyperlinked environment. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.
- [HB98] Monika R. Henzinger and Krishna Bharat. Improved algorithms for topic distillation in a hyperlinked environment. Proceedings of the 21'st International ACM SIGIR Conference on Research and Development in IR, August 1998.
- [BRRT05] A. Borodin, G. Roberts, J. Rosenthal, P. Tsaparas, Link Analysis Ranking: Algorithms, Theory and Experiments, ACM Transactions on Internet Technologies (TOIT), 5(1), 2005
- R. Lempel, S. Moran. The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. 9th International World Wide Web Conference, May 2000.
- A. Y. Ng, A. X. Zheng, and M. I. Jordan. Link analysis, eigenvectors, and stability. International Joint Conference on Artificial Intelligence (IJCAI), 2001.
- A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. 24th International Conference on Research and Development in Information Retrieval (SIGIR 2001).