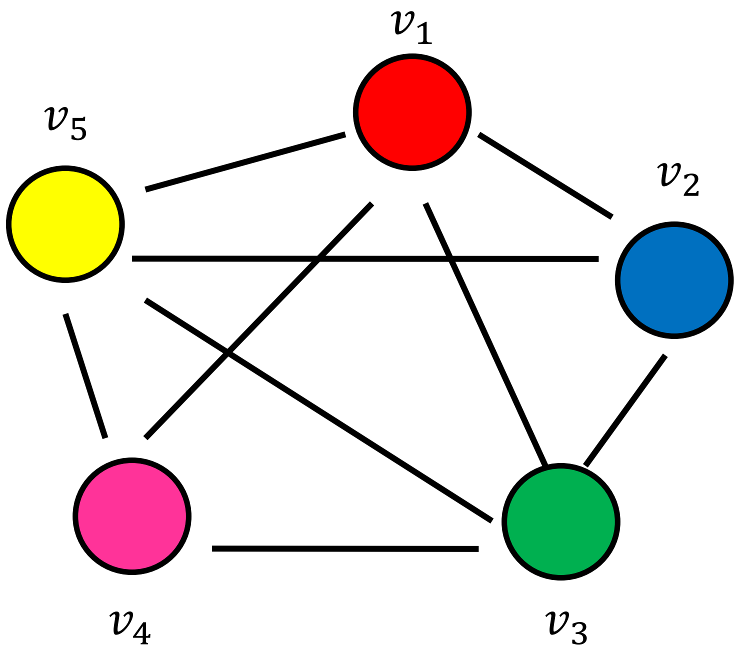
DATA MINING
LECTURE 12

Graphs, Node importance, Link Analysis Ranking, Random walks

RANDOM WALKS AND PAGERANK

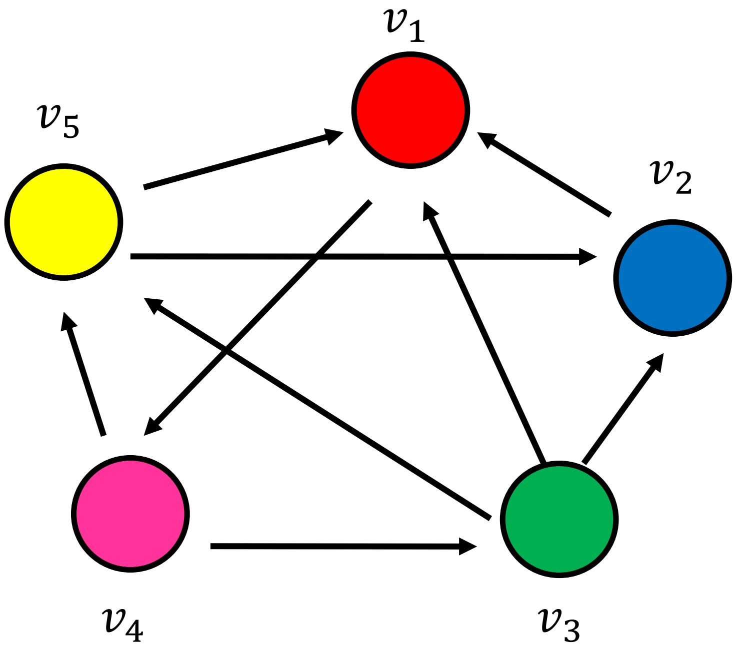
Graphs

- A graph is a powerful abstraction for modeling entities and their pairwise relationships.
- Examples:
  - Social network
  - Collaboration graphs
  - Twitter Followers
  - Web



Graphs

- A graph is a powerful abstraction for modeling entities and their pairwise relationships.
- Examples:
  - Social network
  - Collaboration graphs
  - Twitter Followers
  - Web



Mining the graph structure

- A graph is a combinatorial object, with a certain structure.
- Mining the structure of the graph reveals information about the entities in the graph
  - E.g., if in the Facebook graph I find that there are 100 people that are all linked to each other, then these people are likely to be a community
    - The community discovery problem
  - By measuring the number of friends in the facebook graph I can find the most important nodes
    - The node importance problem
- We will now focus on the node importance problem

Link Analysis

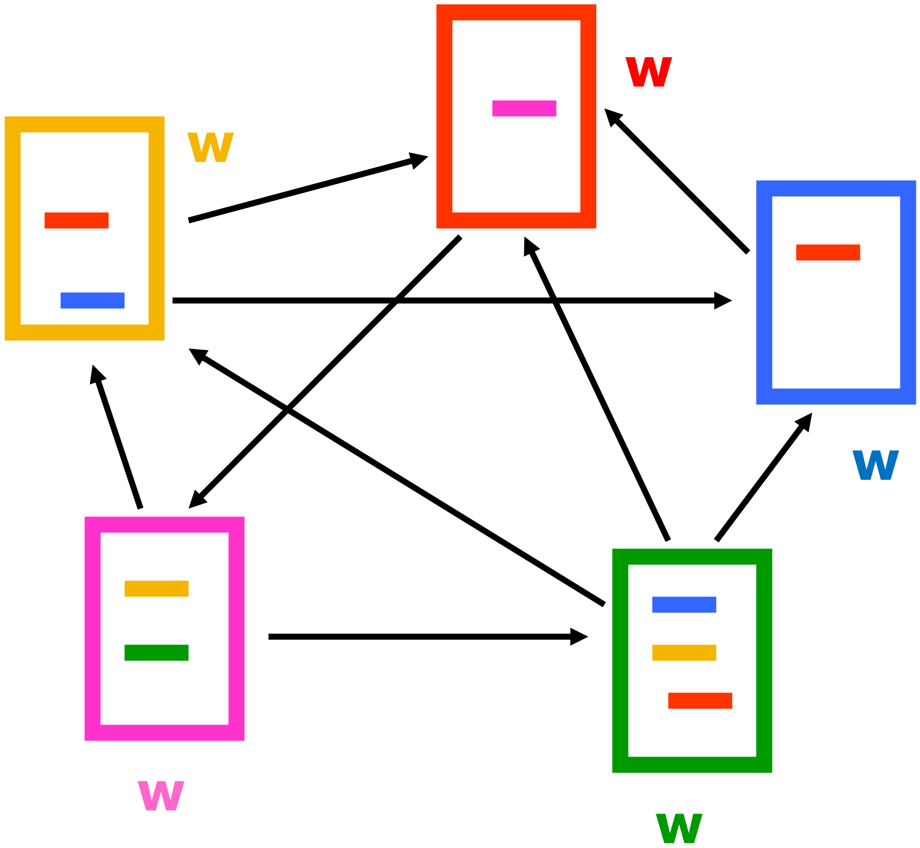
- First generation search engines
  - view documents as flat text files
  - could not cope with size, spamming, user needs
- Second generation search engines
  - Ranking becomes critical
  - shift from relevance to authoritativeness
    - authoritativeness: the static importance of the page
  - use of Web specific data: Link Analysis of the Web graph
  - a success story for the network analysis + a huge commercial success
  - it all started with two graduate students at Stanford

Link Analysis: Intuition

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

- The same idea applies to other graphs as well
  - Twitter graph, where user p follows user q

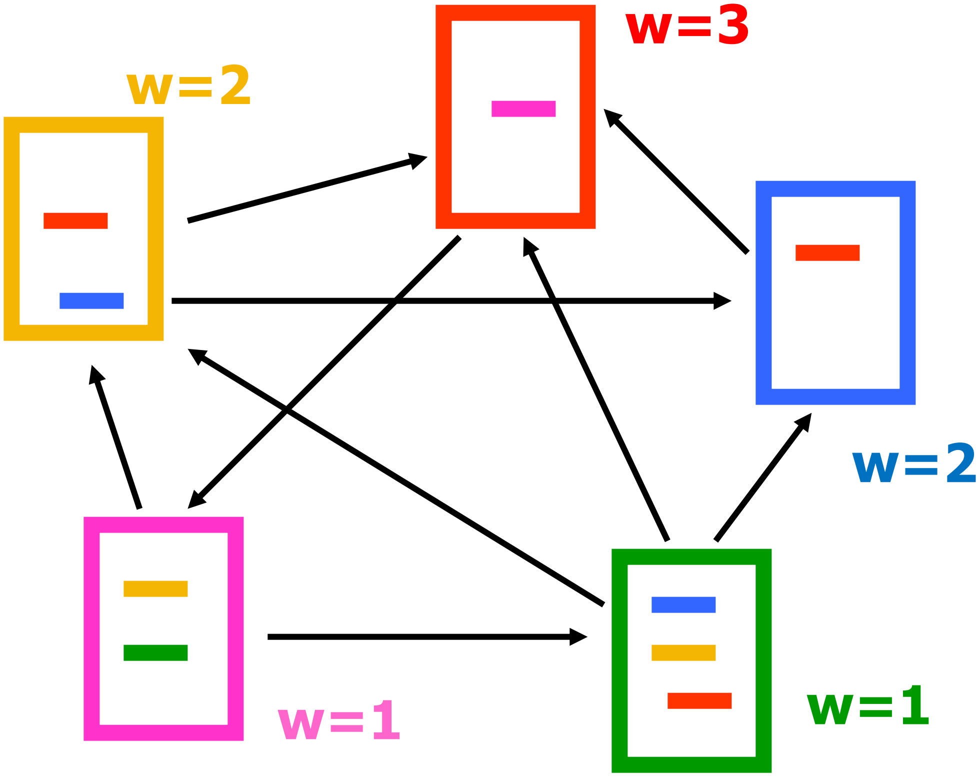
Constructing the graph



- Goal: output an authority weight for each node
  - Also known as centrality, or importance

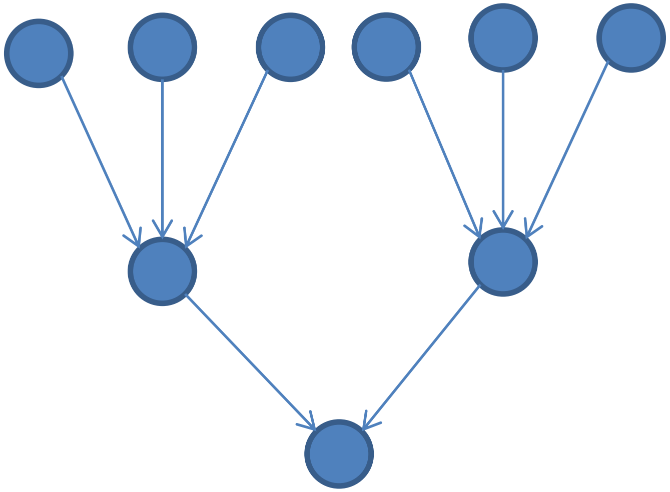
Rank by Popularity

- Rank pages according to the number of incoming edges (in-degree, degree centrality)



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

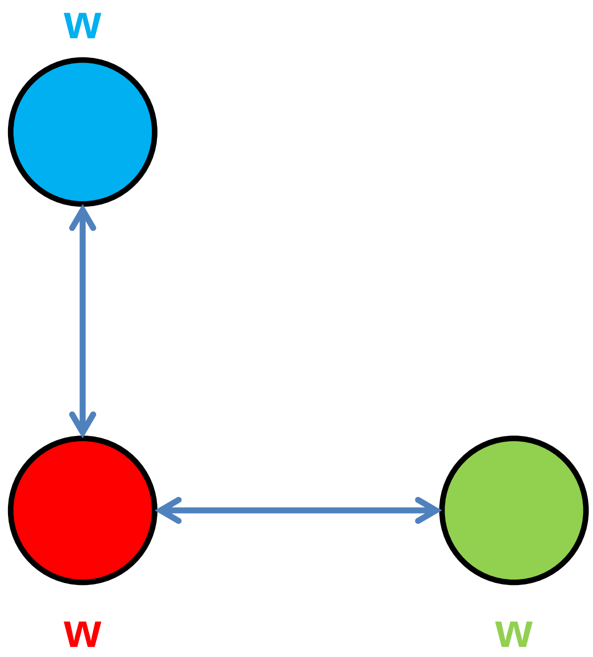
Popularity



- It is not important only how many link to you, but how important are the people that link to you.
- Good authorities are pointed by good authorities
  - Recursive definition of importance

PageRank

- Assume that we have a unity of authority to distribute to all nodes.
- Each node distributes the authority value they have to all their neighbors
- The authority value of each node is the sum of the fractions it collects from its neighbors.
- Solving the system of equations we get the authority values for the nodes
  - w = ½ , w = ¼ , w = ¼



w + w + w = 1

w = w + w

w = ½ w

w = ½ w

A more complex example

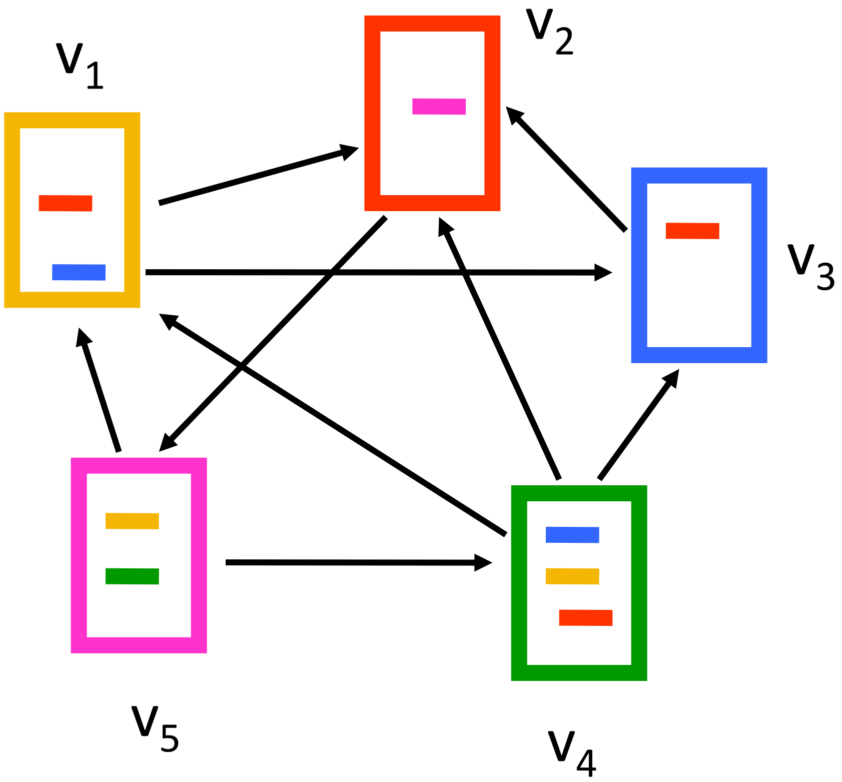
$w_1 = 1/3\ w_4 + 1/2\ w_5$

$w_2 = 1/2\ w_1 + w_3 + 1/3\ w_4$

$w_3 = 1/2\ w_1 + 1/3\ w_4$

$w_4 = 1/2\ w_5$

$w_5 = w_2$



$$PR(p) = \sum_{q \to p} \frac{PR(q)}{|Out(q)|}$$

Random walks on graphs

- The equations above describe a step of a random walk on the graph
  - Random walk: start from some node uniformly at random and then from each node pick a random link to follow.
  - Question: what is the probability of being at a specific node?
    - p_i: probability of being at node i at this step
    - p_i': probability of being at node i in the next step

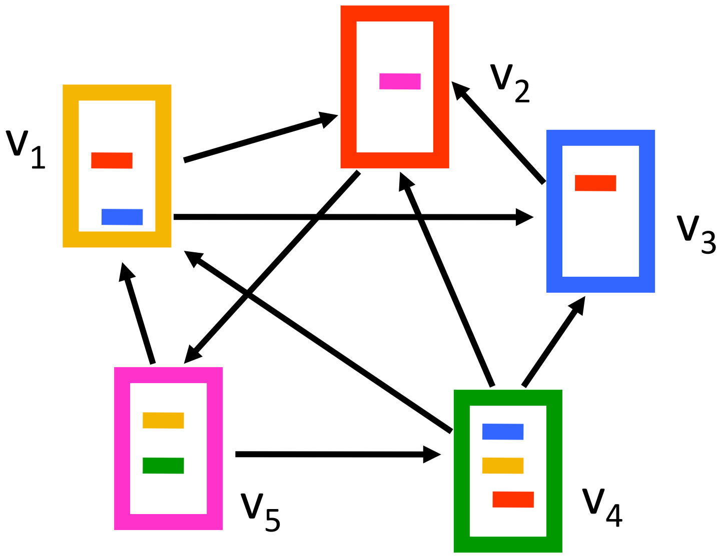
$$p'_1 = 1/3\ p_4 + 1/2\ p_5$$

$$p'_2 = 1/2\ p_1 + p_3 + 1/3\ p_4$$

$$p'_3 = 1/2\ p_1 + 1/3\ p_4$$

$$p'_4 = 1/2\ p_5$$

$$p'_5 = p_2$$

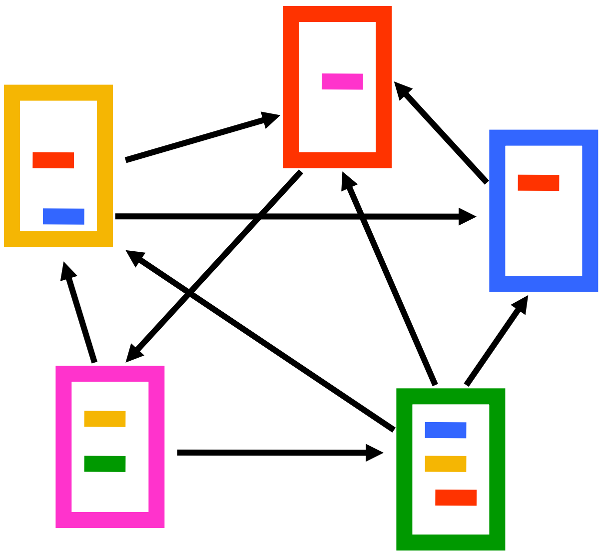


- After many steps the probabilities converge to the stationary distribution of the random walk.

PageRank algorithm [BP98]

- Good authorities should be pointed by good authorities
  - The value of a page is the value of the people that link to you
- How do we implement that?
  - Each page has a value.
  - Proceed in iterations,
    - in each iteration every page distributes the value to the neighbors
  - Continue until there is convergence.

$$PR(p) = \sum_{q \to p} \frac{PR(q)}{|Out(q)|}$$



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

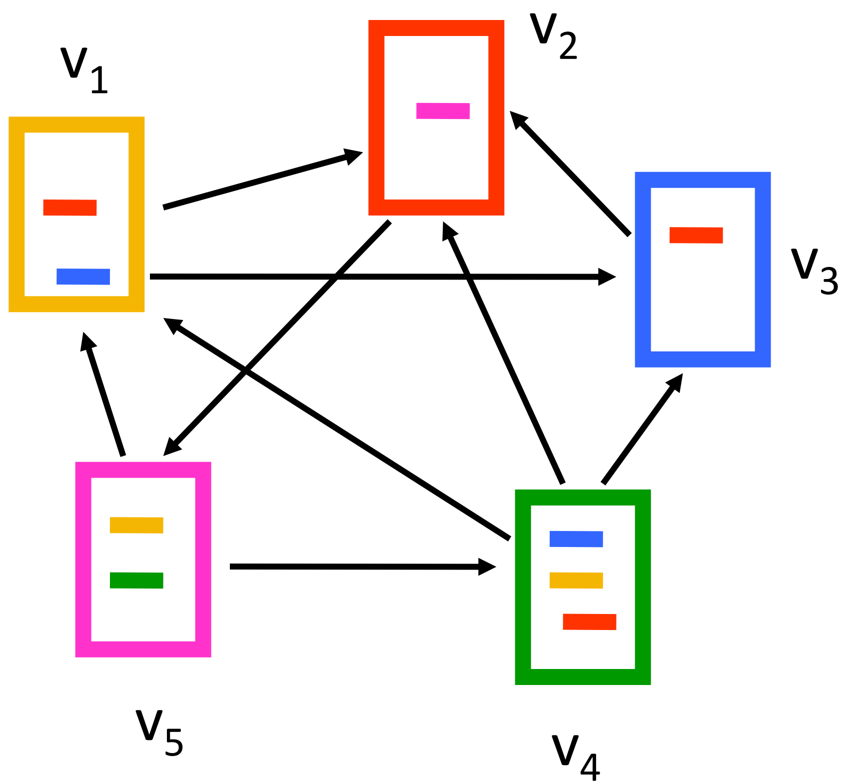
Random walks

- 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

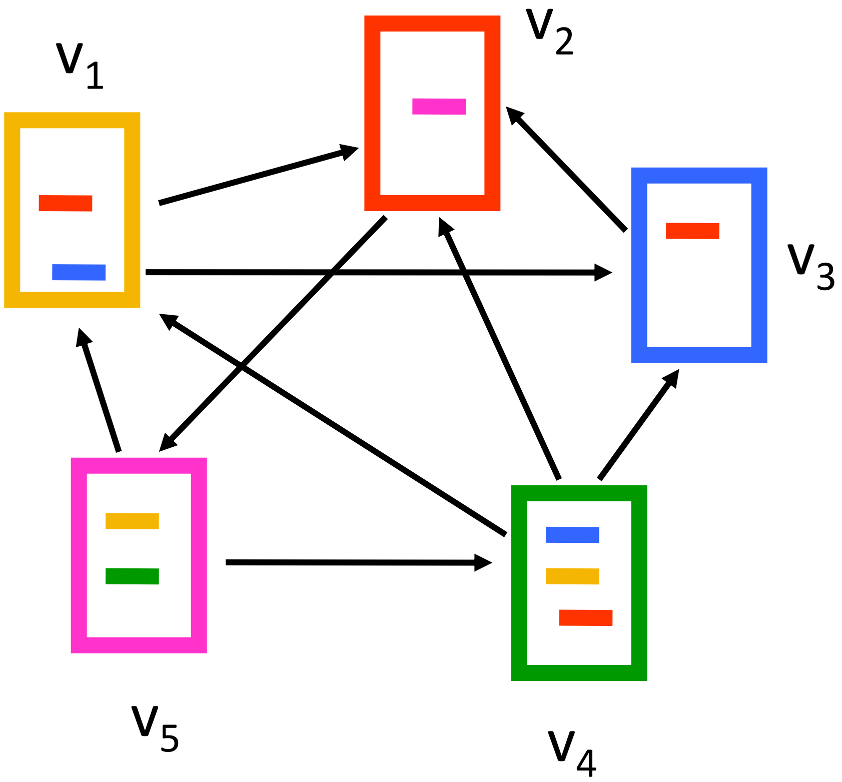
- The vector $q^t = (q^t_1, q^t_2, \ldots, q^t_n)$ that stores the probability of being at state i at time t
  - q^0_i = the probability of starting from state i

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

An example

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

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



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

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

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

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

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

Same equations as before!

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 $t \to \infty$
- The stationary distribution is an eigenvector of matrix P
  - the principal left eigenvector of P – stochastic matrices have maximum eigenvalue 1

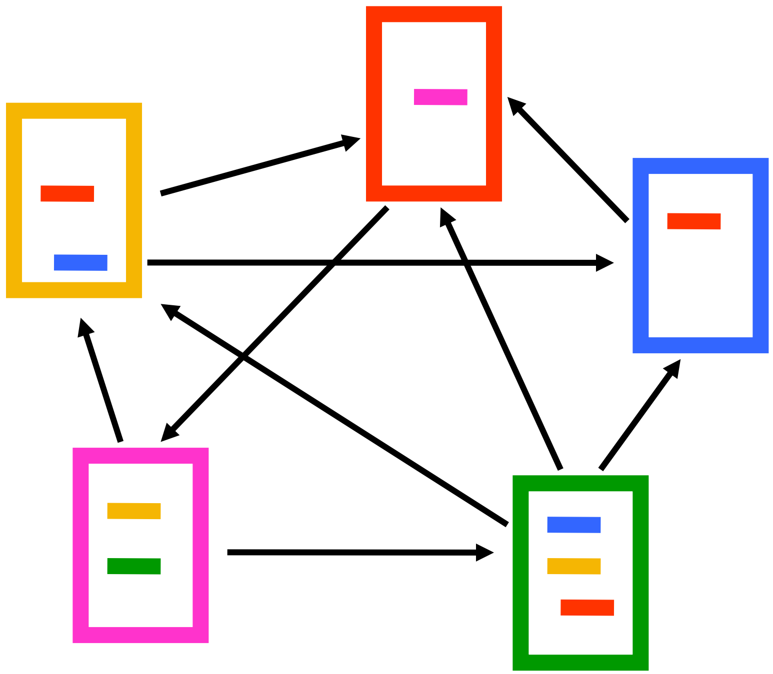
Computing the stationary distribution

- 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

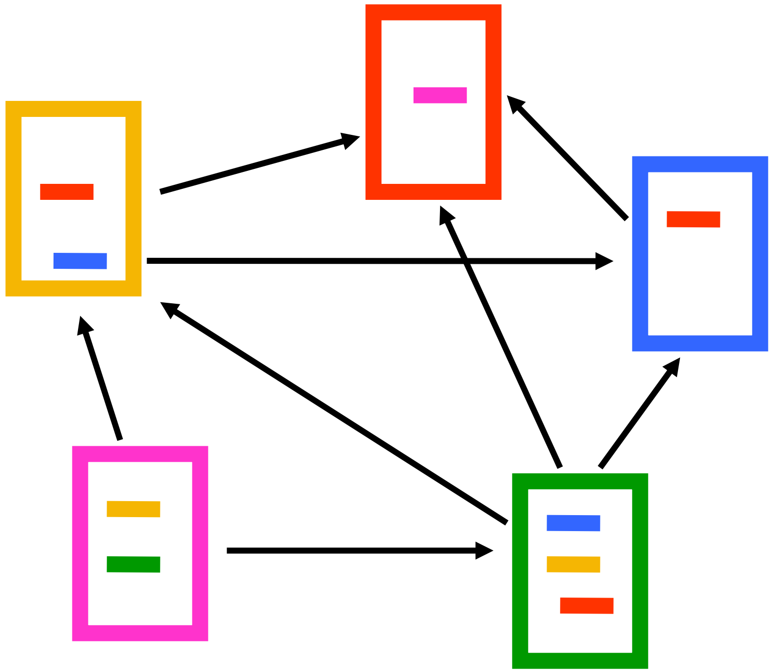
$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



The PageRank random walk

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

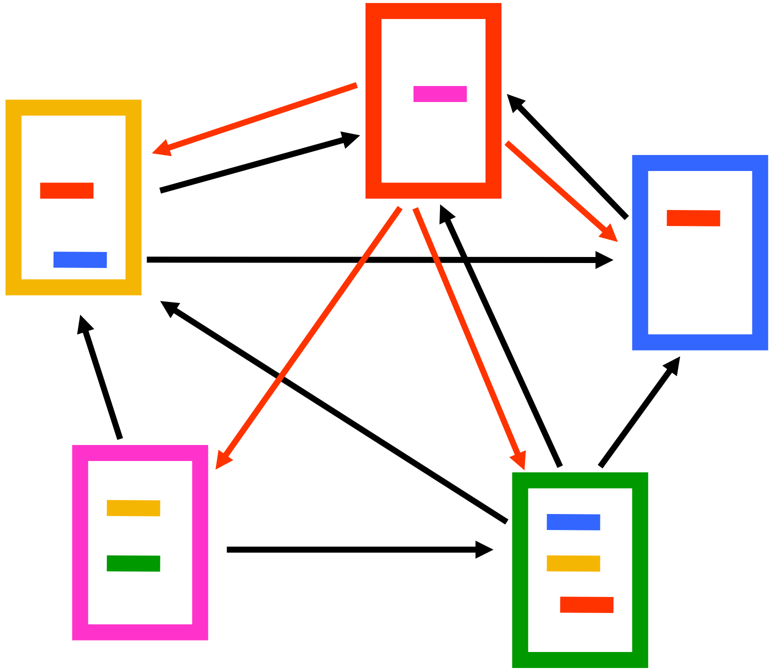
$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



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?
- How do we guarantee not getting stuck in loops?
  - add a random jump to vector v with prob α
    - typically, to a uniform vector

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

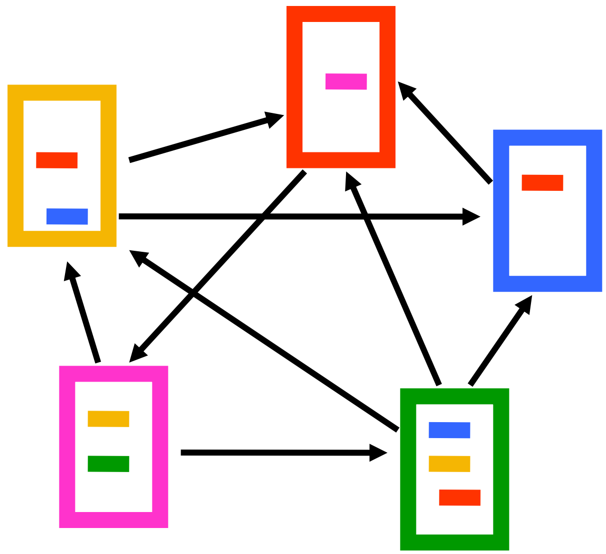
$P'' = \alpha P' + (1-\alpha)uv^T$, where u is the vector of all 1s

Random walk with restarts

PageRank algorithm [BP98]

- The Random Surfer model
  - pick a page at random
  - with probability 1- α jump to a random page
  - with probability ɑ follow a random outgoing link
- Rank according to the stationary distribution
- $$PR(p) = \alpha \sum_{q \to p} \frac{PR(q)}{|Out(q)|} + (1-\alpha)\frac{1}{n}$$

$\alpha = 0.85$ in most cases



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

The stationary distribution

- What is the meaning of the stationary distribution π of a random walk?
- $\pi(i)$: the probability of being at node i after very large (infinite) number of steps
- $\pi = p_0 P^{\infty}$, where P is the transition matrix, p_0 the original vector
  - $P(i,j)$: probability of going from i to j in one step
  - $P^2(i,j)$: probability of going from i to j in two steps (probability of all paths of length 2)
  - $P^{\infty}(i,j) = \pi(j)$: probability of going from i to j in infinite steps – starting point does not matter.

Stationary distribution with random jump

- If v is the jump vector

$$p_0 = v$$
$$p_1 = \alpha p_0 P + (1-\alpha)v = \alpha v P + (1-\alpha)v$$
$$p_2 = \alpha p_1 P + (1-\alpha)v = \alpha^2 v P^2 + (1-\alpha)v\alpha P + (1-\alpha)v$$
$$\vdots$$
$$p^\infty = (1-\alpha)v + (1-\alpha)v\alpha P + (1-\alpha)v\alpha^2 P^2 + \cdots$$
$$= (1-\alpha)(I - \alpha P)^{-1}$$

- With the random jump the shorter paths are more important, since the weight decreases exponentially
  - makes sense when thought of as a restart
- If v is not uniform, we can bias the random walk towards the pages that are close to v
  - Personalized and Topic Specific Pagerank.

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

Random walks on undirected graphs

- For undirected graphs, the stationary distribution is proportional to the degrees of the nodes
  - Thus in this case a random walk is the same as degree popularity
- This is not longer true if we do random jumps
  - Now the short paths play a greater role, and the previous distribution does not hold.

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

Pagerank history

- Huge advantage for Google in the early days
  - It gave a way to get an idea for the value of a page, which was useful in many different ways
    - Put an order to the web.
  - After a while it became clear that the anchor text was probably more important for ranking
  - Also, link spam became a new (dark) art
- Flood of research
  - Numerical analysis got rejuvenated
  - Huge number of variations
  - Efficiency became a great issue.
  - Huge number of applications in different fields
    - Random walk is often referred to as PageRank.

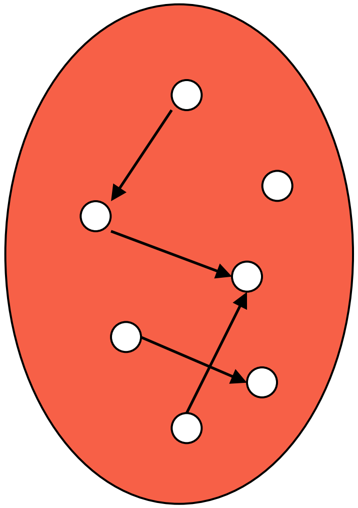
THE HITS ALGORITHM

The HITS algorithm

- Another algorithm proposed around the same time as Pagerank for using the hyperlinks to rank pages
  - Kleinberg: then an intern at IBM Almaden
  - IBM never made anything out of it

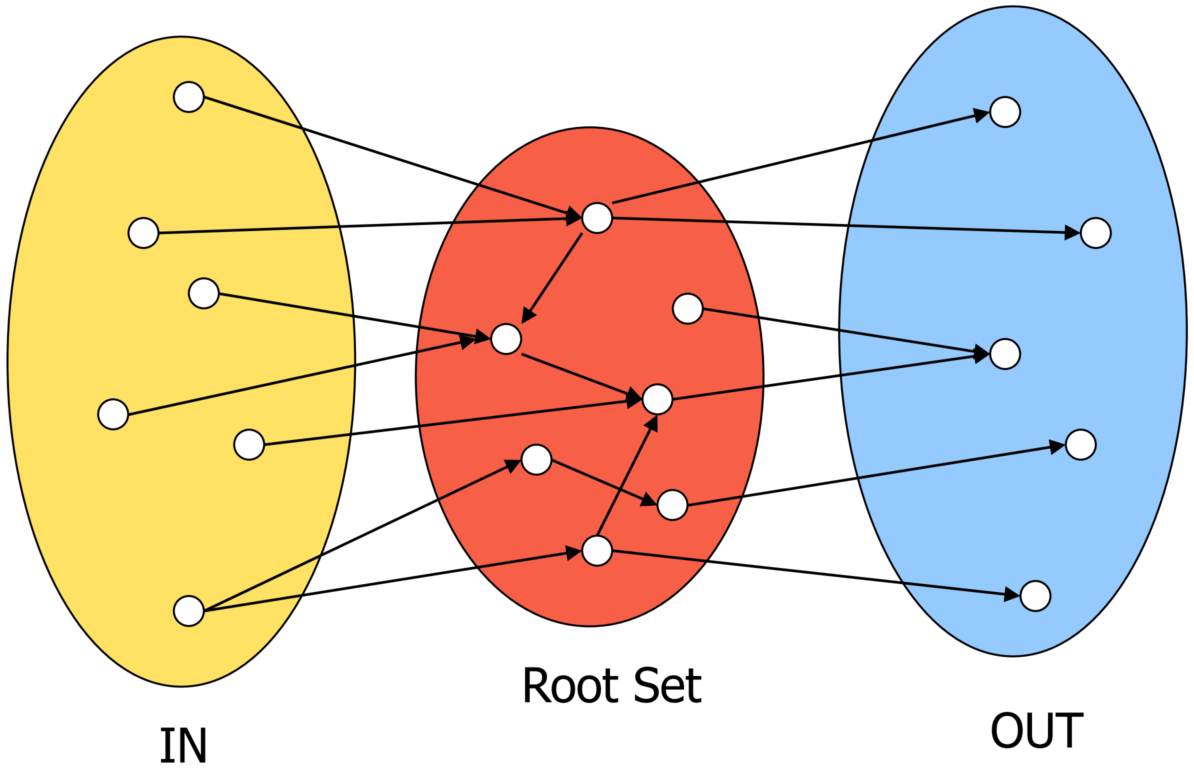
Query dependent input

Root set obtained from a text-only search engine

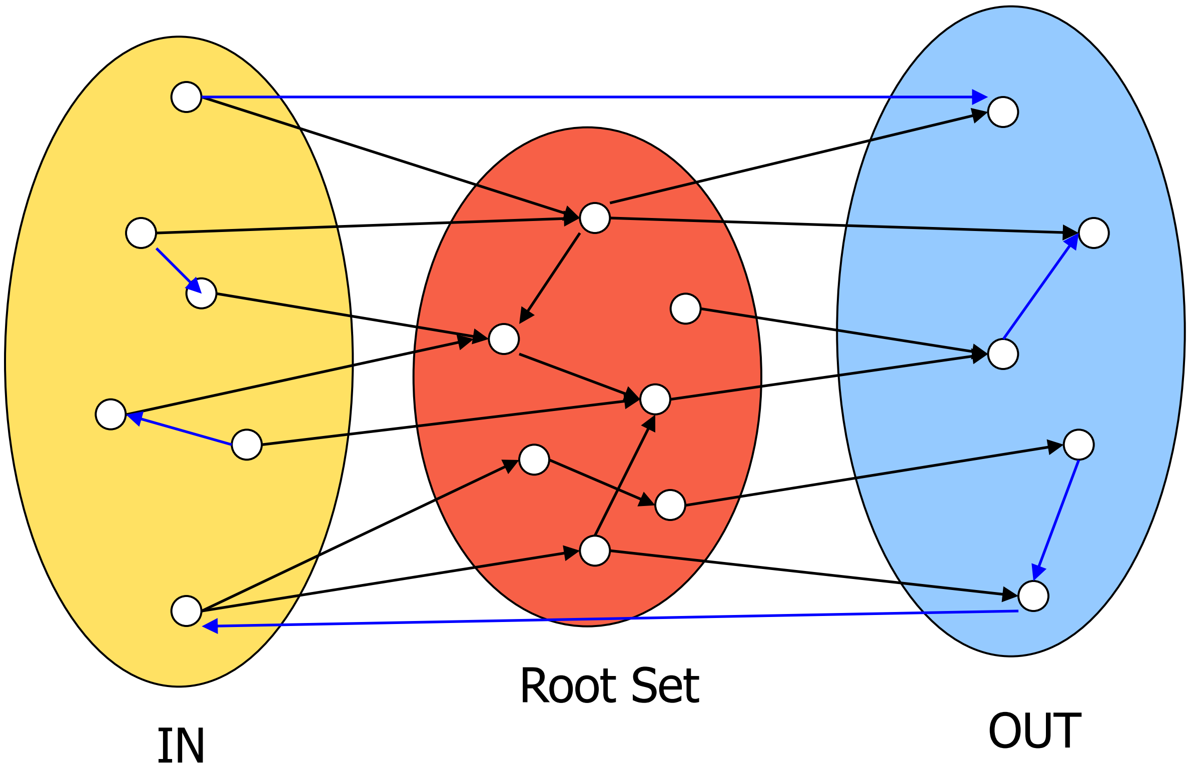


Root Set

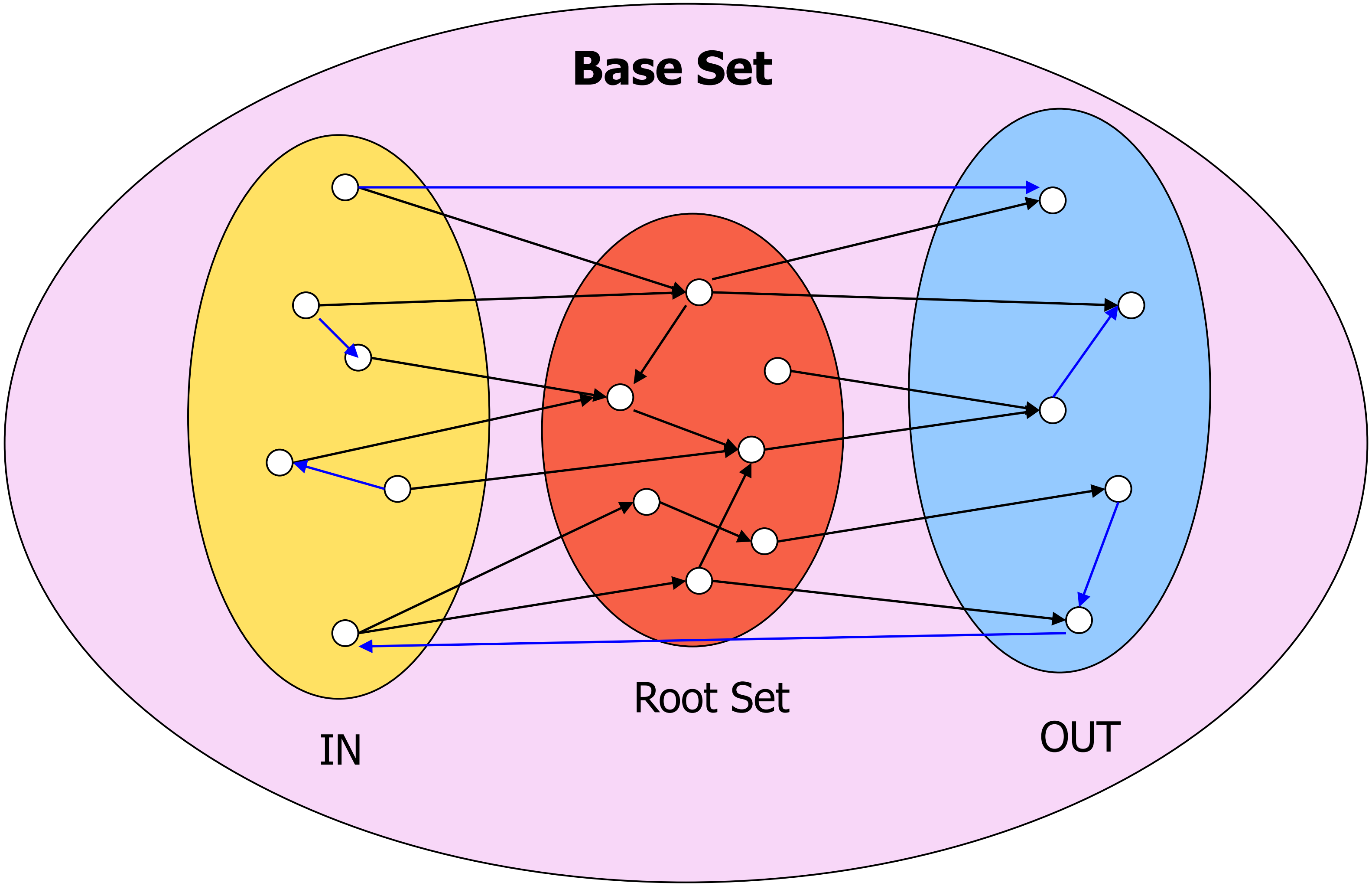
Query dependent input



Query dependent input

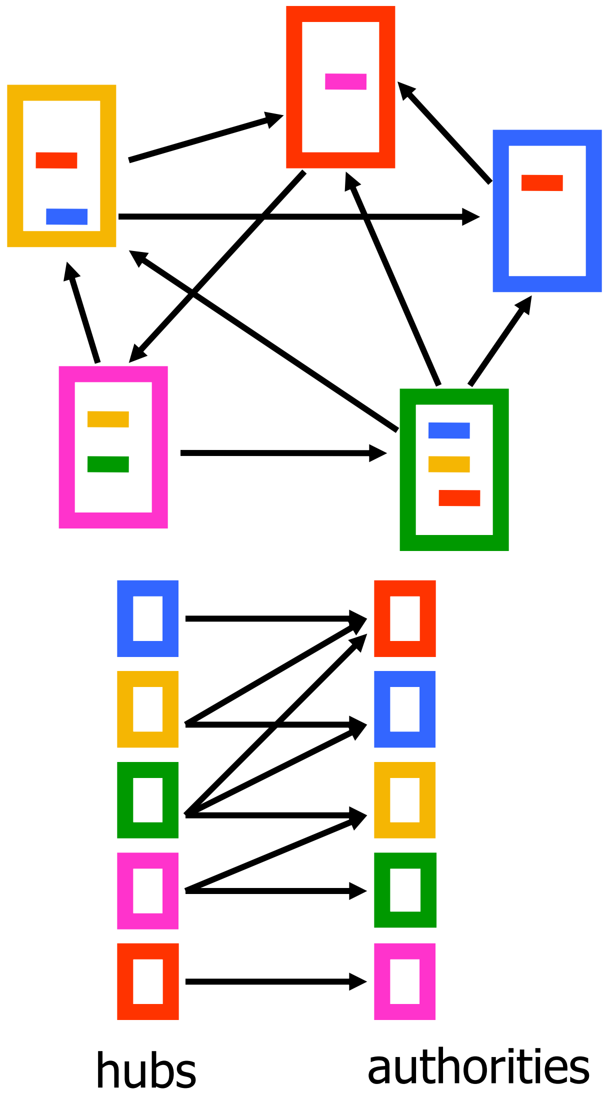


Query dependent input



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



Hubs and Authorities

- Two kind of weights:
  - Hub weight
  - Authority weight

- The hub weight is the sum of the authority weights of the authorities pointed to by the hub

- The authority weight is the sum of the hub weights that point to this authority.

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

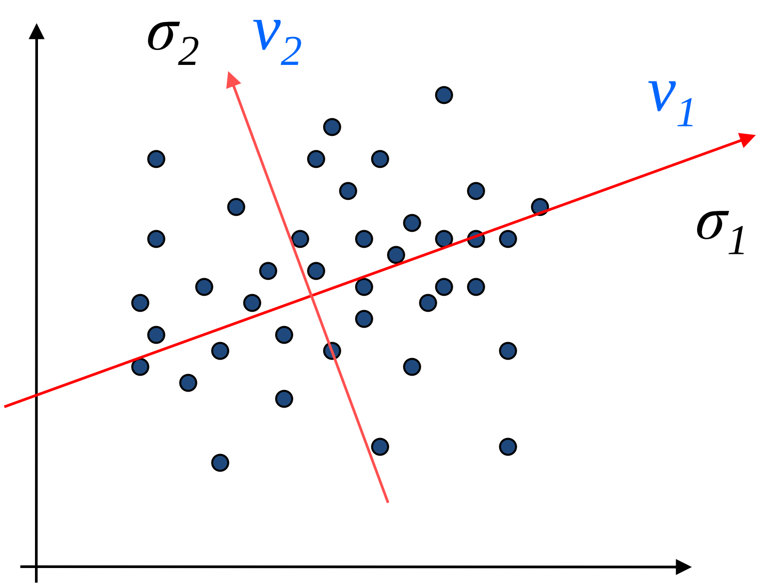
$$A = U \quad \Sigma \quad V^T = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix} \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_r \end{bmatrix}$$

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

- **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)
- $\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_r$: left singular vectors (eig-vectors of AA^T)
- $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_r$: right singular vectors (eig-vectors of A^TA)
- $$A = \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_r \vec{u}_r \vec{v}_r^T$$

Singular Value Decomposition

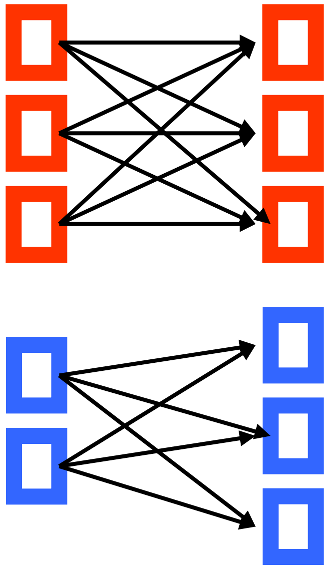
- Linear trend **v** in matrix A:
  - the tendency of the row vectors of A to align with vector **v**
  - strength of the linear trend: A**v**
- SVD discovers the linear trends in the data
- $\mathbf{u}_i$, $\mathbf{v}_i$: the i-th strongest linear trends
- σ_i : the strength of the i-th strongest linear trend



- HITS discovers the strongest linear trend in the authority space

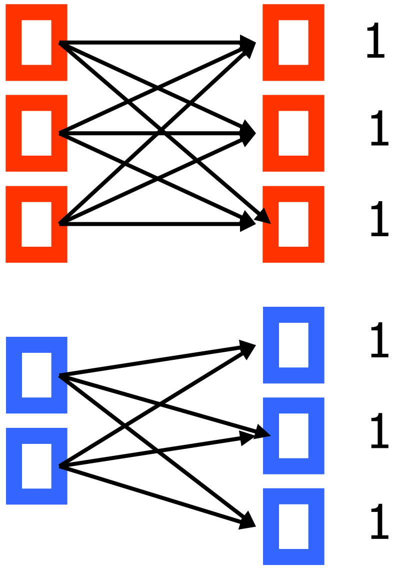
HITS and the TKC effect

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



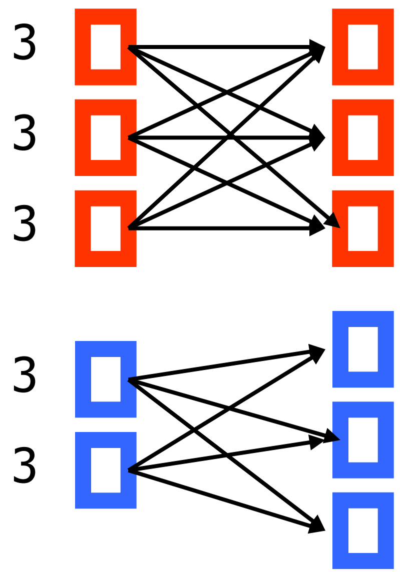
HITS and the TKC effect

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



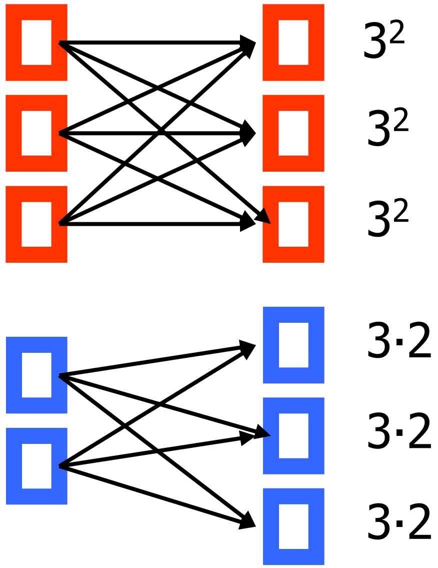
HITS and the TKC effect

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



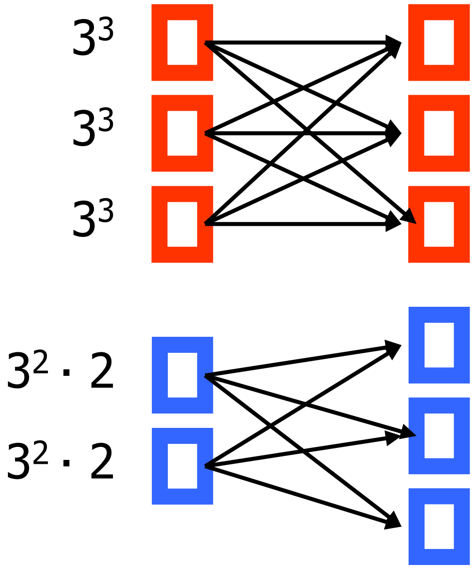
HITS and the TKC effect

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



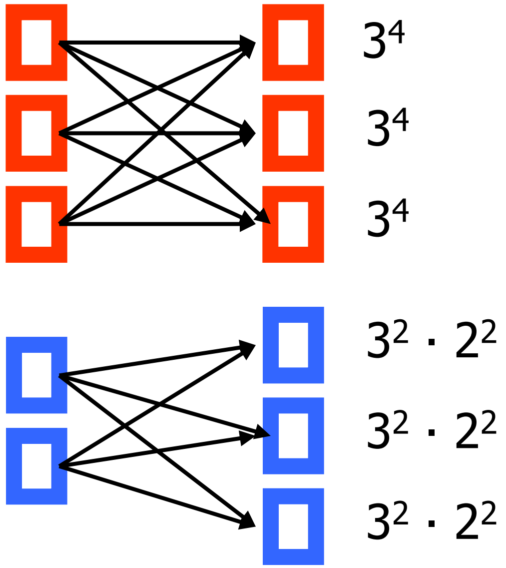
HITS and the TKC effect

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



HITS and the TKC effect

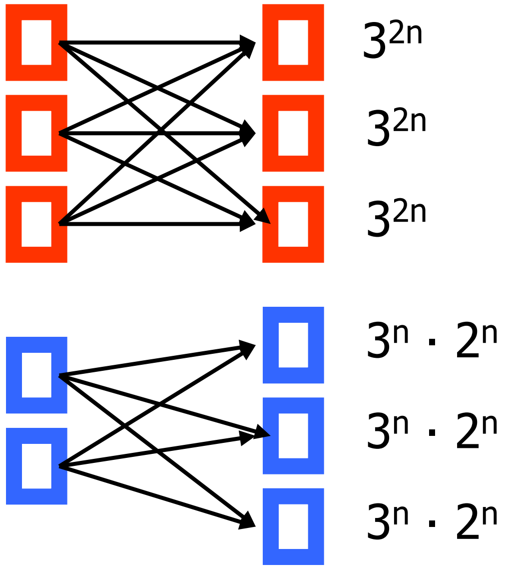
- The HITS algorithm favors the most dense community of hubs and authorities
  - Tightly Knit Community (TKC) effect



HITS and the TKC effect

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

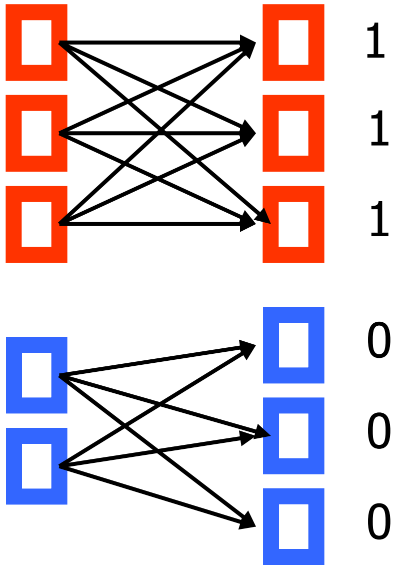
weight of node p is proportional to the number of $(BF)^n$ paths that leave node p



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

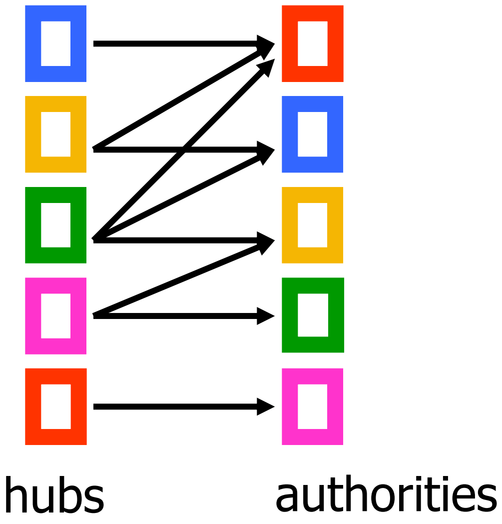


after normalization with the max element as n $\rightarrow \infty$

OTHER ALGORITHMS

The SALSA algorithm [LM00]

- Perform a random walk alternating between hubs and authorities



- What does this random walk converge to?

- The graph is essentially undirected, so it will be proportional to the degree.

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

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

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

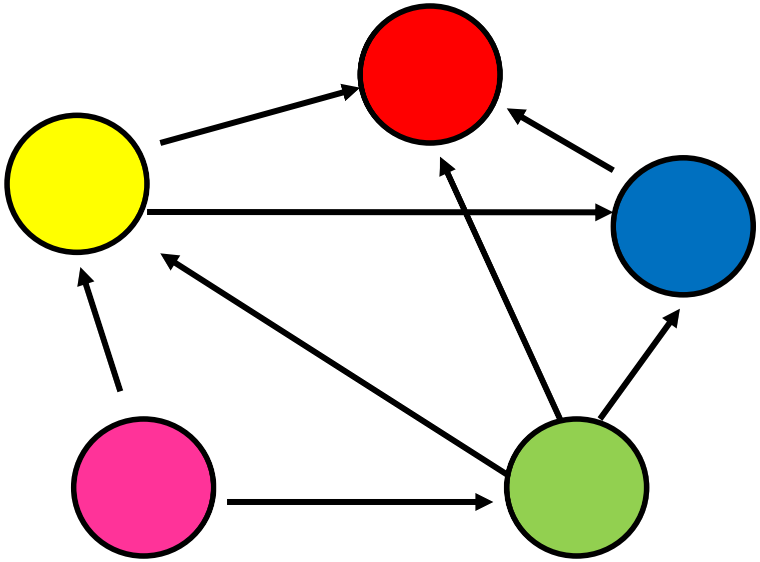
Bibliometrics

- 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

ABSORBING RANDOM WALKS

Random walk with absorbing nodes

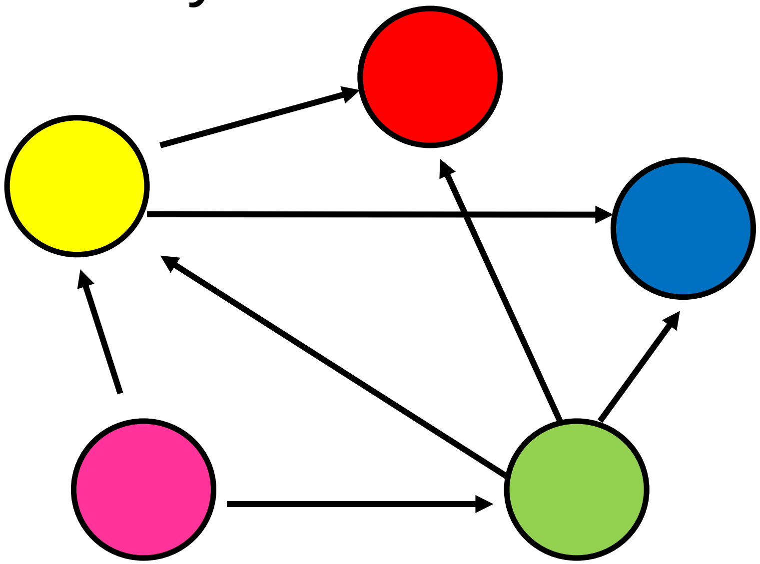
- What happens if we do a random walk on this graph? What is the stationary distribution?



- All the probability mass on the red sink node:
  - The red node is an absorbing node

Random walk with absorbing nodes

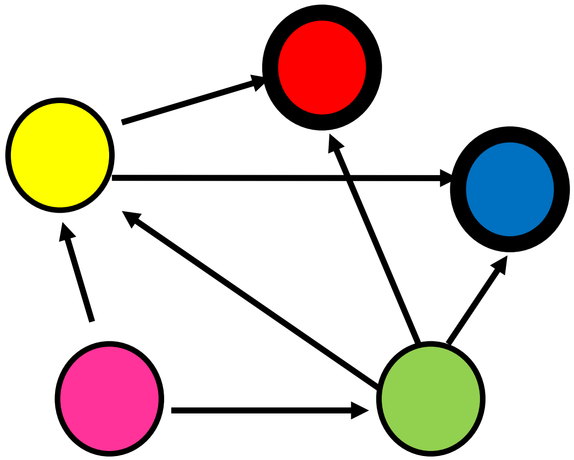
- What happens if we do a random walk on this graph? What is the stationary distribution?



- There are two absorbing nodes: the red and the blue.
- The probability mass will be divided between the two

Absorption probability

- If there are more than one absorbing nodes in the graph a random walk that starts from a non-absorbing node will be absorbed in one of them with some probability
  - The probability of absorption gives an estimate of how close the node is to red or blue



- Why care?
  - Red and Blue may be different categories

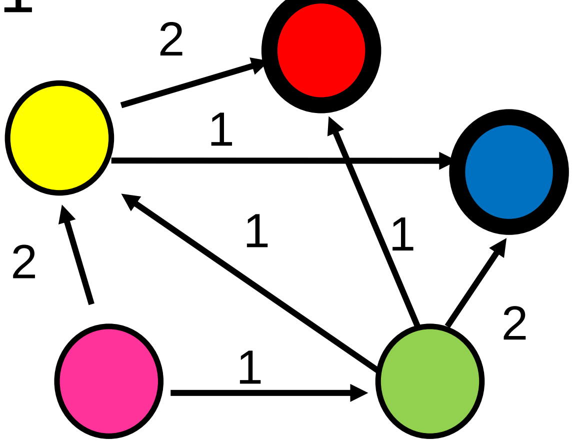
Absorption probability

- Computing the probability of being absorbed is very easy
  - Take the (weighted) average of the absorption probabilities of your neighbors
    - if one of the neighbors is the absorbing node, it has probability 1
  - Repeat until convergence
  - Initially only the absorbing have prob 1

$$P(Red|Pink) = \frac{2}{3}P(Red|Yellow) + \frac{1}{3}P(Red|Green)$$

$$P(Red|Green) = \frac{1}{4}P(Red|Yellow) + \frac{1}{4}$$

$$P(Red|Yellow) = \frac{2}{3}$$



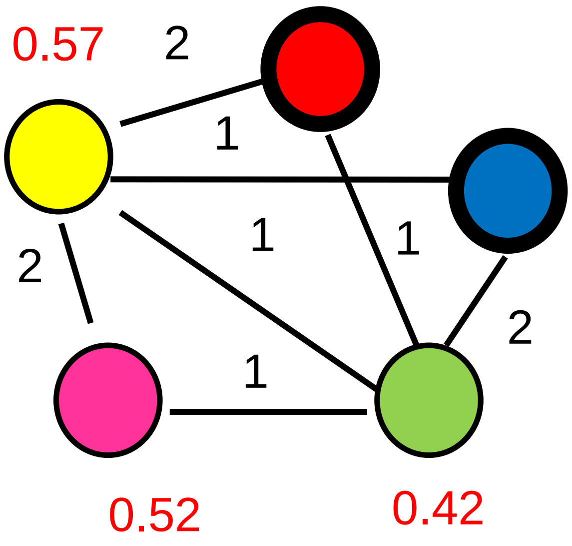
Absorption probability

- The same idea can be applied to the case of undirected graphs
  - The absorbing nodes are still absorbing, so the edges to them are (implicitely) directed.

$$P(Red|Pink) = \frac{2}{3}P(Red|Yellow) + \frac{1}{3}P(Red|Green)$$

$$P(Red|Green) = \frac{1}{5}P(Red|Yellow) + \frac{1}{5}P(Red|Pink) + \frac{1}{5}$$

$$P(Red|Yellow) = \frac{1}{6}P(Red|Green) + \frac{1}{3}P(Red|Pink) + \frac{1}{3}$$



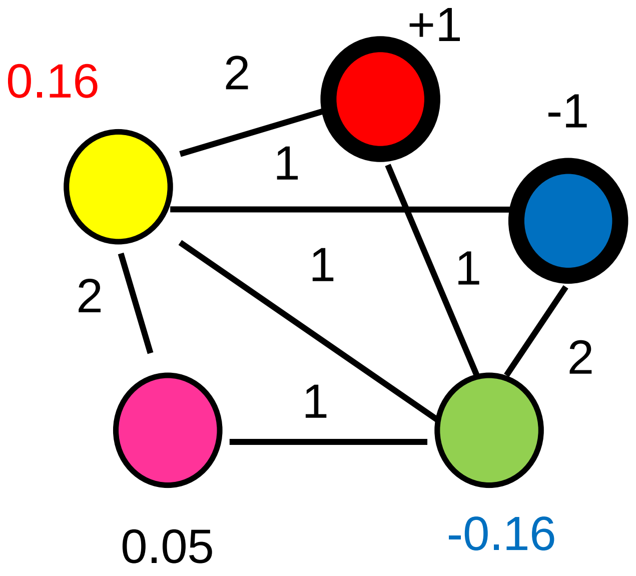
Propagating values

- Assume that Red corresponds to a positive class and Blue to a negative class
  - We can compute a value for all the other nodes in the same way
    - This is the expected value for the node

$$V(Pink) = \frac{2}{3}V(Yellow) + \frac{1}{3}V(Green)$$

$$V(Green) = \frac{1}{5}V(Yellow) + \frac{1}{5}V(Pink) + \frac{1}{5} - \frac{2}{5}$$

$$V(Yellow) = \frac{1}{6}V(Green) + \frac{1}{3}V(Pink) + \frac{1}{3} - \frac{1}{6}$$



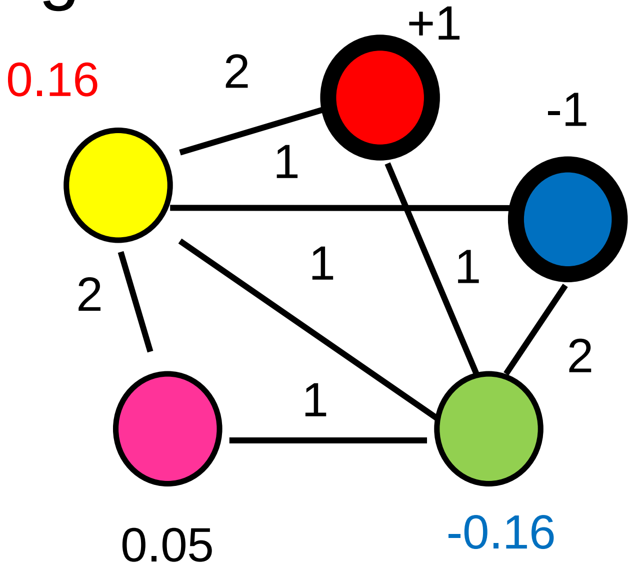
Electrical networks and random walks

- If Red corresponds to a positive voltage and Blue to a negative voltage
- There are resistances on the edges inversely proportional to the weights
- The computed values are the voltages

$$V(Pink) = \frac{2}{3}V(Yellow) + \frac{1}{3}V(Green)$$

$$V(Green) = \frac{1}{5}V(Yellow) + \frac{1}{5}V(Pink) + \frac{1}{5} - \frac{2}{5}$$

$$V(Yellow) = \frac{1}{6}V(Green) + \frac{1}{3}V(Pink) + \frac{1}{3} - \frac{1}{6}$$



Transductive learning

- If we have a graph of relationships and some labels on these edges we can propagate them to the remaining nodes
  - E.g., a social network where some people are tagged as spammers
- This is a form of semi-supervised learning
  - We make use of the unlabeled data, and the relationships
- It is also called transductive learning because it does not produce a model, and labels only what is at hand.