

# DATA MINING

## LECTURE 13

---

**Link Analysis Ranking**

**PageRank -- Random walks**

**HITS**

# How to organize the web

- **First try:** Manually curated Web Directories

YAHOO! DIRECTORY Yahoo! | Help

Search:  the Web |  the Directory


---

Yahoo! Directory Advanced Search | Suggest a Site | Email This Page

<b>Arts &amp; Humanities</b> Photography, History, Literature...	<b>News &amp; Media</b> Newspapers, Radio, Weather, Blogs...
<b>Business &amp; Economy</b> B2B, Finance, Shopping, Jobs...	<b>Recreation &amp; Sports</b> Sports, Travel, Autos, Outdoors...
<b>Computer &amp; Internet</b> Hardware, Software, Web, Games...	<b>Reference</b> Phone Numbers, Dictionaries, Quotes...
<b>Education</b> Colleges, K-12, Distance Learning...	<b>Regional</b> Countries, Regions, U.S. States...
<b>Entertainment</b> Movies, TV Shows, Music, Humor...	<b>Science</b> Animals, Astronomy, Earth Science...
<b>Government</b> Elections, Military, Law, Taxes...	<b>Social Science</b> Languages, Archaeology, Psychology...
<b>Health</b> Disease, Drugs, Fitness, Nutrition...	<b>Society &amp; Culture</b> Sexuality, Religion, Food & Drink...
<b>New Additions</b> 12/3, 12/2, 12/1, 11/30, 11/29...	<b>Subscribe via RSS</b> Arts, Music, Sports, TV, more...

---

Copyright © 2012 Yahoo! Inc. All rights reserved. [Privacy Policy](#) - [About Our Ads](#) - [Terms of Service](#) - [Copyright/IP Policy](#)

 Help us improve the Yahoo! Directory - [Share your ideas](#)

about dmoz | [dmoz blog](#) | [suggest URL](#) | [help](#) | [link](#) | [editor login](#)

[advanced](#)

<a href="#">Arts</a> <a href="#">Movies</a> , <a href="#">Television</a> , <a href="#">Music</a> ...	<a href="#">Business</a> <a href="#">Jobs</a> , <a href="#">Real Estate</a> , <a href="#">Investing</a> ...	<a href="#">Computers</a> <a href="#">Internet</a> , <a href="#">Software</a> , <a href="#">Hardware</a> ...
<a href="#">Games</a> <a href="#">Video Games</a> , <a href="#">RPGs</a> , <a href="#">Gambling</a> ...	<a href="#">Health</a> <a href="#">Fitness</a> , <a href="#">Medicine</a> , <a href="#">Alternative</a> ...	<a href="#">Home</a> <a href="#">Family</a> , <a href="#">Consumers</a> , <a href="#">Cooking</a> ...
<a href="#">Kids and Teens</a> <a href="#">Arts</a> , <a href="#">School Time</a> , <a href="#">Teen Life</a> ...	<a href="#">News</a> <a href="#">Media</a> , <a href="#">Newspapers</a> , <a href="#">Weather</a> ...	<a href="#">Recreation</a> <a href="#">Travel</a> , <a href="#">Food</a> , <a href="#">Outdoors</a> , <a href="#">Humor</a> ...
<a href="#">Reference</a> <a href="#">Maps</a> , <a href="#">Education</a> , <a href="#">Libraries</a> ...	<a href="#">Regional</a> <a href="#">US</a> , <a href="#">Canada</a> , <a href="#">UK</a> , <a href="#">Europe</a> ...	<a href="#">Science</a> <a href="#">Biology</a> , <a href="#">Psychology</a> , <a href="#">Physics</a> ...
<a href="#">Shopping</a> <a href="#">Clothing</a> , <a href="#">Food</a> , <a href="#">Gifts</a> ...	<a href="#">Society</a> <a href="#">People</a> , <a href="#">Religion</a> , <a href="#">Issues</a> ...	<a href="#">Sports</a> <a href="#">Baseball</a> , <a href="#">Soccer</a> , <a href="#">Basketball</a> ...
<a href="#">World</a> <a href="#">Català</a> , <a href="#">Dansk</a> , <a href="#">Deutsch</a> , <a href="#">Español</a> , <a href="#">Français</a> , <a href="#">Italiano</a> , <a href="#">日本語</a> , <a href="#">Nederlands</a> , <a href="#">Polski</a> , <a href="#">Русский</a> , <a href="#">Svenska</a> ...		

[Become an Editor](#) Help build the largest human-edited directory of the web



Copyright © 2012 Netscape

5,114,642 sites - 96,895 editors - over 1,014,858 categories

# How to organize the web

- **Second try:** Web Search

- Information Retrieval investigates:

- Find relevant docs in a small and trusted set e.g., Newspaper articles, Patents, etc. (“needle-in-a-haystack”)
- Limitation of keywords (synonyms, polysemy, etc)

- **But:** Web is huge, full of untrusted documents, random things, web spam, etc.

- Everyone can create a web page of high production value
- Rich diversity of people issuing queries
- Dynamic and constantly-changing nature of web content

# How to organize the web

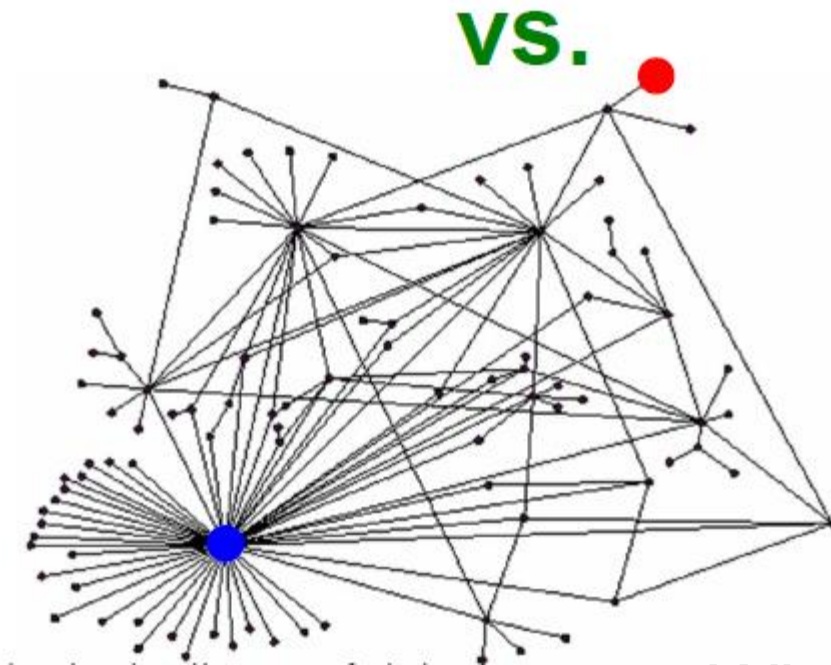
- **Third try** (the **Google** era): using the web graph
  - Sift from **relevance** to **authoritativeness**
  - It is not only important that a page is relevant, but that it is also important on the web
- For example, what kind of results would we like to get for the query “greek newspapers”?

# Link Analysis Ranking

- Use the **graph structure** in order to determine the **relative importance** of the nodes
  - Applications: Ranking on graphs (Web, Twitter, FB, etc)
- **Intuition**: An edge from node **p** to node **q** denotes **endorsement**
  - Node **p** **endorses/recommends/confirm**s the **authority/centrality/importance** of node **q**
  - Use the graph of recommendations to assign an **authority value** to every node

# Link Analysis

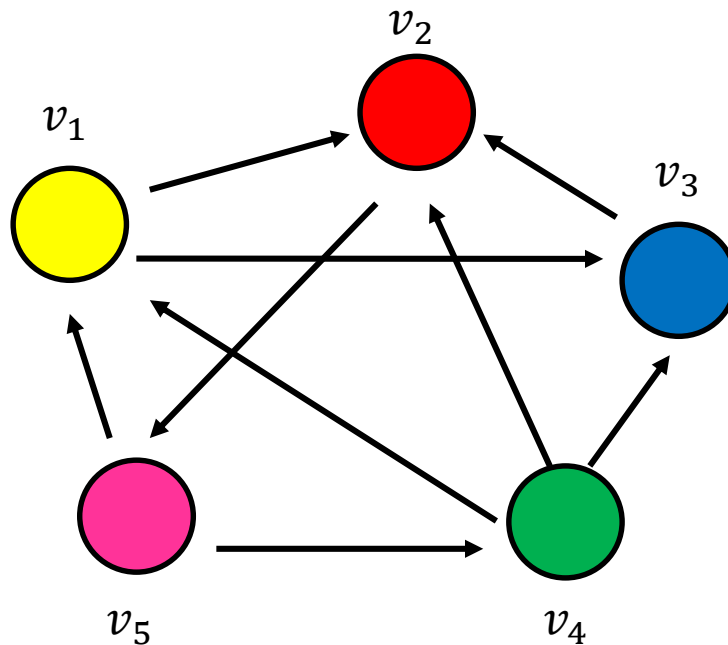
- Not all web pages are equal on the web



What is the simplest way to measure importance of a page on the web?

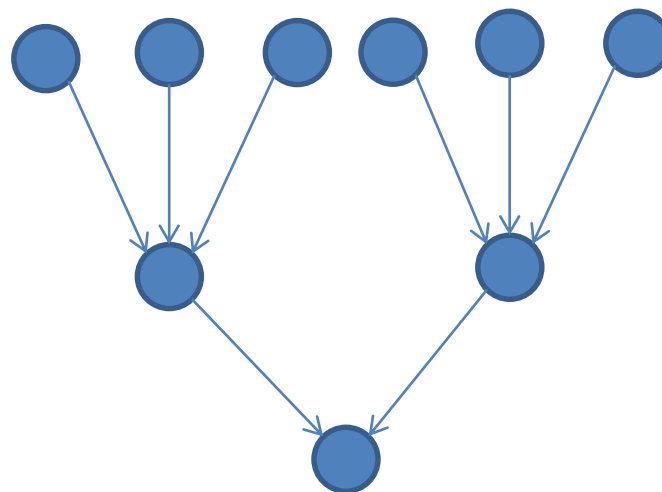
# Rank by Popularity

- 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

---

# PageRank

- **Good** authorities should be pointed by **good** authorities
  - The value of a node is the value of the nodes that point to it.
- How do we implement that?
  - Assume that we have **a unit of authority** to distribute to all nodes.
    - Initially each node gets  $\frac{1}{n}$  amount of authority
  - Each node **distributes** the authority value they have **to their neighbors**
  - The authority value of each node is the sum of the **authority fractions** it collects from its neighbors.

$$w_v = \sum_{u \rightarrow v} \frac{1}{d_{out}(u)} w_u$$

$w_v$ : the **PageRank value** of node  $v$

Recursive definition

# Example

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

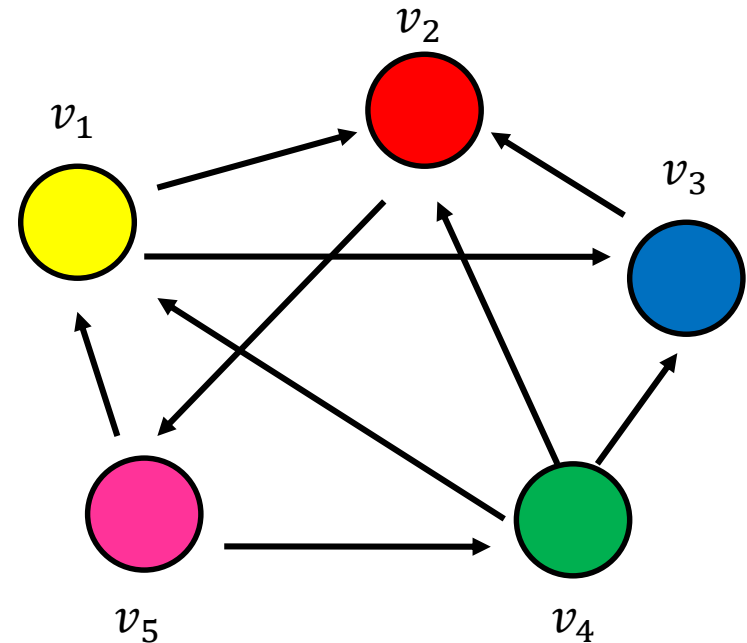
$$w_2 = 1/2 w_1 + w_3 + 1/3 w_4$$

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

$$w_4 = 1/2 w_5$$

$$w_5 = w_2$$

$$w_v = \sum_{u \rightarrow v} \frac{1}{d_{out}(u)} w_u$$



# Computing PageRank weights

- A simple way to compute the weights is by iteratively updating the weights
- PageRank Algorithm

Initialize all PageRank weights to  $\frac{1}{n}$

Repeat:

$$w_v = \sum_{u \rightarrow v} \frac{1}{d_{out}(u)} w_u$$

Until the weights do not change

- This process converges

# Example

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

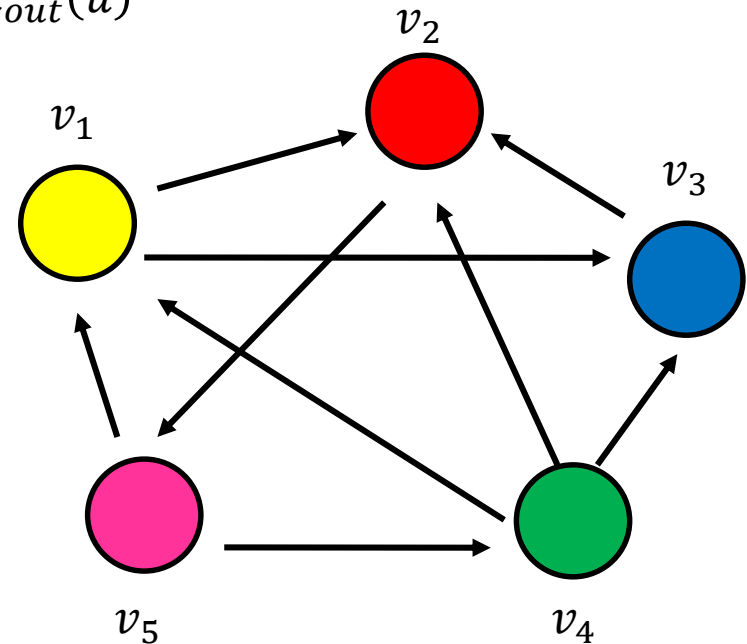
$$w_2 = 1/2 w_1 + w_3 + 1/3 w_4$$

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

$$w_4 = 1/2 w_5$$

$$w_5 = w_2$$

$$w_v = \sum_{u \rightarrow v} \frac{1}{d_{out}(u)} w_u$$



Think of the weight as a **fluid**: there is constant amount of it in the graph, but it moves around until it stabilizes

	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$
$t=0$	0.2	0.2	0.2	0.2	0.2
$t=1$	0.16	0.36	0.16	0.1	0.2
$t=2$	0.13	0.28	0.11	0.1	0.36
$t=3$	0.22	0.22	0.1	0.18	0.28
$t=4$	0.2	0.27	0.17	0.14	0.22

# Example

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

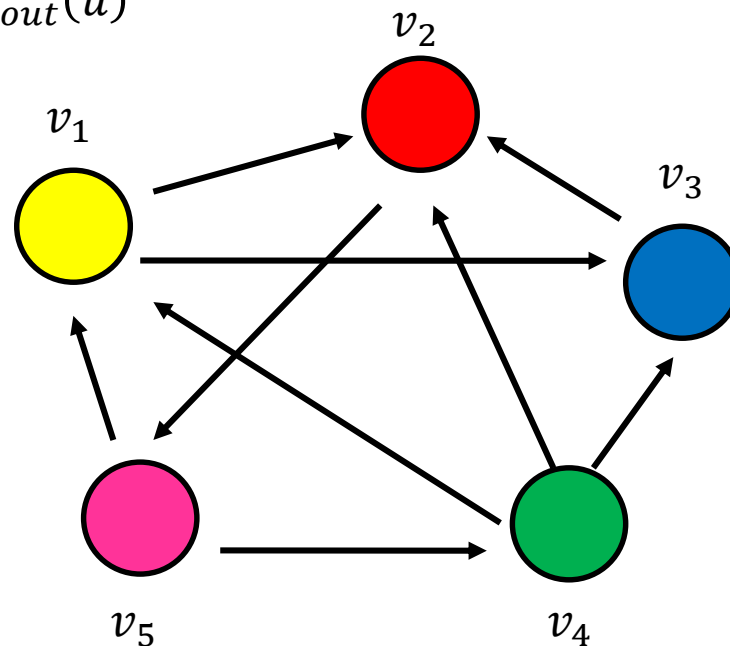
$$w_2 = 1/2 w_1 + w_3 + 1/3 w_4$$

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

$$w_4 = 1/2 w_5$$

$$w_5 = w_2$$

$$w_v = \sum_{u \rightarrow v} \frac{1}{d_{out}(u)} w_u$$



	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$
t=25	0.18	0.27	0.13	0.13	0.27

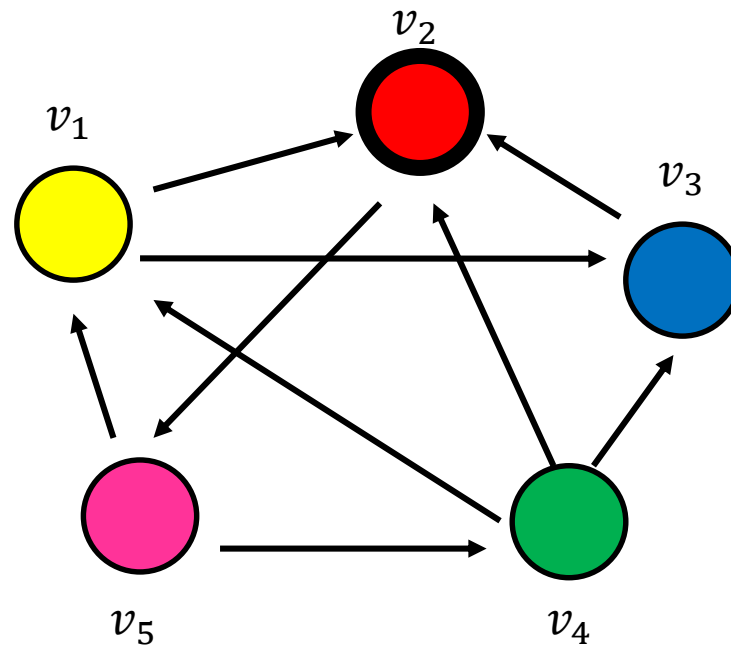
Think of the weight as a **fluid**: there is constant amount of it in the graph, but it moves around until it stabilizes

# Random Walks on Graphs

- The algorithm defines a **random walk** on the graph
- Random walk:
  - **Start** from a node chosen **uniformly at random** with probability  $\frac{1}{n}$ .
  - **Pick** one of the **outgoing edges** **uniformly at random**
  - **Move** to the destination of the edge
  - Repeat.
- The **Random Surfer** model
  - Users wander on the web, following links.

# Example

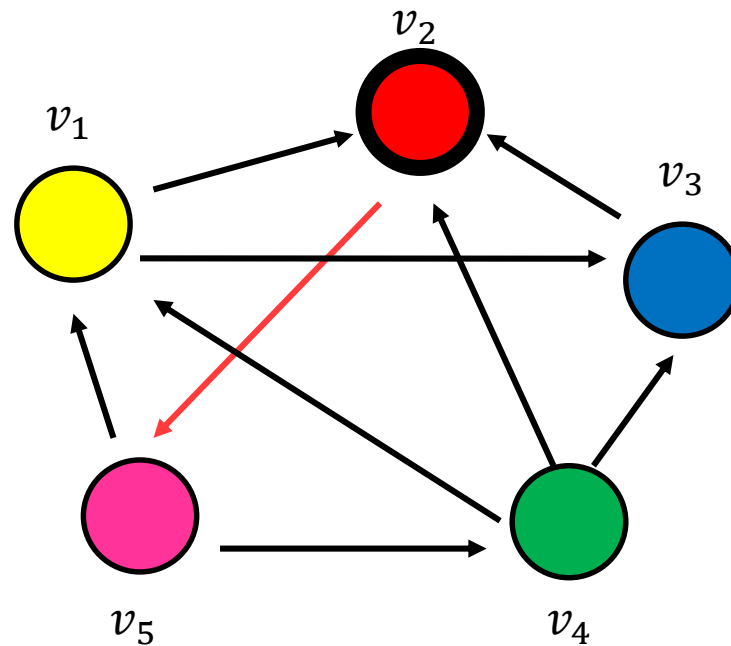
- Step 0





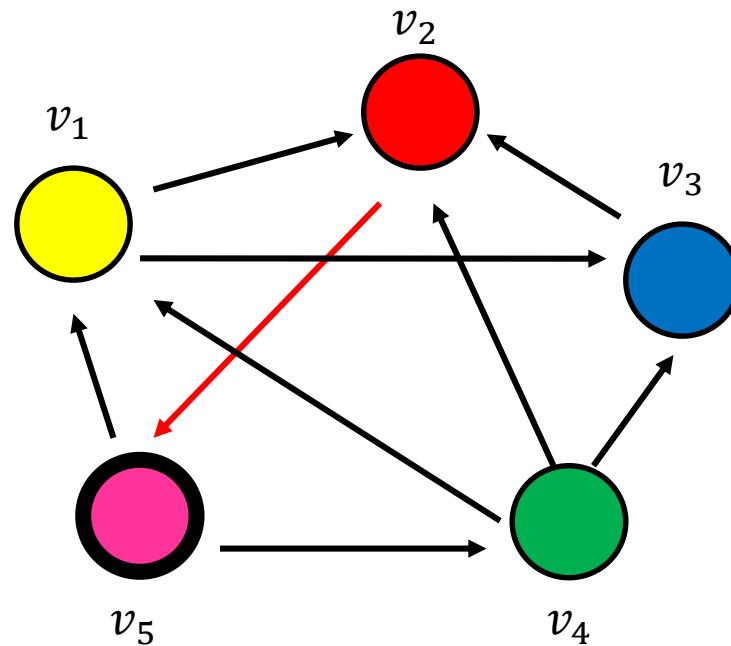
# Example

- Step 0



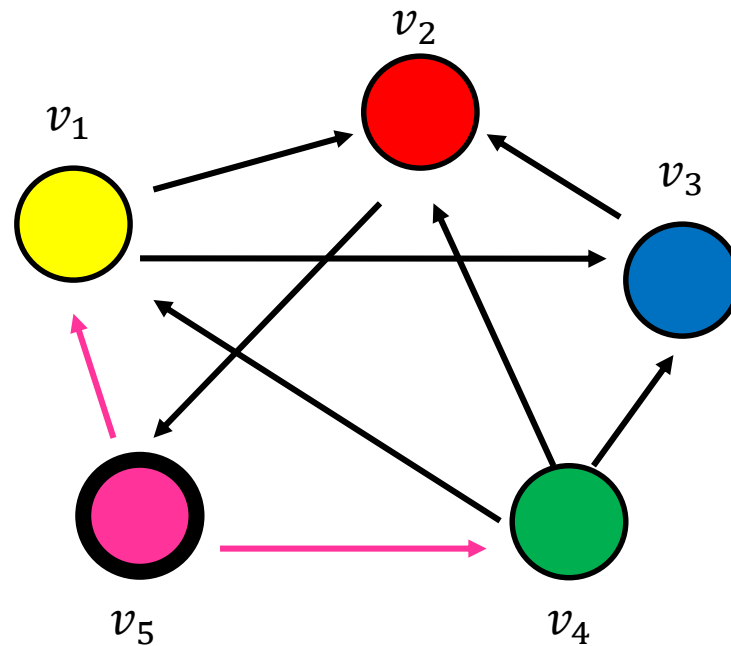
# Example

- Step 1



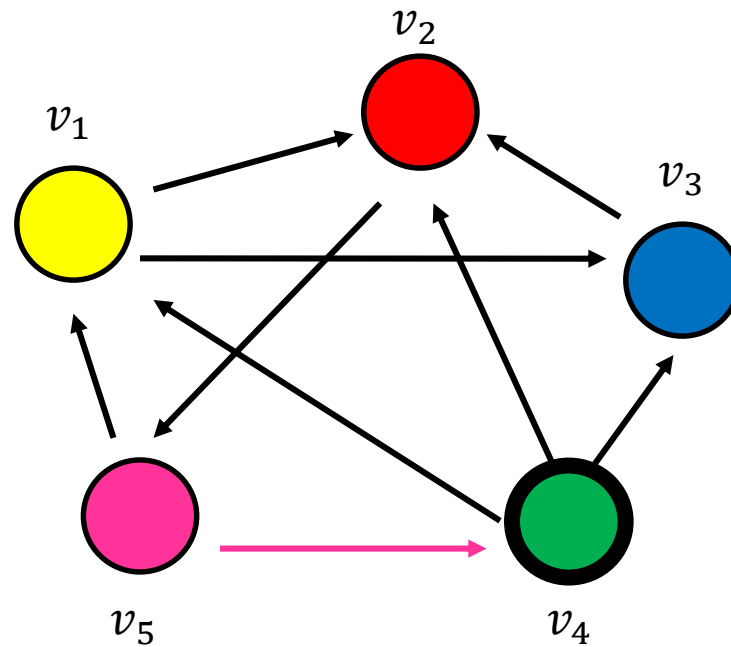
# Example

- Step 1



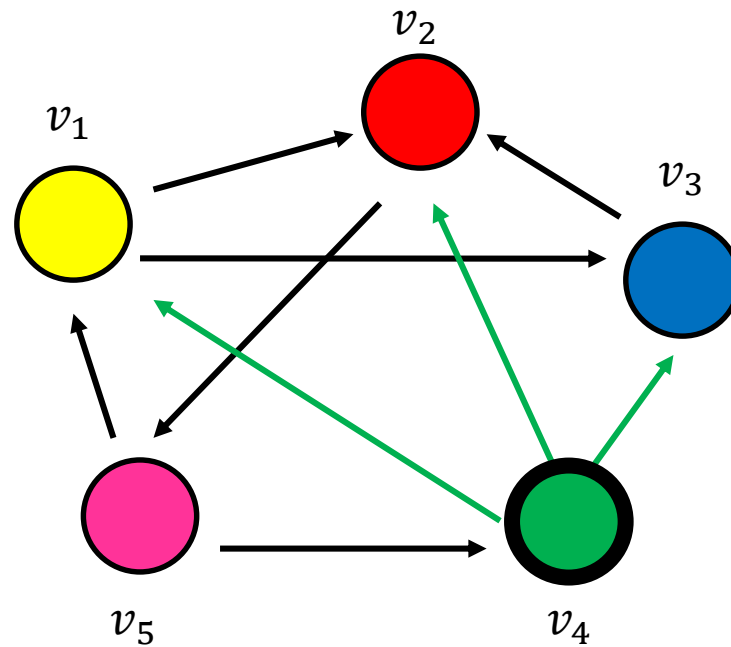
# Example

- Step 2



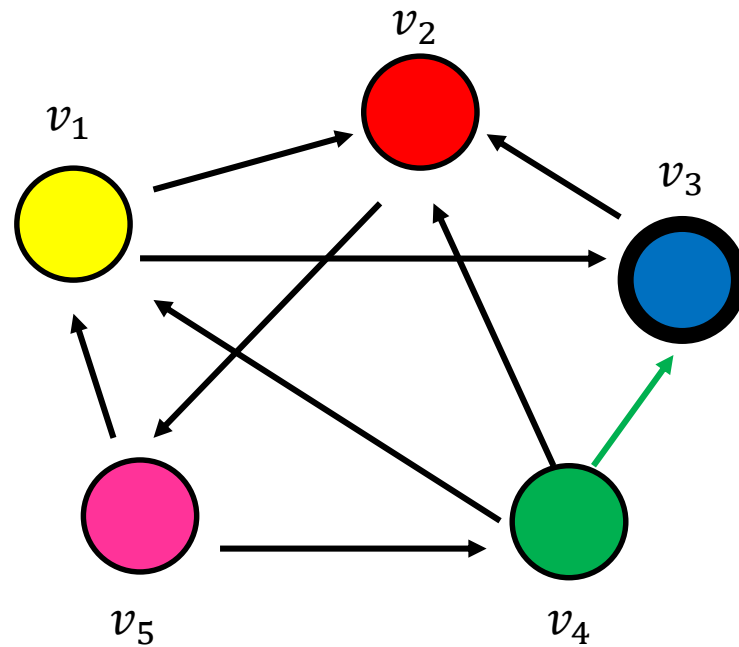
# Example

- Step 2



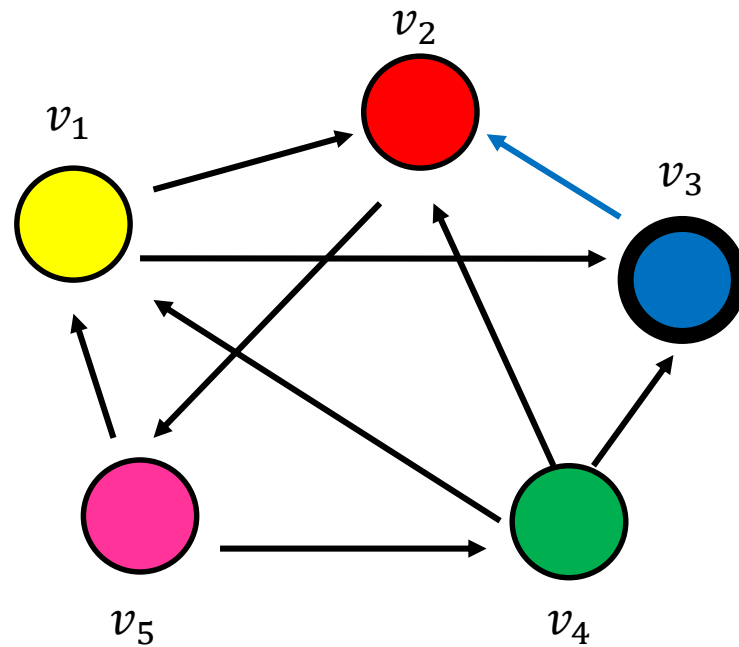
# Example

- Step 3



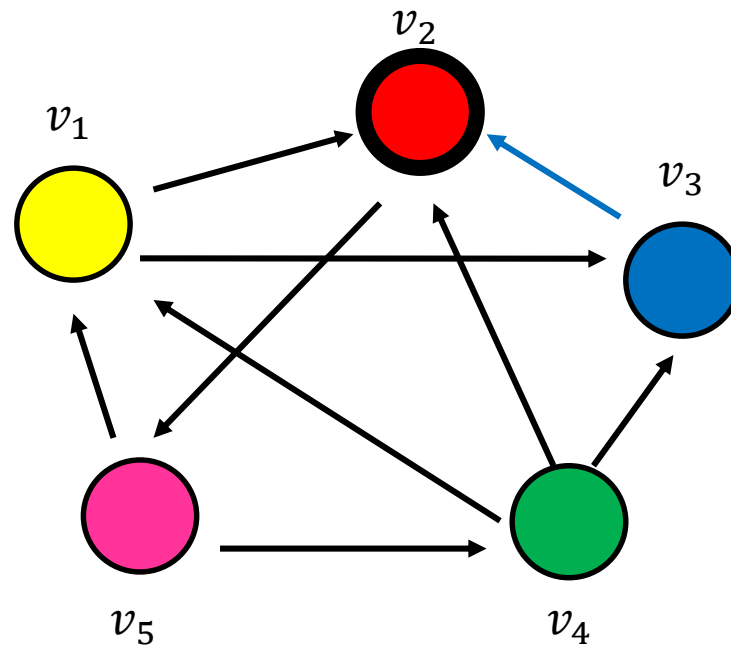
# Example

- Step 3



# Example

- Step 4...





# Random walk

- Question: what is the probability  $p_i^t$  of being at node  $i$  after  $t$  steps?

$$p_1^0 = \frac{1}{5}$$

$$p_2^0 = \frac{1}{5}$$

$$p_3^0 = \frac{1}{5}$$

$$p_4^0 = \frac{1}{5}$$

$$p_5^0 = \frac{1}{5}$$

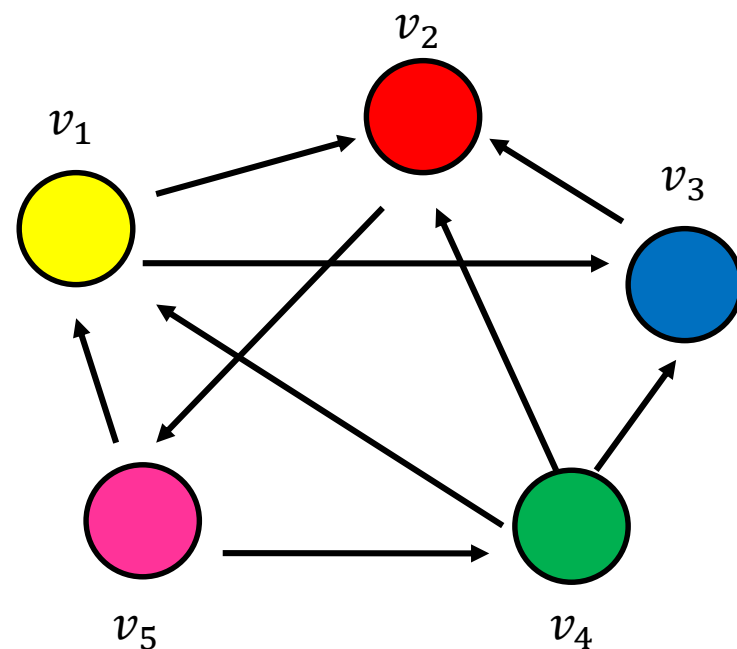
$$p_1^t = \frac{1}{3}p_4^{t-1} + \frac{1}{2}p_5^{t-1}$$

$$p_2^t = \frac{1}{2}p_1^{t-1} + p_3^{t-1} + \frac{1}{3}p_4^{t-1}$$

$$p_3^t = \frac{1}{2}p_1^{t-1} + \frac{1}{3}p_4^{t-1}$$

$$p_4^t = \frac{1}{2}p_5^{t-1}$$

$$p_5^t = p_2^{t-1}$$



The equations are the same as those for the PageRank computation

# 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  $i$

- Matrix  $P$  has the property that the entries of all **rows sum to 1**

$$\sum_j P[i, j] = 1$$

A matrix with this property is called **stochastic**

- **State probability distribution**: The vector  $p^t = (p_i^t, p_2^t, \dots, p_n^t)$  that stores the probability of being at state  $s_i$  after  $t$  steps
- **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
- **Markov Chain Theory**: After infinite steps the **state probability vector converges** to a **unique** distribution if the chain is **irreducible** and **aperiodic**

# 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

$$P[i, j] = \frac{1}{\text{deg}_{out}(i)}$$

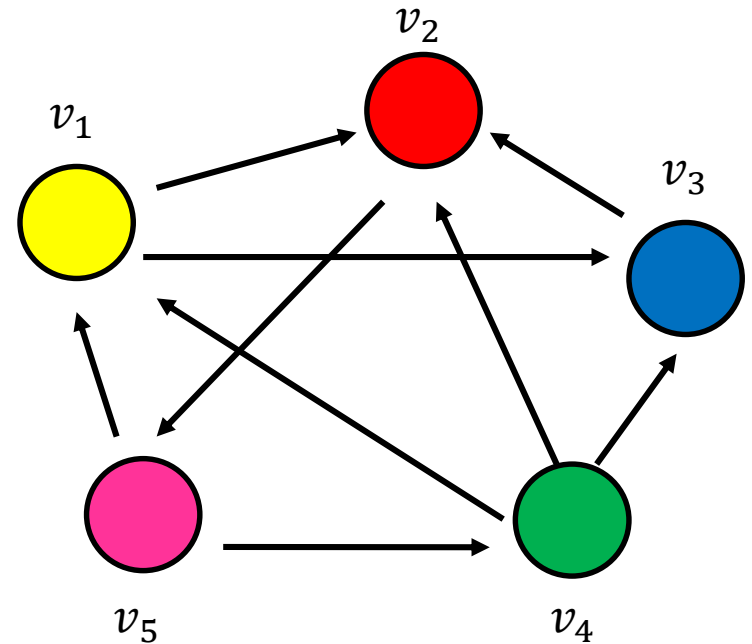
- We can compute the vector  $p^t$  at step  $t$  using a vector-matrix multiplication

$$p^{t+1} = p^t P$$

# An example

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

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



# An example

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

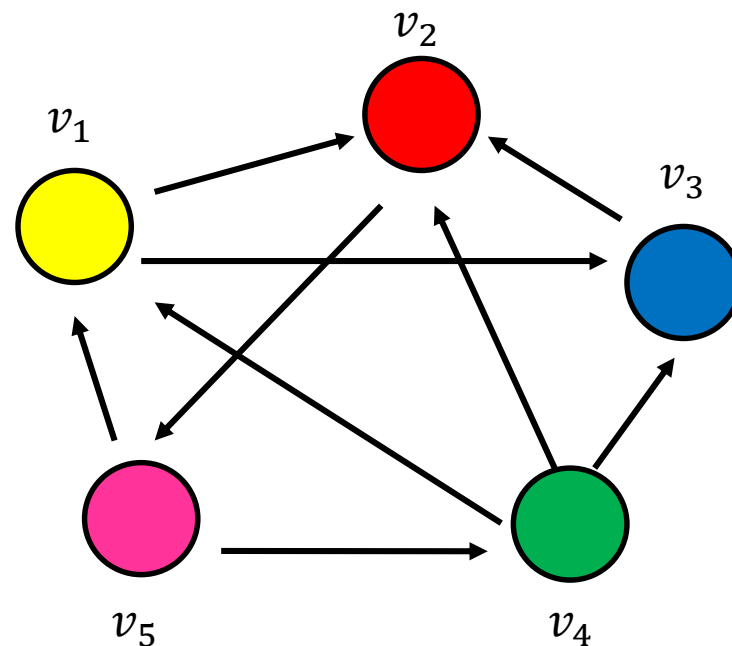
$$p_1^t = \frac{1}{3}p_4^{t-1} + \frac{1}{2}p_5^{t-1}$$

$$p_2^t = \frac{1}{2}p_1^{t-1} + p_3^{t-1} + \frac{1}{3}p_4^{t-1}$$

$$p_3^t = \frac{1}{2}p_1^{t-1} + \frac{1}{3}p_4^{t-1}$$

$$p_4^t = \frac{1}{2}p_5^{t-1}$$

$$p_5^t = p_2^{t-1}$$



# Stationary distribution

- The **stationary distribution** of a random walk with transition matrix  $P$ , is a probability distribution  $\pi$ , such that  $\pi = \pi P$
- The stationary distribution is an **eigenvector** of matrix  $P$ 
  - the **principal left eigenvector** of  $P$  – stochastic matrices have maximum eigenvalue 1
- The probability  $\pi_i$  is the fraction of times that we visited state  $i$  as  $t \rightarrow \infty$
- **Markov Chain Theory**: The random walk converges to a **unique stationary distribution independent of the initial vector** if the graph is **strongly connected**, and **not bipartite**.

# Computing the stationary distribution

- The **Power Method**

Initialize  $p^0$  to some distribution

Repeat

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

Until **convergence**

- After **many** iterations  $p^t \rightarrow \pi$  regardless of the initial vector  $p^0$
- Power method because it computes  $p^t = p^0 P^t$
- Rate of convergence
  - determined by the second eigenvalue  $\lambda_2$

# The stationary distribution

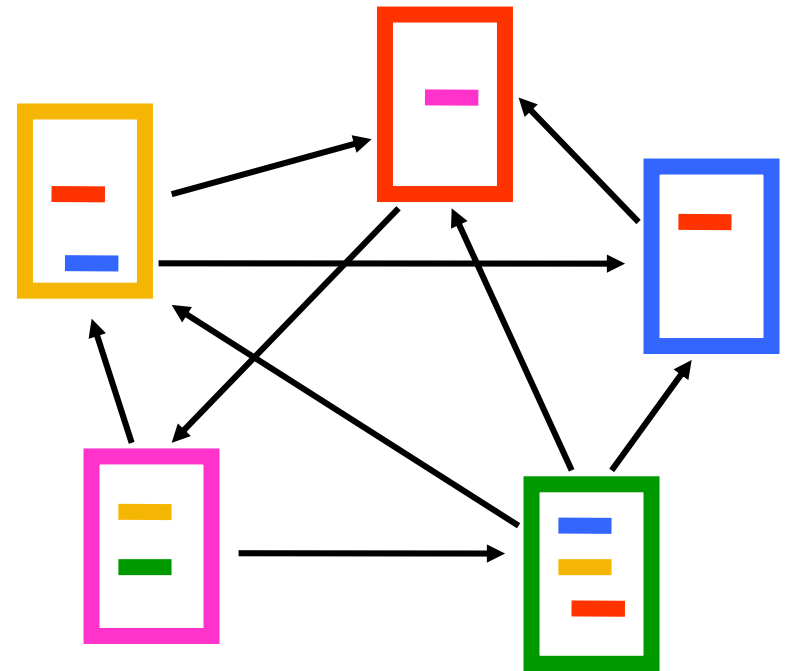
- What is the meaning of the stationary distribution  $\pi$  of a random walk?
- $\pi(i)$ : the probability of being at node  $i$  after very large (infinite) number of steps
- $\pi$  is the left eigenvector of transition matrix  $P$
- $\pi = p_0 P^\infty$ , where  $P$  is the transition matrix,  $p_0$  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.



# The PageRank random walk

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

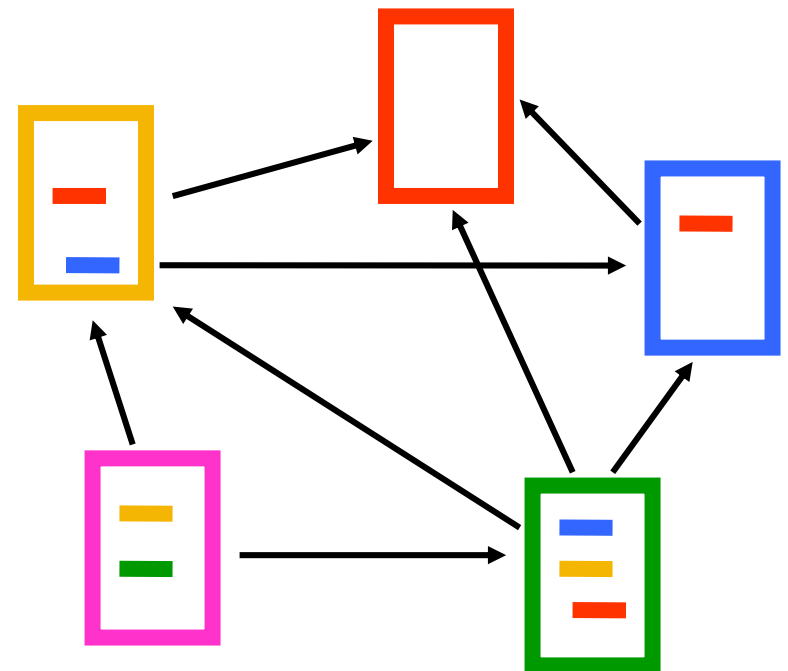
$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



# The PageRank random walk

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

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

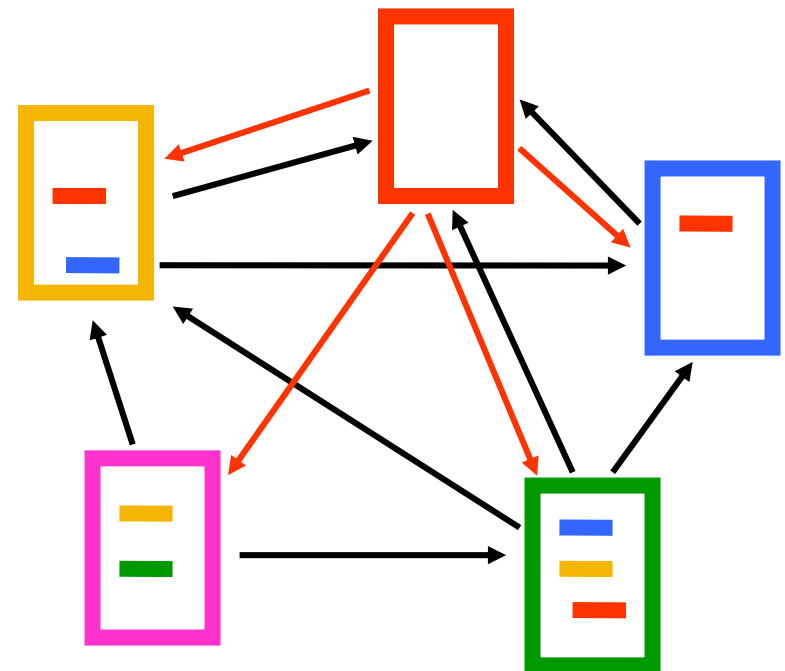


# The PageRank random walk

- Replace these row vectors with a vector  $\mathbf{v}$ 
  - typically, the uniform vector

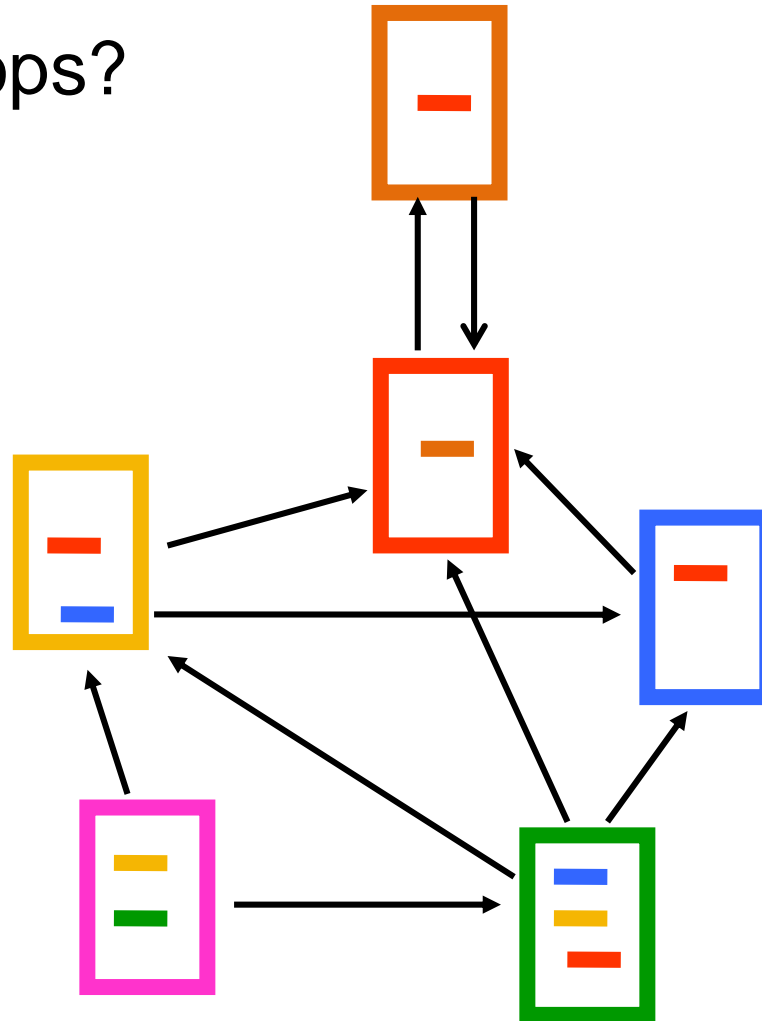
$$P' = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

$$P' = P + d\mathbf{v}^T \quad d = \begin{cases} 1 & \text{if } i \text{ is sink} \\ 0 & \text{otherwise} \end{cases}$$



# The PageRank random walk

- What about loops?
  - Spider traps



# The PageRank random walk

- Add a **random jump** to vector  $\mathbf{v}$  with prob  $1-\alpha$ 
  - typically, to a uniform vector
- Restarts after  $1/(1-\alpha)$  steps in expectation
  - Guarantees irreducibility, convergence

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

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

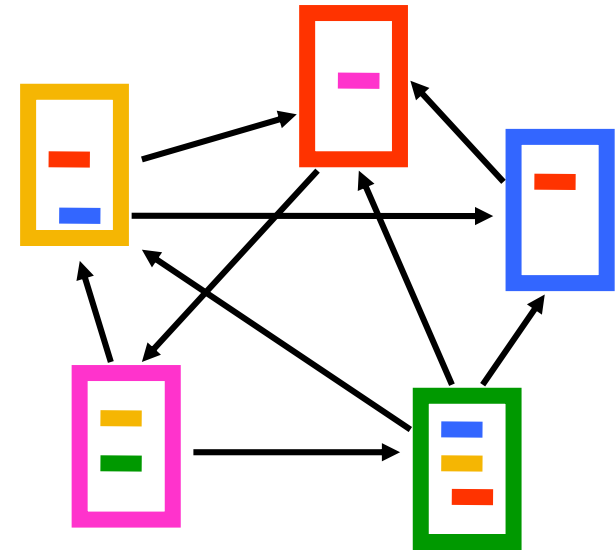
Random walk with restarts

# PageRank algorithm [BP98]

- The Random Surfer model
  - pick a page at random
  - with probability  $1 - \alpha$  jump to a random page
  - with probability  $\alpha$  follow a random outgoing link
- Rank according to the stationary distribution

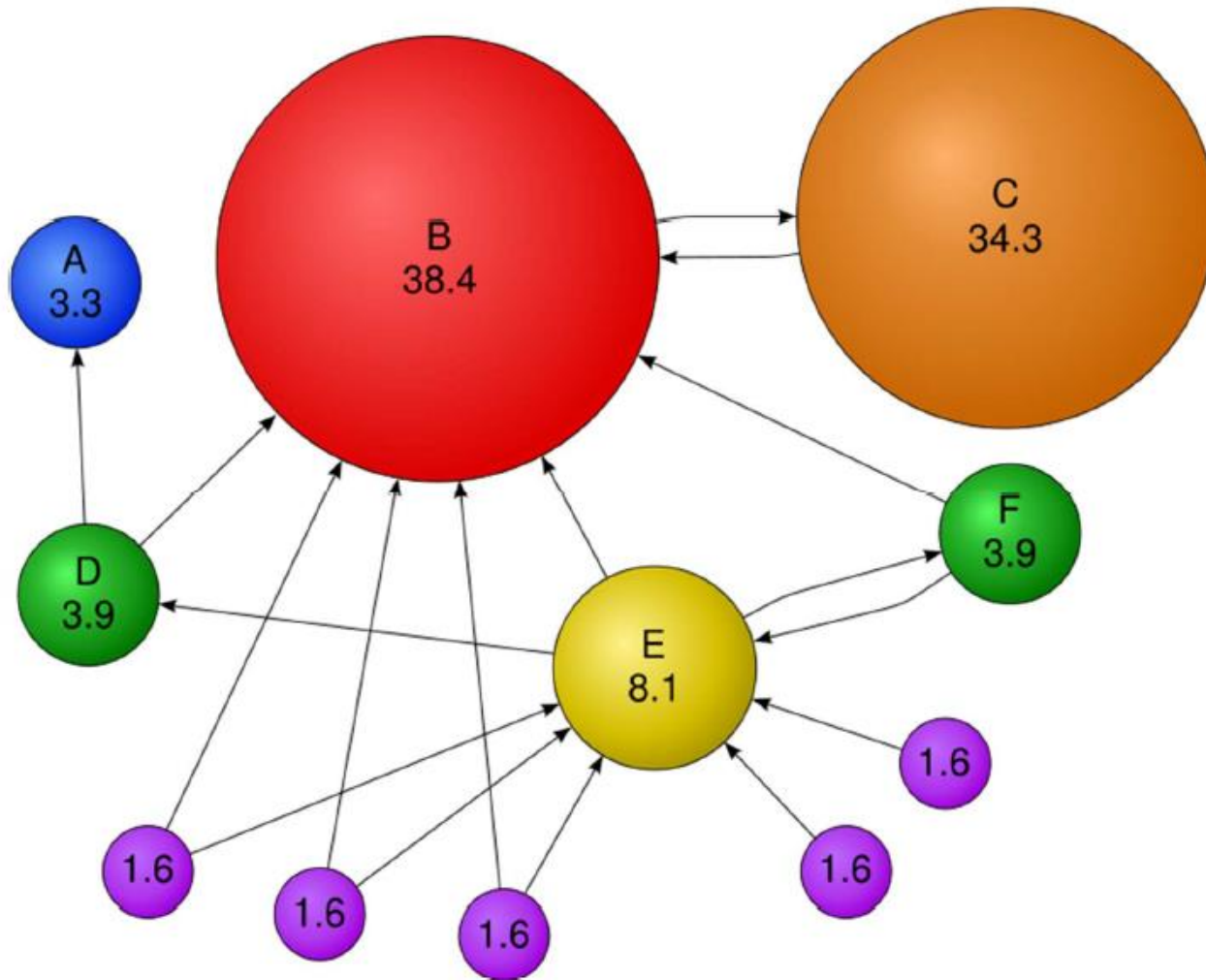
$$PR(p) = \alpha \sum_{q \rightarrow p} \frac{PR(q)}{|Out(q)|} + (1 - \alpha) \frac{1}{n}$$

$\alpha = 0.85$  in most cases



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

# PageRank: Example



# Stationary distribution with random jump

- If  $v$  is the jump vector

$$\begin{aligned}p^0 &= v \\p^1 &= \alpha p^0 P + (1 - \alpha)v = \alpha v P + (1 - \alpha)v \\p^2 &= \alpha p^1 P + (1 - \alpha)v = \alpha^2 v P^2 + (1 - \alpha)v \alpha P + (1 - \alpha)v \\&\vdots \\p^\infty &= (1 - \alpha)v + (1 - \alpha)v \alpha P + (1 - \alpha)v \alpha^2 P^2 + \dots \\&= (1 - \alpha)(I - \alpha P)^{-1}\end{aligned}$$

- 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 nodes that are **close** to  $v$ 
  - **Personalized** and **Topic-Specific** Pagerank.



# Effects of random jump

- Guarantees **convergence** to unique distribution
- 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  $\alpha$

# 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 **no longer true** if we do **random jumps**
  - Now the short paths play a greater role, and the previous distribution does not hold.

# Pagerank implementation

- Store the graph in adjacency list, or list of edges
- Keep current pagerank values and new pagerank values
- Go through edges and update the values of the destination nodes.
- Repeat until the difference ( $L_1$  or  $L_\infty$  difference) is below some small value  $\epsilon$ .

# A (Matlab-friendly) 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  $\delta < \epsilon$

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

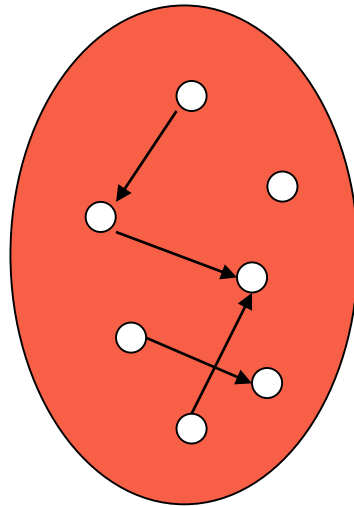
---

# 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

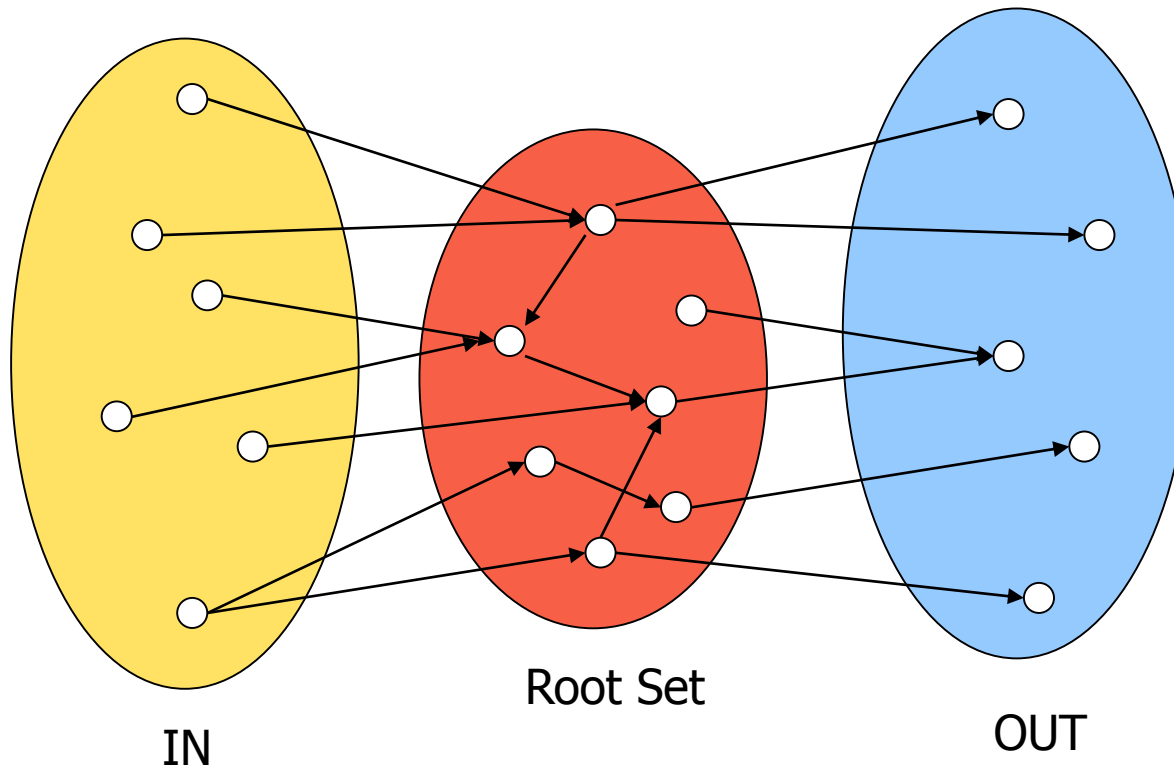
Root set obtained from a text-only search engine



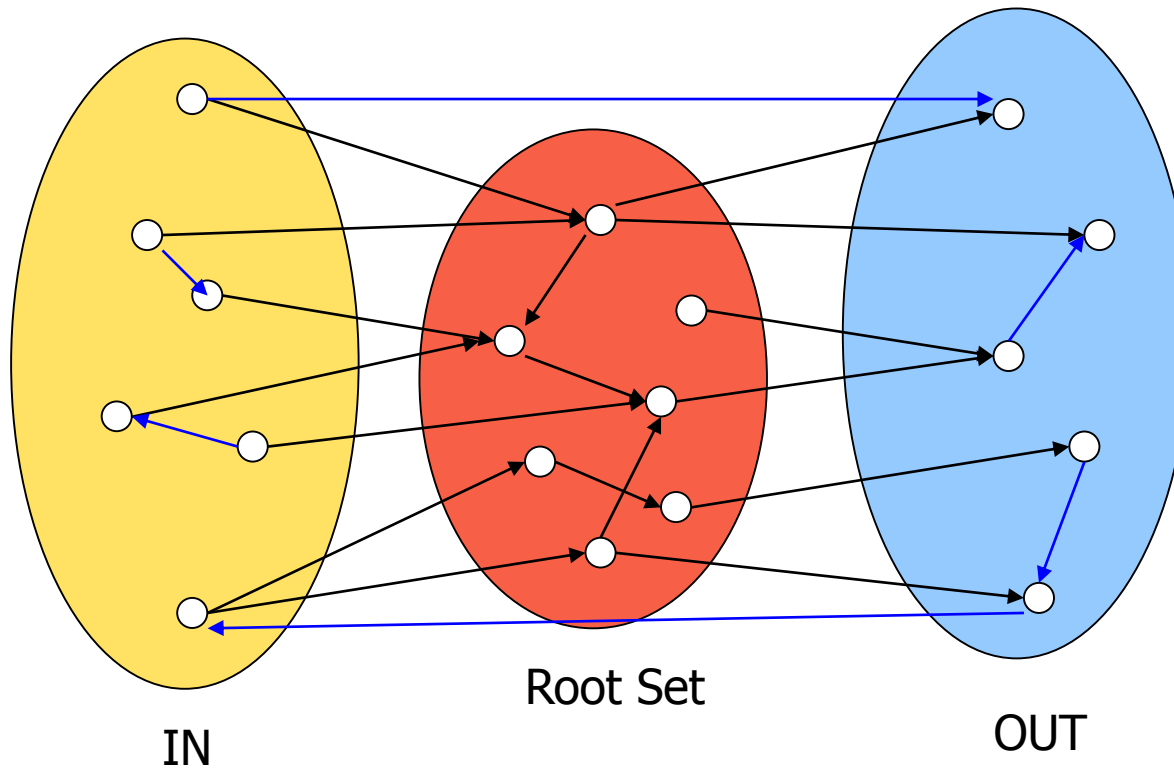
Root Set



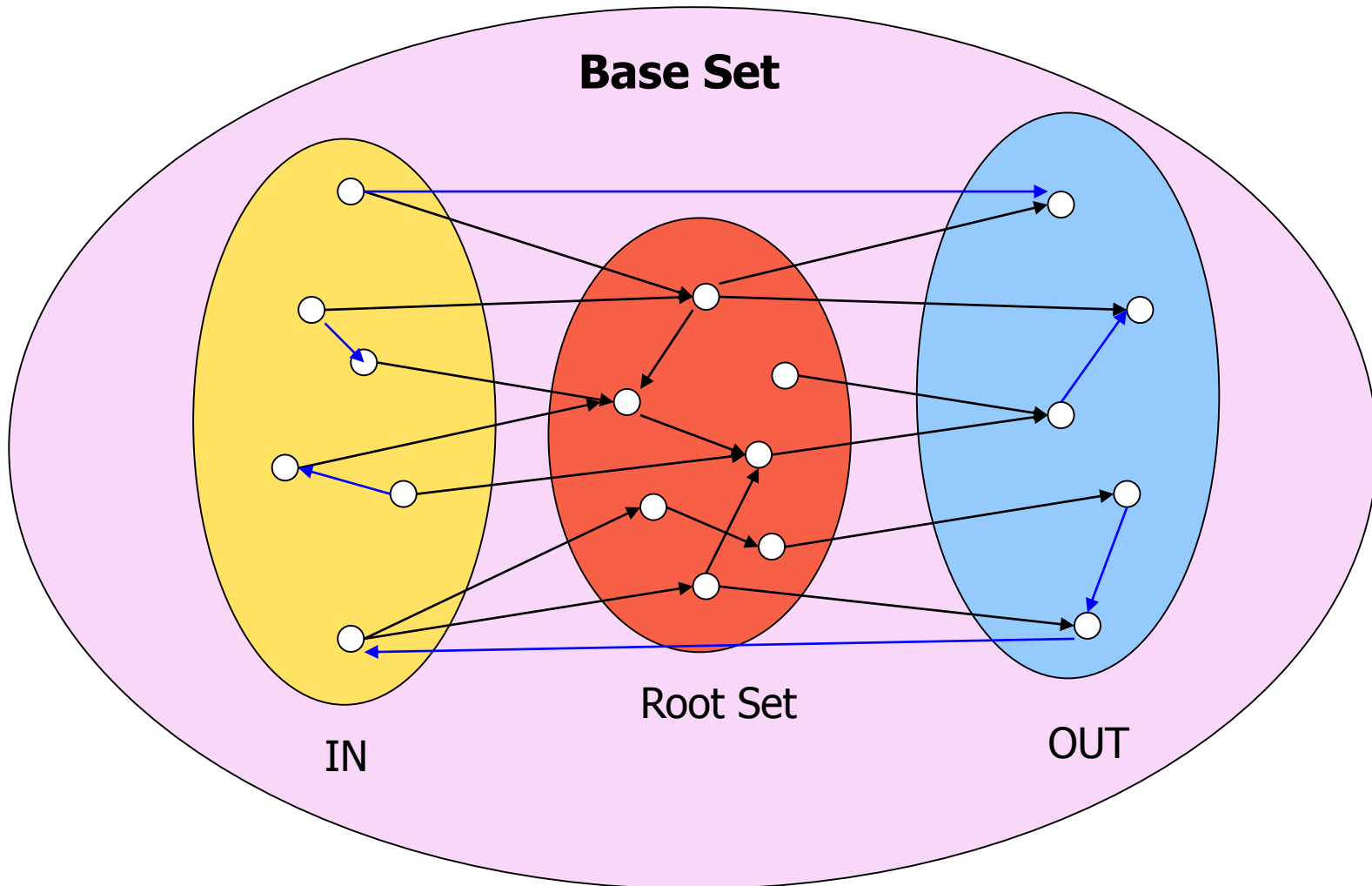
# Query dependent input



# Query dependent input

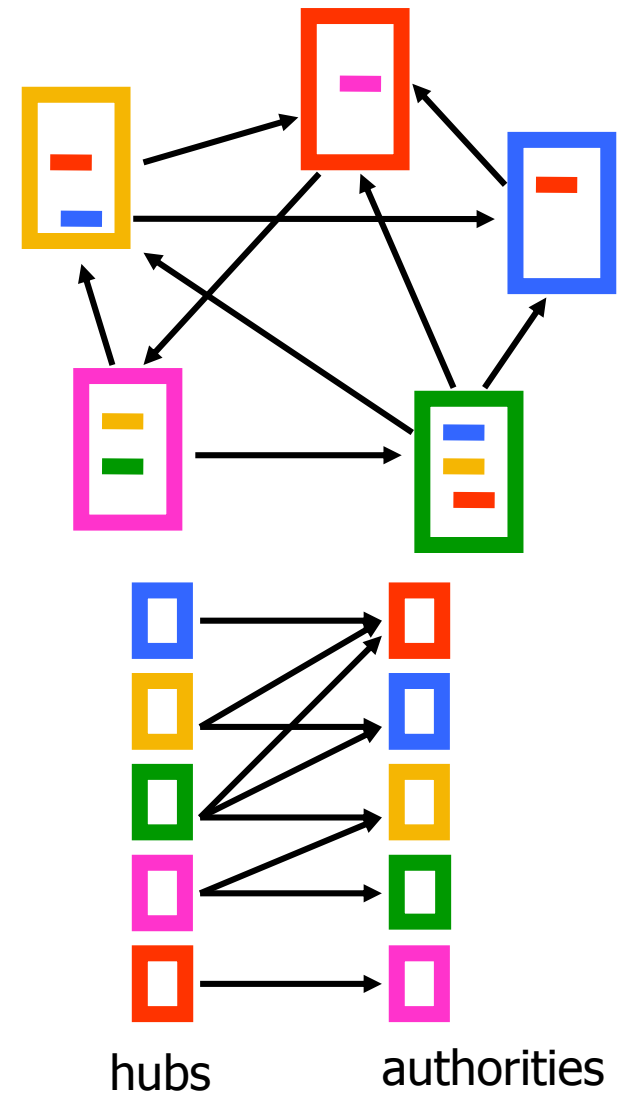


# Query dependent input



# 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

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

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

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

- 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  $h^t = A a^{t-1}$
  - $a^t = A^T A a^{t-1}$  and  $h^t = A A^T h^{t-1}$
  - Repeated iterations will converge to the eigenvectors
- The **authority** weight vector  $a$  is the **eigenvector** of  $A^T A$
- The **hub** weight vector  $h$  is the **eigenvector** of  $A A^T$
- The vectors  $a$  and  $h$  are called the **singular vectors** of the matrix  $A$

# Singular Value Decomposition

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

$[n \times r] \quad [r \times r] \quad [r \times n]$

- $r$  : rank of matrix  $A$
- $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$  : singular values (square roots of eig-vals  $AA^T, A^T A$ )
- $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_r$  : left singular vectors (eig-vectors of  $AA^T$ )
- $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r$  : right singular vectors (eig-vectors of  $A^T A$ )

$$A = \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_r \vec{u}_r \vec{v}_r^T$$

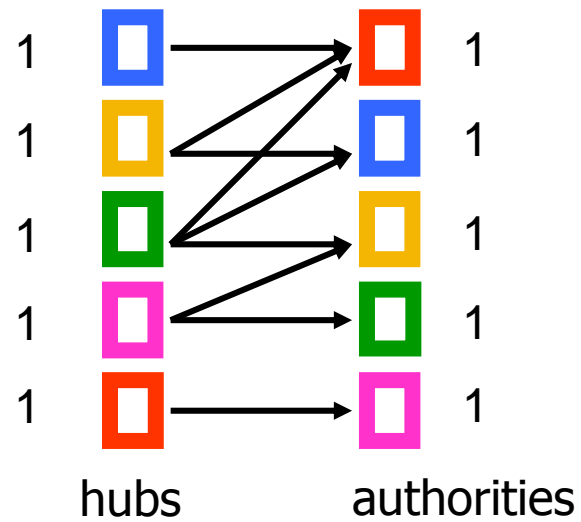


# Why does the Power Method work?

- If a matrix  $R$  is **real and symmetric**, it has real eigenvalues and eigenvectors:  $(\lambda_1, w_1), (\lambda_2, w_2), \dots, (\lambda_r, w_r)$ 
  - $r$  is the rank of the matrix
  - $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_r|$
- For any matrix  $R$ , the eigenvectors  $w_1, w_2, \dots, w_r$  of  $R$  define **a basis of the vector space**
  - For any vector  $x$ ,  $Rx = \alpha_1 w_1 + \alpha_2 w_2 + \dots + \alpha_r w_r$
- After  $t$  multiplications we have:
$$R^t x = \lambda_1^{t-1} \alpha_1 w_1 + \lambda_2^{t-1} \alpha_2 w_2 + \dots + \lambda_r^{t-1} \alpha_r w_r$$
- Normalizing leaves only the term  $w_1$ .

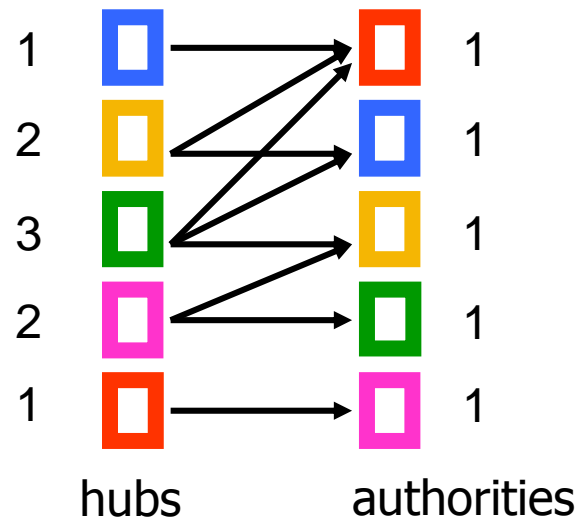
# Example

Initialize



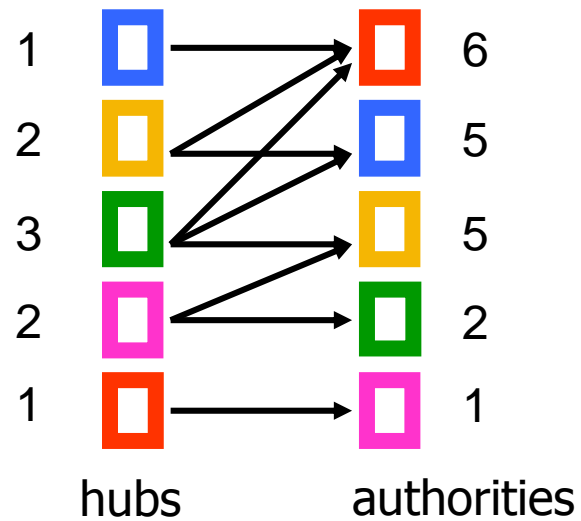
# Example

Step 1: O operation



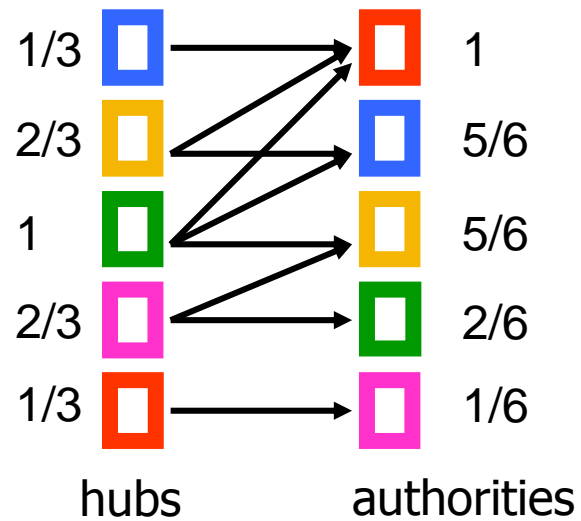
# Example

Step 1: I operation



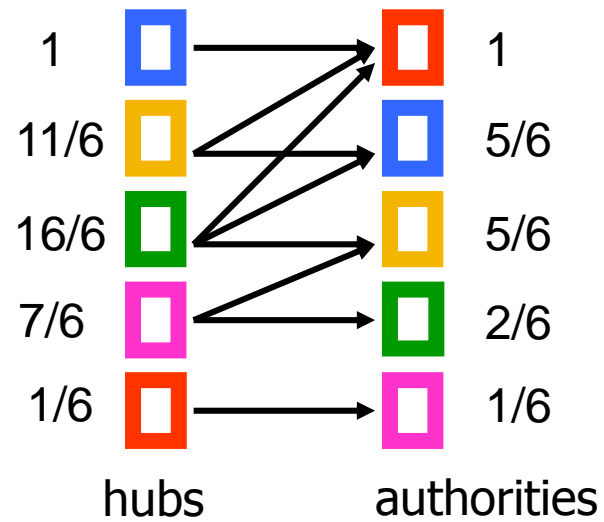
# Example

Step 1: Normalization (Max norm)



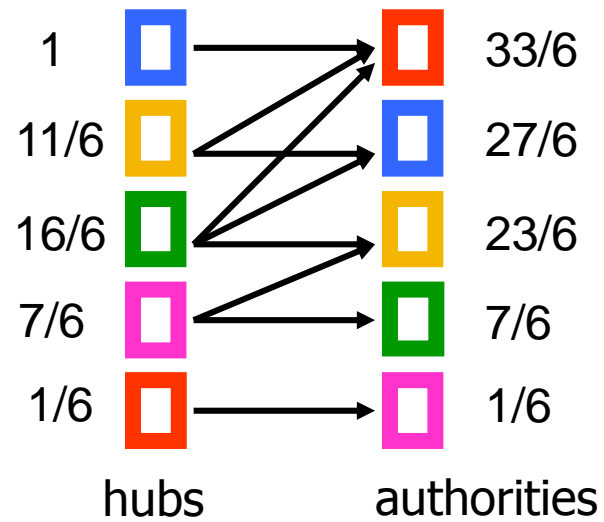
# Example

Step 2: O step



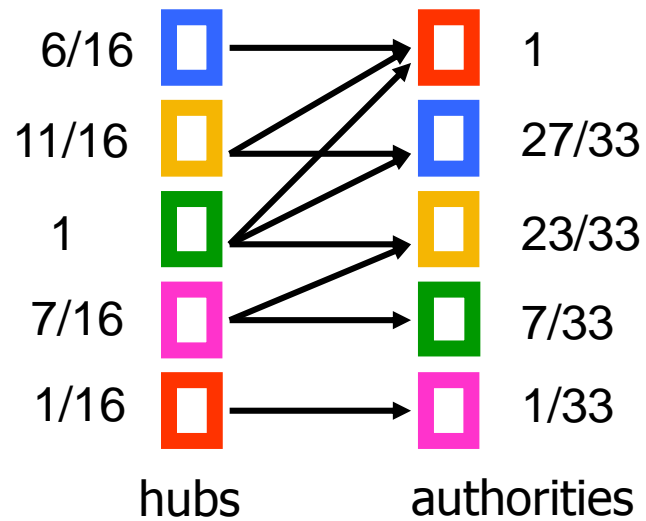
# Example

Step 2: 1 step



# Example

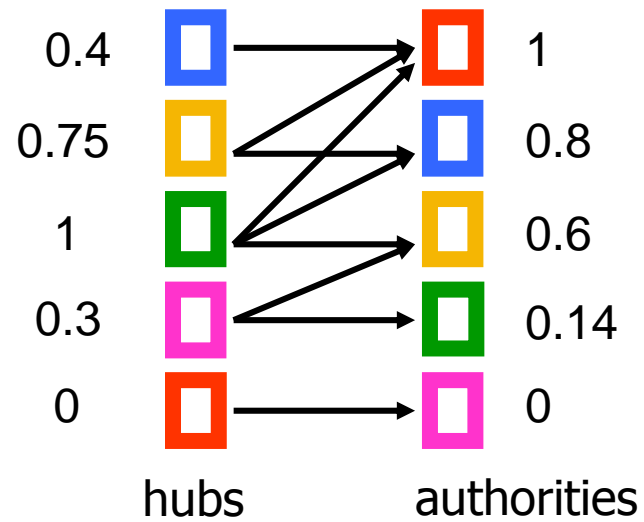
Step 2: Normalization





# Example

Convergence

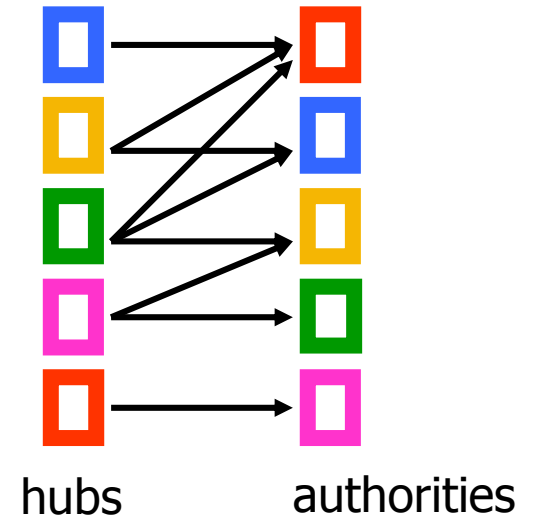


# OTHER ALGORITHMS

---

# 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

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

- 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  $j$