

Community Detection in Graphs

CS246: Mining Massive Datasets
Jure Leskovec, Stanford University
<http://cs246.stanford.edu>



Link Analysis:

HITS: Hubs and Authorities

Hubs and Authorities

- **HITS (Hypertext-Induced Topic Selection)**
 - Is a measure of importance of pages or documents, similar to PageRank
 - Proposed at around same time as PageRank ('98)
- **Goal:** Say we want to find good newspapers
 - Don't just find newspapers. Find "experts" – people who link in a coordinated way to good newspapers
- **Idea: Links as votes**
 - **Page is more important if it has more links**
 - In-coming links? Out-going links?

Finding newspapers

■ Hubs and Authorities

Each page has 2 scores:

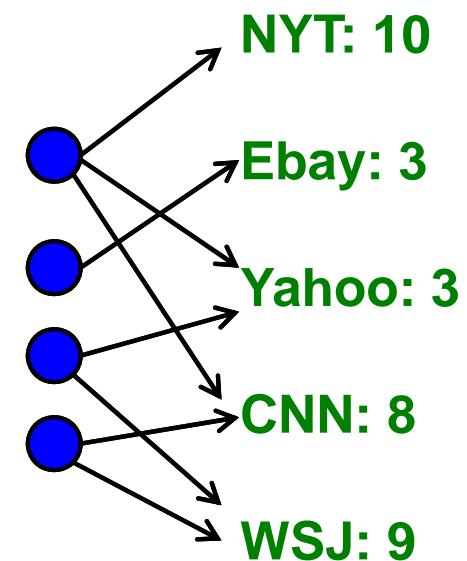
- **Quality as an expert (hub):**

- Total sum of votes of authorities pointed to

- **Quality as a content (authority):**

- Total sum of votes coming from experts

■ Principle of repeated improvement



Hubs and Authorities

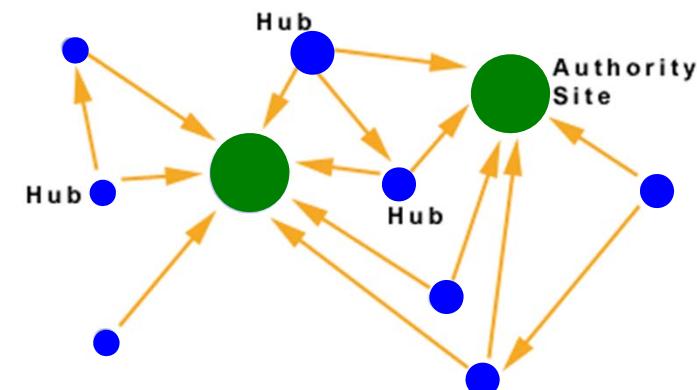
Interesting pages fall into two classes:

1. **Authorities** are pages containing useful information

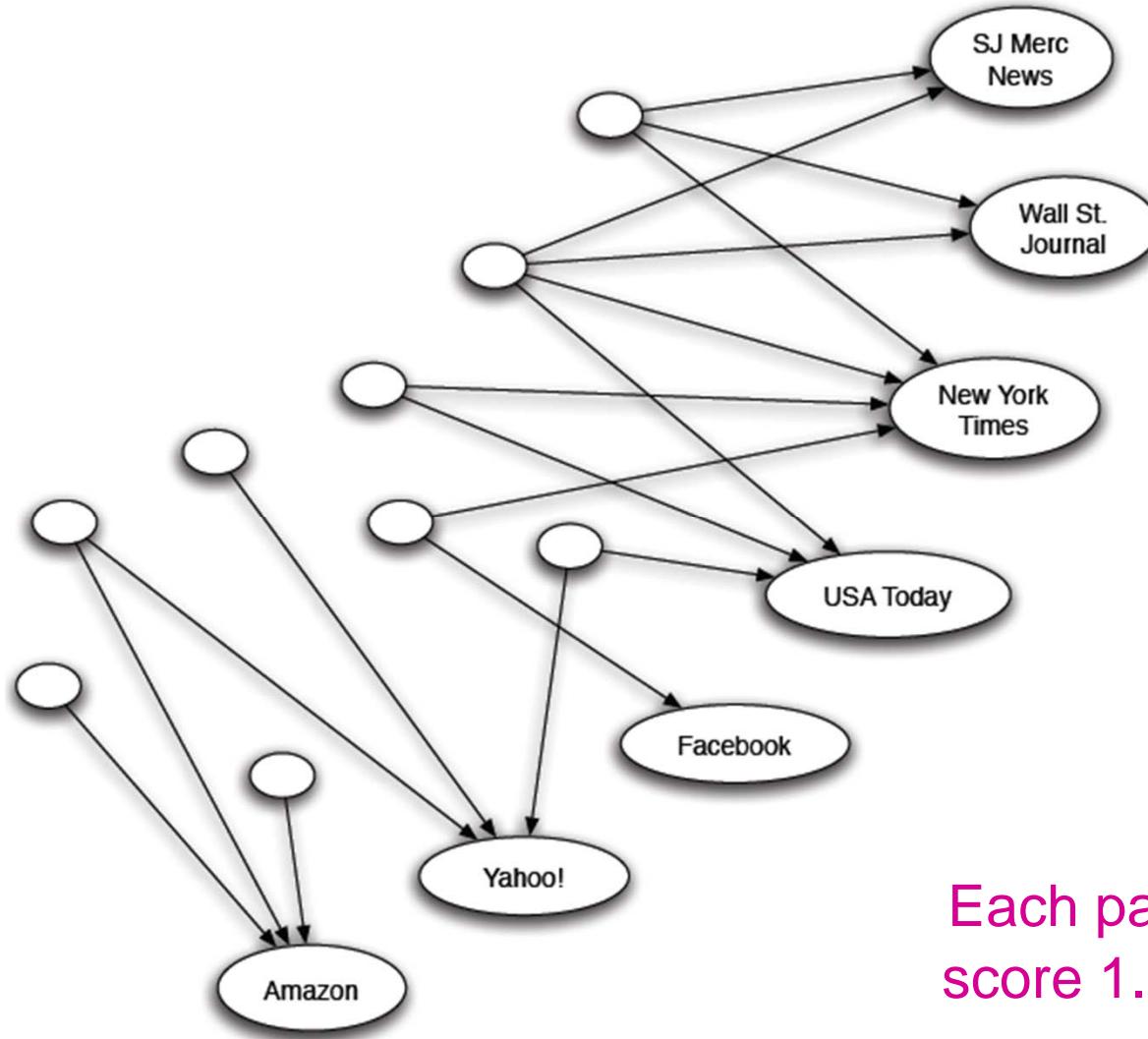
- Newspaper home pages
- Course home pages
- Home pages of auto manufacturers

2. **Hubs** are pages that link to authorities

- List of newspapers
- Course bulletin
- List of US auto manufacturers



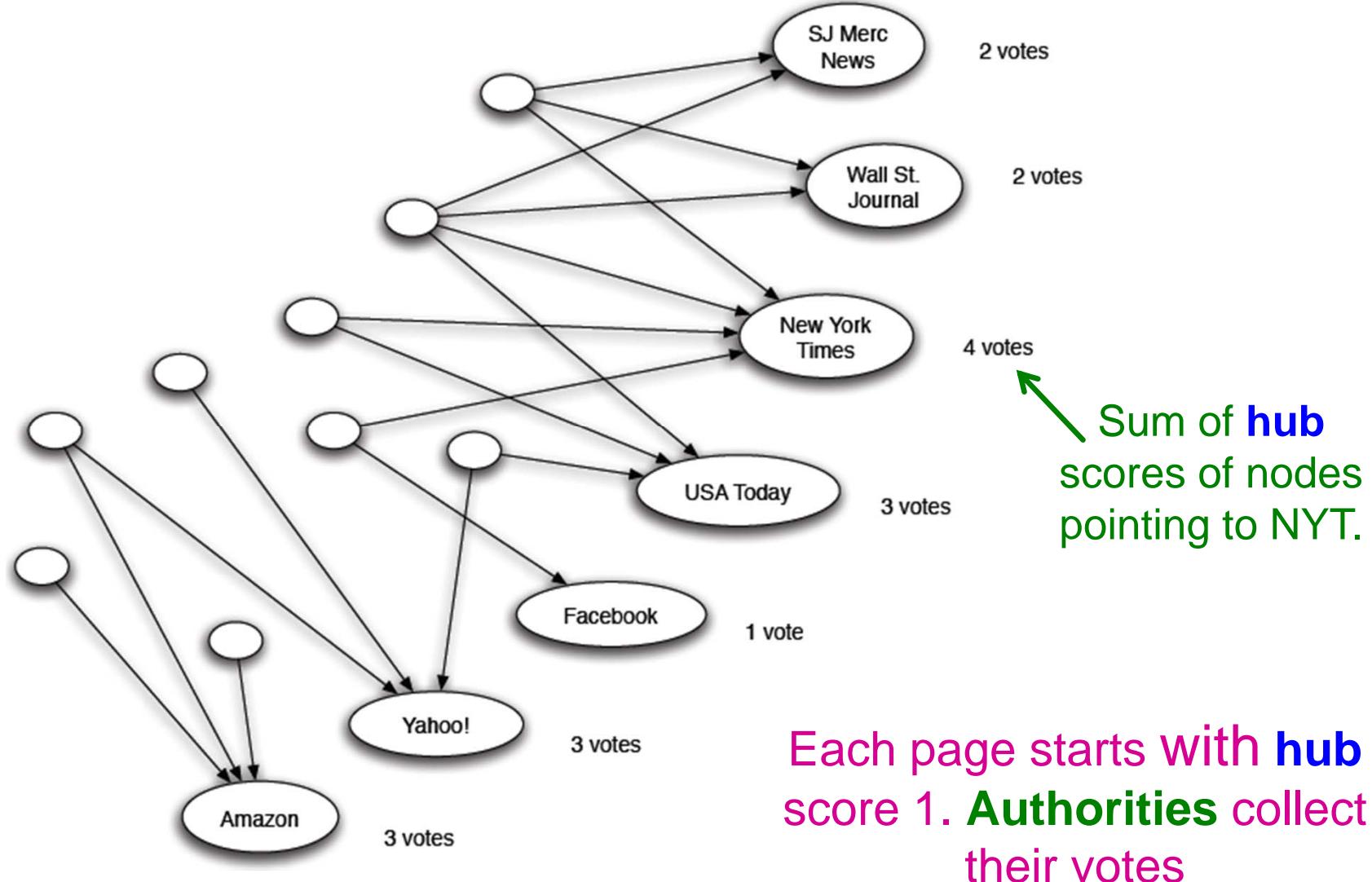
Counting in-links: Authority



Each page starts with hub score 1. Authorities collect their votes

(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

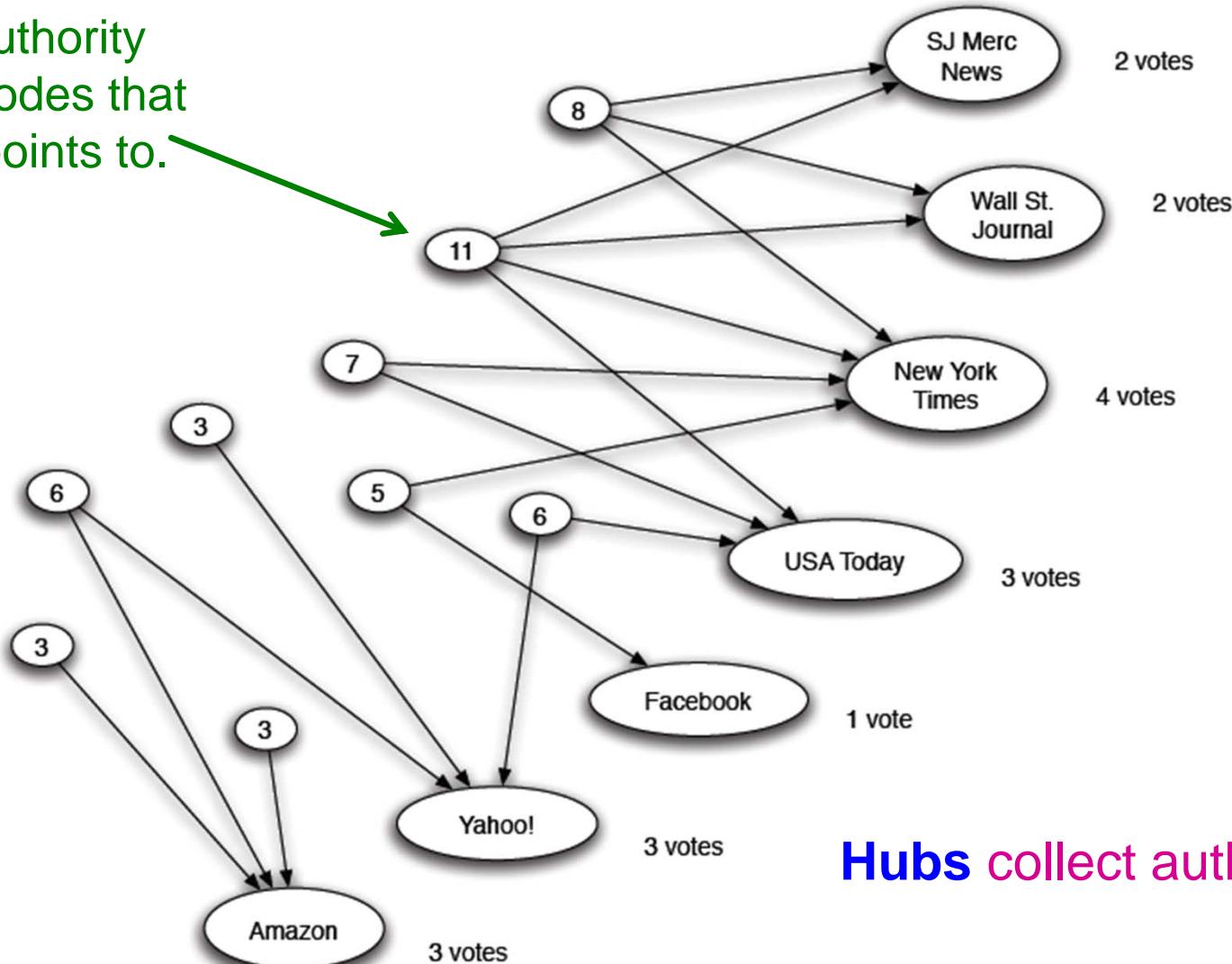
Counting in-links: Authority



(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

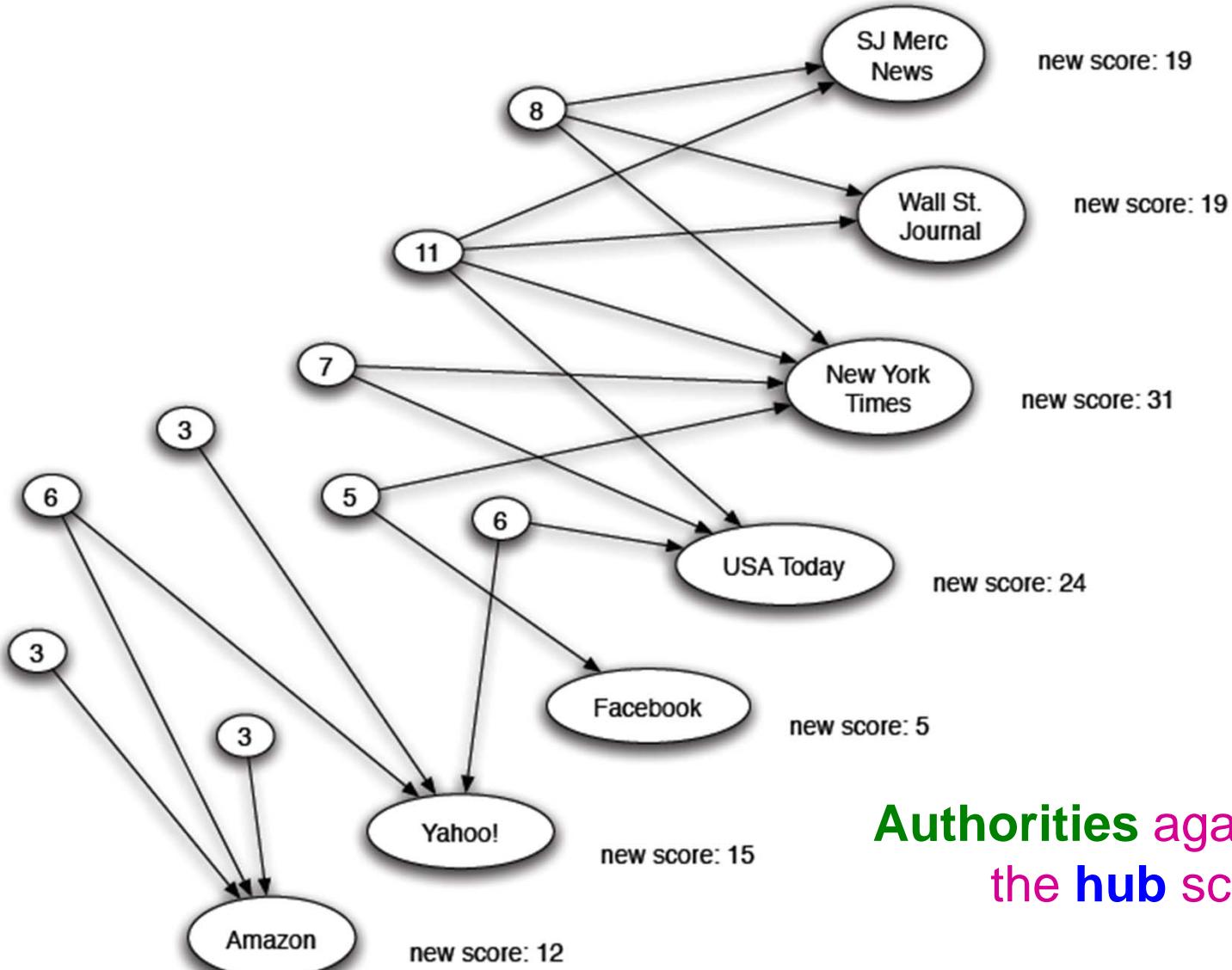
Expert Quality: Hub

Sum of authority scores of nodes that the node points to.



(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Reweighting



**Authorities again collect
the hub scores**

(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Mutually Recursive Definition

- A good hub links to many good authorities
- A good authority is linked from many good hubs
- Model using two scores for each node:
 - Hub score and Authority score
 - Represented as vectors h and a

Hubs and Authorities

- Each page i has 2 scores:

- Authority score: a_i
- Hub score: h_i

HITS algorithm: n...number of node in a graph

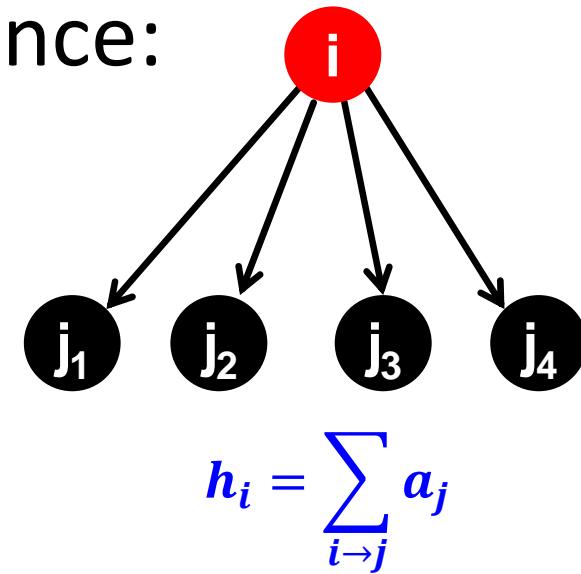
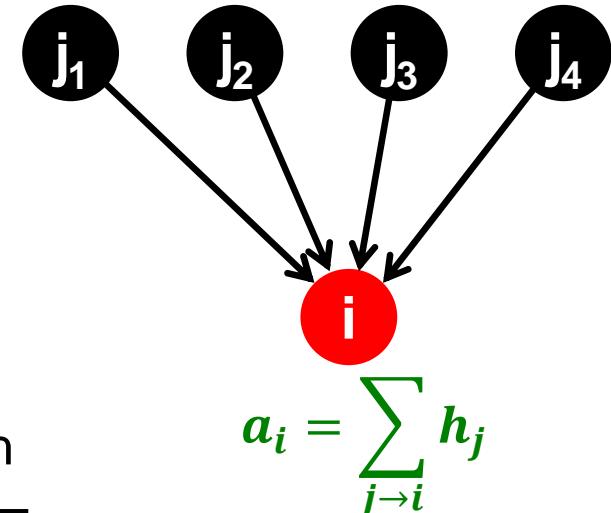
- Initialize: $a_i = 1/\sqrt{n}$, $h_i = 1/\sqrt{n}$
- Then keep iterating until convergence:

- $\forall i$: **Authority**: $a_i = \sum_{j \rightarrow i} h_j$

- $\forall i$: **Hub**: $h_i = \sum_{i \rightarrow j} a_j$

- $\forall i$: Normalize a, h such that:

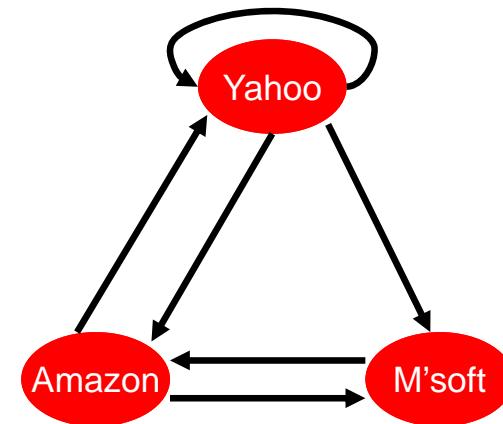
$$\sum_i a_i^2 = 1, \sum_i h_i^2 = 1$$



Example

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

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



$$h(\text{yahoo}) = .58 \quad .80 \quad .80 \quad .79 \quad \dots \quad .788$$

$$h(\text{amazon}) = .58 \quad .53 \quad .53 \quad .57 \quad \dots \quad .577$$

$$h(\text{m'soft}) = .58 \quad .27 \quad .27 \quad .23 \quad \dots \quad .211$$

$$a(\text{yahoo}) = .58 \quad .58 \quad .62 \quad .62 \quad \dots \quad .628$$

$$a(\text{amazon}) = .58 \quad .58 \quad .49 \quad .49 \quad \dots \quad .459$$

$$a(\text{m'soft}) = .58 \quad .58 \quad .62 \quad .62 \quad \dots \quad .628$$

Hubs and Authorities

- HITS converges to a single stable point
- Notation:
 - Vector $a = (a_1 \dots, a_n)$, $h = (h_1 \dots, h_n)$
 - Adjacency matrix A ($n \times n$): $A_{ij} = 1$ if $i \rightarrow j$
- Then $h_i = \sum_{i \rightarrow j} a_j$
can be rewritten as $h_i = \sum_j A_{ij} \cdot a_j$

So: $h = A \cdot a$
- Similarly, $a_i = \sum_{j \rightarrow i} h_j$
can be rewritten as $a_i = \sum_j A_{ji} \cdot h_j = A^T \cdot h$

Hub and Authority Equations

- The **hub** score of page i is proportional to the sum of the **authority** scores of the pages it links to: $\mathbf{h} = \lambda \mathbf{A} \mathbf{a}$
 - λ is a scale factor: $\lambda = \frac{1}{\sqrt{\sum_i h_i^2}}$
- The **authority** score of page i is proportional to the sum of the **hub** scores of the pages it is linked from: $\mathbf{a} = \mu \mathbf{A}^T \mathbf{h}$
 - μ is scale factor: $\mu = \frac{1}{\sqrt{\sum_i a_i^2}}$

Hubs and Authorities

■ HITS algorithm in vector notation:

- Set: $a_i = h_i = \frac{1}{\sqrt{n}}$

Repeat until convergence:

- $h = A \cdot a$
- $a = A^T \cdot h$
- Normalize a and h
- Then: $a = A^T \cdot (\underbrace{A \cdot a}_{\text{new } h})$
- Thus, in $2k$ steps:
$$a = (A^T \cdot A)^k \cdot a$$
$$h = (A \cdot A^T)^k \cdot h$$

Convergence criterion:

$$\sum_i (h_i^{(t)} - h_i^{(t-1)})^2 < \varepsilon$$

$$\sum_i (a_i^{(t)} - a_i^{(t-1)})^2 < \varepsilon$$

a is updated (in 2 steps):

$$a = A^T(Aa) = (A^TA)a$$

h is updated (in 2 steps):

$$h = A(A^Th) = (AA^T)h$$

Repeated matrix powering

Existence and Uniqueness

- $\mathbf{h} = \lambda A \mathbf{a}$ $\lambda = 1/\sqrt{\sum_i h_i^2}$
- $\mathbf{a} = \mu A^T \mathbf{h}$ $\mu = 1/\sqrt{\sum_i a_i^2}$
- $\mathbf{h} = \lambda \mu A A^T \mathbf{h}$
- $\mathbf{a} = \lambda \mu A^T A \mathbf{a}$

- Under reasonable assumptions about A ,
HITS **converges to vectors \mathbf{h}^* and \mathbf{a}^* :**
 - \mathbf{h}^* is the **principal eigenvector** of matrix $A A^T$
 - \mathbf{a}^* is the **principal eigenvector** of matrix $A^T A$

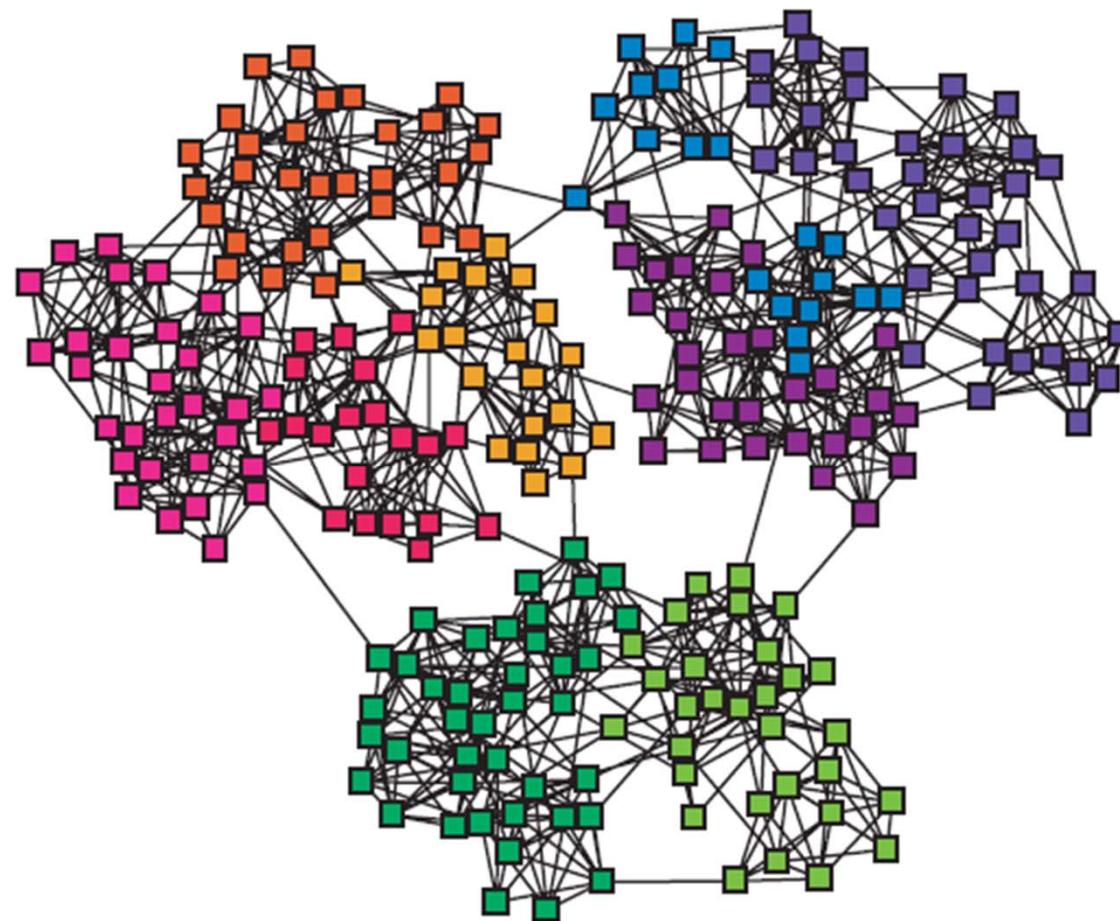
PageRank and HITS

- PageRank and HITS are two solutions to the same problem:
 - What is the value of an in-link from u to v ?
 - In the PageRank model, the value of the link depends on the links into u
 - In the HITS model, it depends on the value of the other links out of u
- The destinies of PageRank and HITS post-1998 were very different

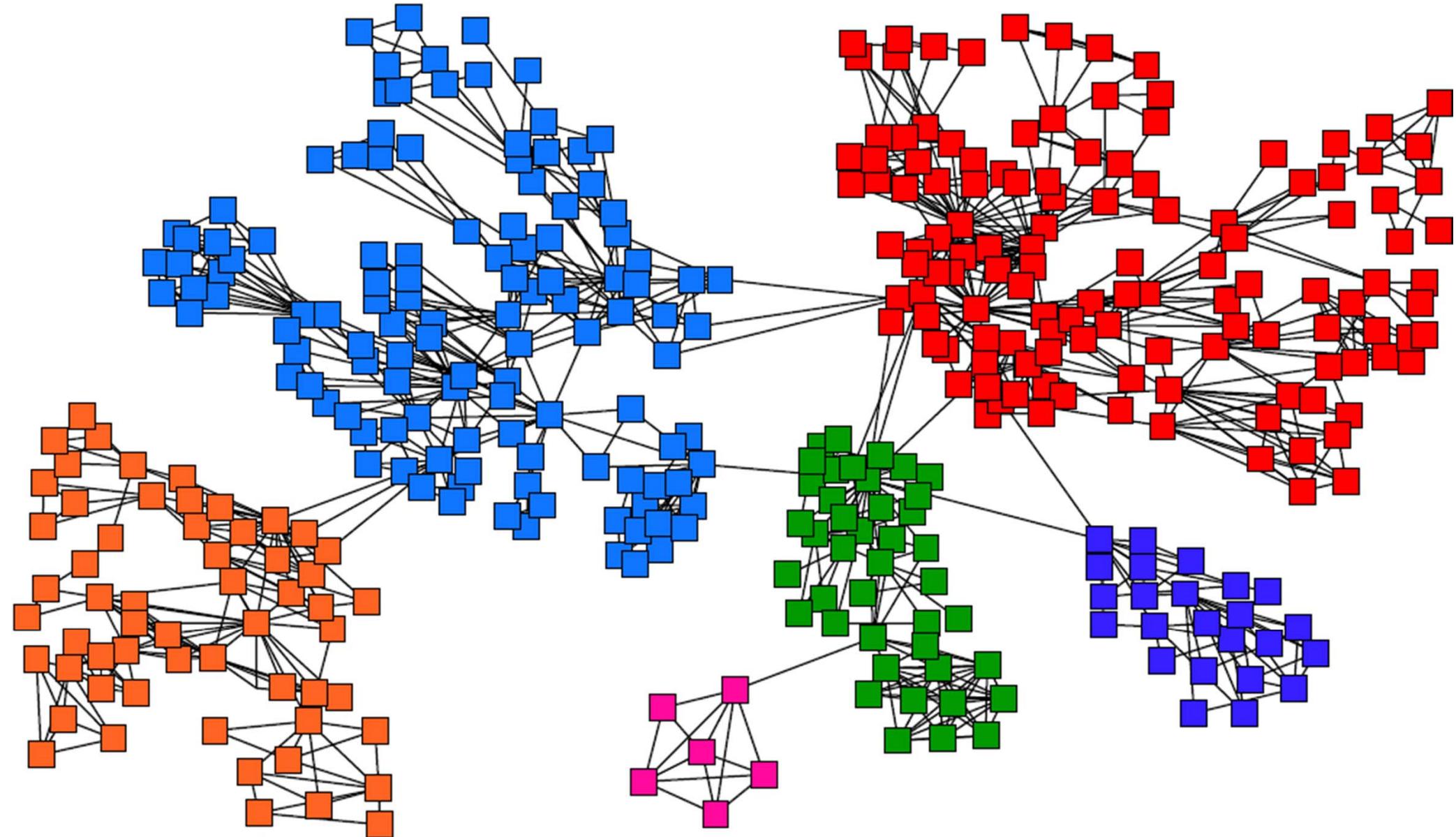
New topic:
**Finding Communities in
Networks using PageRank**

Networks & Communities

- We often think of networks being organized into **modules, cluster, communities**:

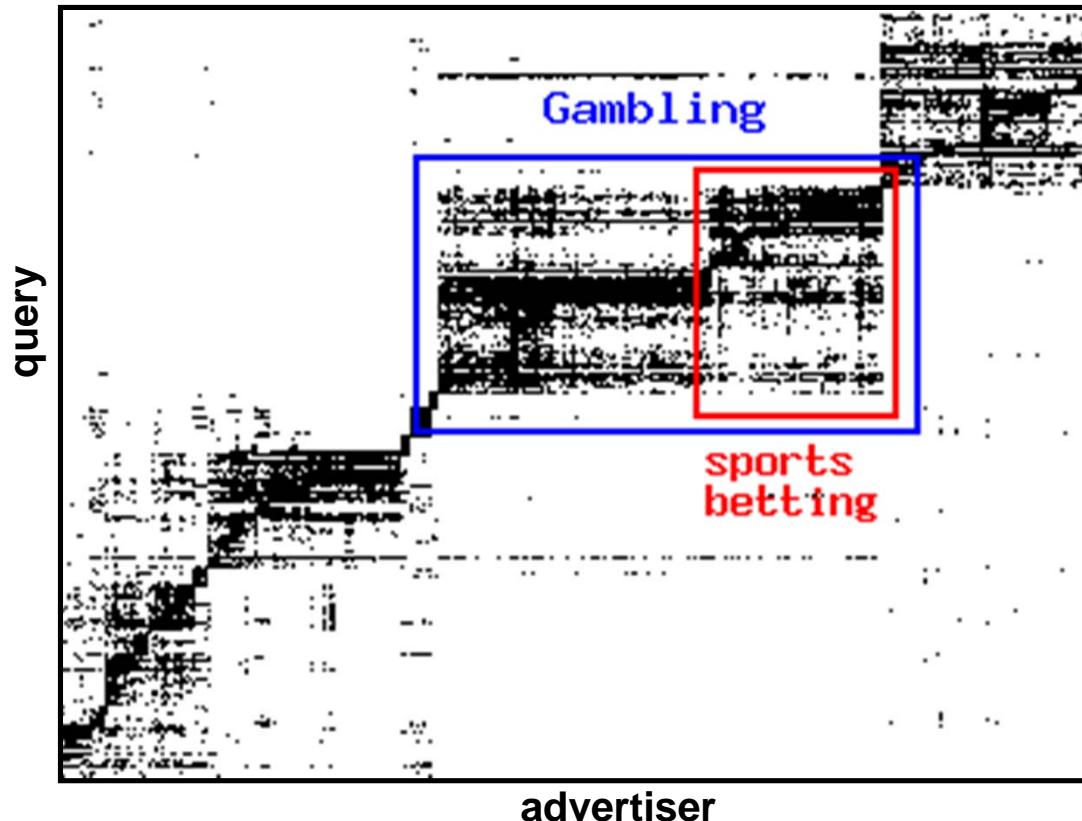


Goal: Find Densely Linked Clusters



Micro-Markets in Sponsored Search

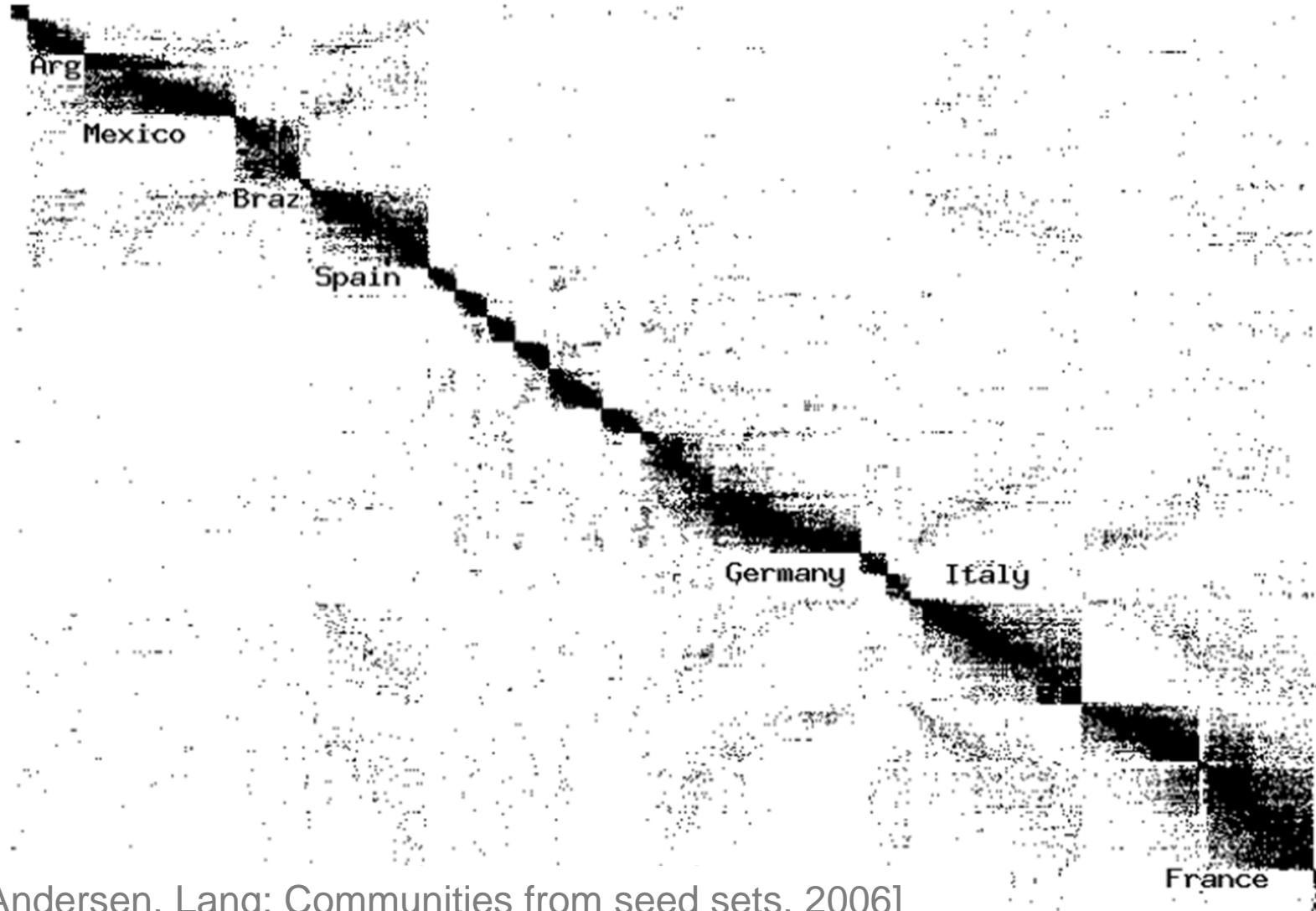
- Find micro-markets by partitioning the query-to-advertiser graph:



[Andersen, Lang: Communities from seed sets, 2006]

Movies and Actors

■ Clusters in Movies-to-Actors graph:

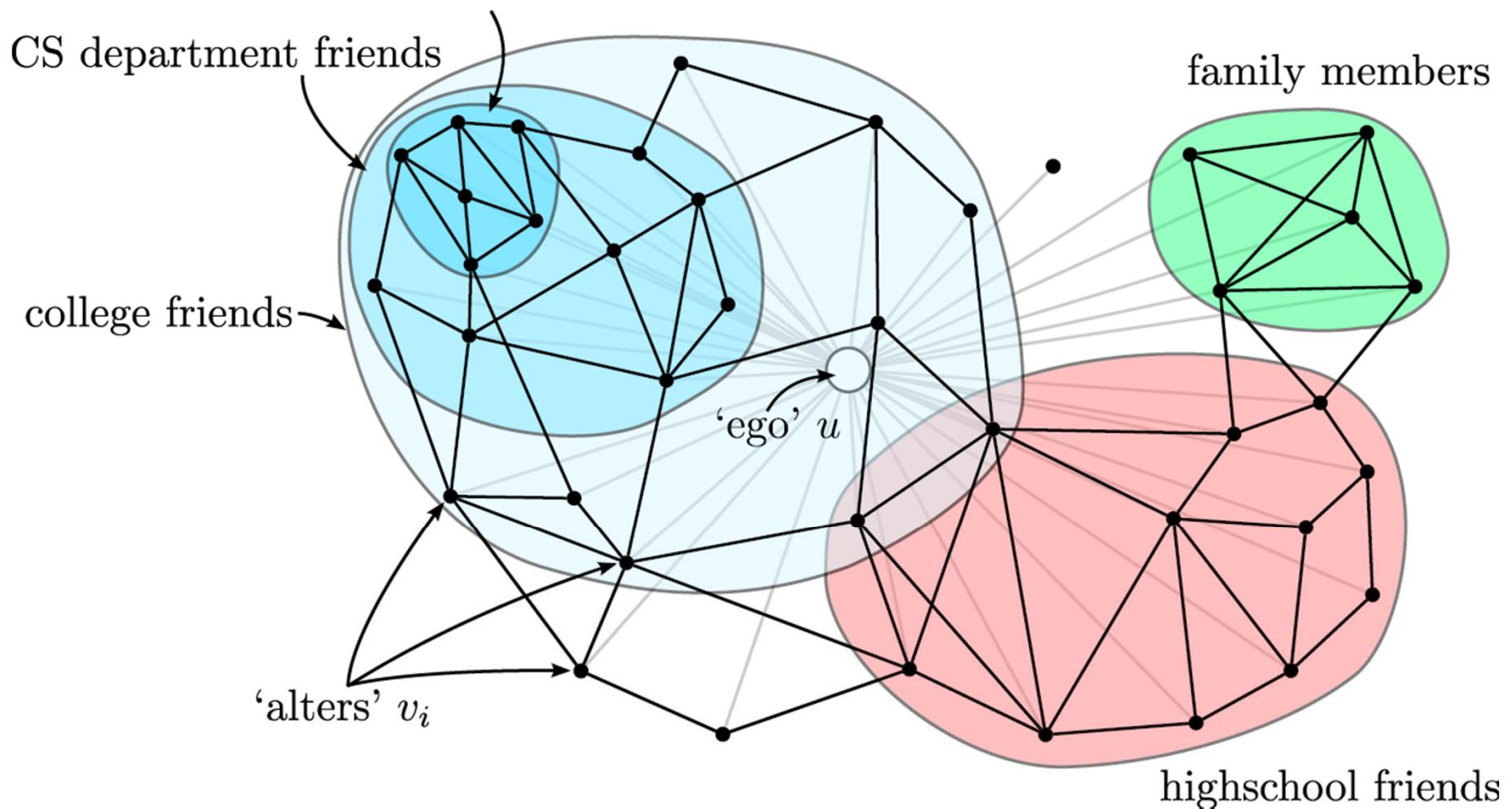


[Andersen, Lang: Communities from seed sets, 2006]

Twitter & Facebook

■ Discovering social circles, circles of trust:

friends under the same advisor



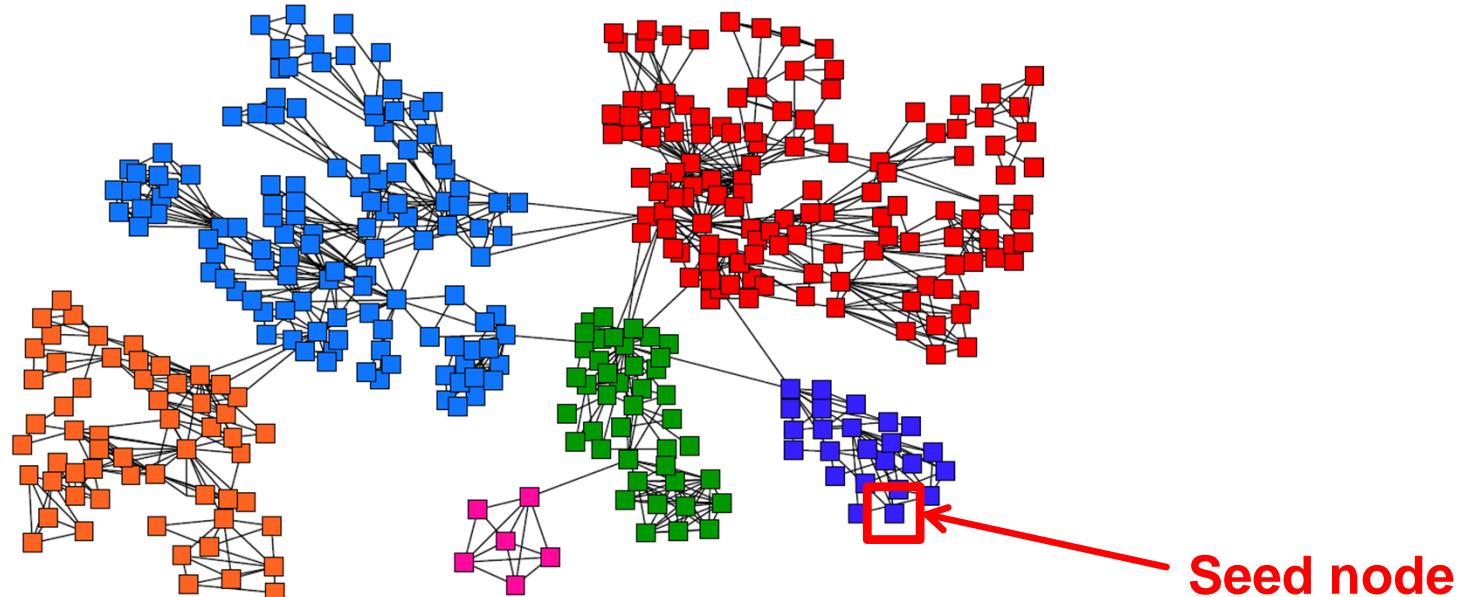
[McAuley, Leskovec: Discovering social circles in ego networks, 2012]

The Setting

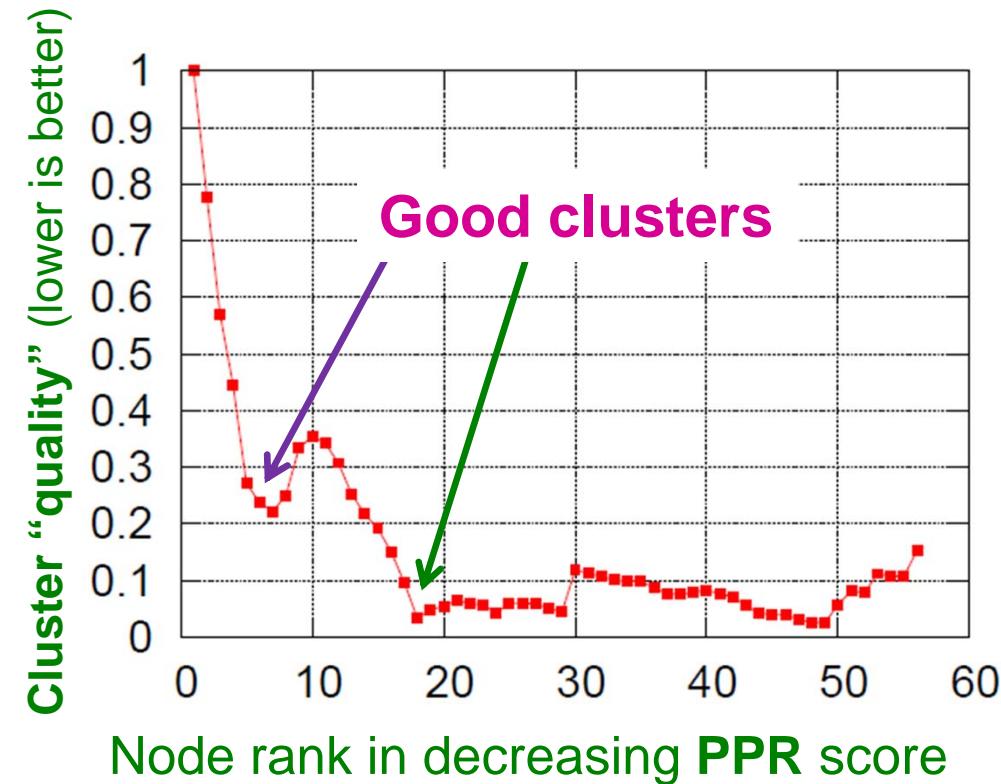
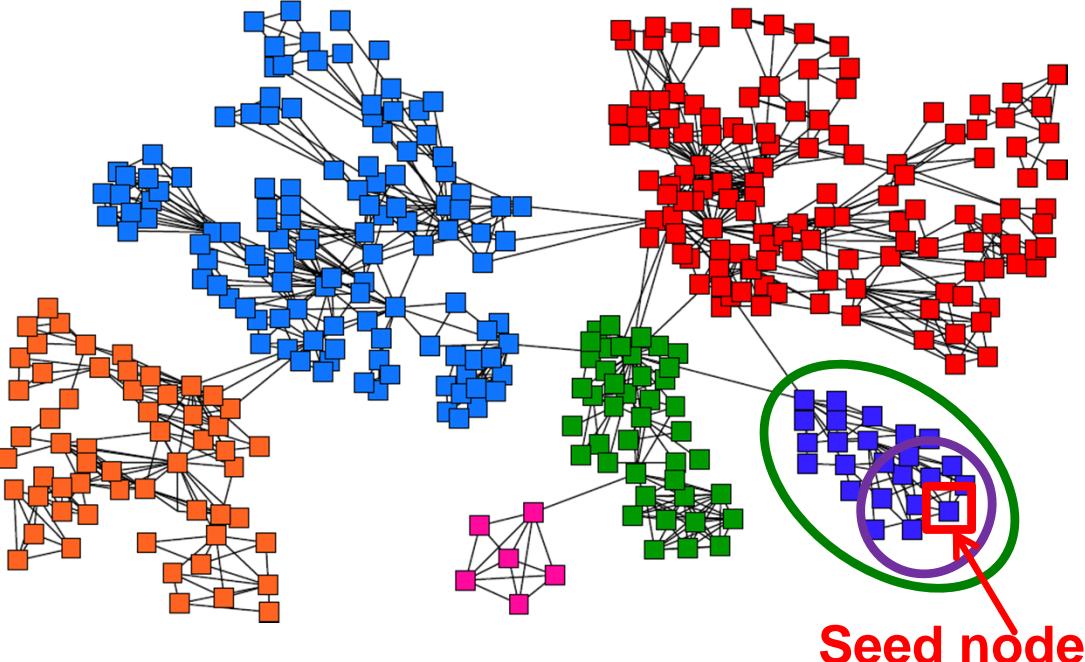
- **Graph is large**
 - Assume the graph fits in main memory
 - For example, to work with a 200M node and 2B edge graph one needs approx. 16GB RAM
 - But the graph is too big for running anything more than linear time algorithms
- **We will cover a PageRank based algorithm for finding dense clusters**
 - The runtime of the algorithm will be proportional to the cluster size (not the graph size!)

Idea: Seed Nodes

- **Discovering clusters based on seed nodes**
 - **Given:** Seed node S
 - Compute (approximate) Personalized PageRank (**PPR**) around node S (teleport set= $\{S\}$)
 - Idea is that if S belongs to a nice cluster, the random walk will get **trapped** inside the cluster



Seed Node: Intuition

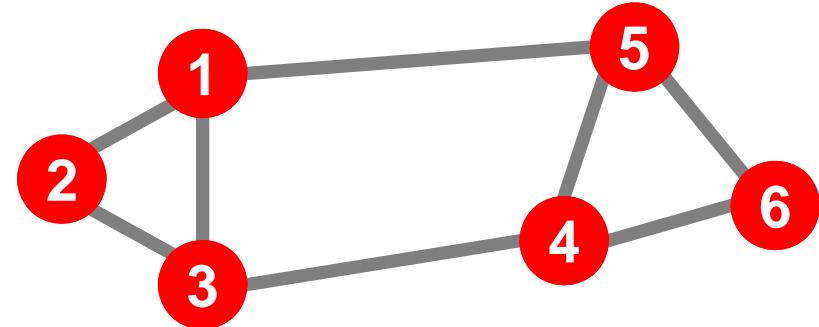


Algorithm outline:

- Pick a seed node S of interest
- Run PPR with teleport set = $\{S\}$
- Sort the nodes by the decreasing PPR score
- Sweep** over the nodes and find **good clusters**

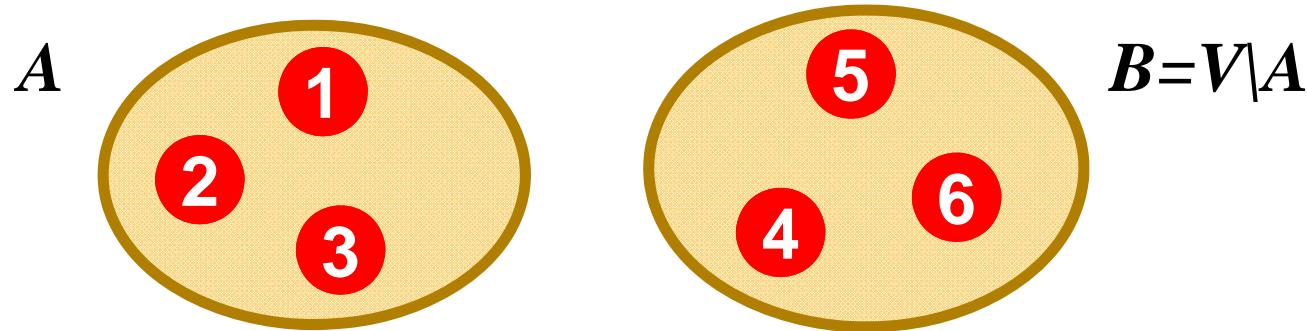
What makes a good cluster?

- Undirected graph $G(V, E)$:



- Partitioning task:

- Divide vertices into 2 disjoint groups $A, B = V \setminus A$

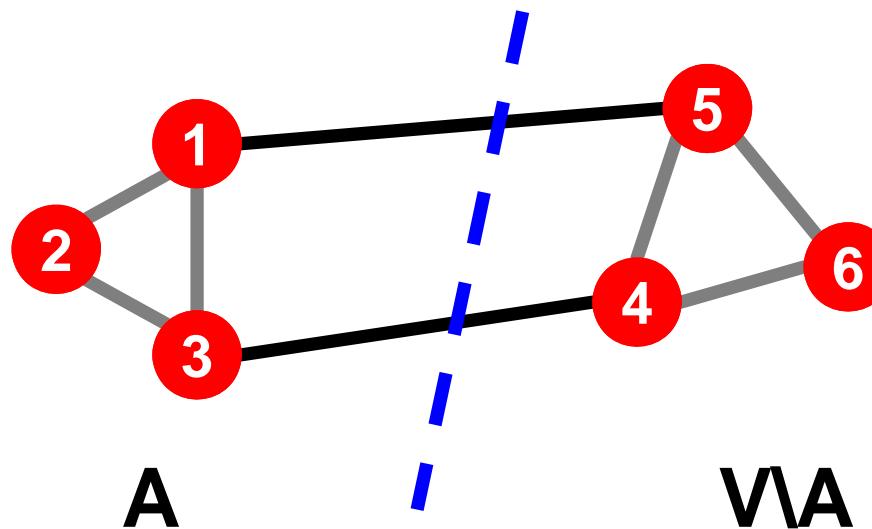


- Question:

- How can we define a “good” cluster in G ?

What makes a good cluster?

- **What makes a good cluster?**
 - Maximize the number of within-cluster connections
 - Minimize the number of between-cluster connections

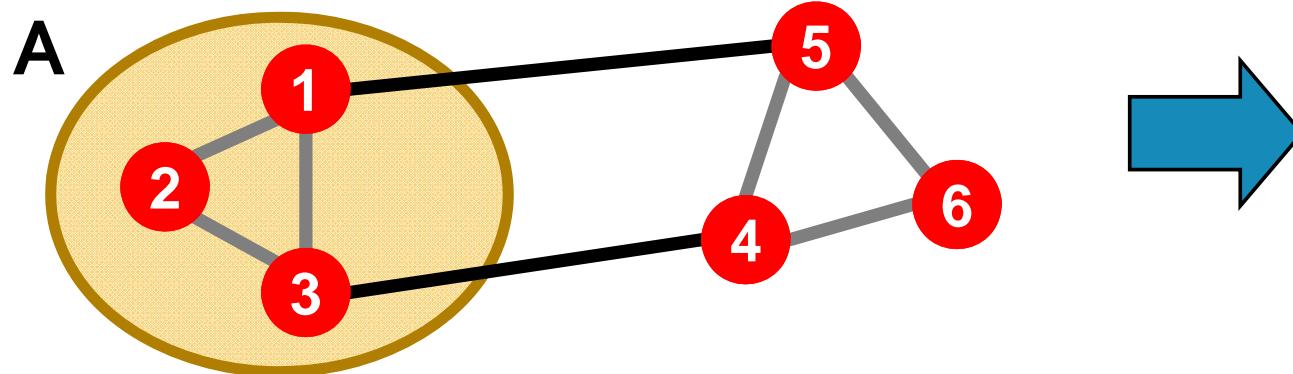


Graph Cuts

- Express cluster quality as a function of the “edge cut” of the cluster
- Cut: Set of edges with only one node in the cluster:

$$cut(A) = \sum_{i \in A, j \notin A} w_{ij}$$

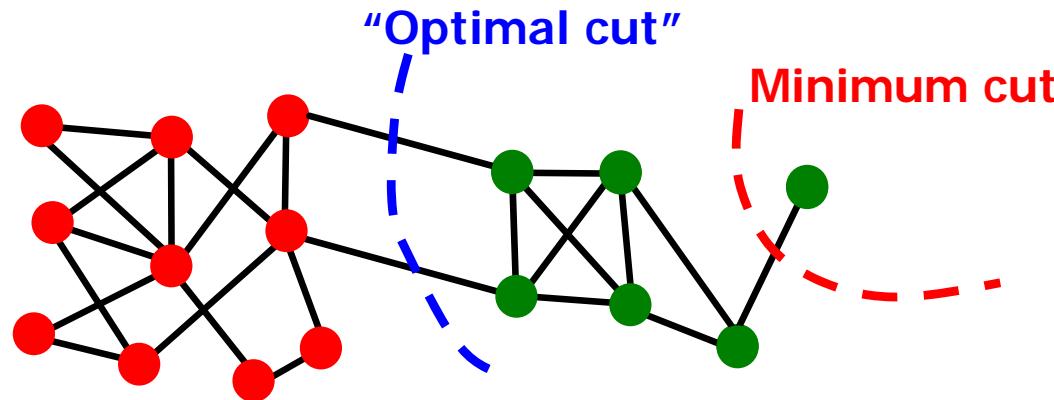
Note: This works for weighed and unweighted (set all $w_{ij}=1$) graphs



$$cut(A) = 2$$

Cut Score

- Partition quality: **Cut score**
 - Quality of a cluster is the weight of connections pointing outside the cluster
- **Degenerate case:**



- **Problem:**
 - Only considers external cluster connections
 - Does not consider internal cluster connectivity

Graph Partitioning Criteria

- Criterion: **Conductance:**

Connectivity of the group to the rest of the network relative to the density of the group

$$\phi(A) = \frac{|\{(i, j) \in E; i \in A, j \notin A\}|}{\min(vol(A), 2m - vol(A))}$$

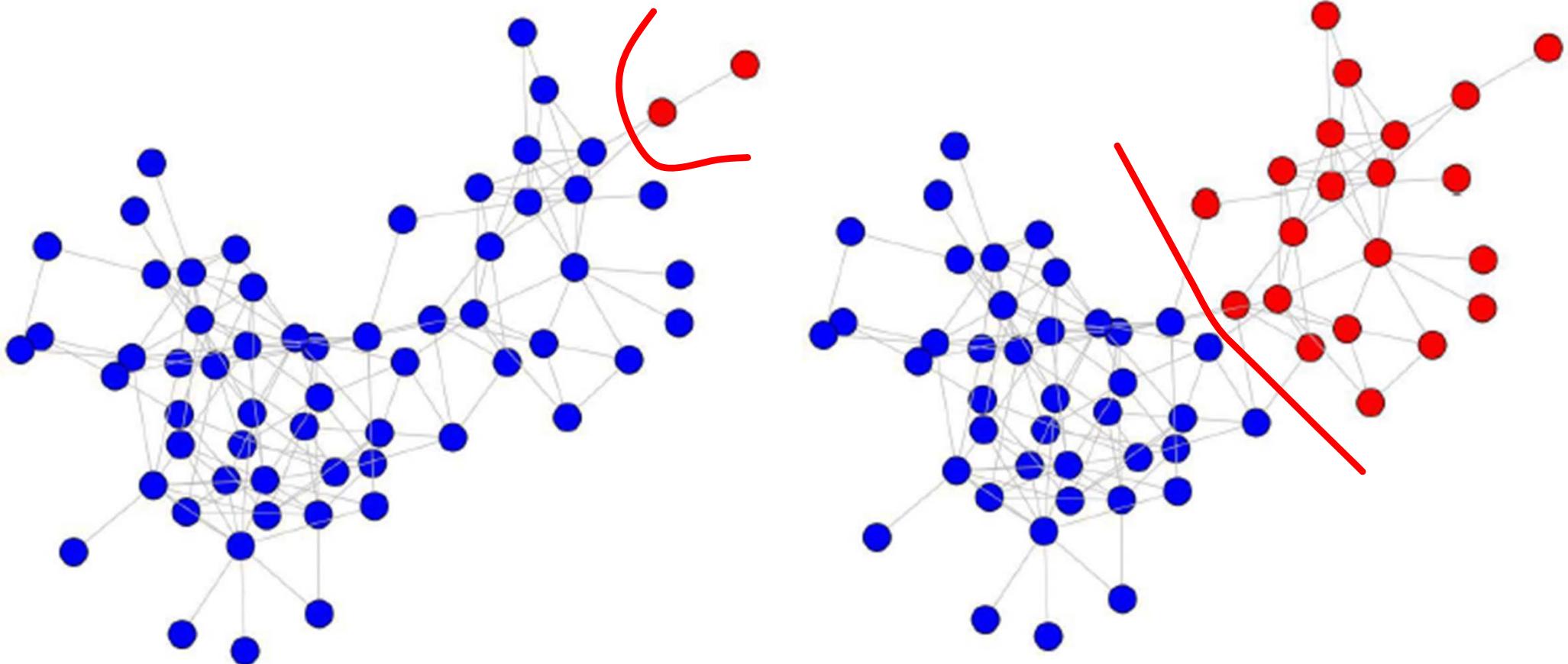
$vol(A)$: total weight of the edges with at least one endpoint in A : $vol(A) = \sum_{i \in A} d_i$

m... number
of edges of
the graph
 d_i ... degree
of node i

- Why use this criterion?

- Produces more balanced partitions

Example: Conductance Score



$$\phi = 2/4 = 0.5$$

$$\phi = 6/92 = 0.065$$

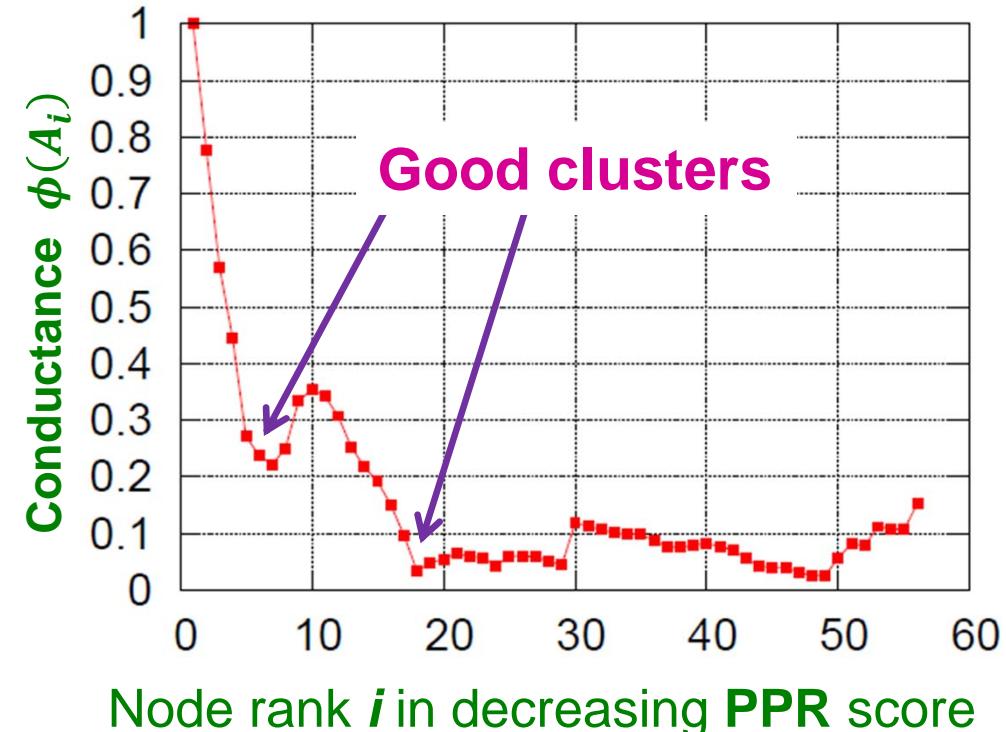
Algorithm Outline: Sweep

Algorithm outline:

- Pick a seed node S of interest
- Run PPR w/ teleport= $\{S\}$
- Sort the nodes by the decreasing PPR score
- Sweep** over the nodes and find good clusters

Sweep:

- Sort nodes in decreasing PPR score $r_1 > r_2 > \dots > r_n$
- For each i compute $\phi(A_i = \{r_1, \dots, r_i\})$
- Local minima** of $\phi(A_i)$ correspond to good clusters



Computing PPR

- How to compute Personalized PageRank (PPR) without touching the whole graph?
 - Power method won't work since each single iteration accesses all nodes of the graph:
$$\mathbf{r}^{(t+1)} = \beta \mathbf{M} \cdot \mathbf{r}^{(t)} + (1 - \beta) \mathbf{a}$$
 - \mathbf{a} is a teleport vector: $\mathbf{a} = [0 \dots 0 \downarrow 1 0 \dots 0]^T$ At index \mathbf{S}
 - \mathbf{r} is the personalized PageRank vector
- PageRank-Nibble [Andersen, Chung, Lang, '07]
 - A fast method for computing approximate Personalized PageRank (PPR) with teleport set = $\{\mathbf{S}\}$
 - ApproxPageRank($\mathbf{S}, \beta, \epsilon$)
 - \mathbf{S} ... seed node
 - β ... teleportation parameter
 - ϵ ... approximation error parameter

Approximate PPR: Overview

■ Overview of the approximate PPR

- **Lazy random walk**, which is a variant of a random walk that stays put with probability 1/2 at each time step, and walks to a random neighbor the other half of the time:

$$r_u^{(t+1)} = \frac{1}{2}r_u^{(t)} + \frac{1}{2}\sum_{i \rightarrow u} \frac{1}{d_i} r_i^{(t)}$$

- Keep track of **residual PPR score** $q_u = p_u - r_u^{(t)}$
 - Residual tells us how well PPR score of u is approximated
 - p_u ... is the ‘true’ PageRank of node u
 - $r_u^{(t)}$... is PageRank estimate at around t
- If **residual** q_u of node u is too big $\frac{q_u}{d_u} \geq \epsilon$ then spread the walk further, else don’t touch the node

Towards approximate PPR

- **A different way to look at PageRank:**

[Jeh&Widom. *Scaling Personalized Web Search*, 2002]

$$\mathbf{p} = (1 - \beta)\mathbf{a} + \beta R_\beta(\mathbf{M}\mathbf{a})$$

- \mathbf{p} is the true PageRank vector
- $R_\beta(\mathbf{M}\mathbf{a})$ is the PageRank vector with teleportation vector (set) $\mathbf{M}\mathbf{a}$ and teleportation parameter β

- **Gives an idea how to compute PageRank:**

- Node S 's “view” of the graph (r_S) is the average of its out-neighbors views plus S 's own importance

“Push” Operation

- Idea:
 - r ... approx. PageRank, q ... its residual PageRank
 - Start with $r = 0$ and $q = a$
 - Iteratively **push** PageRank from q to r until q is small enough
 - **Maintain invariant:** $r + R_\beta(q) = p$ (**the true PageRank**)
[Andersen, Chung, Lang. *Local graph partitioning using PageRank vectors*, 2007]
- Push: 1 step of a lazy random walk from node u :

$\text{Push}(u, r, q)$:

$$r' = r, \quad q' = q$$

$$r'_u = r_u + (1 - \beta)q_u$$

$$q'_u = \frac{1}{2}\beta q_u$$

for each v such that $(u, v) \in E$:

$$q'_v = q_v + \frac{1}{2}\beta \frac{q_u}{d_u}$$

return r' , q'

Update r
Do 1 step of a walk:
Stay at u with prob. $\frac{1}{2}$
Spread remaining $\frac{1}{2}$
fraction of q_u as if a
single step of random
walk were applied to u

Approximate PPR

■ ApproxPageRank(S, β, ε):

Set $r = \vec{0}$, $q = [0..0\ 1\ 0\dots 0]$

While $\max_{u \in V} \frac{q_u}{d_u} \geq \varepsilon$:



At index S

Choose any vertex u where $\frac{q_u}{d_u} \geq \varepsilon$

Push(u, r, q):

$$r' = r, \quad q' = q$$

$$r'_u = r_u + (1 - \beta)q_u$$

$$q'_u = \frac{1}{2}\beta q_u$$

For each v such that $(u, v) \in E$:

$$q'_v = q_v + \frac{1}{2}\beta q_u / d_u$$

Update $r = r'$, $q = q'$

Return r

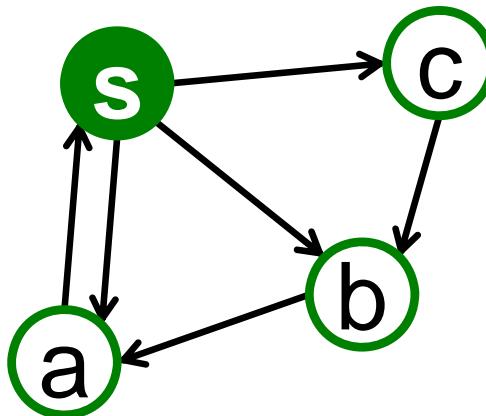
r ... PPR vector
 r_u ... PPR score of u
 q ... residual PPR vector
 q_u ... residual of node u
 d_u ... degree of u

Update r : Move $(1 - \beta)$ of the prob. from q_u to r_u

1 step of a lazy random walk:

- Stay at q_u with prob. $\frac{1}{2}$
- Spread remaining $\frac{1}{2}\beta$ fraction of q_u as if a single step of random walk were applied to u

Example: $\beta = 0.5$



ApproxPageRank(S, β, ϵ):

Set $r = \vec{0}$, $q = [0 \dots 0 \ 1 \ 0 \ \dots \ 0]$

while $\max_{u \in V} \frac{q_u}{d_u} \geq \epsilon$:

Choose any vertex u where $\frac{q_u}{d_u} \geq \epsilon$

Push(u, r, q):

$$r' = r, \quad q' = q$$

$$r'_u = r_u + (1 - \beta)q_u$$

$$q'_u = \frac{1}{2}\beta q_u$$

For each v such that $(u, v) \in E$:

$$q'_v = q_v + \frac{1}{2}\beta \frac{q_u}{d_u}$$

Update $r = r'$, $q = q'$

return r

s a b c

Init:

$$r = [0, 0, 0, 0]$$

$$q = [1, 0, 0, 0]$$

Push(s, r, q):

$$r = [.5, 0, 0, 0]$$

$$q = [.25, .08, .08, .08]$$

Push(s, r, q):

$$r = [.62, 0, 0, 0]$$

$$q = [.06, .10, .10, .10]$$

Push(a, r, q):

$$r = [0.62, .05, 0, 0]$$

$$q = [.09, .03, .10, .10]$$

Push(b, r, q):

$$r = [0.62, .05, .05, 0]$$

$$q = [.09, .05, .03, .10]$$

....

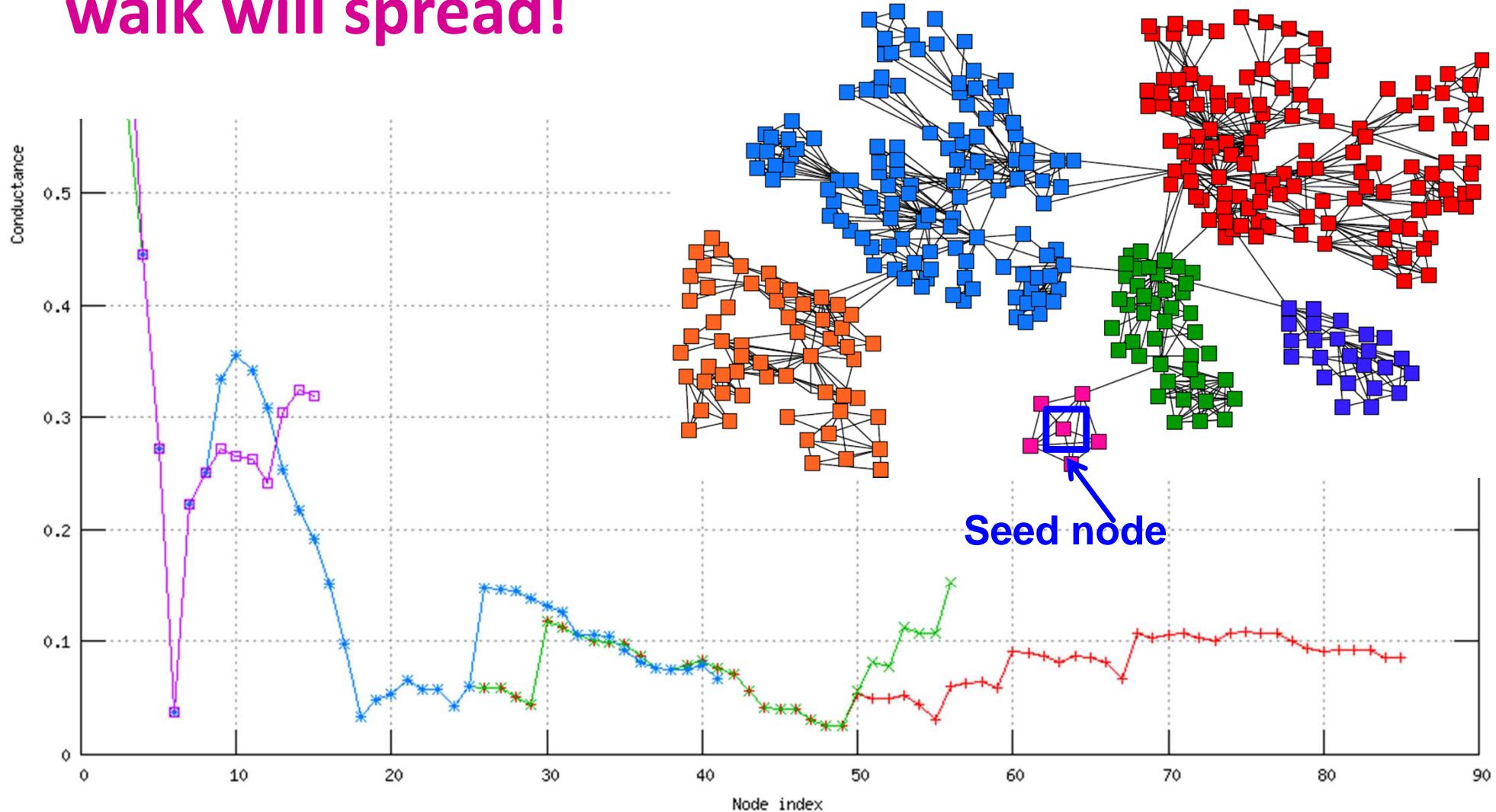
$$r = [.57, .19, .14, .09]$$

Observations (1)

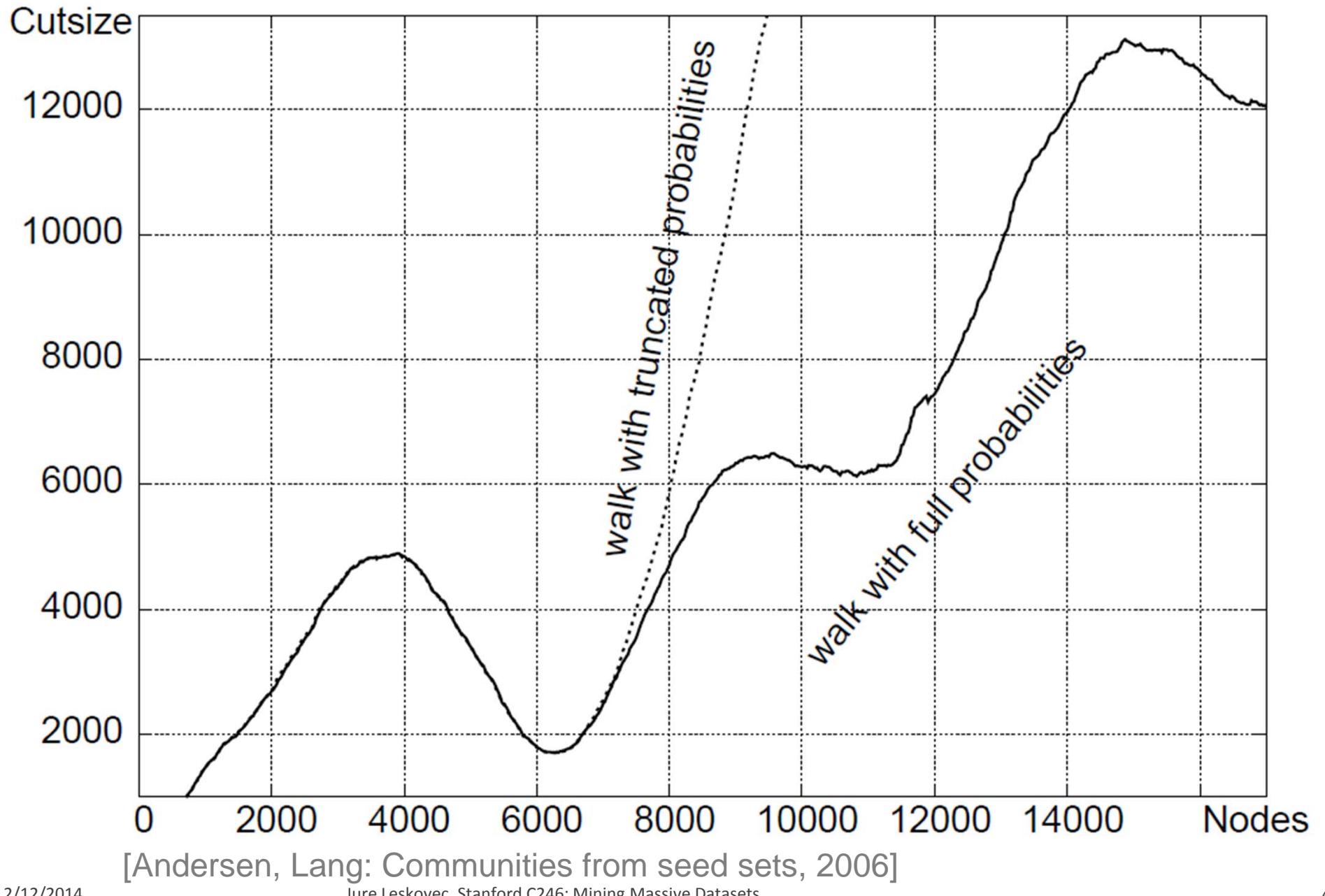
- **Runtime:**
 - PageRank-Nibble computes PPR in time $\left(\frac{1}{\varepsilon(1-\beta)}\right)$ with residual error $\leq \varepsilon$
 - Power method would take time $O\left(\frac{\log n}{\varepsilon(1-\beta)}\right)$
- **Graph cut approximation guarantee:**
 - If there exists a cut of conductance ϕ then the method finds a cut of conductance $\Omega(\phi^2 / \log m)$
 - Details in [Andersen, Chung, Lang. *Local graph partitioning using PageRank vectors*, 2007]
<http://www.math.ucsd.edu/~fan/wp/localpartfull.pdf>

Observations (2)

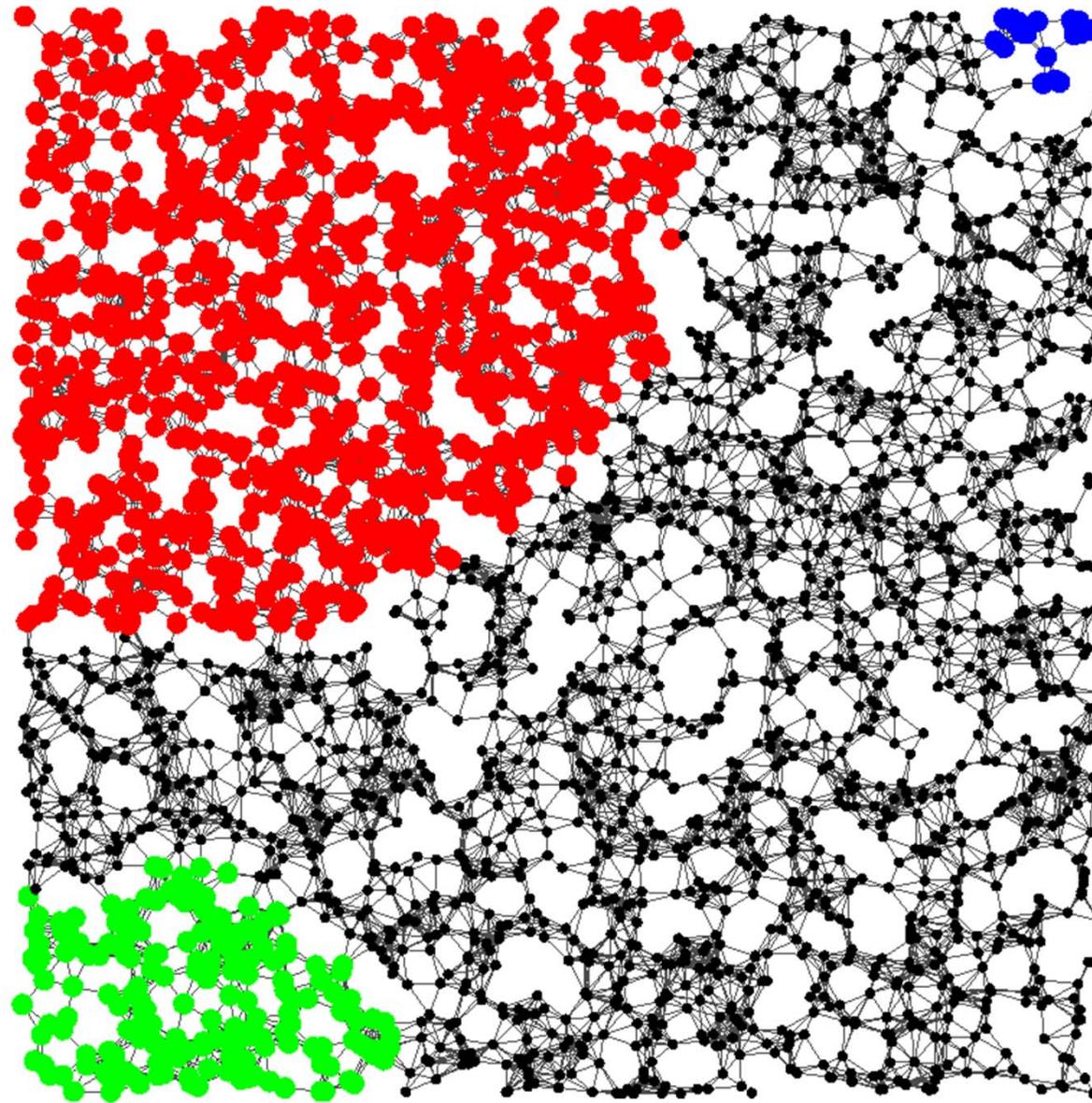
- The smaller the ϵ the farther the random walk will spread!



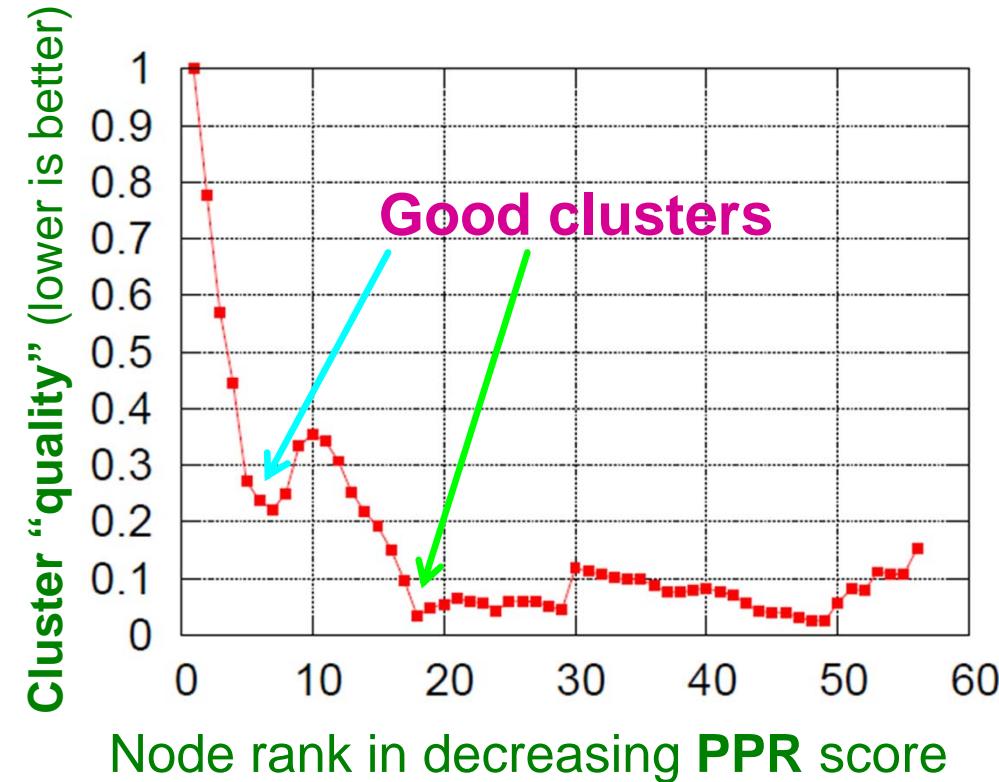
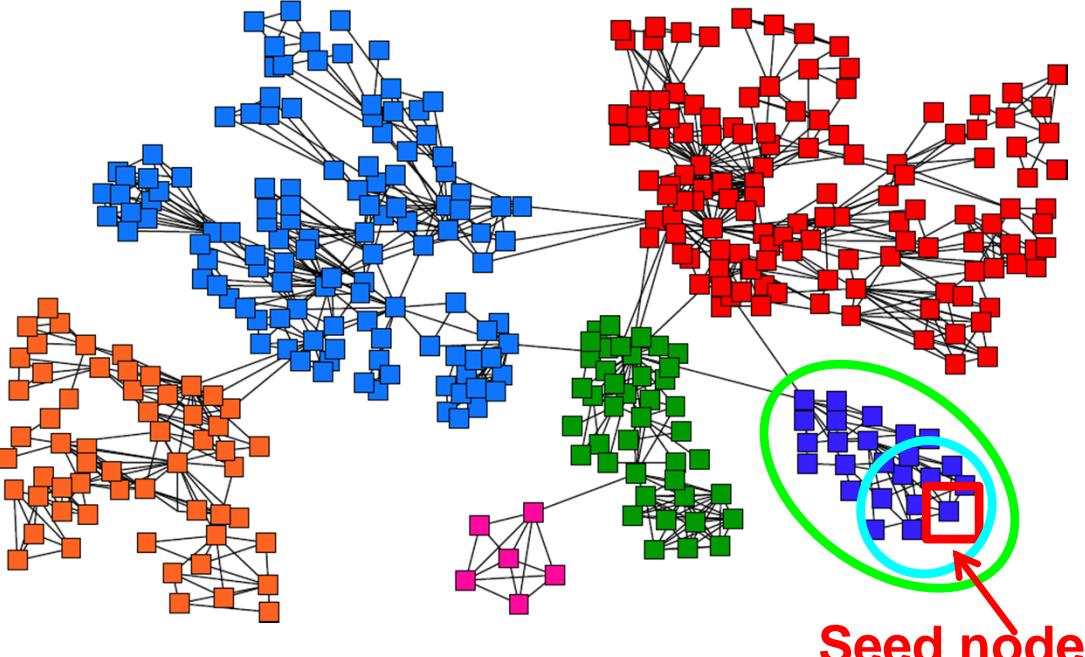
Observations (3)



Example



Summary



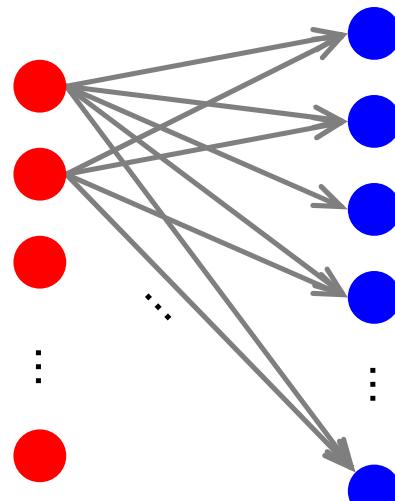
Algorithm summary:

- Pick a seed node S of interest
- Run **PPR** with teleport set = $\{S\}$
- Sort the nodes by the decreasing **PPR** score
- **Sweep** over the nodes and find good clusters

Analysis of Large Graphs: Trawling

Trawling

- Searching for small communities in the Web graph
- What is the signature of a community / discussion in a Web graph?



Dense 2-layer graph

Use this to define “topics”:
What the same people on
the left talk about on the right
Remember HITS!

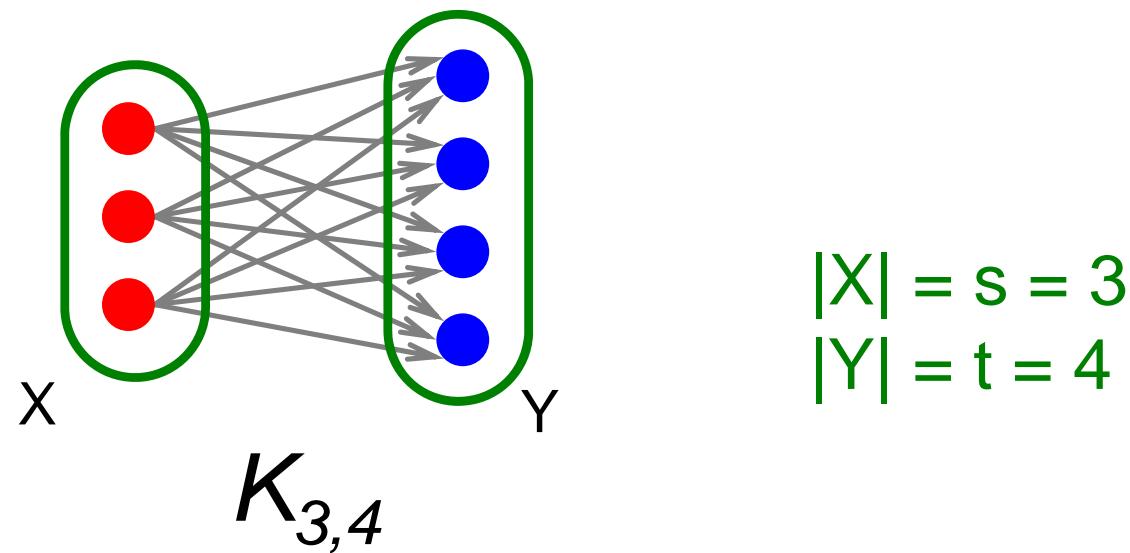
Intuition: Many people all talking about the same things

Searching for Small Communities

- **A more well-defined problem:**

Enumerate complete bipartite subgraphs $K_{s,t}$

- Where $K_{s,t}$: s nodes on the “left” where each links to the same t other nodes on the “right”



Fully connected

Frequent Itemset Enumeration

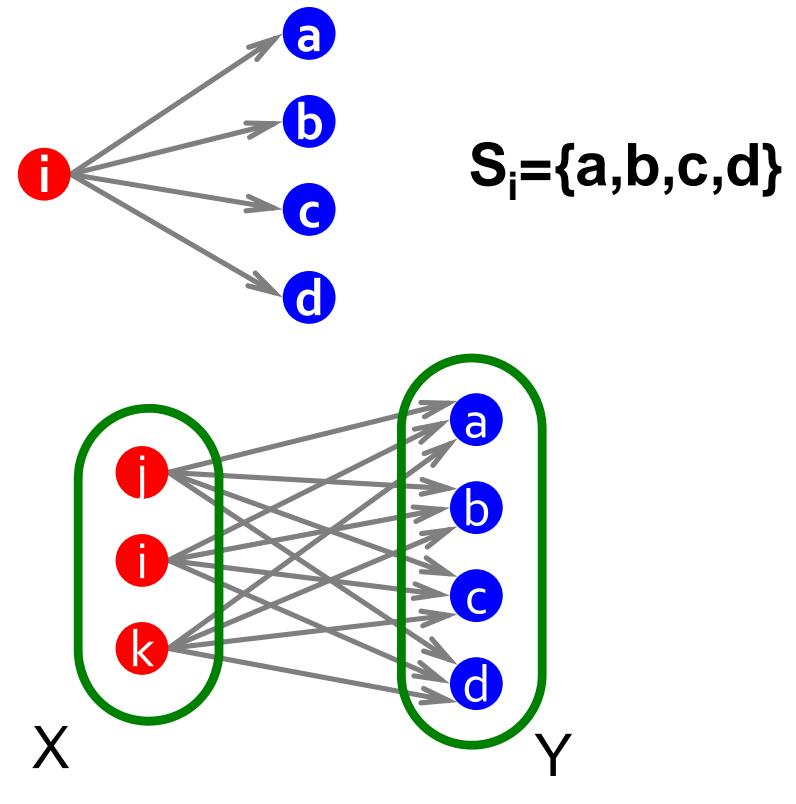
- **Market basket analysis.** Setting:
 - **Market:** Universe U of n items
 - **Baskets:** m subsets of U : $S_1, S_2, \dots, S_m \subseteq U$
(S_i is a set of items one person bought)
 - **Support:** Frequency threshold f
- **Goal:**
 - Find all subsets T s.t. $T \subseteq S_i$ of at least f sets S_i
(items in T were bought together at least f times)
 - **What's the connection between the itemsets and complete bipartite graphs?**

From Itemsets to Bipartite $K_{s,t}$

Frequent itemsets = complete bipartite graphs!

■ How?

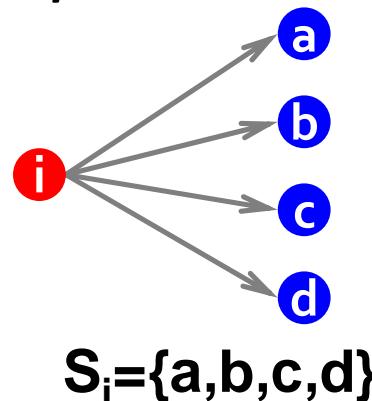
- View each node i as a set S_i of nodes i points to
- $K_{s,t} =$ a set Y of size t that occurs in s sets S_i
- Looking for $K_{s,t} \rightarrow$ set of frequency threshold to s and look at layer t – all frequent sets of size t



s ... minimum support ($|X|=s$)
 t ... itemset size ($|Y|=t$)

From Itemsets to Bipartite $K_{s,t}$

View each node i as a set S_i of nodes i points to

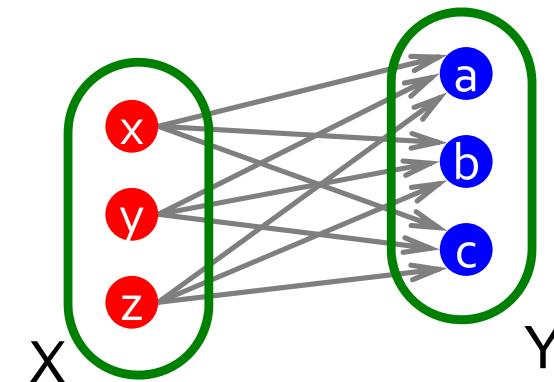
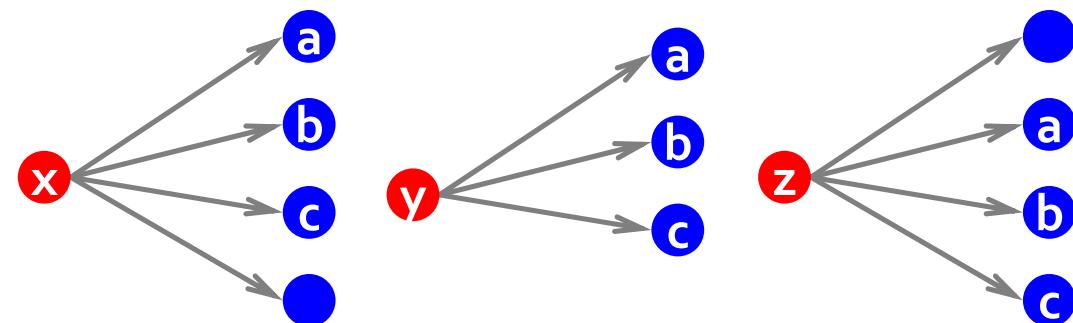


Find frequent itemsets:
 s ... minimum support
 t ... itemset size

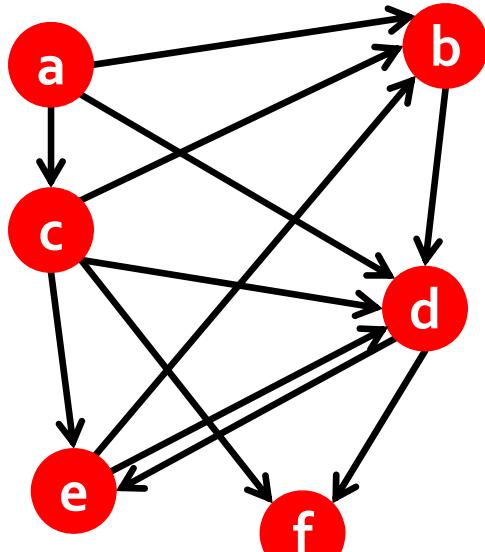
We found $K_{s,t}$!

$K_{s,t}$ = a set Y of size t
 that occurs in s sets S_i

Say we find a **frequent itemset** $Y=\{a,b,c\}$ of supp s
 So, there are s nodes that link to all of $\{a,b,c\}$:



Example (1)



Itemsets:

$$a = \{b, c, d\}$$

$$b = \{d\}$$

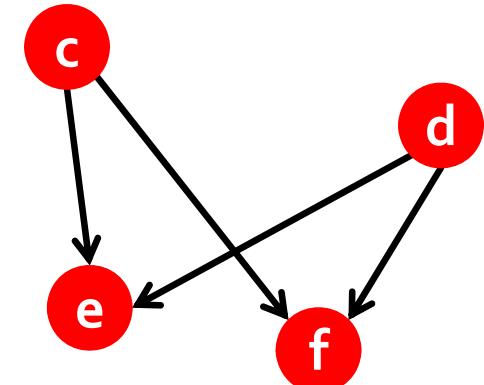
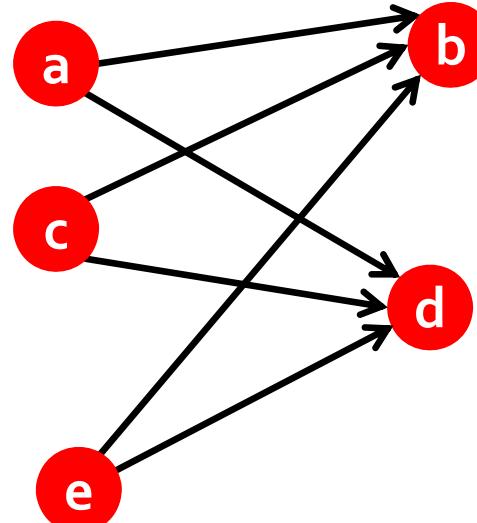
$$c = \{b, d, e, f\}$$

$$d = \{e, f\}$$

$$e = \{b, d\}$$

$$f = \{\}$$

- **Support threshold $s=2$**
 - $\{b, d\}$: support 3
 - $\{e, f\}$: support 2
- **And we just found 2 bipartite subgraphs:**



Example (2)

■ Example of a community from a web graph

A community of Australian fire brigades

Nodes on the right

NSW Rural Fire Service Internet Site
NSW Fire Brigades
Sutherland Rural Fire Service
CFA: County Fire Authority
“The National Cente...ted Children’s Ho...
CRAFTI Internet Connexions-INFO
Welcome to Blackwoo... Fire Safety Serv...
The World Famous Guestbook Server
Wilberforce County Fire Brigade
NEW SOUTH WALES FIR...ES 377 STATION
Woronora Bushfire Brigade
Mongarlowe Bush Fire – Home Page
Golden Square Fire Brigade
FIREBREAK Home Page
Guises Creek Volunt...fficial Home Page...

Nodes on the left

New South Wales Fir...ial Australian Links
Feuerwehrlinks Australien
FireNet Information Network
The Cherrybrook Rur...re Brigade Home Page
New South Wales Fir...ial Australian Links
Fire Departments, F... Information Network
The Australian Firefighter Page
Kristiansand brannv...dens brannvesener...
Australian Fire Services Links
The 911 F,P,M., Fir...mp; Canada A Section
Feuerwehrlinks Australien
Sanctuary Point Rural Fire Brigade
Fire Trails “l...ghters around the...
FireSafe – Fire and Safety Directory
Kristiansand Firede...departments of th...