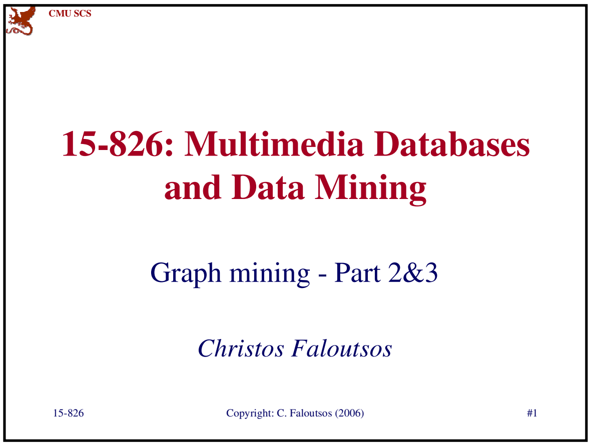
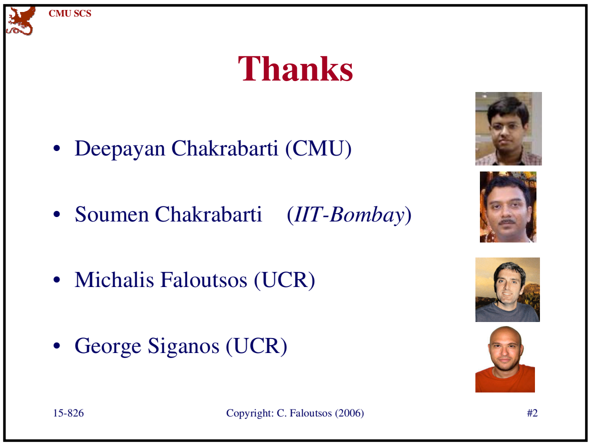


15-826: Multimedia Databases and Data Mining

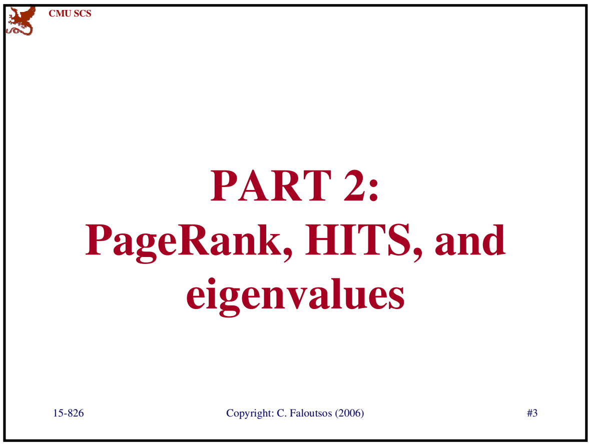
Graph mining - Part 2&3

Christos Faloutsos

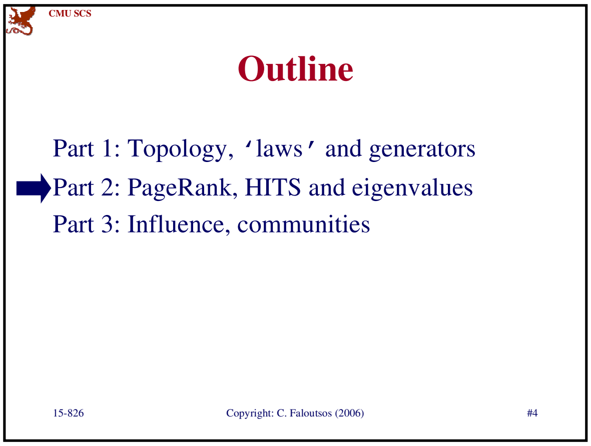


Thanks

- Deepayan Chakrabarti (CMU)
- Soumen Chakrabarti (*IIT-Bombay*)
- Michalis Faloutsos (UCR)
- George Siganos (UCR)



PART 2: PageRank, HITS, and eigenvalues

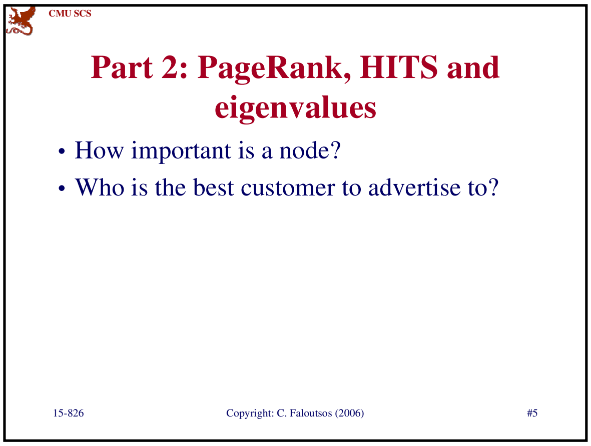


Outline

Part 1: Topology, ‘laws’ and generators

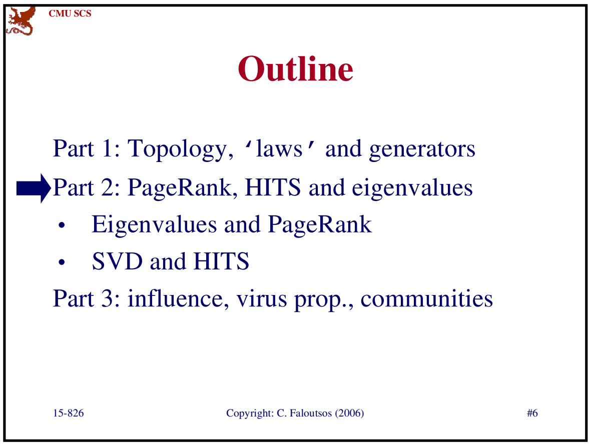
➡Part 2: PageRank, HITS and eigenvalues

Part 3: Influence, communities



Part 2: PageRank, HITS and eigenvalues

- How important is a node?
- Who is the best customer to advertise to?



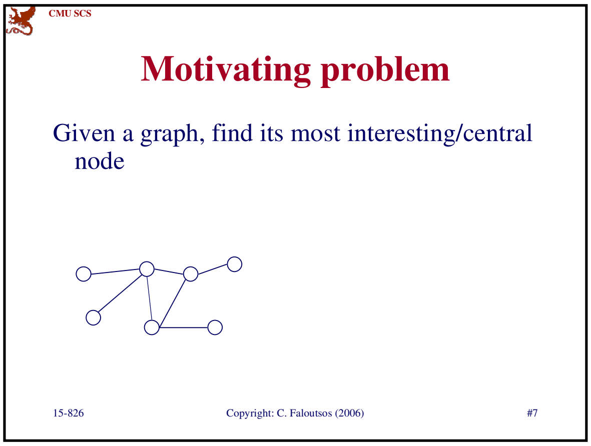
Outline

Part 1: Topology, ‘laws’ and generators

➡Part 2: PageRank, HITS and eigenvalues

- Eigenvalues and PageRank
- SVD and HITS

Part 3: influence, virus prop., communities

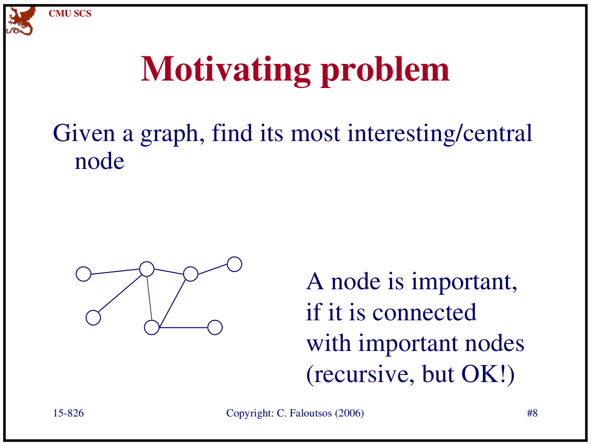


CMU SCS

Motivating problem

Given a graph, find its most interesting/central node

15-826 Copyright: C. Faloutsos (2006) #7



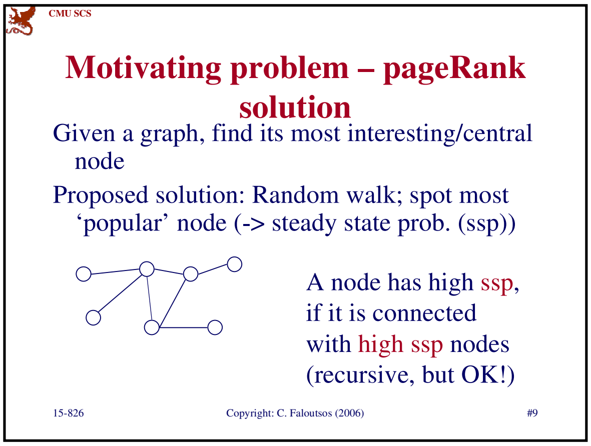
CMU SCS

Motivating problem

Given a graph, find its most interesting/central node

A node is important,
if it is connected
with important nodes
(recursive, but OK!)

15-826 Copyright: C. Faloutsos (2006) #8



CMU SCS

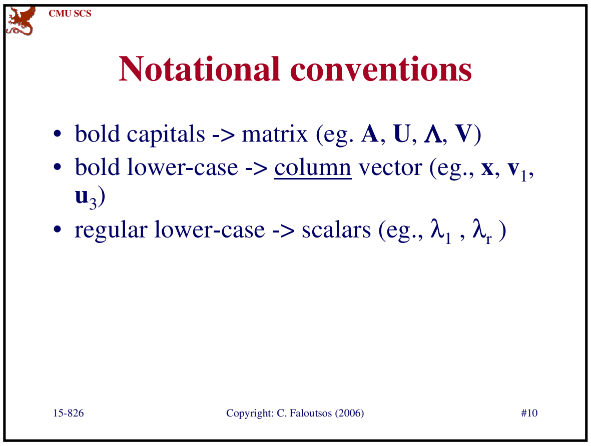
Motivating problem – pageRank solution

Given a graph, find its most interesting/central node

Proposed solution: Random walk; spot most 'popular' node (-> steady state prob. (ssp))

A node has high ssp,
if it is connected
with high ssp nodes
(recursive, but OK!)

15-826 Copyright: C. Faloutsos (2006) #9

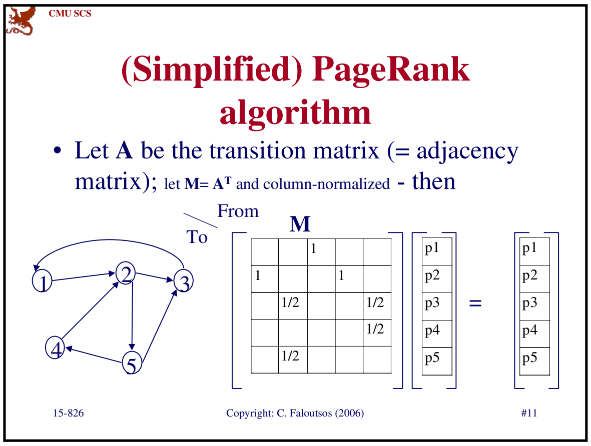


CMU SCS

Notational conventions

- bold capitals -> matrix (eg. $\mathbf{A}$, $\mathbf{U}$, $\mathbf{\Lambda}$, $\mathbf{V}$)
- bold lower-case -> column vector (eg., $\mathbf{x}$, $\mathbf{v}_1$, $\mathbf{u}_3$)
- regular lower-case -> scalars (eg., λ_1 , λ_r)

15-826 Copyright: C. Faloutsos (2006) #10



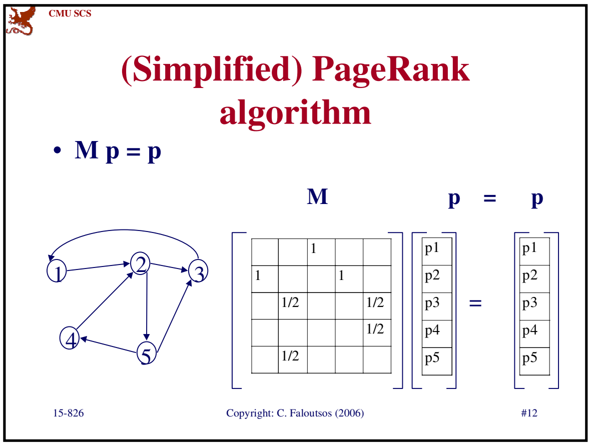
CMU SCS

(Simplified) PageRank algorithm

- Let $\mathbf{A}$ be the transition matrix (= adjacency matrix); let $\mathbf{M} = \mathbf{A}^T$ and column-normalized - then

$$\begin{array}{c} \mathbf{M} \\ \left[\begin{array}{ccccc} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{array}\right] \end{array} \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} = \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}$$

15-826 Copyright: C. Faloutsos (2006) #11



CMU SCS

(Simplified) PageRank algorithm

- $\mathbf{M}\ \mathbf{p} = \mathbf{p}$

$$\begin{array}{c} \mathbf{M} \\ \left[\begin{array}{ccccc} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{array}\right] \end{array} \begin{array}{c} \mathbf{p} \\ \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} \end{array} \begin{array}{c} = \\ = \end{array} \begin{array}{c} \mathbf{p} \\ \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} \end{array}$$

15-826 Copyright: C. Faloutsos (2006) #12

(Simplified) PageRank algorithm

- **M p** = 1 * **p**
- thus, **p** is the eigenvector that corresponds to the highest eigenvalue (=1, since the matrix is column-normalized)

(Simplified) PageRank algorithm

- In short: imagine a particle randomly moving along the edges
- compute its steady-state probabilities (ssp)

Full version of algo: with occasional random jumps – see later

Formal definition

If **M** is a (n x n) square matrix
(λ , **x**) is an **eigenvalue/eigenvector** pair
of **M** if

$$\boxed{\mathbf{M}\,\mathbf{x} = \lambda\,\mathbf{x}}$$

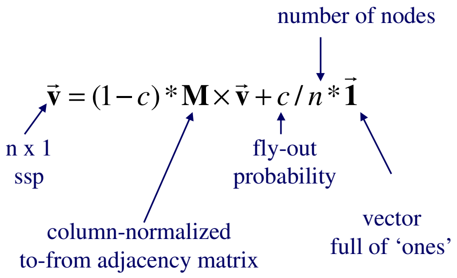
(Published) PageRank

- Do a random walk, but
- with probability *c*, fly-out to a random node
- Then, the ssp vector **v** obeys:

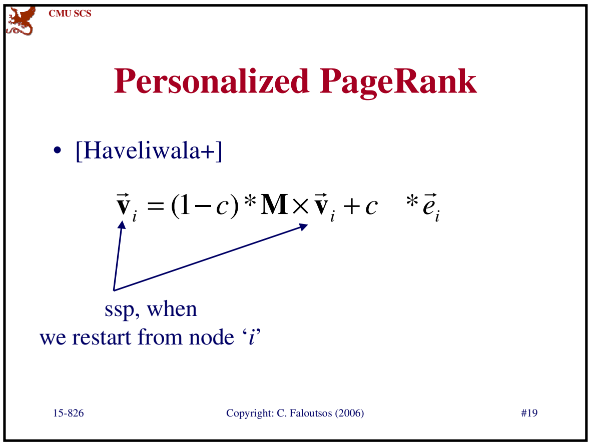
(Published) PageRank

$$\vec{\mathbf{v}} = (1-c) * \mathbf{M} \times \vec{\mathbf{v}} + c/n * \vec{\mathbf{1}}$$

(Published) PageRank



$$\vec{\mathbf{v}} = (1-c) * \mathbf{M} \times \vec{\mathbf{v}} + c/n * \vec{\mathbf{1}}$$

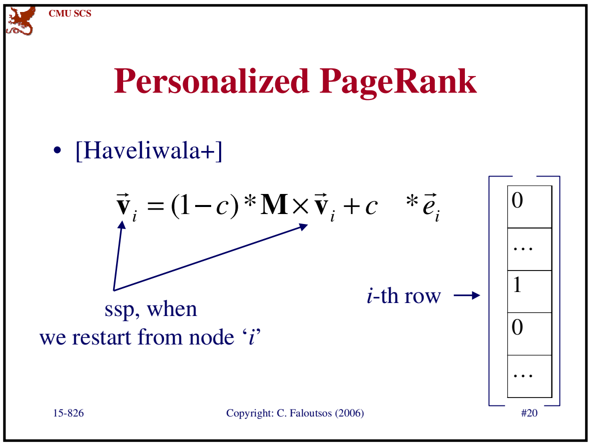


Personalized PageRank

- [Haveliwala+]

$$\vec{\mathbf{v}}_i = (1-c)*\mathbf{M}\times\vec{\mathbf{v}}_i + c \quad *\vec{e}_i$$

ssp, when
we restart from node ‘*i*’



Personalized PageRank

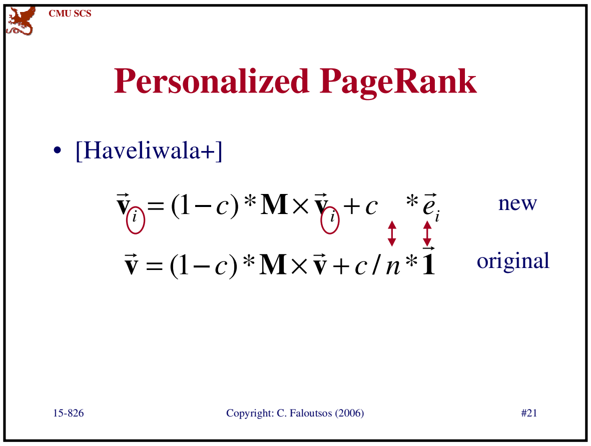
- [Haveliwala+]

$$\vec{\mathbf{v}}_i = (1-c)*\mathbf{M}\times\vec{\mathbf{v}}_i + c \quad *\vec{e}_i$$

ssp, when
we restart from node ‘*i*’

i-th row →

$$\begin{bmatrix}0\\ \cdots\\ 1\\ 0\\ \cdots\end{bmatrix}$$



Personalized PageRank

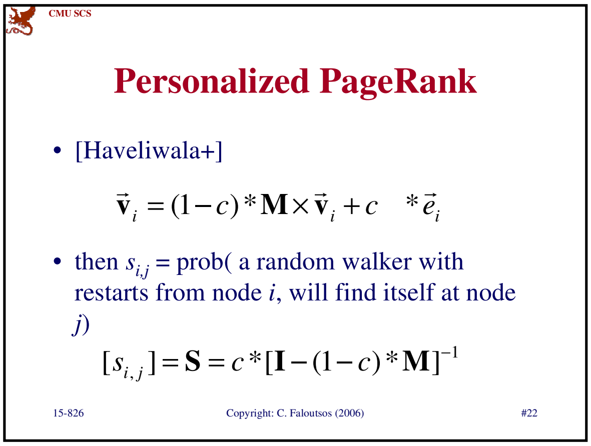
- [Haveliwala+]

$$\vec{\mathbf{v}}_{i} = (1-c)*\mathbf{M}\times\vec{\mathbf{v}}_{i} + c \quad *\vec{e}_i$$

new

$$\vec{\mathbf{v}} = (1-c)*\mathbf{M}\times\vec{\mathbf{v}} + c/n*\vec{\mathbf{1}}$$

original



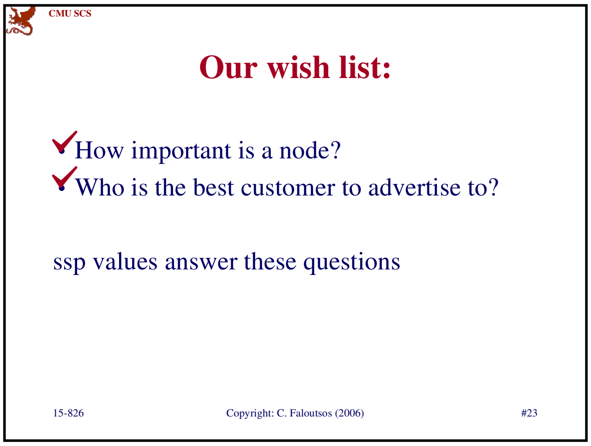
Personalized PageRank

- [Haveliwala+]

$$\vec{\mathbf{v}}_i = (1-c)*\mathbf{M}\times\vec{\mathbf{v}}_i + c \quad *\vec{e}_i$$

- then $s_{i,j}$ = prob(a random walker with restarts from node *i*, will find itself at node *j*)

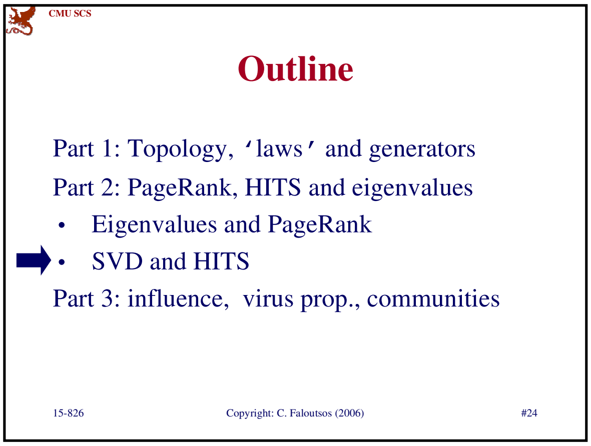
$$[s_{i,j}] = \mathbf{S} = c*[\mathbf{I}-(1-c)*\mathbf{M}]^{-1}$$



Our wish list:

✓How important is a node?
✓Who is the best customer to advertise to?

ssp values answer these questions



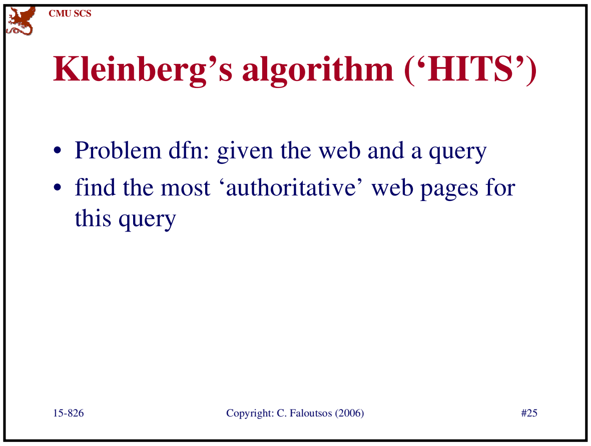
Outline

Part 1: Topology, ‘laws’ and generators

Part 2: PageRank, HITS and eigenvalues

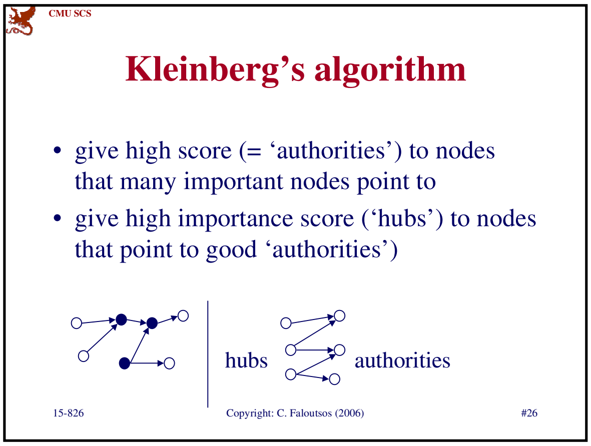
- Eigenvalues and PageRank
- ➡ SVD and HITS

Part 3: influence, virus prop., communities



Kleinberg’s algorithm (‘HITS’)

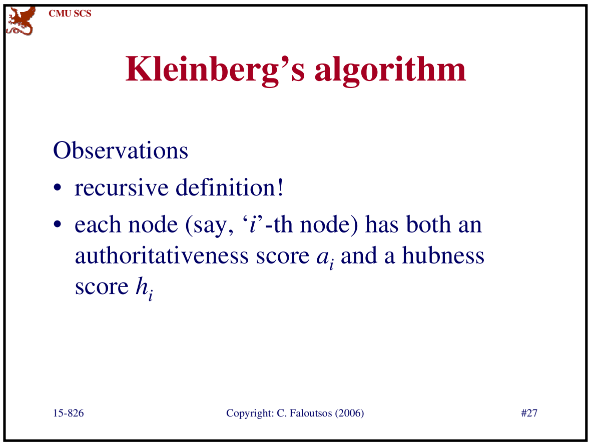
- Problem dfn: given the web and a query
- find the most ‘authoritative’ web pages for this query



Kleinberg’s algorithm

- give high score (= ‘authorities’) to nodes that many important nodes point to
- give high importance score (‘hubs’) to nodes that point to good ‘authorities’)

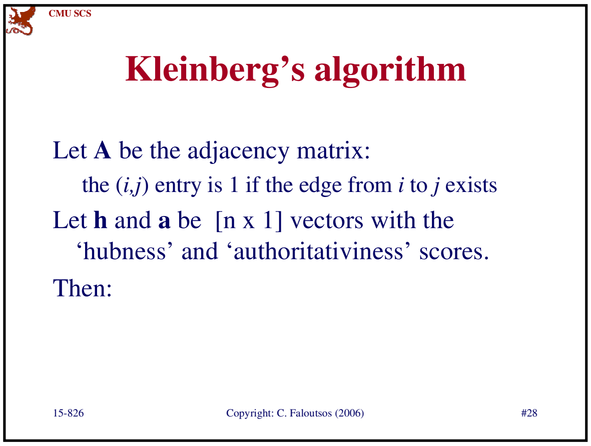
hubs
authorities



Kleinberg’s algorithm

Observations

- recursive definition!
- each node (say, ‘*i*’-th node) has both an authoritativeness score a_i and a hubness score h_i



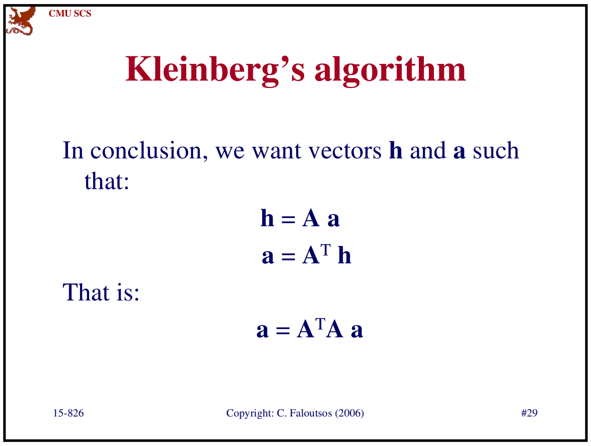
Kleinberg’s algorithm

Let **A** be the adjacency matrix:

the (i,j) entry is 1 if the edge from i to j exists

Let **h** and **a** be [n x 1] vectors with the ‘hubness’ and ‘authoritativiness’ scores.

Then:



Kleinberg’s algorithm

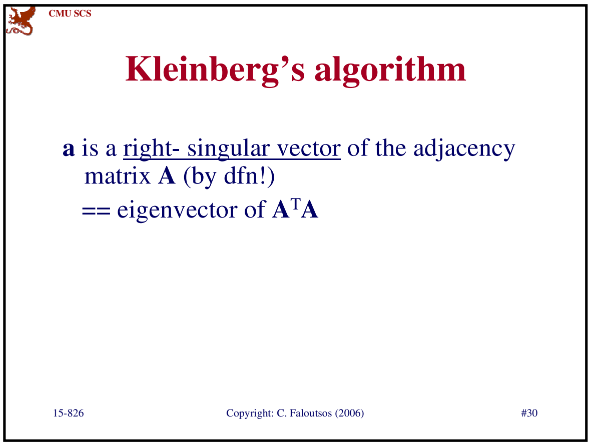
In conclusion, we want vectors **h** and **a** such that:

$$\mathbf{h} = \mathbf{A}\ \mathbf{a}$$

$$\mathbf{a} = \mathbf{A}^T\ \mathbf{h}$$

That is:

$$\mathbf{a} = \mathbf{A}^T\mathbf{A}\ \mathbf{a}$$



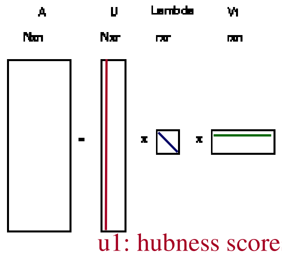
Kleinberg’s algorithm

a is a right- singular vector of the adjacency matrix **A** (by dfn!)

== eigenvector of $\mathbf{A}^T\mathbf{A}$

SVD & HITS

- $\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{V}^T$ - example:



v1: author. scores

u1: hubness scores

Conclusions

eigenvalues/eigenvectors: vital for

- PageRank,
- (virus propagation - coming up next!)
- (graph partitioning - not mentioned here)

Conclusions, cont'd

SVD

- closely related: HITS/Kleinberg
- (and also LSI, KLT, PCA, Least squares, ...)

Both are **extremely useful**, **well understood** tools for graphs / matrices.

PART 3: Influence, virus propagation, communities

Outline

Part 1: Topology, 'laws' and generators

Part 2: PageRank, HITS and eigenvalues

- Eigenvalues and PageRank
- SVD and HITS

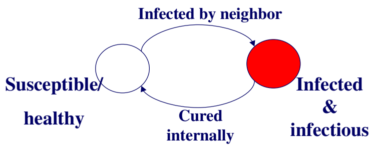
➡ Part 3: influence, virus prop., communities

Problem definition

- Q1: How does a virus spread across an arbitrary network?
- Q2: will it create an epidemic?

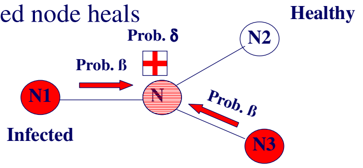
Framework

- Susceptible-Infected-Susceptible (SIS) model
  - Cured nodes immediately become susceptible



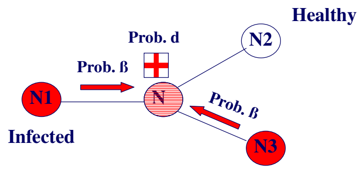
The model

- (virus) Birth rate β: probability than an infected neighbor attacks
- (virus) Death rate δ: probability that an infected node heals



The model

- Virus ‘strength’ s= β / δ



Other models:

- SIR: Susceptible - infected & infectious - recovered/removed
  - eg., mumps, chickenpox; black plague

Other models:

- and many more:
- SEIR: Susceptible; Exposed (= infected, but not infectious yet); I; R
- variations:
  - M: passively immune, like infants
  - with births/newcomers
  - ...

Epidemic threshold τ

of a graph, defined as the value of τ, such that

if strength $s = \beta / \delta < \tau$

an epidemic can not happen

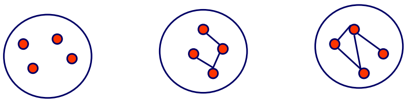
Thus,

- given a graph
- compute its epidemic threshold

Epidemic threshold τ

What should τ depend on?

- avg. degree? and/or highest degree?
- and/or variance of degree?
- and/or third moment of degree?



Epidemic threshold

- [Theorem] We have no epidemic, if

$$\boxed{\beta/\delta < \tau = 1/\lambda_{1,A}}$$

Epidemic threshold

- [Theorem] We have no epidemic, if

recovery prob.

epidemic threshold

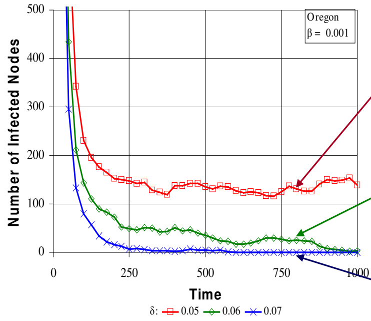
$$\boxed{\beta/\delta < \tau = 1/\lambda_{1,A}}$$

attack prob.

largest eigenvalue of adj. matrix ***A***

Proof: [Wang+03]

Experiments (Oregon)



β/δ > t (above threshold)

β/δ = t (at the threshold)

β/δ < t (below threshold)

Our wish list:

- Who is the best person/computer to immunize against a virus?

Our wish list:

✓ Who is the best person/computer to immunize against a virus? Highest diff in λ1

Outline

Part 1: Topology, 'laws' and generators

Part 2: PageRank, HITS and eigenvalues

- Eigenvalues and PageRank
- SVD and HITS

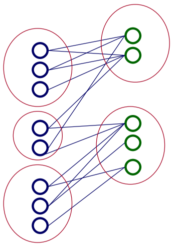
Part 3: influence, virus prop., communities

Graph clustering & mining

- Q1: which edges/nodes are ‘abnormal’?
- Q2: split a graph in *k* ‘natural’ communities
  - but how to determine *k*?

Graph partitioning

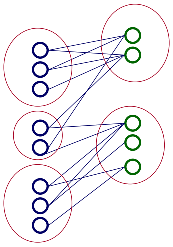
- Documents x terms
- Customers x products
- Users x web-sites



Graph partitioning

- Documents x terms
- Customers x products
- Users x web-sites

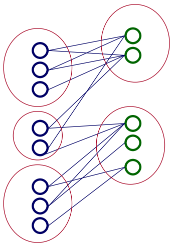
- Q: HOW MANY PIECES?



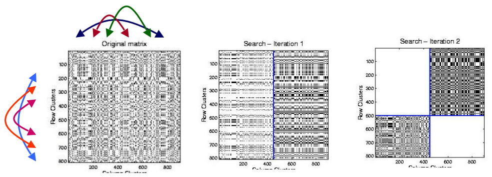
Graph partitioning

- Documents x terms
- Customers x products
- Users x web-sites

- Q: HOW MANY PIECES?
- A: MDL/ compression



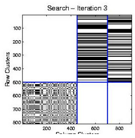
Cross-associations



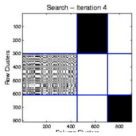
1x2

2x2

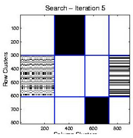
Cross-associations



2x3

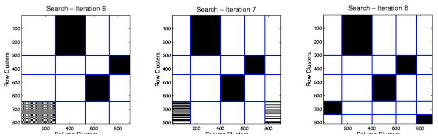


3x3

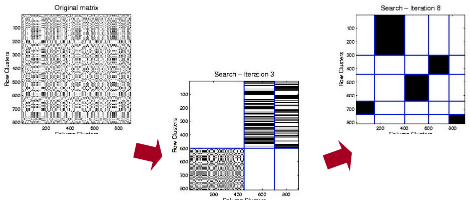


3x4

Cross-associations



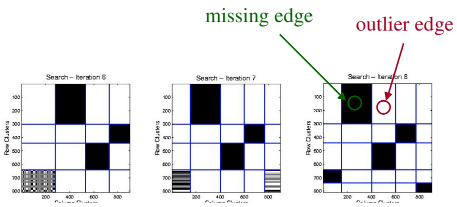
Cross-associations



Graph clustering & mining

- Q1: which edges/nodes are ‘abnormal’?
- ✓ Q2: split a graph in *k* ‘natural’ communities - but how to determine *k*?
- A2: choose the *k* that leads to best overall compression (= MDL = Minimum Description Language)

Cross-associations



Conclusions

- virus propagation: eigenvalue determines the epidemic threshold (SIS model)
- communities/graph partitioning: MDL

Resources: Software and urls

- SVD packages: in **many** systems (matlab, mathematica, LINPACK, LAPACK)
- stand-alone, free code: SVDPACK from Michael Berry

  http://www.cs.utk.edu/~berry/projects.html

Books

- Faloutsos, C. (1996). *Searching Multimedia Databases by Content*, Kluwer Academic Inc.
- Jolliffe, I. T. (1986). *Principal Component Analysis*, Springer Verlag.

Books

- [Press+92] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery: *Numerical Recipes in C*, Cambridge University Press, 1992, 2nd Edition. (Great description, intuition and code for SVD)

References

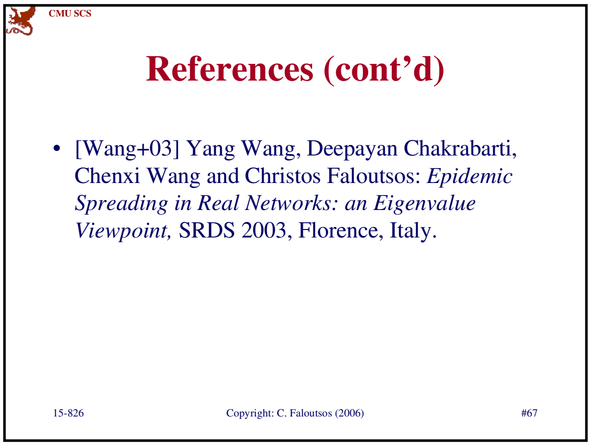
- Berry, Michael: http://www.cs.utk.edu/~lsi/
- [Brin+98] Brin, S. and L. Page (1998). *Anatomy of a Large-Scale Hypertextual Web Search Engine*. 7th Intl World Wide Web Conf.
- [Chakrabarti'04] D. Chakrabarti, *AutoPart: Parameter-Free Graph Partitioning and Outlier Detection*, PKDD 2004 (pages 112-124), Pisa, Italy

References (cont'd)

- [Chakrabarti+,04a] D. Chakrabarti, S. Papadimitriou, D. Modha and C. Faloutsos, *Fully Automatic Cross-Associations*, KDD 2004 (pp. 79-88), Washington, USA
- [Haveliwala02] Taher H. Haveliwala, *Topic-Sensitive PageRank* World Wide Web Conference, 2002

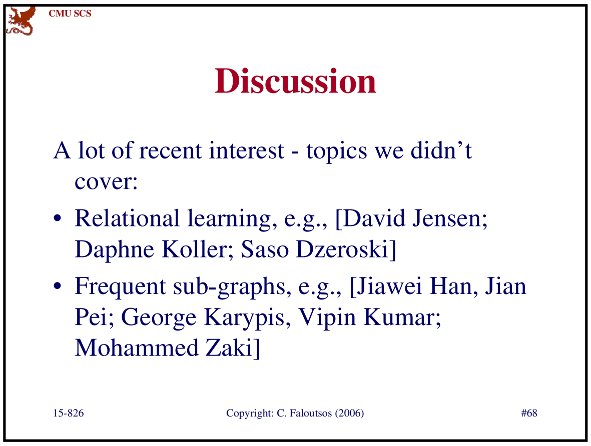
References (cont'd)

- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press.
- Kleinberg, J. (1998). *Authoritative sources in a hyperlinked environment*. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms.



References (cont'd)

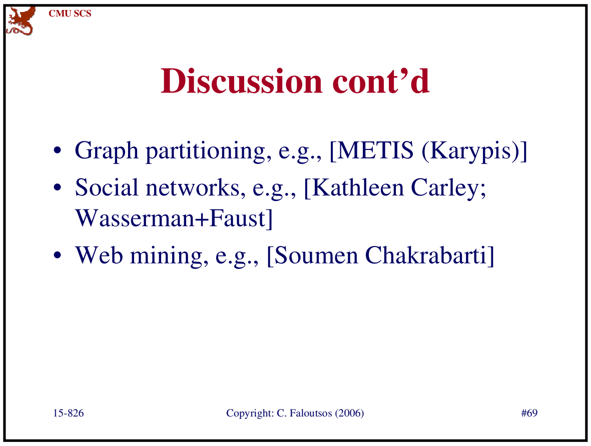
- [Wang+03] Yang Wang, Deepayan Chakrabarti, Chenxi Wang and Christos Faloutsos: *Epidemic Spreading in Real Networks: an Eigenvalue Viewpoint,* SRDS 2003, Florence, Italy.



Discussion

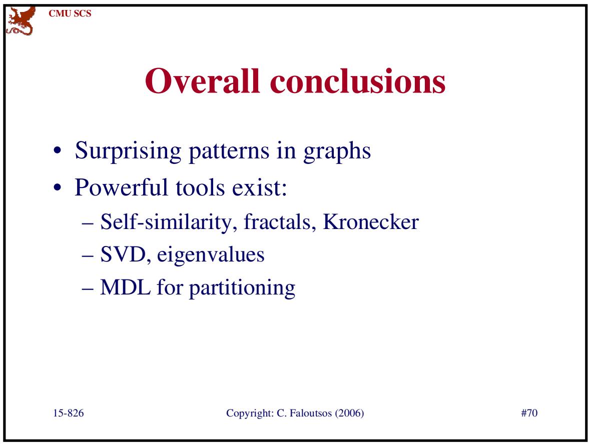
A lot of recent interest - topics we didn't cover:

- Relational learning, e.g., [David Jensen; Daphne Koller; Saso Dzeroski]
- Frequent sub-graphs, e.g., [Jiawei Han, Jian Pei; George Karypis, Vipin Kumar; Mohammed Zaki]



Discussion cont'd

- Graph partitioning, e.g., [METIS (Karypis)]
- Social networks, e.g., [Kathleen Carley; Wasserman+Faust]
- Web mining, e.g., [Soumen Chakrabarti]



Overall conclusions

- Surprising patterns in graphs
- Powerful tools exist:
  - Self-similarity, fractals, Kronecker
  - SVD, eigenvalues
  - MDL for partitioning