CMU SCS

15-826: Multimedia Databases and Data Mining

Guest Lecture: Graph mining
Parts 2&3

Christos Faloutsos

CMU SCS

Thanks

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




CMU SCS

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?

Part 2: PageRank, HITS and eigenvalues

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

Answers:

- PageRank (random surfer model)
- HITS (hubs and authorities)

CMU SCS

reminder

(Published) PageRank

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

CMU SCS

reminder

(Published) PageRank

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

CMU SCS

(Published) PageRank

number of nodes

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

n x 1 ssp

fly-out probability

column-normalized to-from adjacency matrix

vector full of 'ones'

CMU SCS

A clever variation

- Personalized PageRank:
- Who is the most important node in the vicinity of node *i*?

CMU SCS

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*'

CMU SCS

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 →

0
…
1
0
…

Personalized PageRank

- [Haveliwala+]

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

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

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)

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

Personalized PageRank

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

$s_{i,j}$: good measure of how close is node j to node i

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

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')

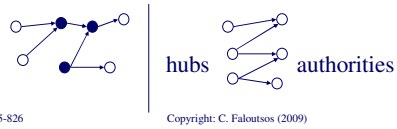
- Problem dfn: given the web and a query
- find the most 'authoritative' web pages for this query

CMU SCS

Kleinberg's algorithm

reminder

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

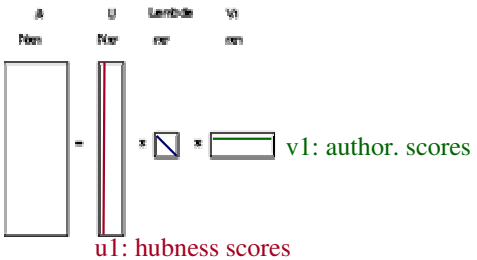


CMU SCS

SVD & HITS

reminder

- Adjacency matrix $\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{V}^T$



CMU SCS

Conclusions

eigenvalues/eigenvectors: vital for

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

CMU SCS

Conclusions, cont'd

SVD

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

Both eigen- and singular decompositions are **extremely useful**, **well understood** tools for graphs / matrices.

CMU SCS

PART 3: Influence, virus propagation, communities

CMU SCS

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

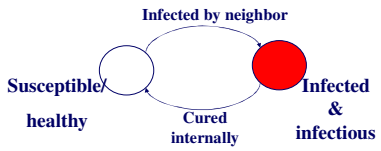
Problem definition

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

CMU SCS

Framework

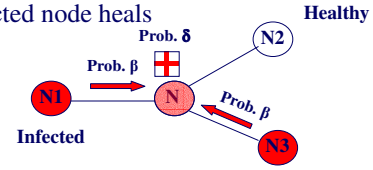
- Susceptible-Infected-Susceptible (SIS) model
  - Cured nodes immediately become susceptible



CMU SCS

The model

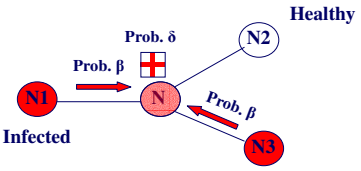
- (virus) Birth rate β: probability than an infected neighbor attacks
- (virus) Death rate δ: probability that an infected node heals



CMU SCS

The model

- Virus 'strength' s= β/δ



CMU SCS

Other models:

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

CMU SCS

Other models:

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

CMU SCS

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

CMU SCS

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?

CMU SCS

Epidemic threshold

- [Theorem] We have no epidemic, if

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

CMU SCS

Epidemic threshold

- [Theorem] We have no epidemic, if

recovery prob. — epidemic threshold

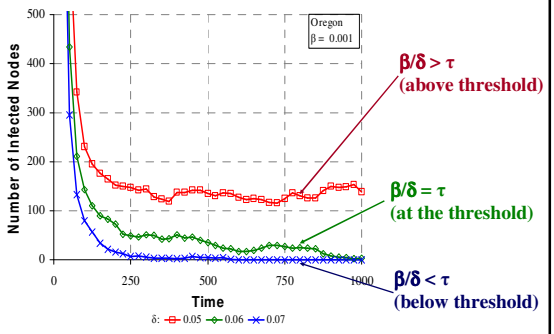
$$\beta/\delta < \tau = 1/\lambda_{1,A}$$

attack prob. — largest eigenvalue of adj. matrix A

Proof: [Wang+03]

CMU SCS

Experiments (Oregon)



CMU SCS

Our wish list:

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

CMU SCS

Our wish list:

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

CMU SCS

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

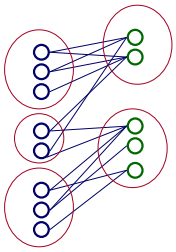
Graph clustering & mining

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

CMU SCS

Graph partitioning

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

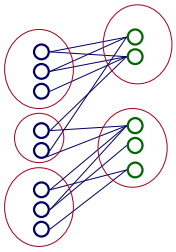


CMU SCS

Graph partitioning

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

- Q: HOW MANY PIECES?

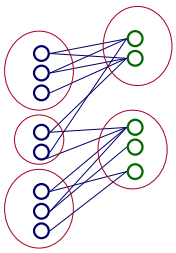


CMU SCS

Graph partitioning

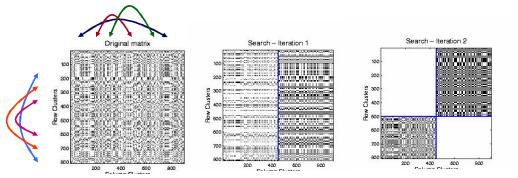
- Documents x terms
- Customers x products
- Users x web-sites

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



CMU SCS

Cross-associations

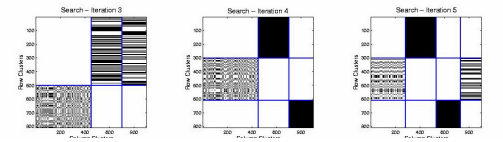


1x2

2x2

CMU SCS

Cross-associations



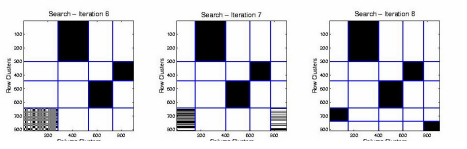
2x3

3x3

3x4

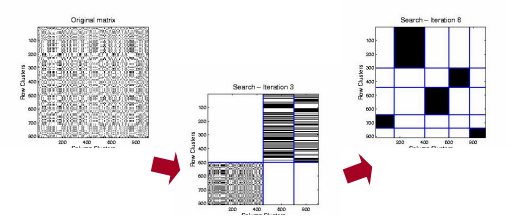
CMU SCS

Cross-associations



CMU SCS

Cross-associations



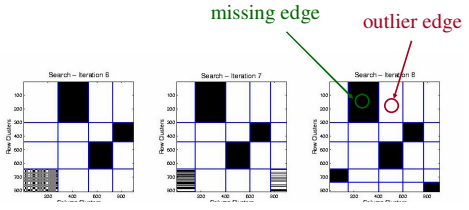
CMU SCS

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)

CMU SCS

Cross-associations



CMU SCS

Conclusions

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

CMU SCS

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

CMU SCS

Books

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

CMU SCS

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)

CMU SCS

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

CMU SCS

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]

CMU SCS

Discussion cont'd

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

CMU SCS

Overall conclusions

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