15-826: Multimedia Databases and Data Mining

Lecture #29: Graph mining - Generators & tools

Christos Faloutsos

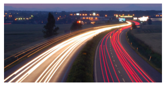
Must-read material (1 of 2)

Fully Automatic Cross-Associations, by D. Chakrabarti, S. Papadimitriou, D. Modha and C. Faloutsos, in KDD 2004 (pages 79-88), Washington, USA

Must-read material (2 of 2)

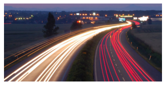
J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos, *Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication*, in PKDD 2005, Porto, Portugal

Main outline



- Introduction
- Indexing
- Mining
  - – Graphs – patterns
  - ➡ – Graphs – generators and tools
  - – Association rules
  - – …

Detailed outline



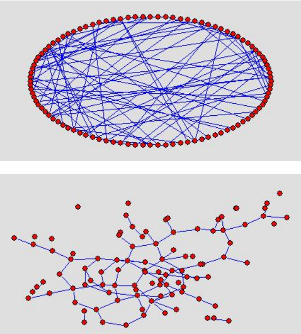
- Graphs – generators
  - ➡ Erdos-Renyi
  - Other generators
  - Kronecker
- Graphs - tools

Generators

- How to generate random, realistic graphs?
  - Erdos-Renyi model: beautiful, but unrealistic
  - degree-based generators
  - process-based generators
  - recursive/self-similar generators

Erdos-Renyi

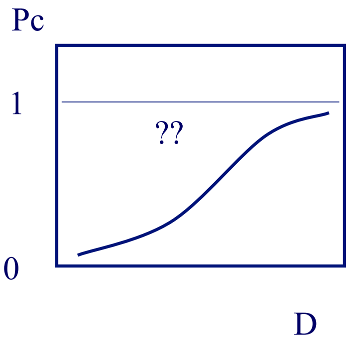
- random graph – 100 nodes, avg degree = 2
- Fascinating properties (phase transition)
- But: unrealistic (Poisson degree distribution != power law)



E-R model & Phase transition

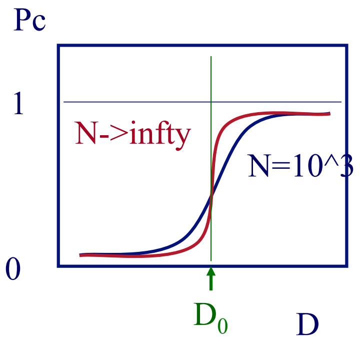
- vary avg degree D
- watch Pc =

  Prob(there is a giant connected component)
- How do you expect it to be?



E-R model & Phase transition

- vary avg degree D
- watch Pc =
  Prob(there is a giant connected component)
- How do you expect it to be?



Degree-based

- Figure out the degree distribution (eg., ‘Zipf’)
- Assign degrees to nodes
- Put edges, so that they match the original degree distribution

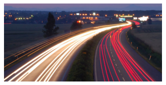
Process-based

- Barabasi; Barabasi-Albert: Preferential attachment -> power-law tails!
  - ‘rich get richer’
- [Kumar+]: preferential attachment + mimick
  - Create ‘communities’

Process-based (cont’d)

- [Fabrikant+, ‘02]: H.O.T.: connect to closest, high connectivity neighbor
- [Pennock+, ‘02]: Winner does NOT take all

Detailed outline



- Graphs – generators
  - Erdos-Renyi
  - Other generators
  - ➡ Kronecker
- Graphs - tools

Recursive generators

- (RMAT [Chakrabarti+,'04])
- Kronecker product

Wish list for a generator:

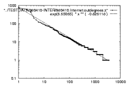
- Power-law-tail in- and out-degrees
- Power-law-tail scree plots
- **shrinking/constant** diameter
- Densification Power Law
- communities-within-communities

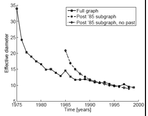
Q: how to achieve all of them?

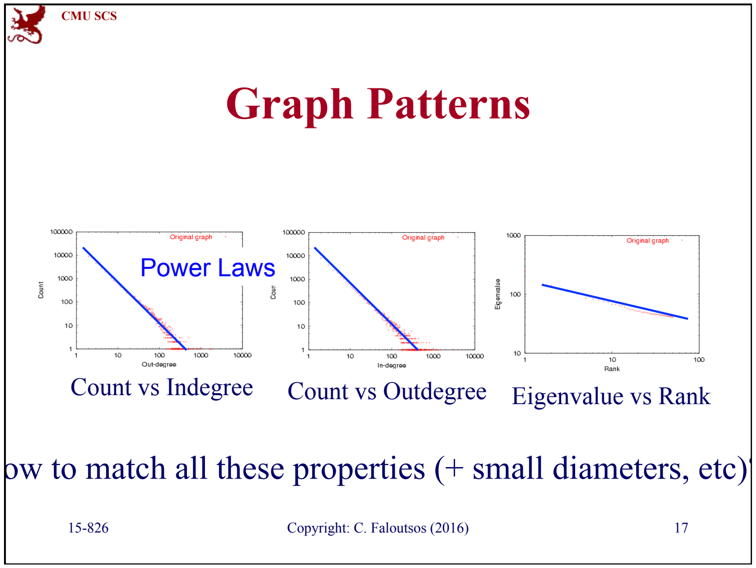
A: Kronecker matrix product [Leskovec+05b]

Graph gen.: Problem dfn

- Given a growing graph with count of nodes N_1, N_2, …
- Generate a realistic sequence of graphs that will obey all the patterns
  - Static Patterns
    - S1 Power Law Degree Distribution
    - S2 Power Law eigenvalue and eigenvector distribution
      - Small Diameter
  - Dynamic Patterns
    - T2 Growth Power Law (2x nodes; 3x edges)
    - T1 Shrinking/Stabilizing Diameters







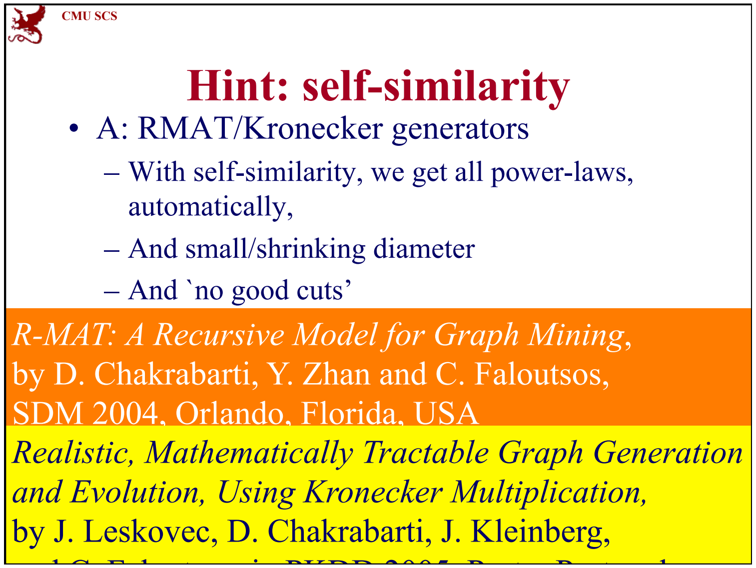
Graph Patterns

Count vs Indegree

Count vs Outdegree

Eigenvalue vs Rank

ow to match all these properties (+ small diameters, etc)

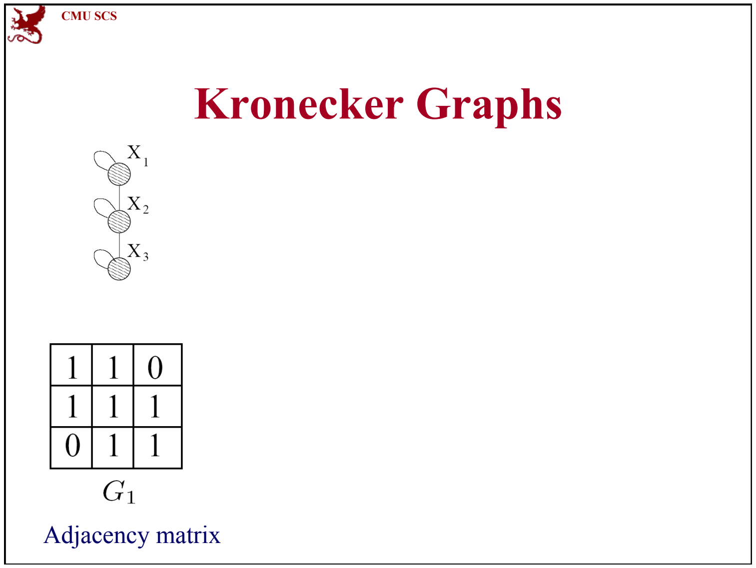


Hint: self-similarity

- A: RMAT/Kronecker generators
  - With self-similarity, we get all power-laws, automatically,
  - And small/shrinking diameter
  - And `no good cuts'

R-MAT: A Recursive Model for Graph Mining,
by D. Chakrabarti, Y. Zhan and C. Faloutsos,
SDM 2004, Orlando, Florida, USA

Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication,
by J. Leskovec, D. Chakrabarti, J. Kleinberg,

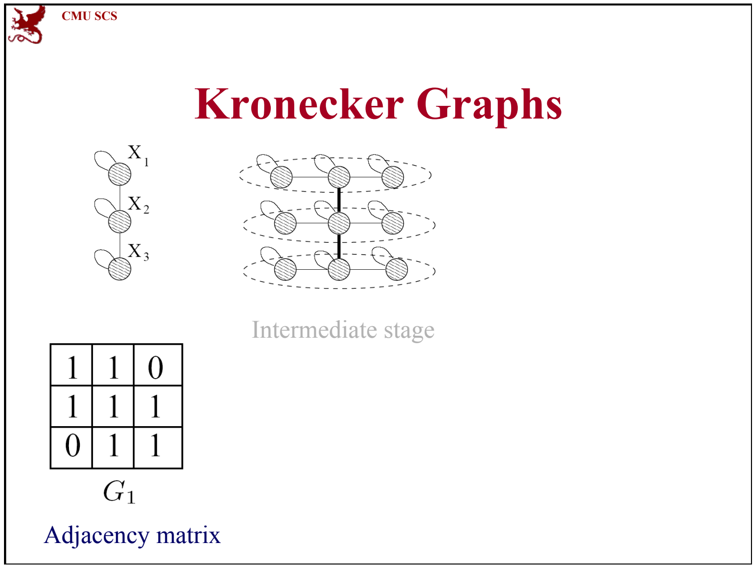


Kronecker Graphs

1	1	0
1	1	1
0	1	1

G_1

Adjacency matrix



Kronecker Graphs

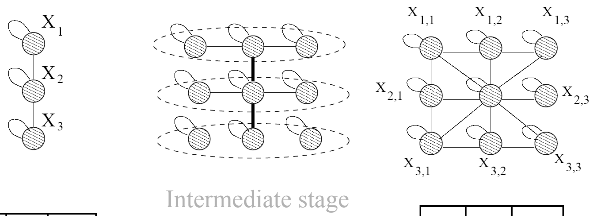
Intermediate stage

1	1	0
1	1	1
0	1	1

G_1

Adjacency matrix

Kronecker Graphs



1	1	0
1	1	1
0	1	1

G_1

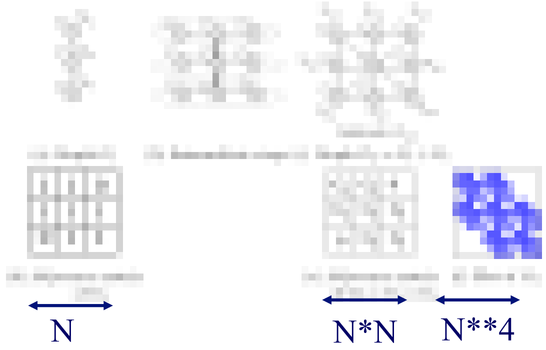
Adjacency matrix

G_1	G_1	0
G_1	G_1	G_1
0	G_1	G_1

$G_2 = G_1 \otimes G_1$

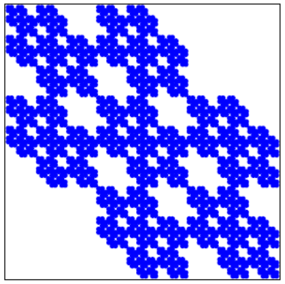
Adjacency matrix

Kronecker product



Kronecker Graphs

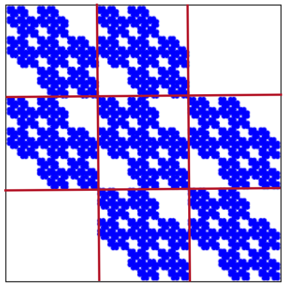
- Continuing multiplying with G_1 we obtain G_4 and so on …



G_4 adjacency matrix

Kronecker Graphs

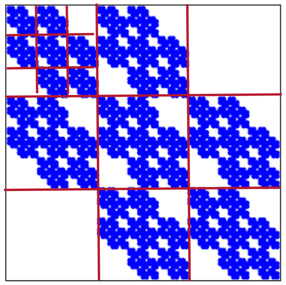
- Continuing multiplying with G_1 we obtain G_4 and so on …



G_4 adjacency matrix

Kronecker Graphs

- Continuing multiplying with G_1 we obtain G_4 and so on …

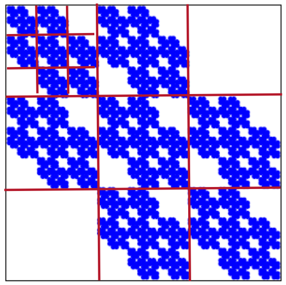


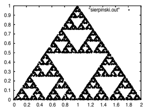
G_4 adjacency matrix

Kronecker Graphs

- Continuing multiplying with G_1 we obtain G_4 and so on …

oles within holes;
Communities
ithin communities





G_4 adjacency matrix

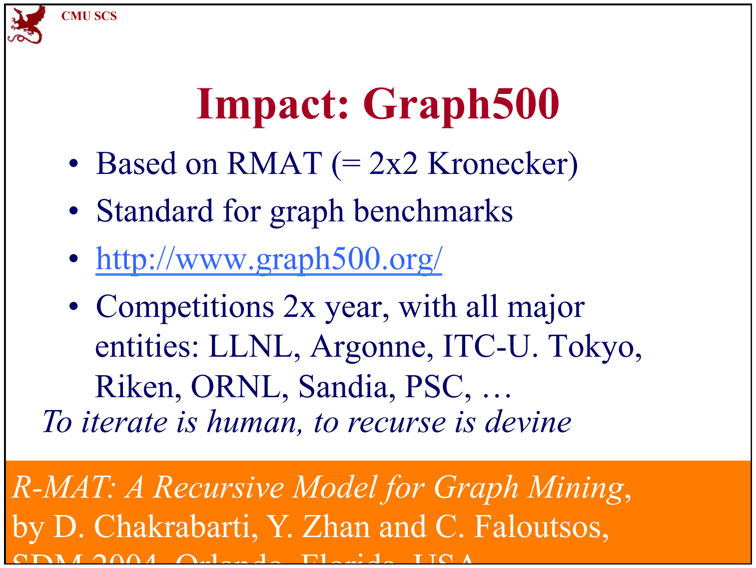
Self-similarity -> power laws

Properties:

- We can PROVE that
  - Degree distribution is multinomial ~ power law
  - new – Diameter: constant
  - Eigenvalue distribution: multinomial
  - First eigenvector: multinomial

Problem Definition

- Given a growing graph with nodes N_1, N_2, …
- Generate a realistic sequence of graphs that will obey all the patterns
  - Static Patterns
    - ✓ Power Law Degree Distribution
    - ✓ Power Law eigenvalue and eigenvector distribution
    - ✓ Small Diameter
  - Dynamic Patterns
    - ✓ Growth Power Law
    - ✓ Shrinking/Stabilizing Diameters
- First generator for which we can **prove** all these properties



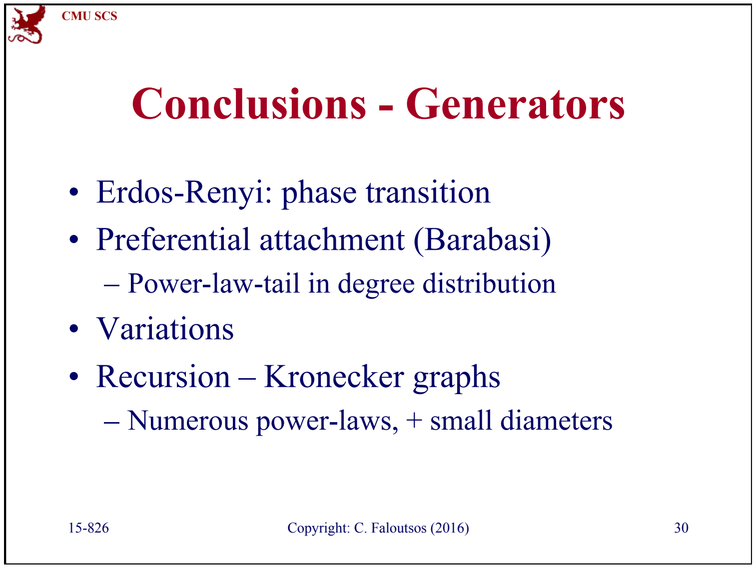
CMU SCS

Impact: Graph500

- Based on RMAT (= 2x2 Kronecker)
- Standard for graph benchmarks
- http://www.graph500.org/
- Competitions 2x year, with all major entities: LLNL, Argonne, ITC-U. Tokyo, Riken, ORNL, Sandia, PSC, …

To iterate is human, to recurse is devine

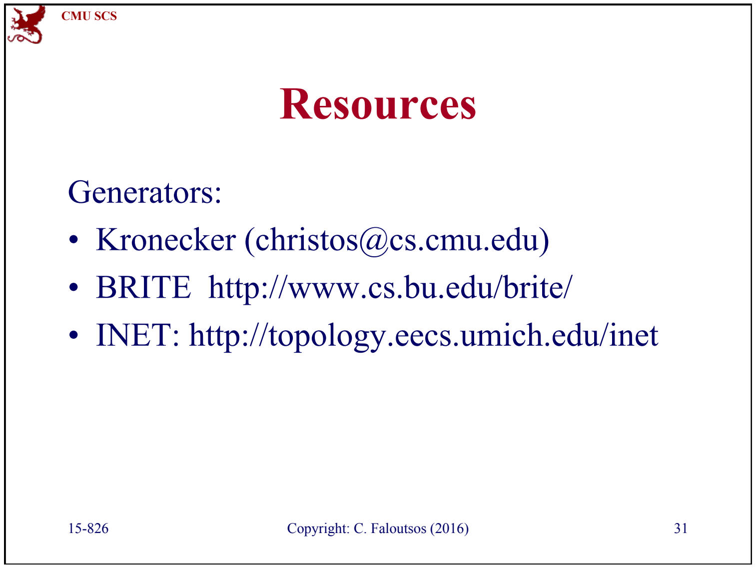
R-MAT: A Recursive Model for Graph Mining, by D. Chakrabarti, Y. Zhan and C. Faloutsos, SDM 2004, Orlando, Florida, USA



CMU SCS

Conclusions - Generators

- Erdos-Renyi: phase transition
- Preferential attachment (Barabasi)
  - Power-law-tail in degree distribution
- Variations
- Recursion – Kronecker graphs
  - Numerous power-laws, + small diameters

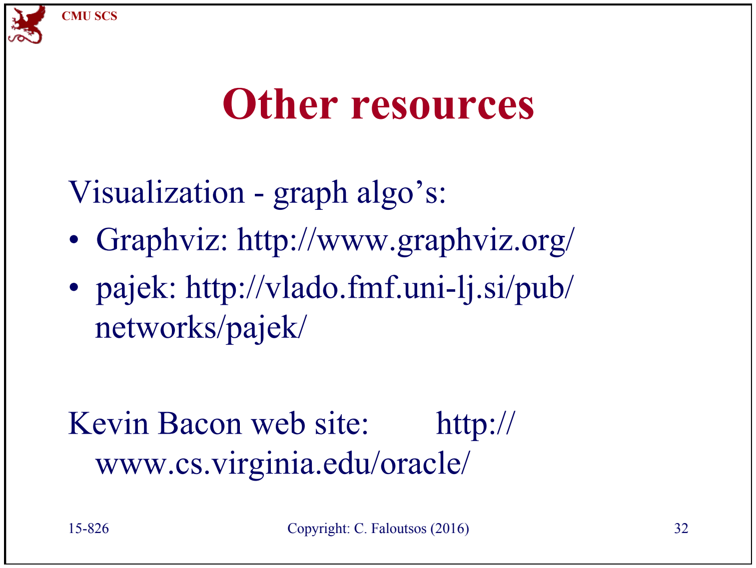


CMU SCS

Resources

Generators:

- Kronecker (christos@cs.cmu.edu)
- BRITE http://www.cs.bu.edu/brite/
- INET: http://topology.eecs.umich.edu/inet



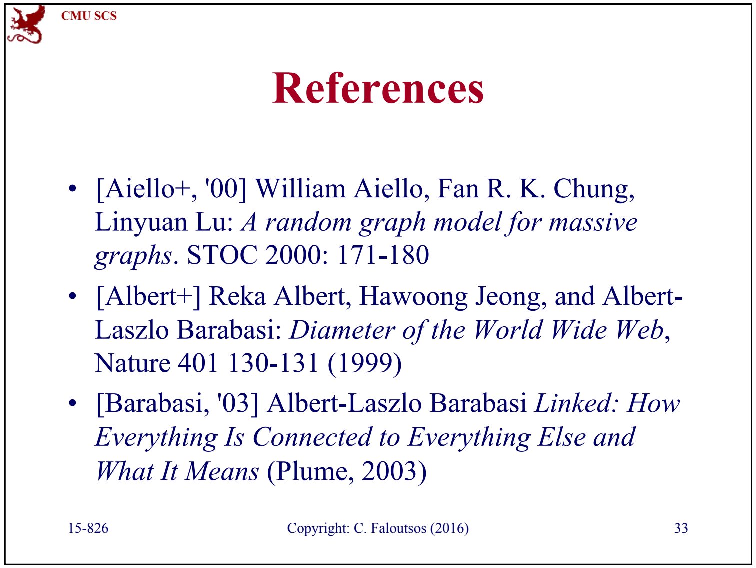
CMU SCS

Other resources

Visualization - graph algo's:

- Graphviz: http://www.graphviz.org/
- pajek: http://vlado.fmf.uni-lj.si/pub/networks/pajek/

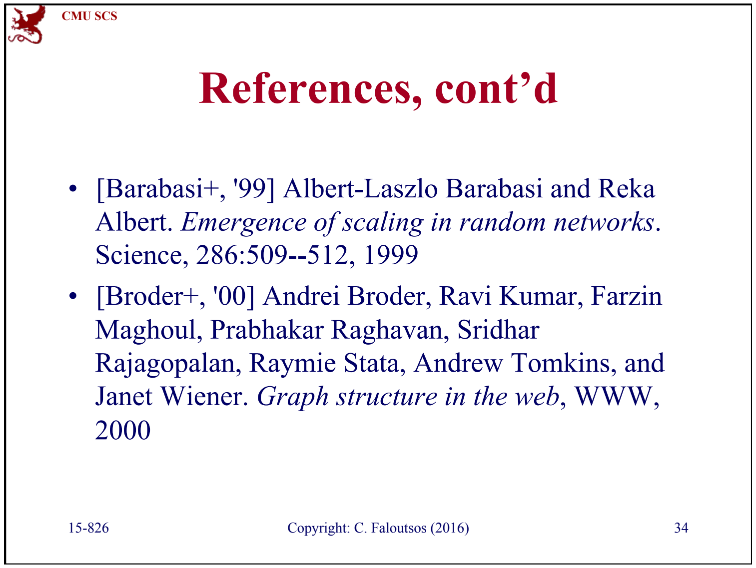
Kevin Bacon web site: http://www.cs.virginia.edu/oracle/



CMU SCS

References

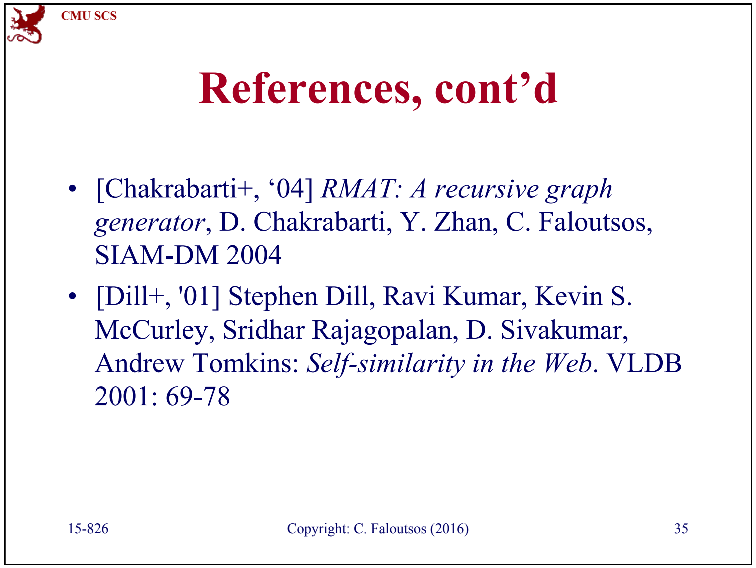
- [Aiello+, '00] William Aiello, Fan R. K. Chung, Linyuan Lu: *A random graph model for massive graphs*. STOC 2000: 171-180
- [Albert+] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi: *Diameter of the World Wide Web*, Nature 401 130-131 (1999)
- [Barabasi, '03] Albert-Laszlo Barabasi *Linked: How Everything Is Connected to Everything Else and What It Means* (Plume, 2003)



CMU SCS

References, cont'd

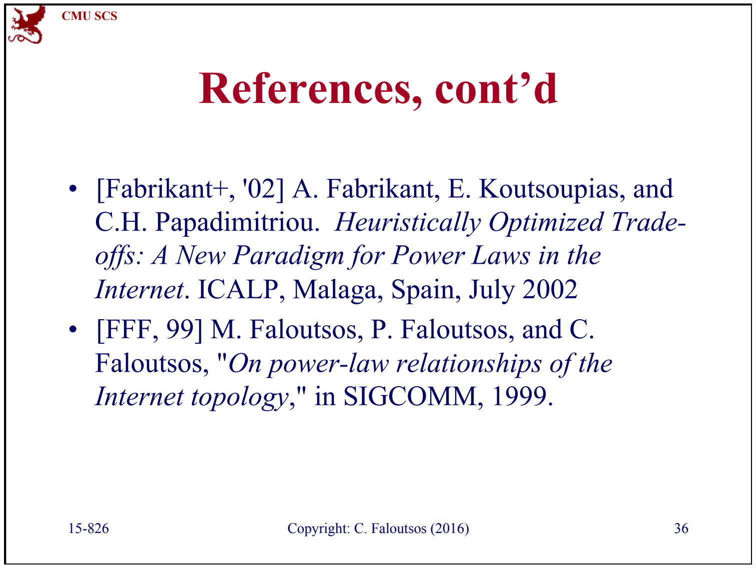
- [Barabasi+, '99] Albert-Laszlo Barabasi and Reka Albert. *Emergence of scaling in random networks*. Science, 286:509--512, 1999
- [Broder+, '00] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. *Graph structure in the web*, WWW, 2000



CMU SCS

References, cont'd

- [Chakrabarti+, '04] *RMAT: A recursive graph generator*, D. Chakrabarti, Y. Zhan, C. Faloutsos, SIAM-DM 2004
- [Dill+, '01] Stephen Dill, Ravi Kumar, Kevin S. McCurley, Sridhar Rajagopalan, D. Sivakumar, Andrew Tomkins: *Self-similarity in the Web*. VLDB 2001: 69-78



CMU SCS

References, cont'd

- [Fabrikant+, '02] A. Fabrikant, E. Koutsoupias, and C.H. Papadimitriou. *Heuristically Optimized Trade-offs: A New Paradigm for Power Laws in the Internet*. ICALP, Malaga, Spain, July 2002
- [FFF, 99] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "*On power-law relationships of the Internet topology*," in SIGCOMM, 1999.

References, cont’d

- [Jovanovic+, '01] M. Jovanovic, F.S. Annexstein, and K.A. Berman. *Modeling Peer-to-Peer Network Topologies through "Small-World" Models and Power Laws*. In TELFOR, Belgrade, Yugoslavia, November, 2001
- [Kumar+ '99] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins: *Extracting Large-Scale Knowledge Bases from the Web*. VLDB 1999: 639-650

References, cont’d

- [Leskovec+05b] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos *Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication* (ECML/PKDD 2005), Porto, Portugal, 2005.

References, cont’d

- [Leskovec+07] Jure Leskovec and Christos Faloutsos, Scalable Modeling of Real Graphs using Kronecker Multiplication, ICML 2007.

References, cont’d

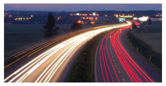
- [Pennock+, '02] David M. Pennock, Gary William Flake, Steve Lawrence, Eric J. Glover, C. Lee Giles: *Winners don't take all: Characterizing the competition for links on the web* Proc. Natl. Acad. Sci. USA 99(8): 5207-5211 (2002)

References, cont’d

- [Watts+ Strogatz, '98] D. J. Watts and S. H. Strogatz *Collective dynamics of 'small-world' networks*, Nature, 393:440-442 (1998)
- [Watts, '03] Duncan J. Watts *Six Degrees: The Science of a Connected Age* W.W. Norton & Company; (February 2003)

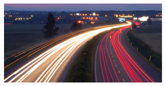
Graph mining: tools

Main outline



- Introduction
- Indexing
- Mining
  - – Graphs – patterns
  - ➡ – Graphs – generators and tools
  - – Association rules
  - – …

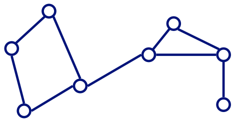
Detailed outline



- Graphs – generators
- Graphs – tools
  - ➡ – Community detection / graph partitioning
    - Algo’s
    - Observation: ‘no good cuts’
  - – Node proximity – personalized RWR
  - – Influence/virus propagation & immunization
  - – ‘Belief Propagation’ & fraud detection
  - – Anomaly detection

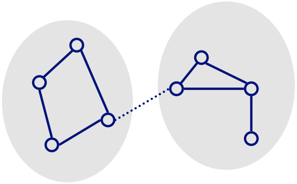
Problem

- Given a graph, and *k*
- Break it into *k* (disjoint) communities



Problem

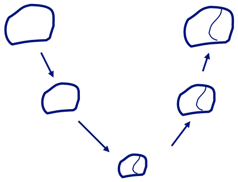
- Given a graph, and *k*
- Break it into *k* (disjoint) communities



$k = 2$

Solution #1: METIS

- Arguably, the best algorithm
- Open source, at
  - http://www.cs.umn.edu/~metis
- and *many* related papers, at same url
- Main idea:
  - coarsen the graph;
  - partition;
  - un-coarsen



Solution #1: METIS

- G. Karypis and V. Kumar. *METIS 4.0: Unstructured graph partitioning and sparse matrix ordering system*. TR, Dept. of CS, Univ. of Minnesota, 1998.
- <and many extensions>

Solution #2

(problem: hard clustering, *k* pieces)

Spectral partitioning:

- Consider the 2nd smallest eigenvector of the (normalized) Laplacian

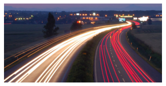
---

Solutions #3, …

Many more ideas:

- Clustering on the **A**^{2} (square of adjacency matrix) [Zhou, Woodruff, PODS’04]
- Minimum cut / maximum flow [Flake+, KDD’00]
- …

Detailed outline



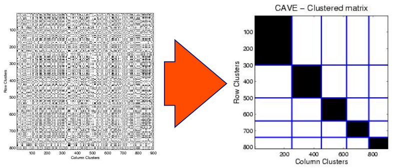
- Motivation
- Hard clustering – *k* pieces
- ➡ Hard co-clustering – (*k,l*) pieces
- Hard clustering – optimal # pieces
- Soft clustering – matrix decompositions
- Observations

Problem definition

- Given a bi-partite graph, and *k*, *l*
- Divide it into *k* row groups and *l* row groups
- (Also applicable to uni-partite graph)

Co-clustering

- Given data matrix and the number of row and column groups *k* and *l*
- Simultaneously
  - Cluster rows into *k* disjoint groups
  - Cluster columns into *l* disjoint groups



---

details

Co-clustering

- Let *X* and *Y* be discrete random variables
  - *X* and *Y* take values in *{1, 2, …, m}* and *{1, 2, …, n}*
  - *p(X, Y)* denotes the joint probability distribution—if not known, it is often estimated based on co-occurrence data
  - Application areas: text mining, market-basket analysis, analysis of browsing behavior, etc.
- Key Obstacles in Clustering Contingency Tables
  - High Dimensionality, Sparsity, Noise
  - Need for robust and scalable algorithms

Reference:

1. Dhillon et al. Information-Theoretic Co-clustering, KDD'03

$$\begin{array}{cc} & n \\ m & \begin{bmatrix} .05 & .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ .04 & .04 & 0 & .04 & .04 & .04 \\ .04 & .04 & .04 & 0 & .04 & .04 \end{bmatrix} \end{array}$$

eg, terms x documents

$$\begin{array}{cc} & k \\ m & \begin{bmatrix} .5 & 0 & 0 \\ .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \\ 0 & 0 & .5 \end{bmatrix} \end{array} \begin{array}{cc} & l \\ k & \begin{bmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{bmatrix} \end{array} \begin{array}{cc} & n \\ l & \begin{bmatrix} .36 & .36 & .28 & 0 & 0 & 0 \\ 0 & 0 & 0 & .28 & .36 & .36 \end{bmatrix} \end{array} = \begin{bmatrix} .054 & .054 & .042 & 0 & 0 & 0 \\ .054 & .054 & .042 & 0 & 0 & 0 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ .036 & .036 & 028 & .028 & .036 & .036 \\ .036 & .036 & .028 & .028 & .036 & .036 \end{bmatrix}$$

---

med. doc

cs doc

$$\begin{bmatrix} .05 & .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ .04 & .04 & 0 & .04 & .04 & .04 \\ .04 & .04 & .04 & 0 & .04 & .04 \end{bmatrix}$$

med. terms

cs terms

common terms

term group x doc. group

$$\begin{bmatrix} .5 & 0 & 0 \\ .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \\ 0 & 0 & .5 \end{bmatrix} \begin{bmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{bmatrix} \begin{bmatrix} .36 & .36 & .28 & 0 & 0 & 0 \\ 0 & 0 & 0 & .28 & .36 & .36 \end{bmatrix} = \begin{bmatrix} .054 & .054 & .042 & 0 & 0 & 0 \\ .054 & .054 & .042 & 0 & 0 & 0 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ .036 & .036 & 028 & .028 & .036 & .036 \\ .036 & .036 & .028 & .028 & .036 & .036 \end{bmatrix}$$

doc x doc group

term x term-group

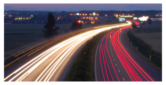
Co-clustering

Observations

- uses KL divergence, instead of L2
- the middle matrix is **not** diagonal
  - Like in the Tucker tensor decomposition
- s/w at:

www.cs.utexas.edu/users/dml/Software/cocluster.html

Detailed outline



- Motivation
- Hard clustering – k pieces
- Hard co-clustering – (k,l) pieces
- ➡ Hard clustering – optimal # pieces
- Soft clustering – matrix decompositions
- Observations

Problem with Information Theoretic Co-clustering

- Number of row and column groups must be specified

Desiderata:

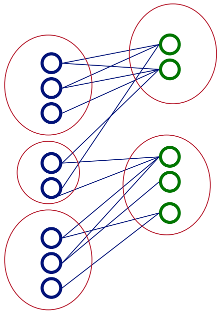
✓ Simultaneously discover row and column groups

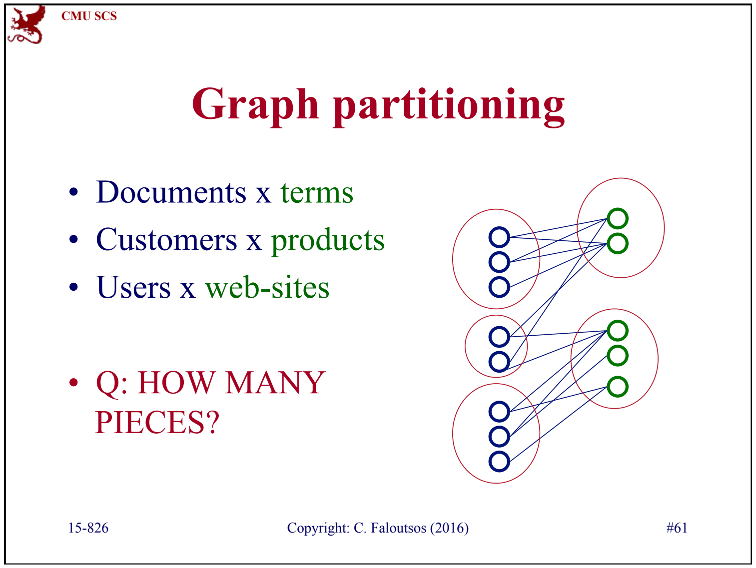
✗ Fully Automatic: No “magic numbers”

✓ Scalable to large graphs

Graph partitioning

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



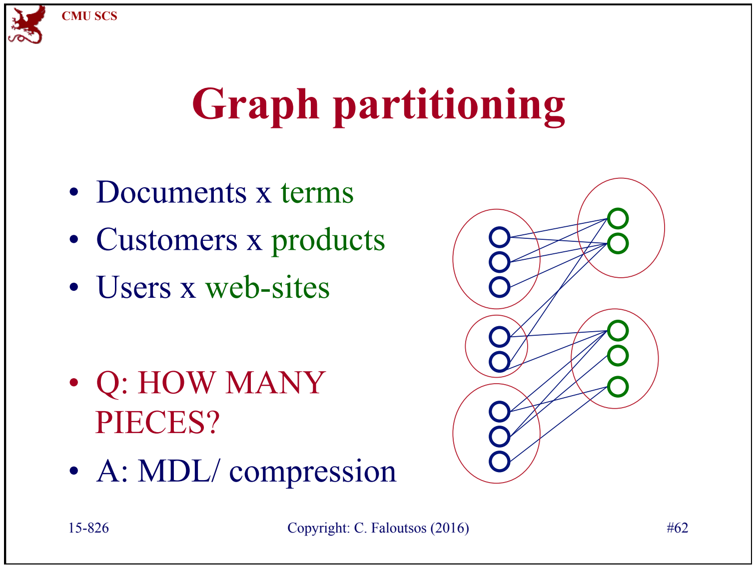


CMU SCS

Graph partitioning

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

- Q: HOW MANY PIECES?

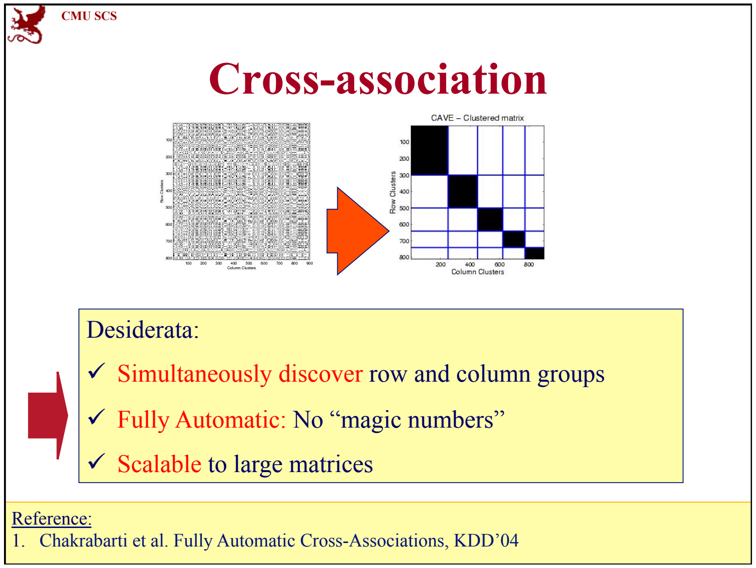


CMU SCS

Graph partitioning

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

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



CMU SCS

Cross-association

Desiderata:

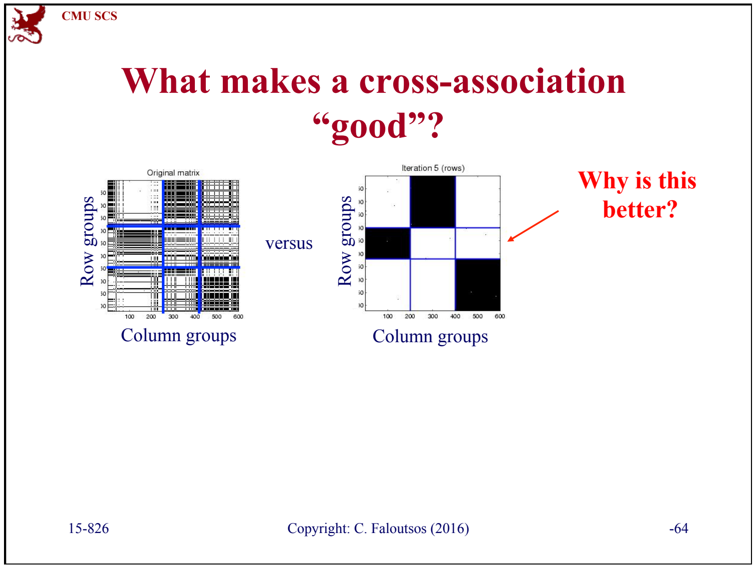
✓ Simultaneously discover row and column groups

✓ Fully Automatic: No “magic numbers”

✓ Scalable to large matrices

Reference:

1. Chakrabarti et al. Fully Automatic Cross-Associations, KDD’04

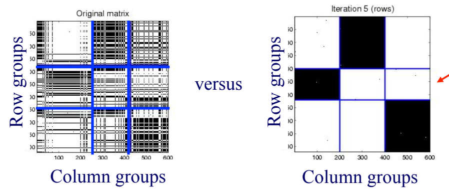


CMU SCS

What makes a cross-association “good”?

Why is this better?

What makes a cross-association "good"?

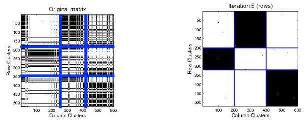


Why is this better?

simpler; easier to describe

easier to compress!

What makes a cross-association "good"?



Problem definition: given an encoding scheme
- decide on the # of col. and row groups *k* and *l*
- and reorder rows and columns,
- to achieve best compression

---

details

Main Idea

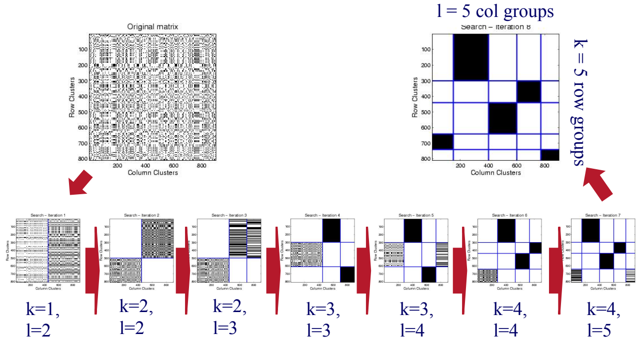
Good Compression → Better Clustering

$$\text{Total Encoding Cost} = \underbrace{\Sigma_i \; size_i * H(x_i)}_{\text{Code Cost}} + \underbrace{\text{Cost of describing cross-associations}}_{\text{Description Cost}}$$

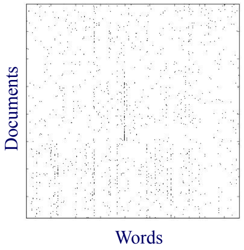
Minimize the total cost (# bits)

for lossless compression

Algorithm



Experiments



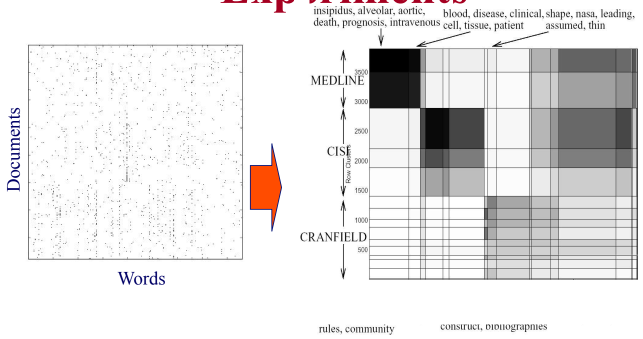
"CLASSIC"

- 3,893 documents
- 4,303 words
- 176,347 "dots"

Combination of 3 sources:

- MEDLINE (medical)
- CISI (info. retrieval)
- CRANFIELD (aerodynamics)

Experiments



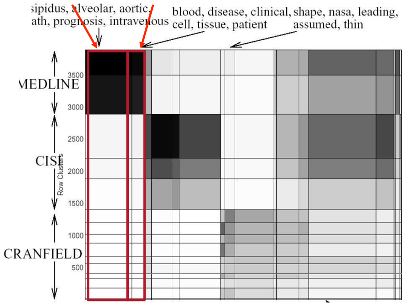
"CLASSIC" graph of documents & words: k=15, l=19

Experiments

insipidus, alveolar, aortic, death, prognosis, intravenous

blood, disease, clinical, cell, tissue, patient

MEDLINE (medical)



"CLASSIC" graph of documents & words: k=15, l=19

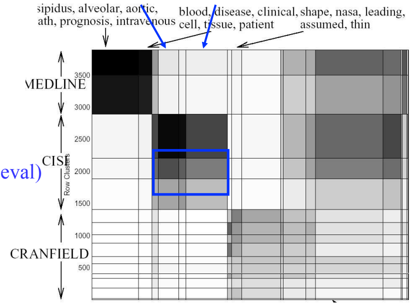
Experiments

providing, studying, records, development, students, rules

abstract, notation, works, construct, bibliographies

MEDLINE (medical)

CISI (Information Retrieval)



"CLASSIC" graph of documents & words: k=15, l=19

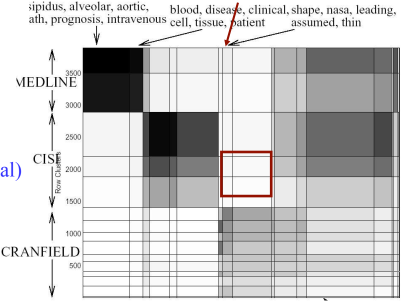
Experiments

shape, nasa, leading, assumed, thin

MEDLINE (medical)

CISI (Information Retrieval)

CRANFIELD (aerodynamics)



"CLASSIC" graph of documents & words: k=15, l=19

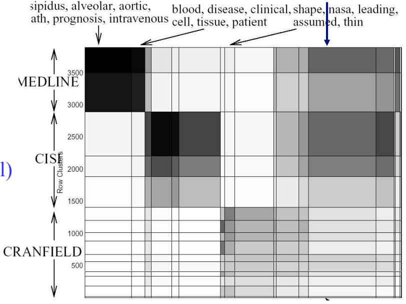
Experiments

paint, examination, fall, raise, leave, based

MEDLINE (medical)

CISI (Information Retrieval)

CRANFIELD (aerodynamics)



"CLASSIC" graph of documents & words: k=15, l=19

Algorithm

Code for cross-associations (matlab):

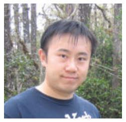
www.cs.cmu.edu/~deepay/mywww/software/CrossAssociations-01-27-2005.tgz

Variations and extensions:

- 'Autopart' [Chakrabarti, PKDD'04]
- www.cs.cmu.edu/~deepay

Algorithm

- Hadoop implementation [ICDM'08]



Spiros Papadimitriou, Jimeng Sun: DisCo: Distributed Co-clustering with Map-Reduce: A Case Study towards Petabyte-Scale End-to-End Mining. ICDM 2008: 512-521

Detailed outline

- Motivation
- Hard clustering – *k* pieces
- Hard co-clustering – (*k*,*l*) pieces
- Hard clustering – optimal # pieces
- ➡ (Soft clustering – matrix decompositions
  - PCA, ICA, non-negative matrix factorization, …)
- Observations

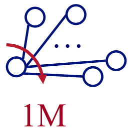
Detailed outline

- Motivation
- Hard clustering – *k* pieces
- Hard co-clustering – (*k*,*l*) pieces
- Hard clustering – optimal # pieces
- (Soft clustering)
- ➡ Observations

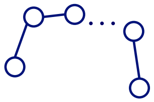
Observation #1

- Skewed degree distributions – there are nodes with huge degree (>O(10^4), in facebook/linkedIn popularity contests!)
- TRAP: '*find all pairs of nodes, within 2 steps from each other*'

Gaussian trap



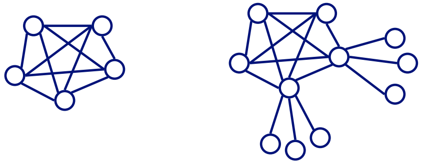
Observation #2



- TRAP: *shortest-path between two nodes*
- (cheat: look for 2, at most 3-step paths)
- Why:
  - If they are close (within 2-3 steps): solved
  - If not, after ~6 steps, you'll have ~ the whole graph, and the path won't be very meaningful, anyway.

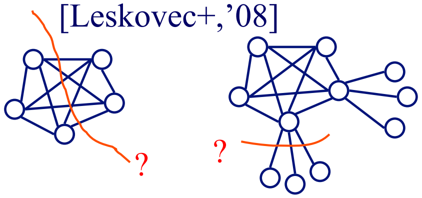
Observation #3

- Maybe there are no good cuts: ``jellyfish'' shape [Tauro+'01], [Siganos+,'06], strange behavior of cuts [Chakrabarti+'04], [Leskovec+,'08]

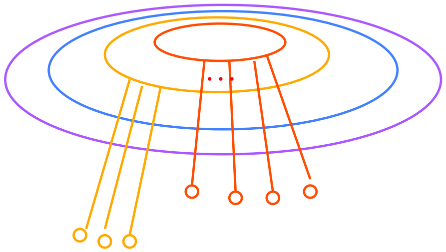


Observation #3

- Maybe there are no good cuts: ``jellyfish'' shape [Tauro+'01], [Siganos+,'06], strange behavior of cuts [Chakrabarti+,'04], [Leskovec+,'08]



Jellyfish model [Tauro+]



A Simple Conceptual Model for the Internet Topology, L. Tauro, C. Palmer, G. Siganos, M. Faloutsos, Global Internet, November 25-29, 2001

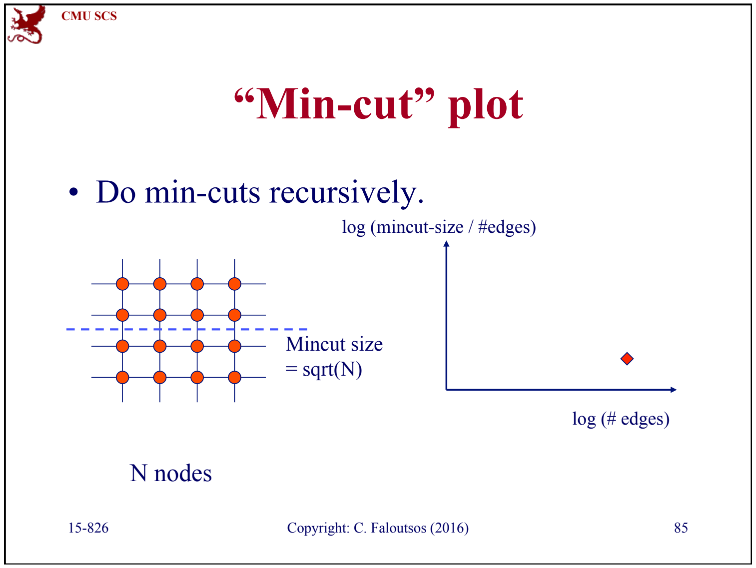
Jellyfish: A Conceptual Model for the AS Internet Topology G. Siganos, Sudhir L Tauro, M. Faloutsos, J. of Communications and Networks, Vol. 8, No. 3, pp 339-350, Sept. 2006.

Strange behavior of min cuts

- 'negative dimensionality' (!)

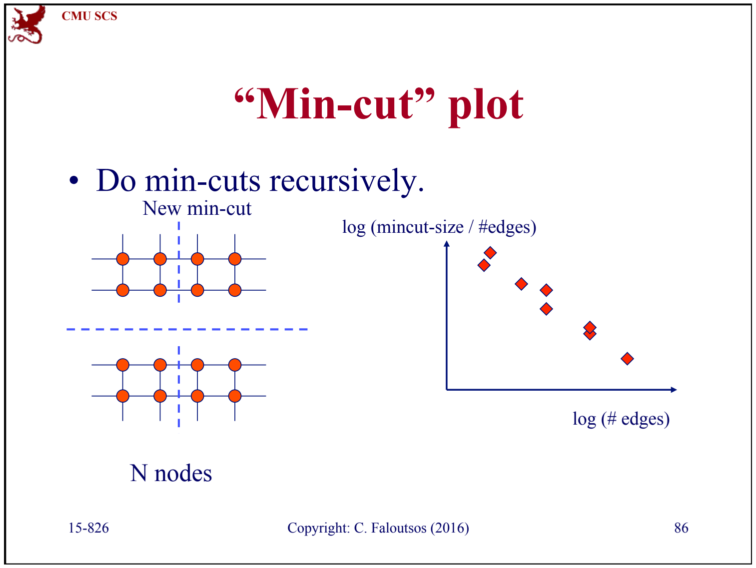
NetMine: New Mining Tools for Large Graphs, by D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos and G. Blelloch, in the SDM 2004 Workshop on Link Analysis, Counter-terrorism and Privacy

Statistical Properties of Community Structure in Large Social and Information Networks, J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. WWW 2008.



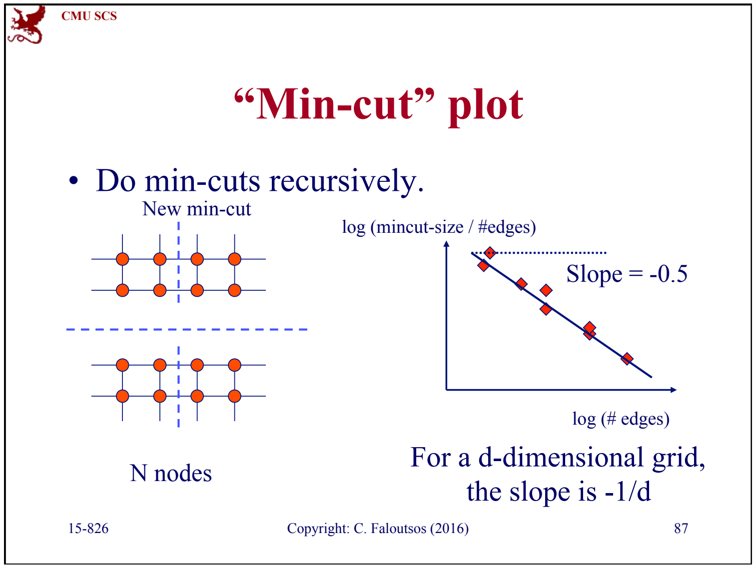
"Min-cut" plot

- Do min-cuts recursively.



"Min-cut" plot

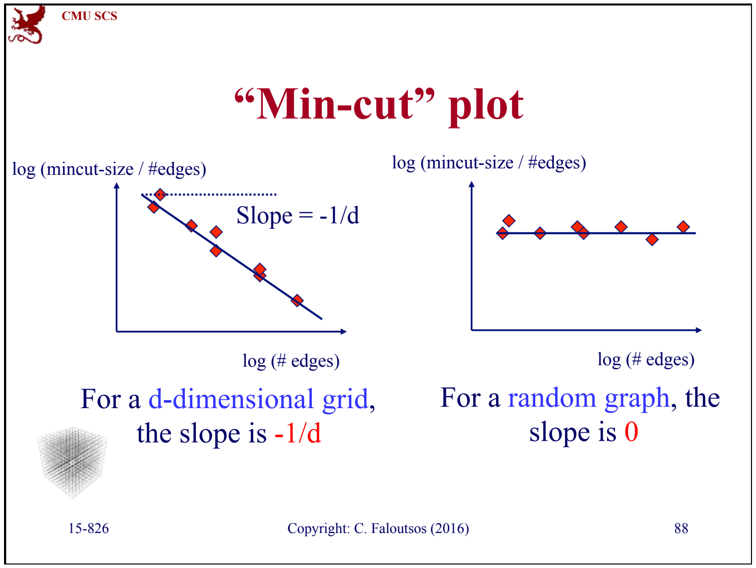
- Do min-cuts recursively.



"Min-cut" plot

- Do min-cuts recursively.

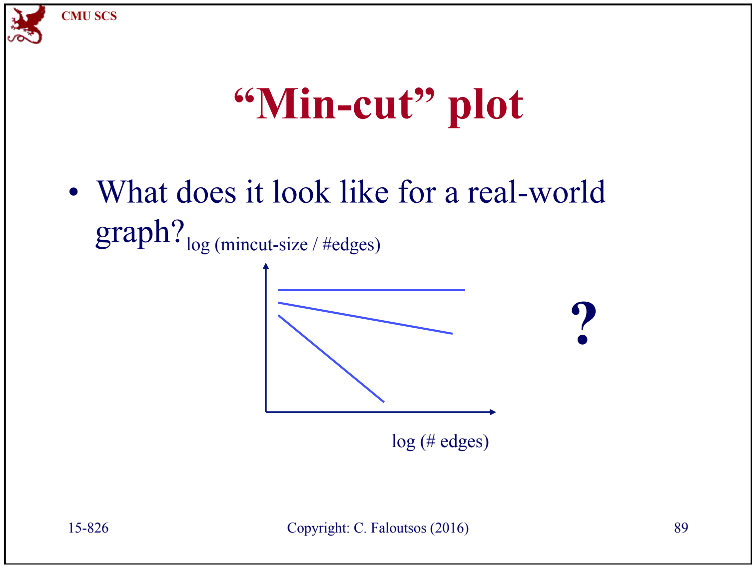
For a d-dimensional grid,
the slope is -1/d



"Min-cut" plot

For a d-dimensional grid,
the slope is -1/d

For a random graph, the
slope is 0



CMU SCS

"Min-cut" plot

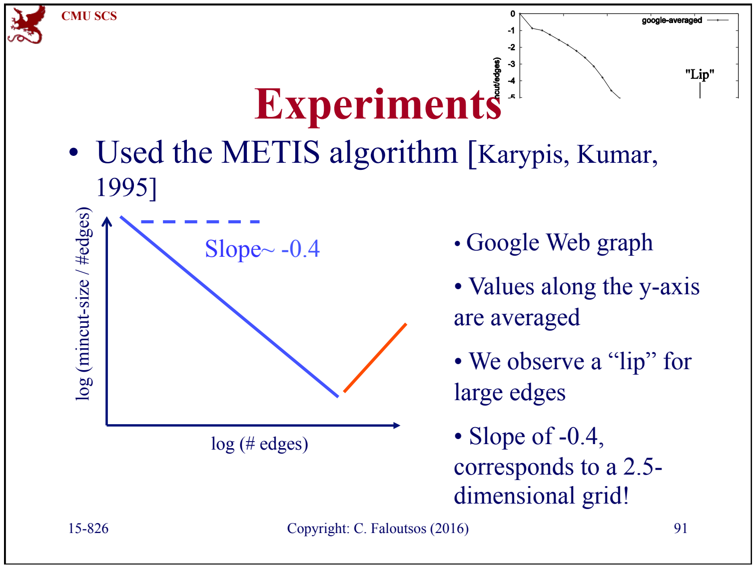
- What does it look like for a real-world graph?

CMU SCS

Experiments

- Datasets:
  - Google Web Graph: 916,428 nodes and 5,105,039 edges
  - Lucent Router Graph: Undirected graph of network routers from www.isi.edu/scan/mercator/maps.html; 112,969 nodes and 181,639 edges
  - User ➔ Website Clickstream Graph: 222,704 nodes and 952,580 edges

NetMine: New Mining Tools for Large Graphs, by D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos and G. Blelloch, in the SDM 2004 Workshop on Link Analysis, Counter-terrorism and Privacy

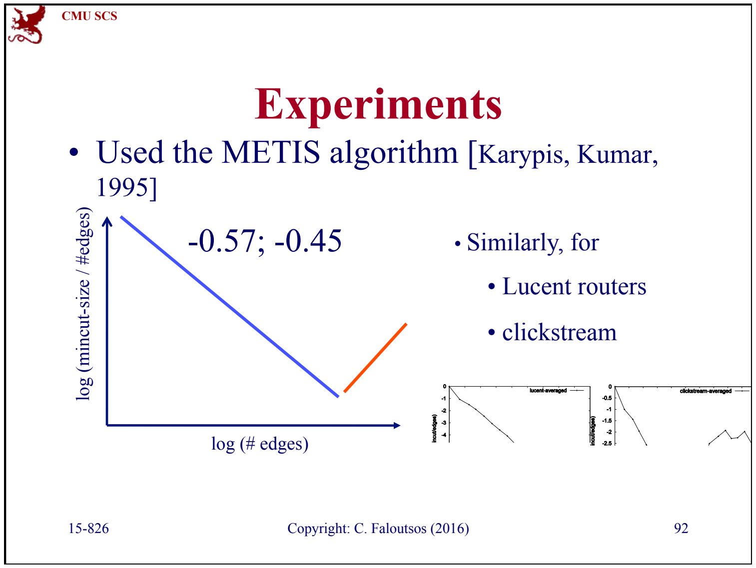


CMU SCS

Experiments

- Used the METIS algorithm [Karypis, Kumar, 1995]

- Google Web graph
- Values along the y-axis are averaged
- We observe a "lip" for large edges
- Slope of -0.4, corresponds to a 2.5-dimensional grid!

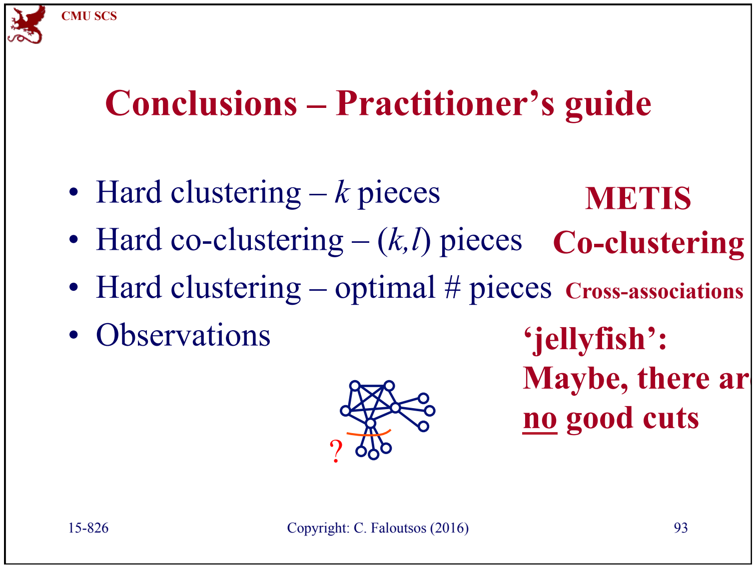


CMU SCS

Experiments

- Used the METIS algorithm [Karypis, Kumar, 1995]

- Similarly, for
  - Lucent routers
  - clickstream



Conclusions – Practitioner’s guide

- Hard clustering – *k* pieces **METIS**
- Hard co-clustering – (*k,l*) pieces **Co-clustering**
- Hard clustering – optimal # pieces **Cross-associations**
- Observations

‘jellyfish’:
Maybe, there ar
no good cuts