Large Graph Mining: Power Tools and a Practitioner's guide

Task 2: Community Detection

Faloutsos, Miller, Tsourakakis

CMU

Outline

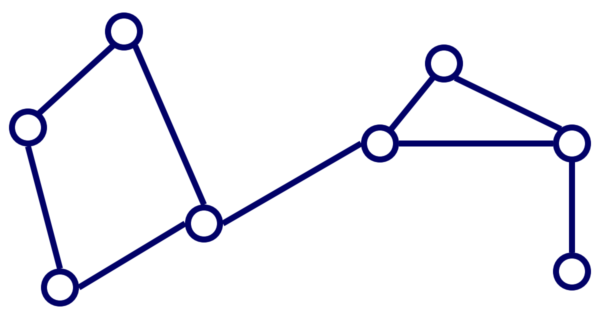
- Introduction – Motivation
- Task 1: Node importance
- Task 2: Community detection
- Task 3: Recommendations
- Task 4: Connection sub-graphs
- Task 5: Mining graphs over time
- Task 6: Virus/influence propagation
- Task 7: Spectral graph theory
- Task 8: Tera/peta graph mining: hadoop
- Observations – patterns of real graphs
- Conclusions

Detailed outline

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

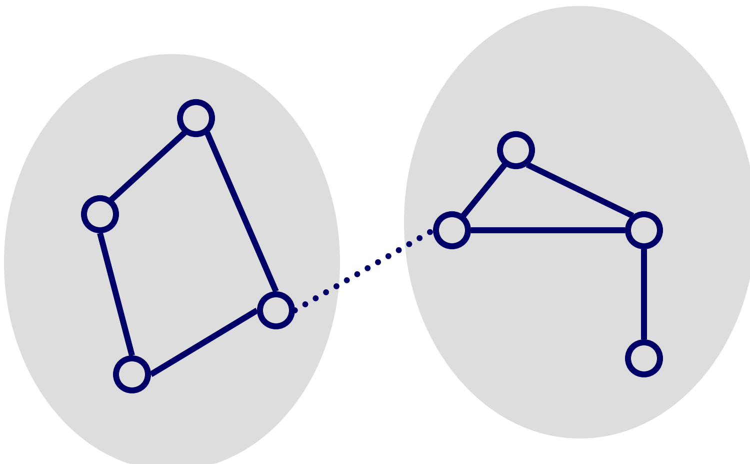
Problem

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



Problem

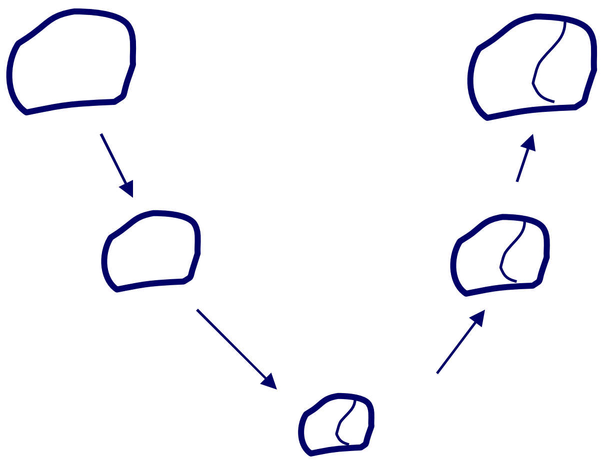
- Given a graph, and *k*
- Break it into *k* (disjoint) communities



$k = 2$

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

See details in 'Task 7', later

Solutions #3, …

Many more ideas:

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

Detailed outline

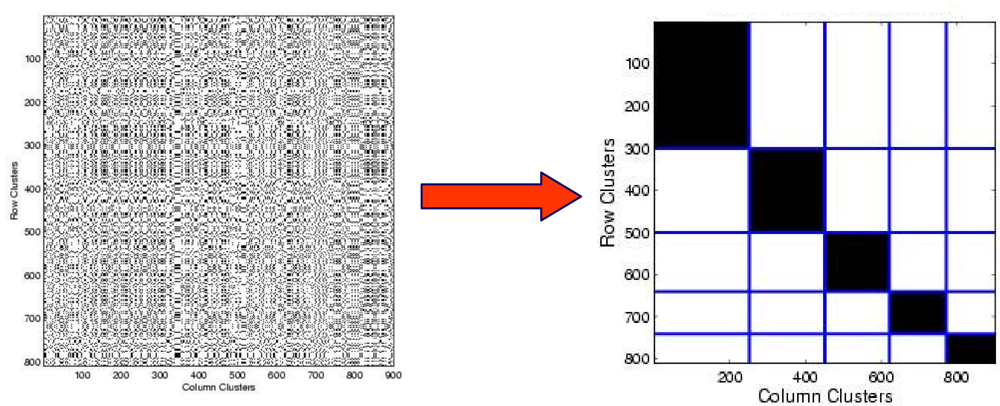
- Motivation
- Hard clustering – *k* pieces
- Hard co-clustering – (*k*,*l*) pieces
- Hard clustering – optimal # pieces
- Soft clustering – matrix decompositions
- Observations

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 $\{1, 2, \ldots, m\}$ and $\{1, 2, \ldots, n\}$
  - $p(X, Y)$ 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

$$
\begin{array}{cc}
 & k \\
m & \begin{bmatrix} .5 & 0 & 0 \\ .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \\ 0 & 0 & .5 \end{bmatrix}
\end{array}
\begin{array}{cc}
 & l \\
k & \begin{bmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{bmatrix}
\end{array}
\begin{array}{cc}
 & n \\
l & \begin{bmatrix} .36 & .36 & .28 & 0 & 0 & 0 \\ 0 & 0 & 0 & .28 & .36 & .36 \end{bmatrix}
\end{array}
=
\left[\begin{array}{ccc|ccc} .054 & .054 & .042 & 0 & 0 & 0 \\ .054 & .054 & .042 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & .042 & .054 & .054 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ \hline .036 & .036 & 028 & .028 & .036 & .036 \\ .036 & .036 & .028 & .028 & .036 & .036 \end{array}\right]
$$

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

$$\begin{bmatrix} .5 & 0 & 0 \\ .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \\ 0 & 0 & .5 \end{bmatrix} \begin{bmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{bmatrix} \begin{bmatrix} .36 & .36 & .28 & 0 & 0 & 0 \\ 0 & 0 & 0 & .28 & .36 & .36 \end{bmatrix} = \left[\begin{array}{ccc|ccc} .054 & .054 & .042 & 0 & 0 & 0 \\ .054 & .054 & .042 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & .042 & .054 & .054 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ \hline .036 & .036 & 028 & .028 & .036 & .036 \\ .036 & .036 & .028 & .028 & .036 & .036 \end{array}\right]$$

doc x doc group

term x term-group

Co-clustering

Observations

- uses KL divergence, instead of L2
- the middle matrix is **not** diagonal
  - we'll see that again 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

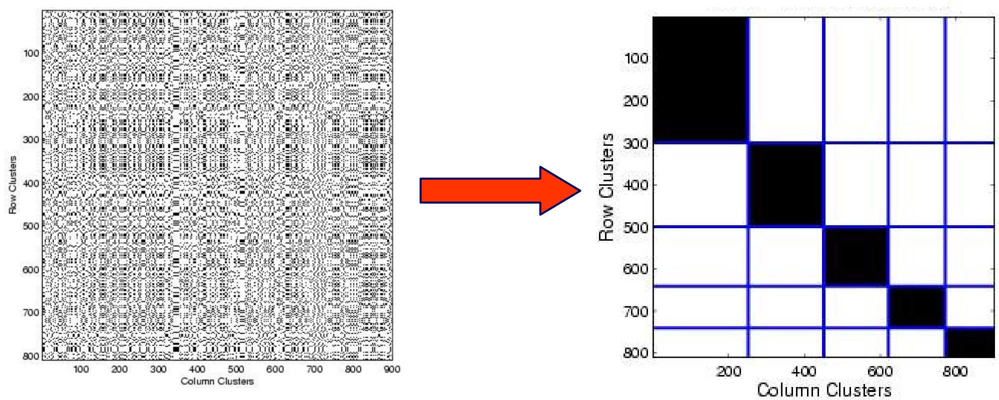
Desiderata:

✓ Simultaneously discover row and column groups

✗ Fully Automatic: No “magic numbers”

✓ Scalable to large graphs

Cross-association



Desiderata:

✓ Simultaneously discover row and column groups

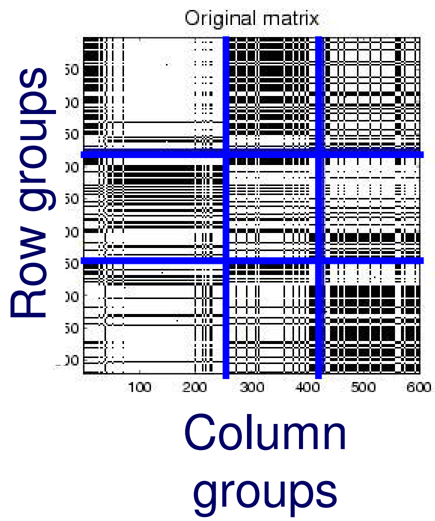
✓ Fully Automatic: No "magic numbers"

✓ Scalable to large matrices

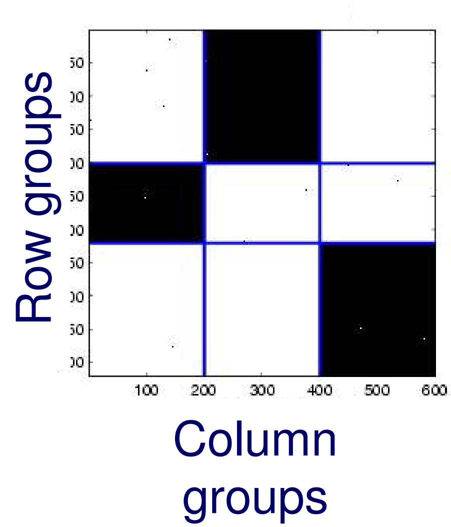
Reference:

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

What makes a cross-association "good"?

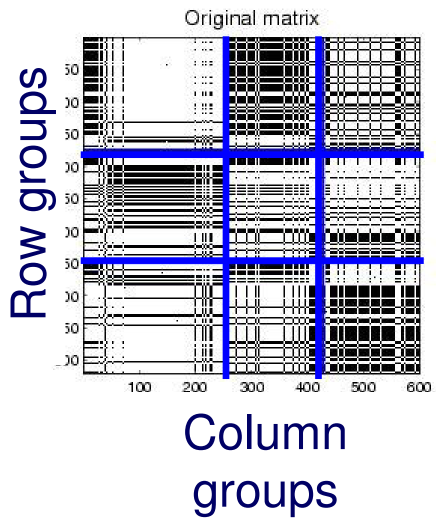


versus

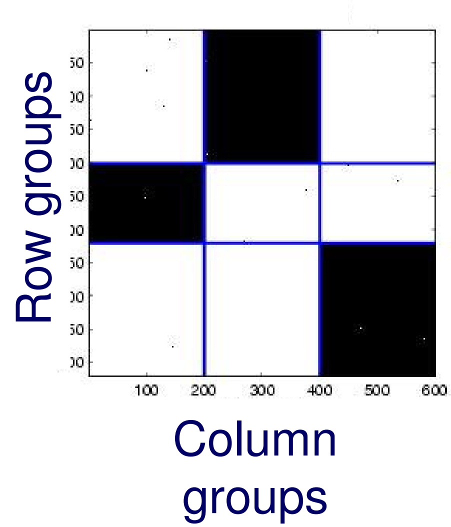


Why is this better?

What makes a cross-association "good"?



versus

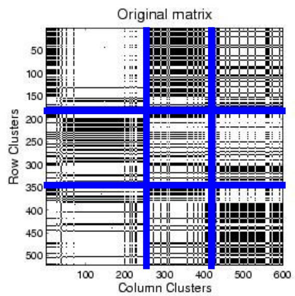


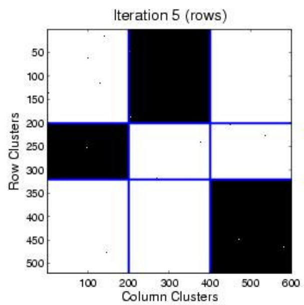
Why is this better?

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

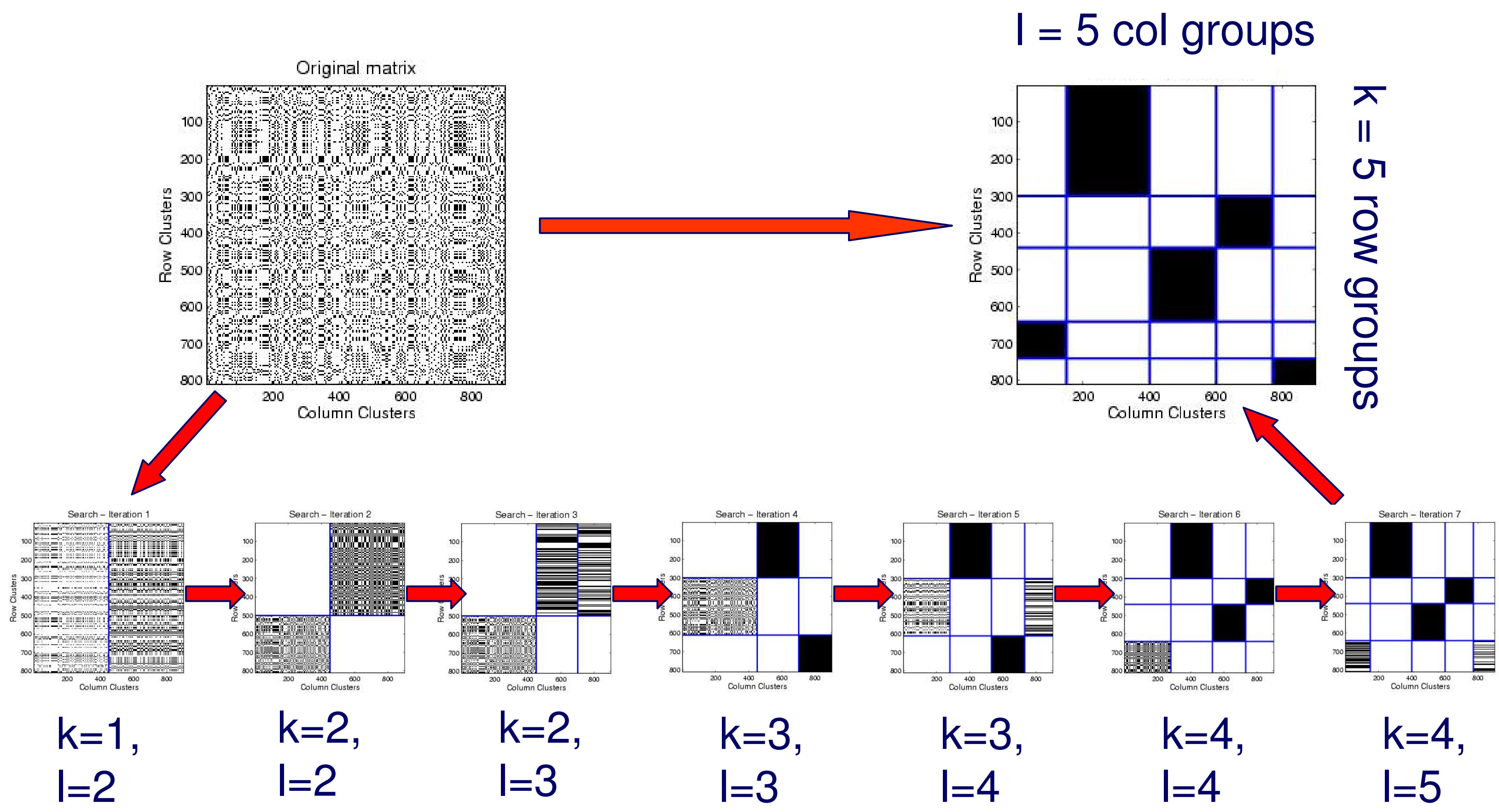
Good Compression	→	Better Clustering

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

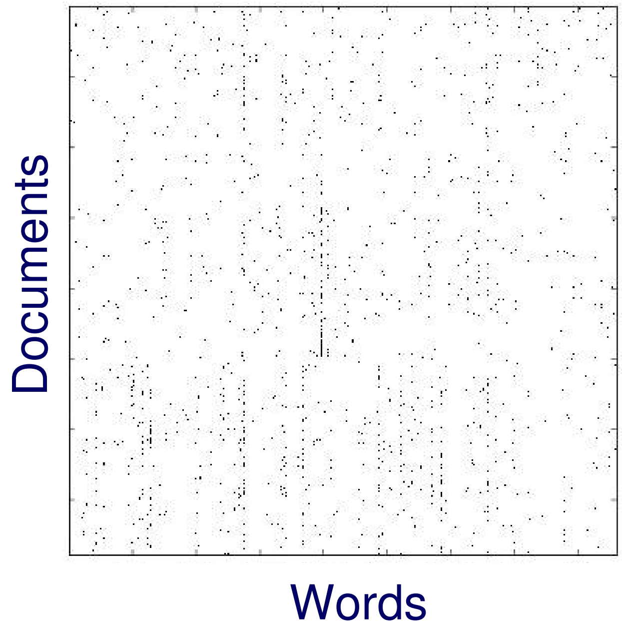
Minimize the total cost (# bits)

for lossless compression

Algorithm


l = 5 col groups
k = 5 row groups
Original matrix
Row Clusters
Column Clusters
Search – Iteration 1
Search – Iteration 2
Search – Iteration 3
Search – Iteration 4
Search – Iteration 5
Search – Iteration 6
Search – Iteration 7
k=1, l=2
k=2, l=2
k=2, l=3
k=3, l=3
k=3, l=4
k=4, l=4
k=4, l=5

Experiments



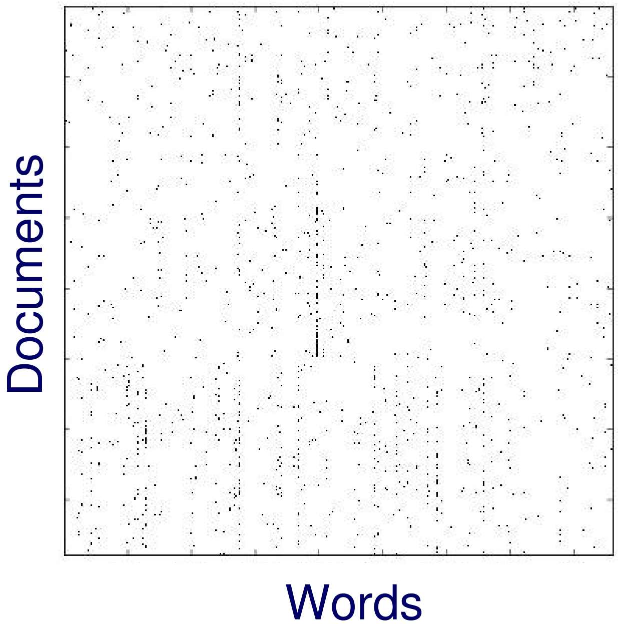
"CLASSIC"

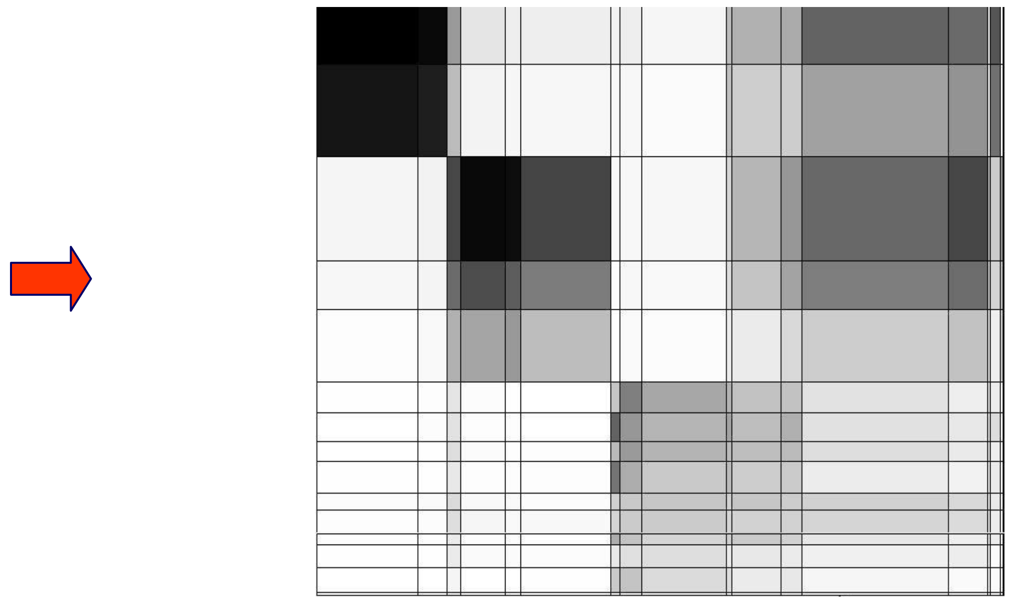
- 3,893 documents
- 4,303 words
- 176,347 "dots"

Combination of 3 sources:

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

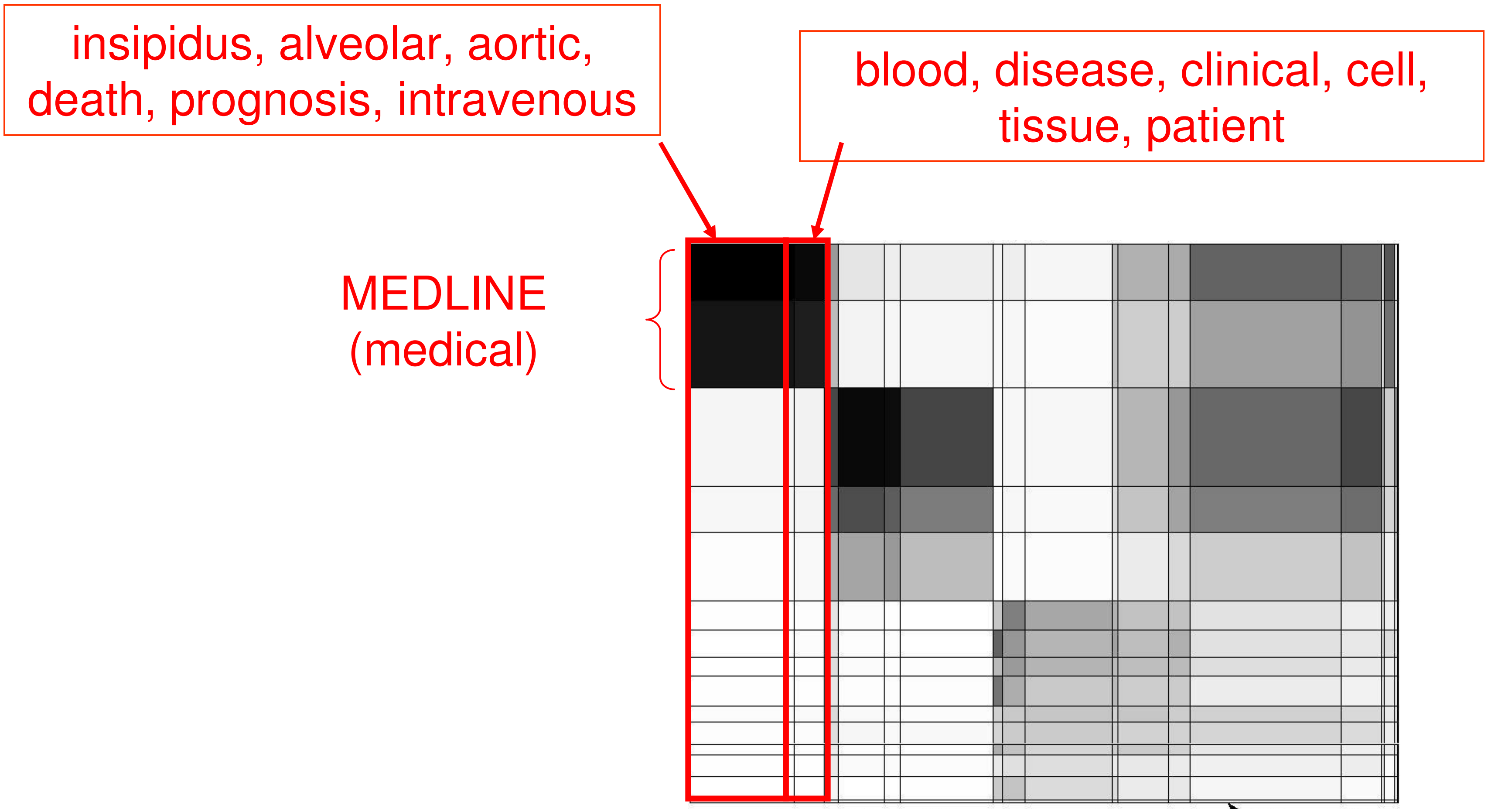
Experiments





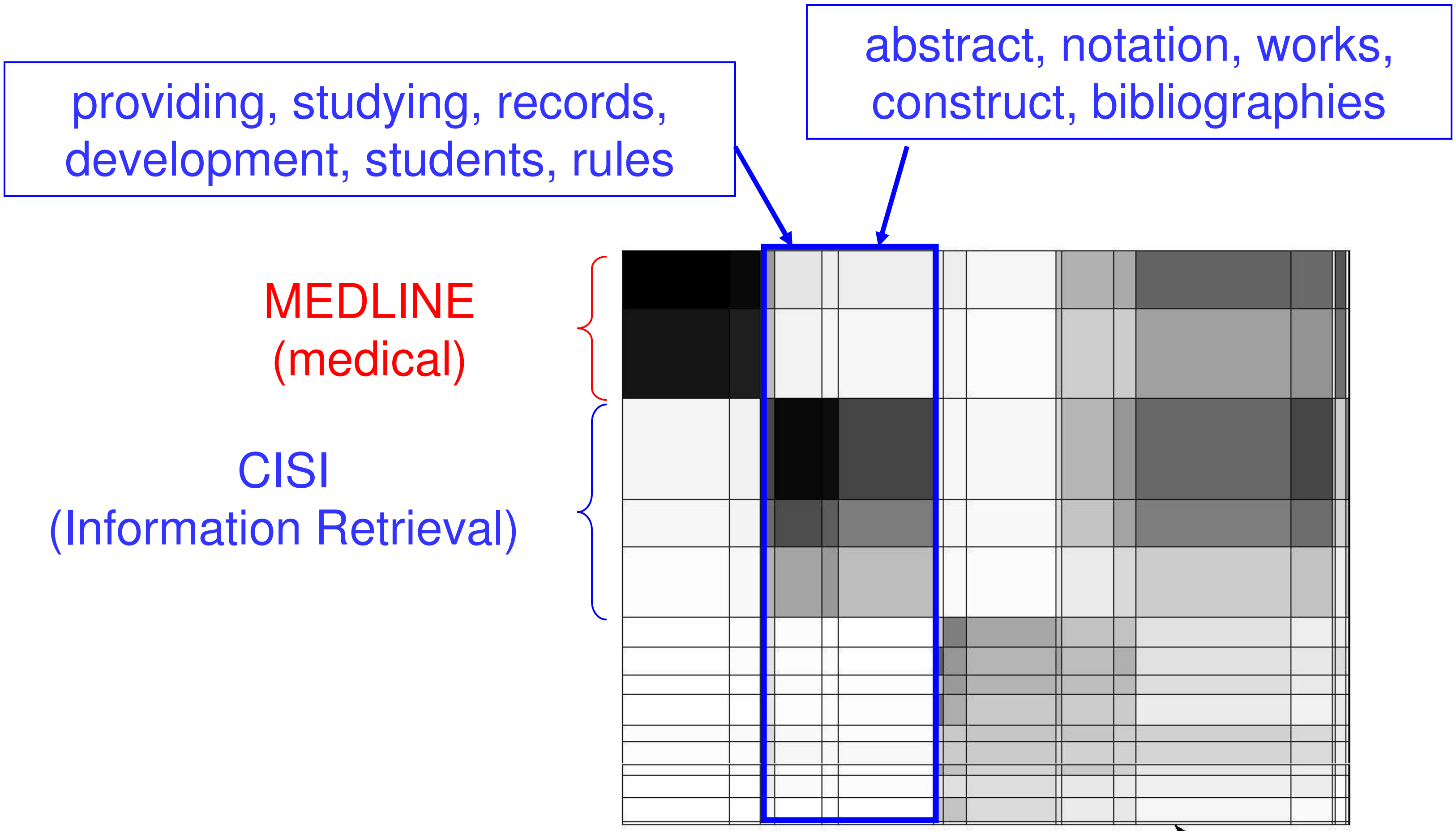
"CLASSIC" graph of documents & words: k=15, l=19

Experiments



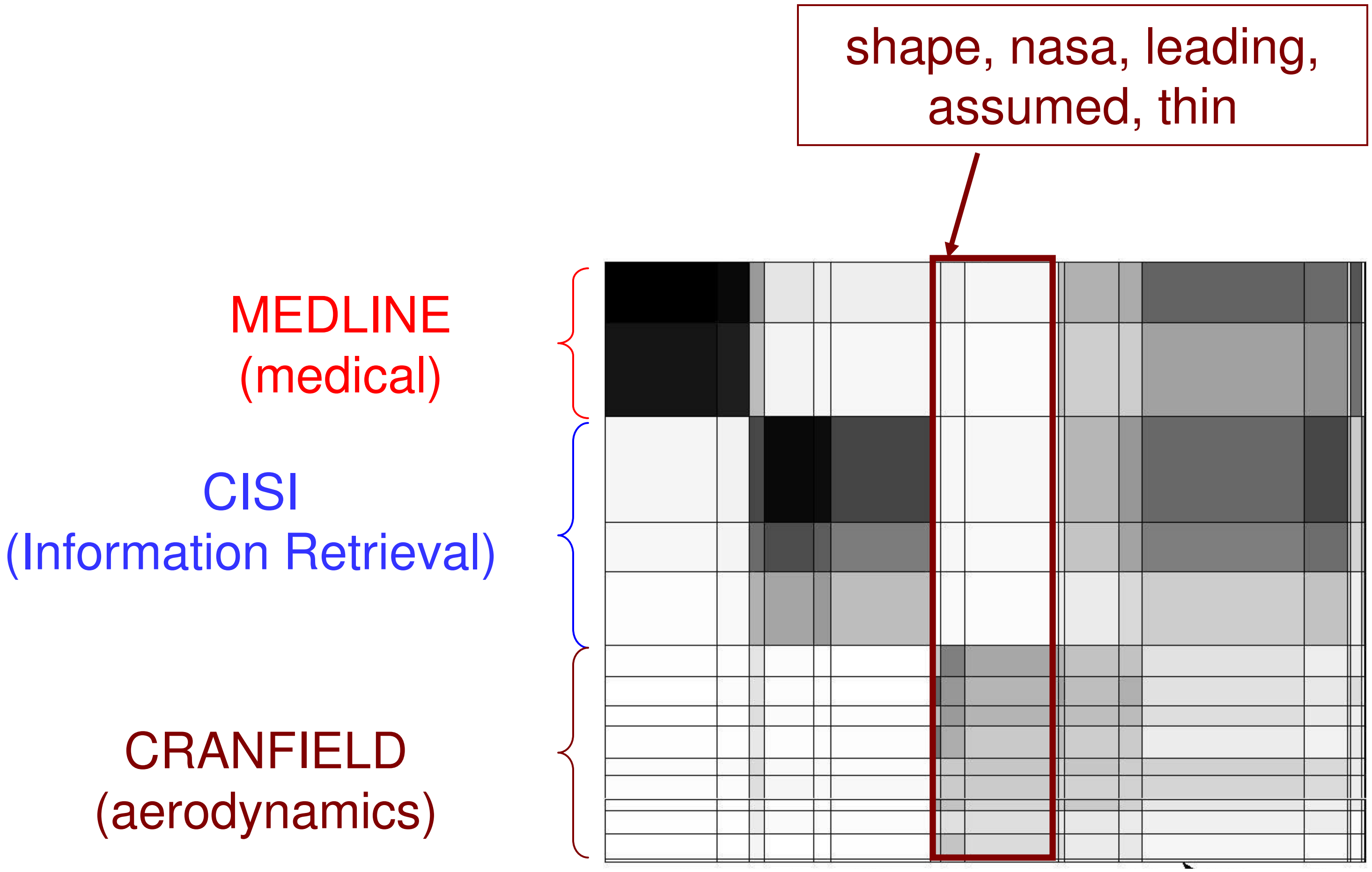
"CLASSIC" graph of documents & words: k=15, l=19

Experiments



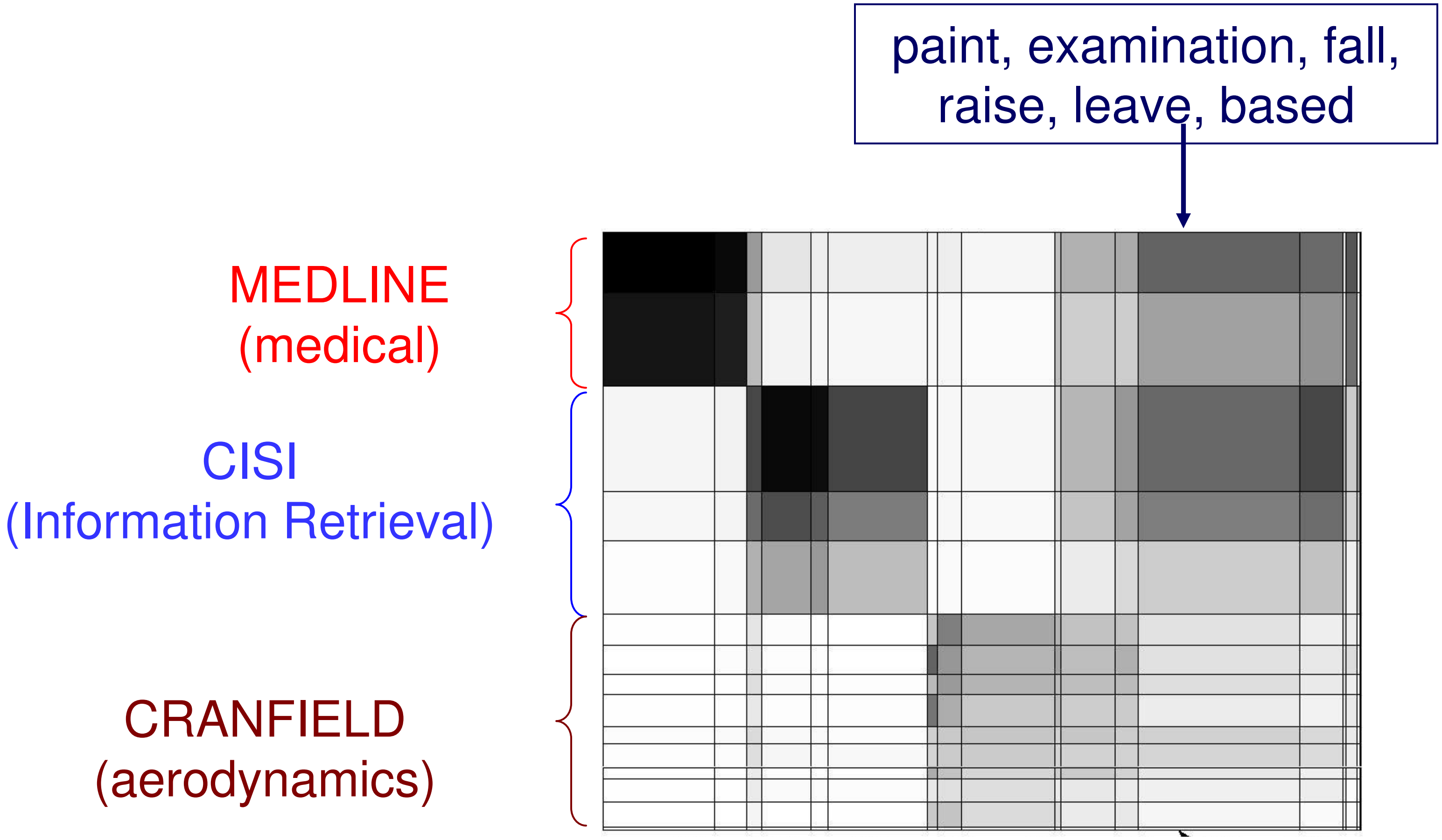
"CLASSIC" graph of documents & words: k=15, l=19

Experiments



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

Experiments



"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

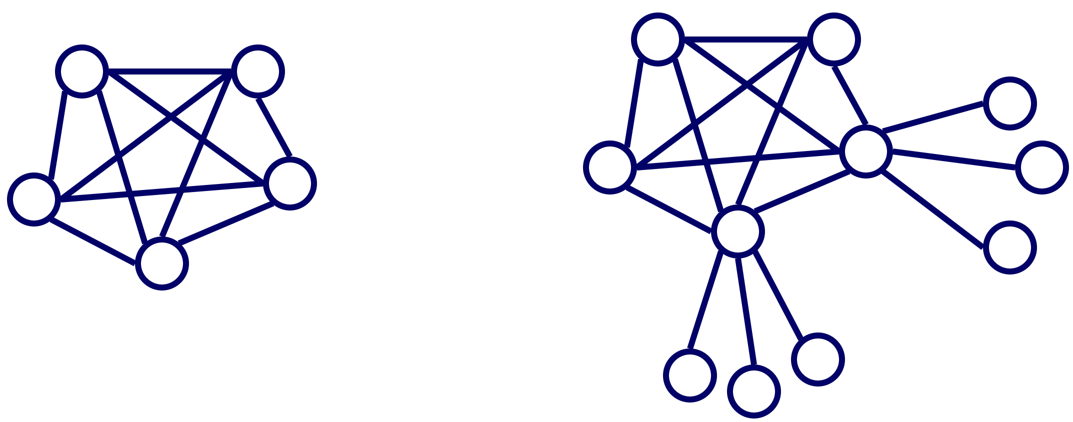
- Motivation
- Hard clustering – k pieces
- Hard co-clustering – (k,l) pieces
- Hard clustering – optimal # pieces
- Observations

Observation #1

- Skewed degree distributions – there are nodes with huge degree (>O(10^4), in facebook/linkedIn popularity contests!)

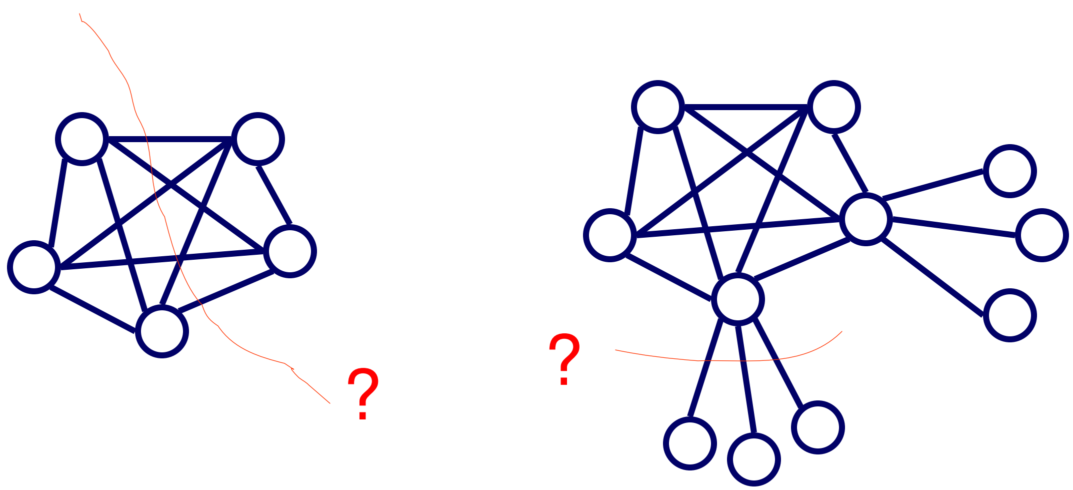
Observation #2

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

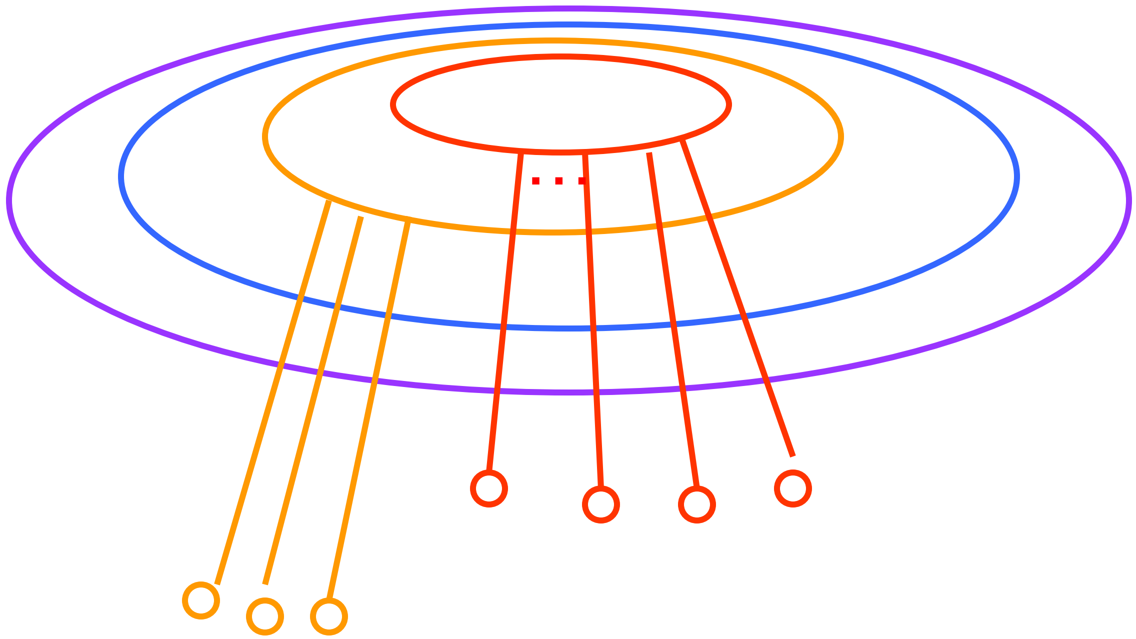


Observation #2

- 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

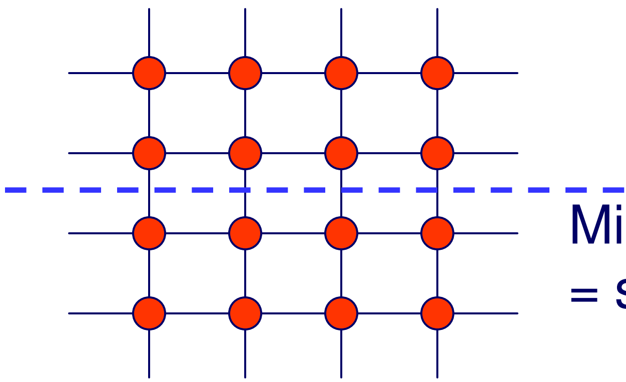
- 'negative dimensionality' (!)

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.

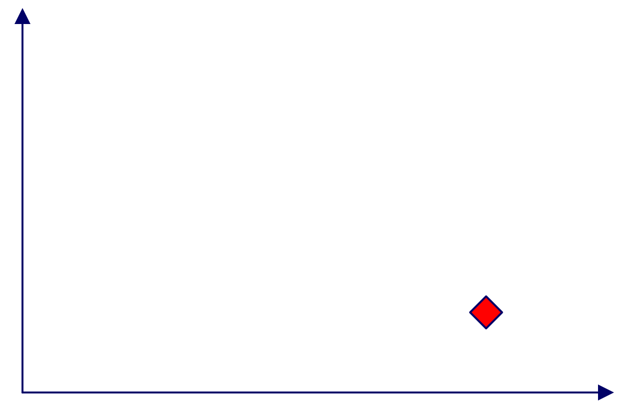
"Min-cut" plot

- Do min-cuts recursively.



Mincut size
= sqrt(N)

log (mincut-size / #edges)

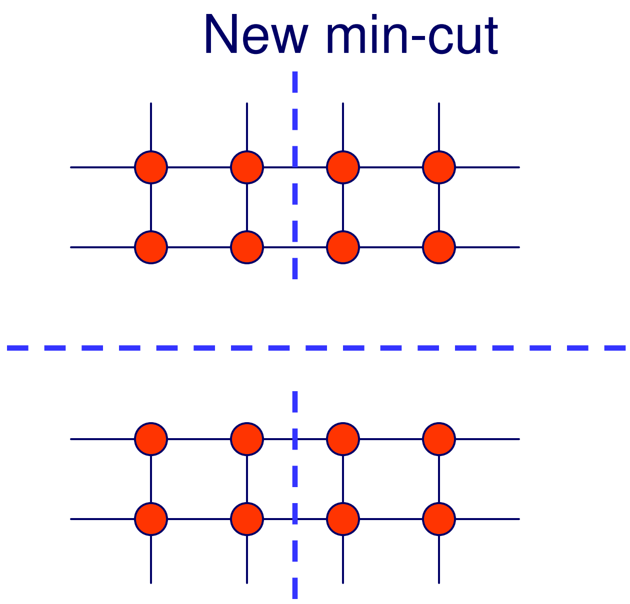


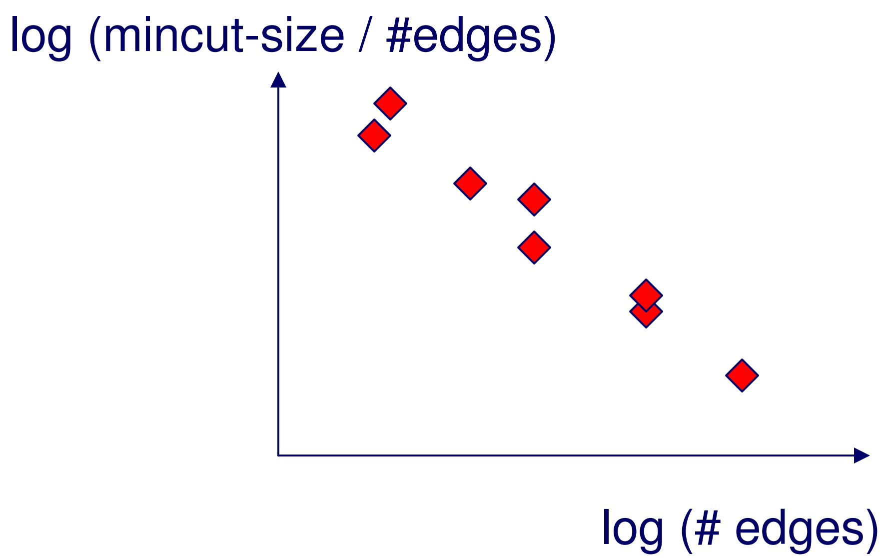
log (# edges)

N nodes

"Min-cut" plot

- Do min-cuts recursively.

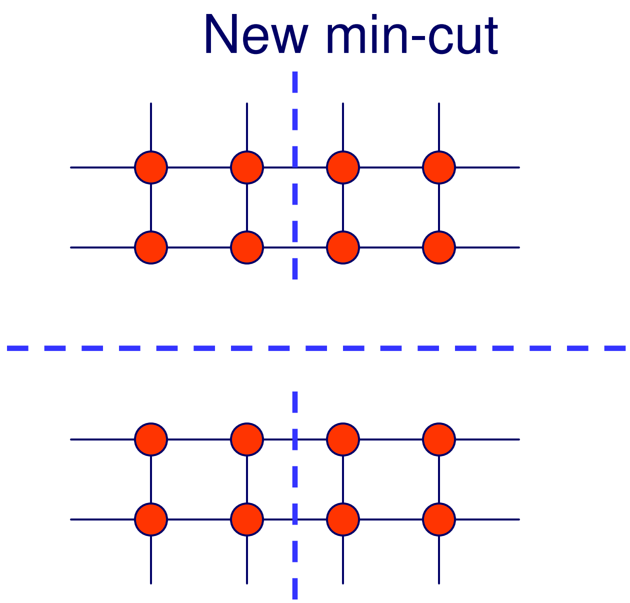




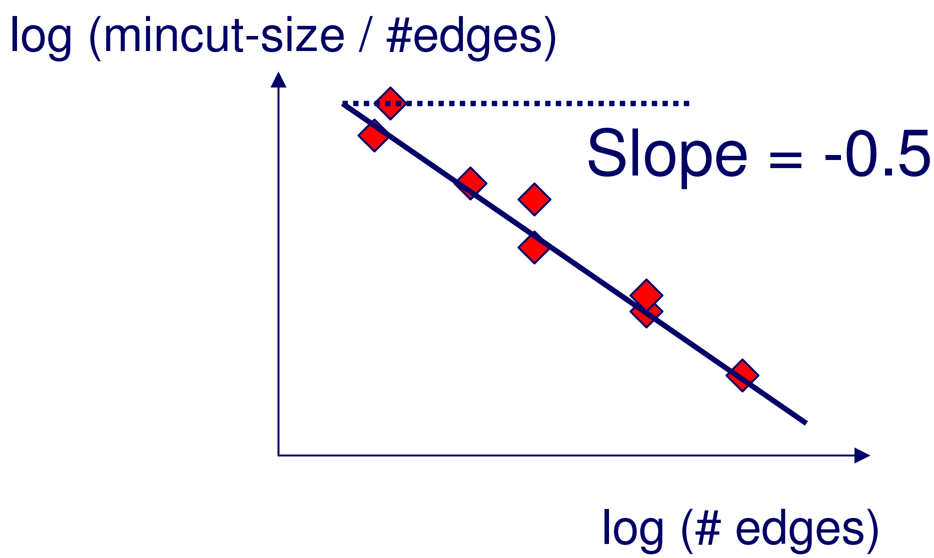
N nodes

"Min-cut" plot

- Do min-cuts recursively.

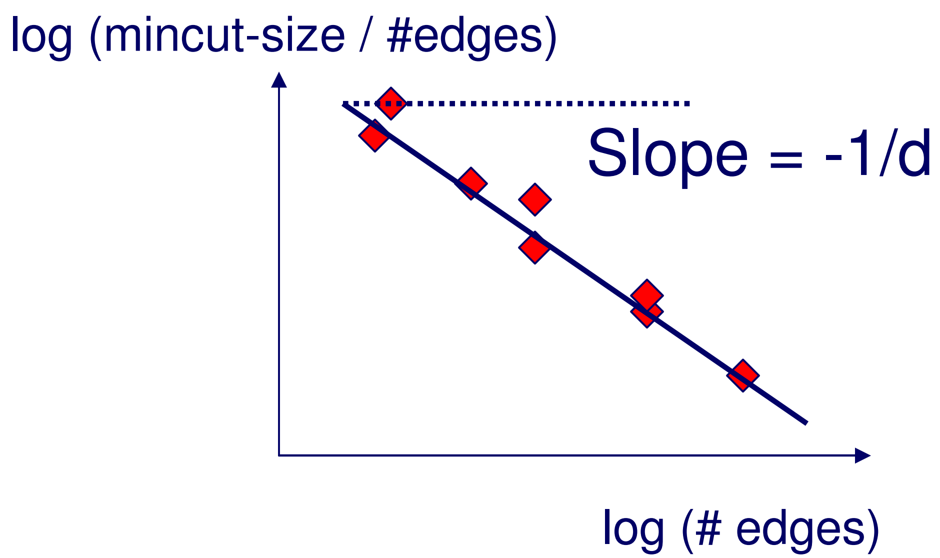


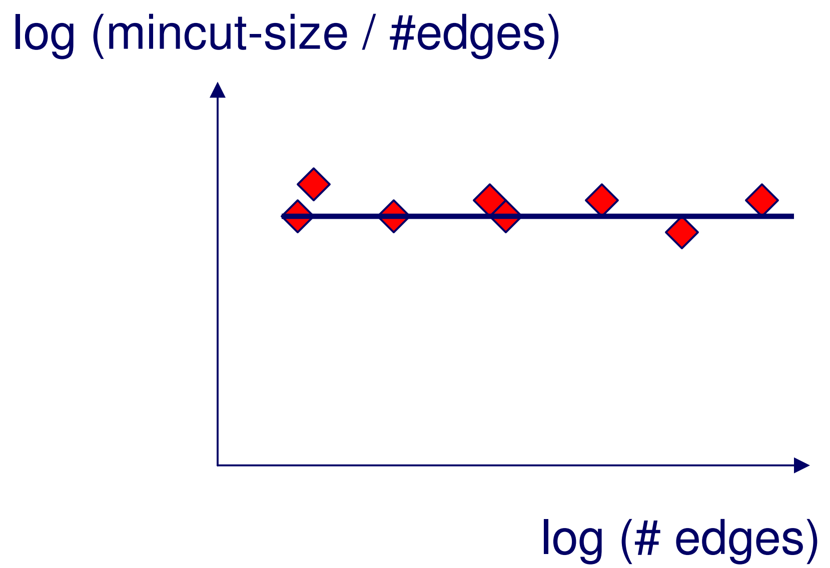
N nodes



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

"Min-cut" plot



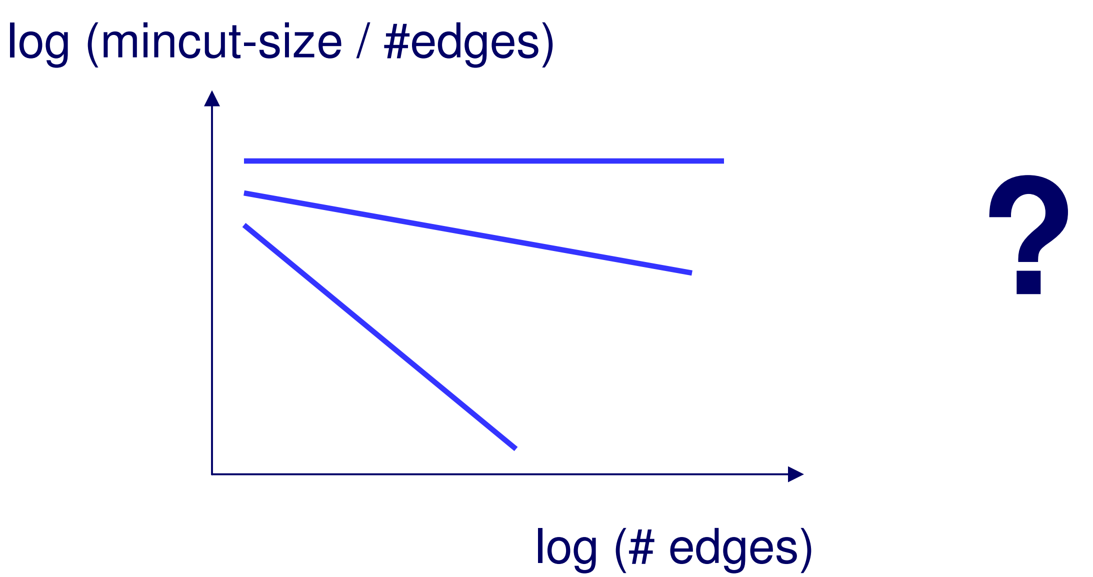


For a d-dimensional grid, the slope is -1/d

For a random graph, the slope is 0

"Min-cut" plot

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



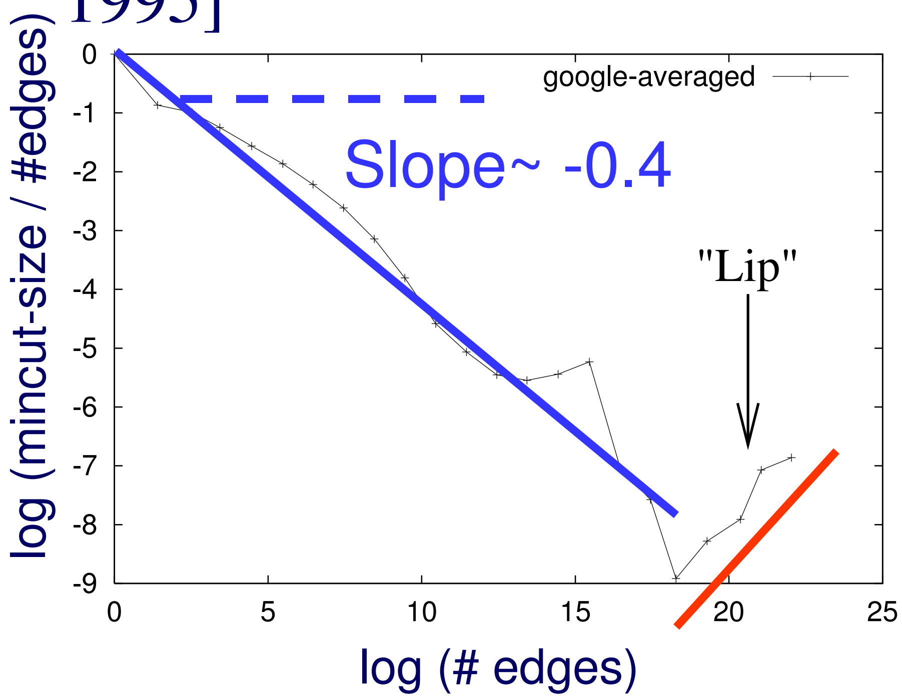
Experiments

- 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

Experiments

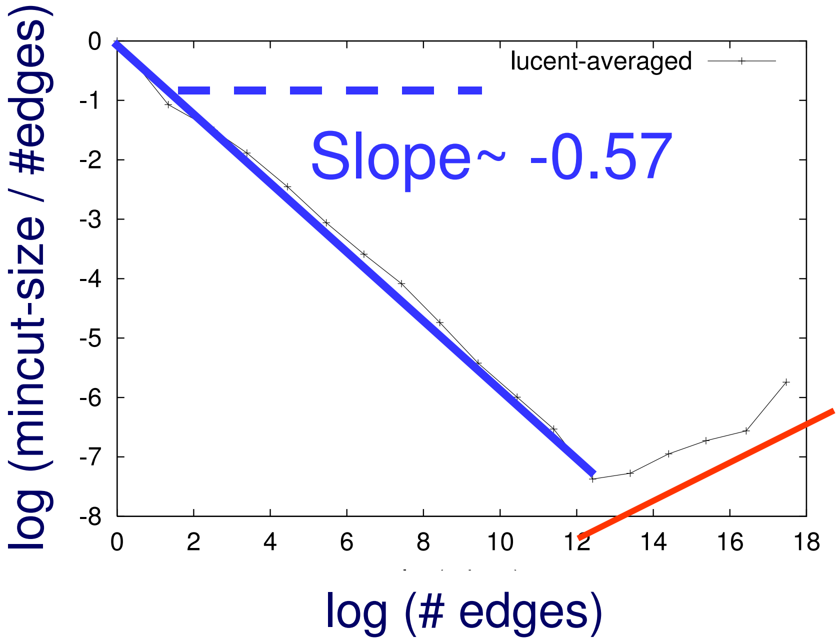
- Used the METIS algorithm [Karypis, Kumar, 1995]



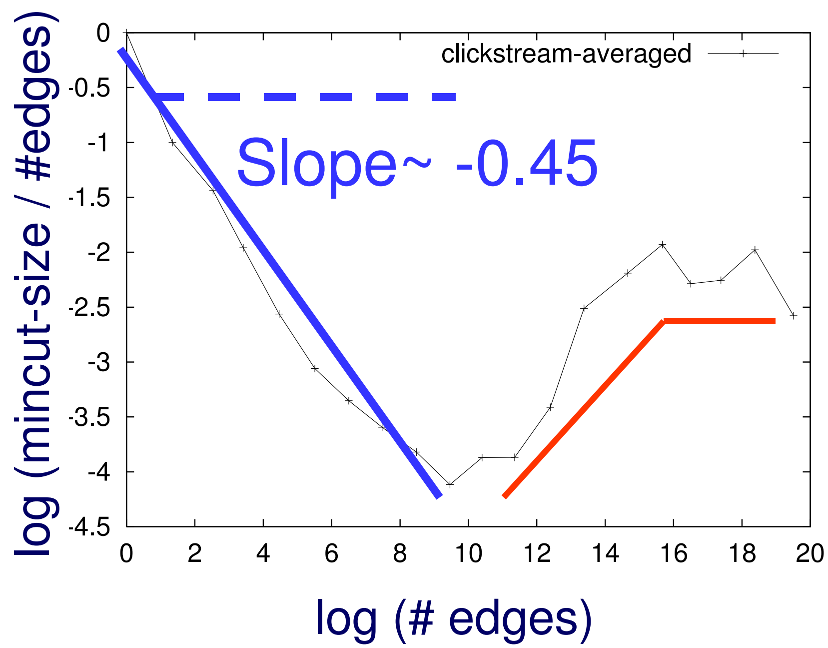
- Google Web graph
- Values along the y-axis are averaged
- We observe a “lip” for large edges
- Slope of -0.4, corresponds to a 2.5-dimensional grid!

Experiments

- Same results for other graphs too…



Lucent Router graph



Clickstream graph

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 are no good cuts**