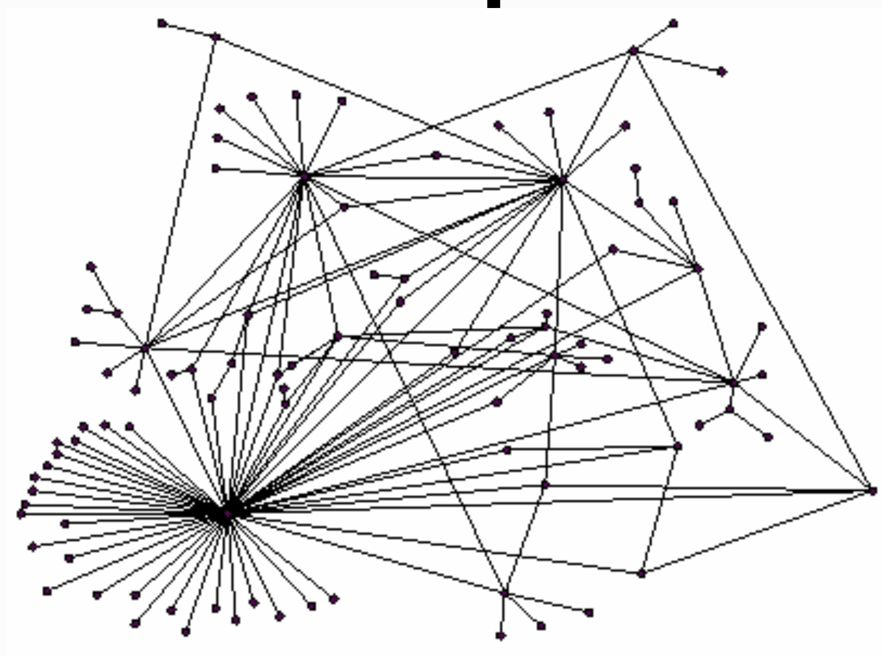
Carnegie Mellon

Part 2:
Tools for Graph Mining



Outline

- Part 1: How do networks form, evolve, collapse?
- Part 2: What tools can we use to study networks?
  - Matrix decomposition
  - Principal Component Analysis
  - Random walks and ranking algorithms
  - Co-clustering and cross-association
  - Self-similarity
  - Entropy plots
- Part 3: Case studies

Carnegie Mellon

Examples of Matrices

- Example/Intuition: Documents and terms
- Find patterns, groups, concepts

	data	info.	brain	lung	...
Paper#1	13	11	22	55	...
Paper#2	5	4	6	7	...
Paper#3	...	...	...	...	...
Paper#4	...	...	...	...	...
...	...	...	...	...	...

SVD - Example

- **A = U Σ V**^{T} - example:

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Example

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

$$
\begin{array}{c}
 \\
\text{CS} \\
 \\
\text{MD} \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{cc} \text{CS-concept} & \text{MD-concept} \end{array} \\
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\end{array}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Example

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

doc-to-concept similarity matrix

$$\begin{array}{cc} & \begin{array}{ccccc}\text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung}\end{array} \\ \begin{array}{c}\text{CS}\\ \\ \\ \\ \text{MD}\\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{cc}\text{CS-concept} & \text{MD-concept}\end{array} \\ \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \end{array} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Example

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

'strength' of CS-concept

$$
\begin{array}{c c}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \\ \\ \text{MD} \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Example

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

term-to-concept similarity matrix

CS-concept

$$
\begin{array}{c|ccccc}
 & \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \\
\hline
\text{CS} & 1 & 1 & 1 & 0 & 0 \\
\text{CS} & 2 & 2 & 2 & 0 & 0 \\
\text{CS} & 1 & 1 & 1 & 0 & 0 \\
\text{CS} & 5 & 5 & 5 & 0 & 0 \\
\text{MD} & 0 & 0 & 0 & 2 & 2 \\
\text{MD} & 0 & 0 & 0 & 3 & 3 \\
\text{MD} & 0 & 0 & 0 & 1 & 1
\end{array}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Example

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

term-to-concept similarity matrix

CS-concept

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Interpretation

‘documents’, ‘terms’ and ‘concepts’:

Q: if **A** is the document-to-term matrix, what is $\mathbf{A}^T \mathbf{A}$?

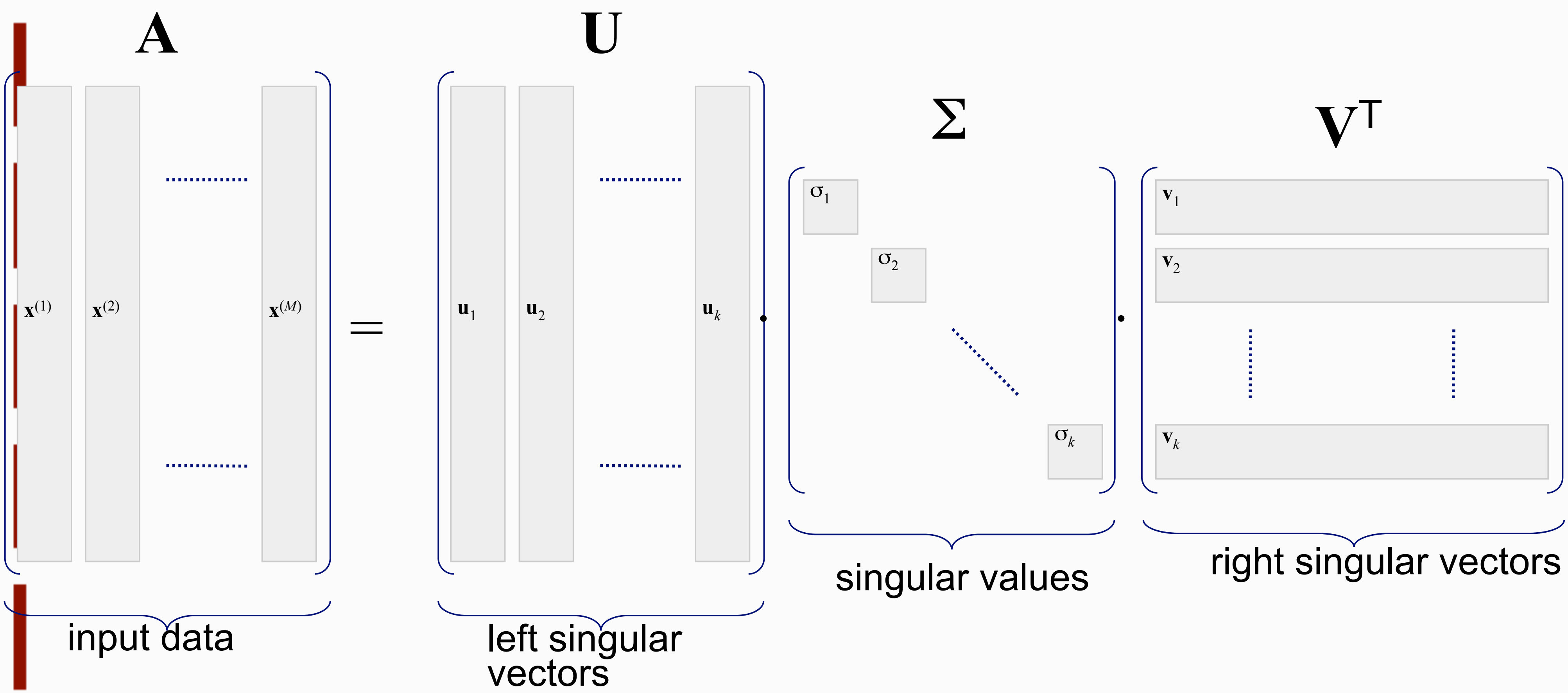
A: term-to-term ([m x m]) similarity matrix

Q: $\mathbf{A} \mathbf{A}^T$?

A: document-to-document ([n x n]) similarity matrix

Singular Value Decomposition (SVD)

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}}$$



Decomposition for 3+ “modes”

- A tensor is a N-D generalization of matrix:

WWW ‘05

WWW ‘06

WWW ‘07

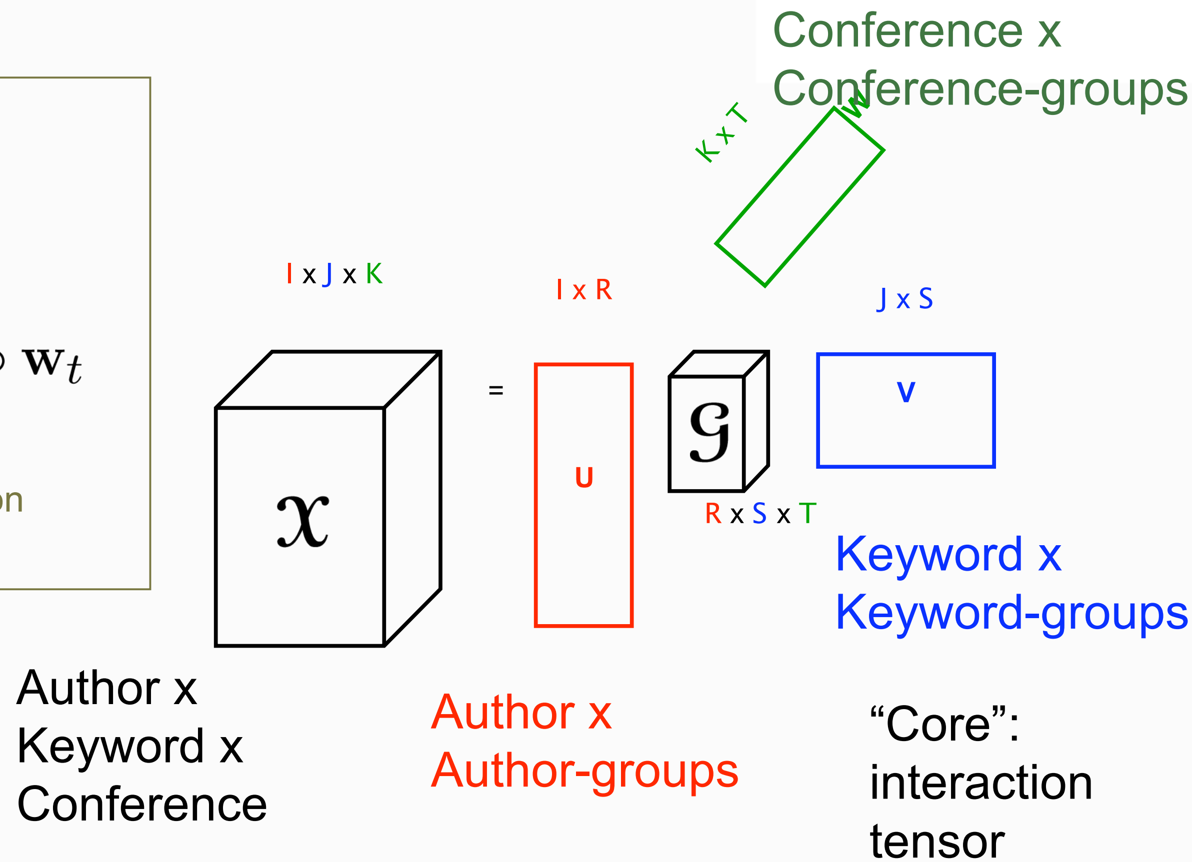
	data	mining	classif.	tree	…
Author #1	13	11	22	55	…
Author #2	5	4	6	7	…
Author #3	…	…	…	…	…
Author #4	…	…	…	…	…
…	…	…	…	…	…

Specially Structured Tensors

- Tucker Tensor

$$\begin{aligned}\mathcal{X} &= \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \\ &= \sum_r \sum_s \sum_t g_{rst}\, \mathbf{u}_r \circ \mathbf{v}_s \circ \mathbf{w}_t \\ &\equiv [\![\mathcal{G}\, ; \mathbf{U}, \mathbf{V}, \mathbf{W}]\!] \end{aligned}$$

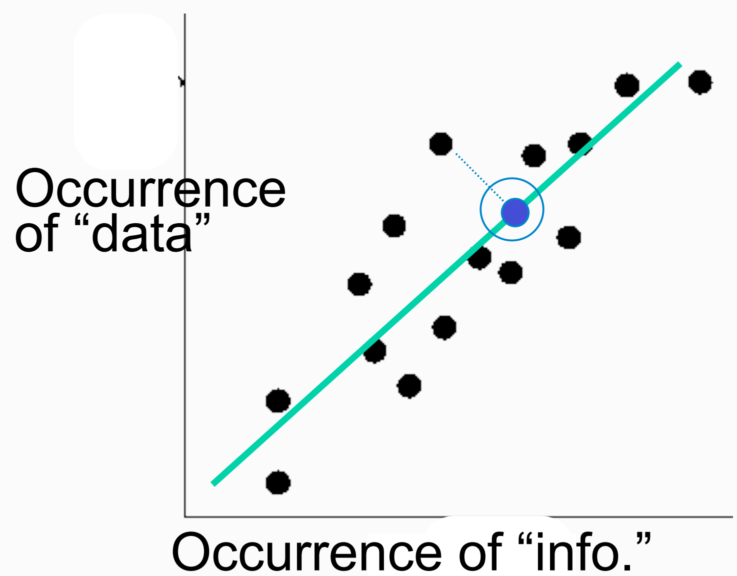
Our Notation



For details, refer to Jimeng Sun and Tamara Kolda’s tutorial:
http://www.cs.cmu.edu/~jimeng/papers/ICMLtutorial.pdf

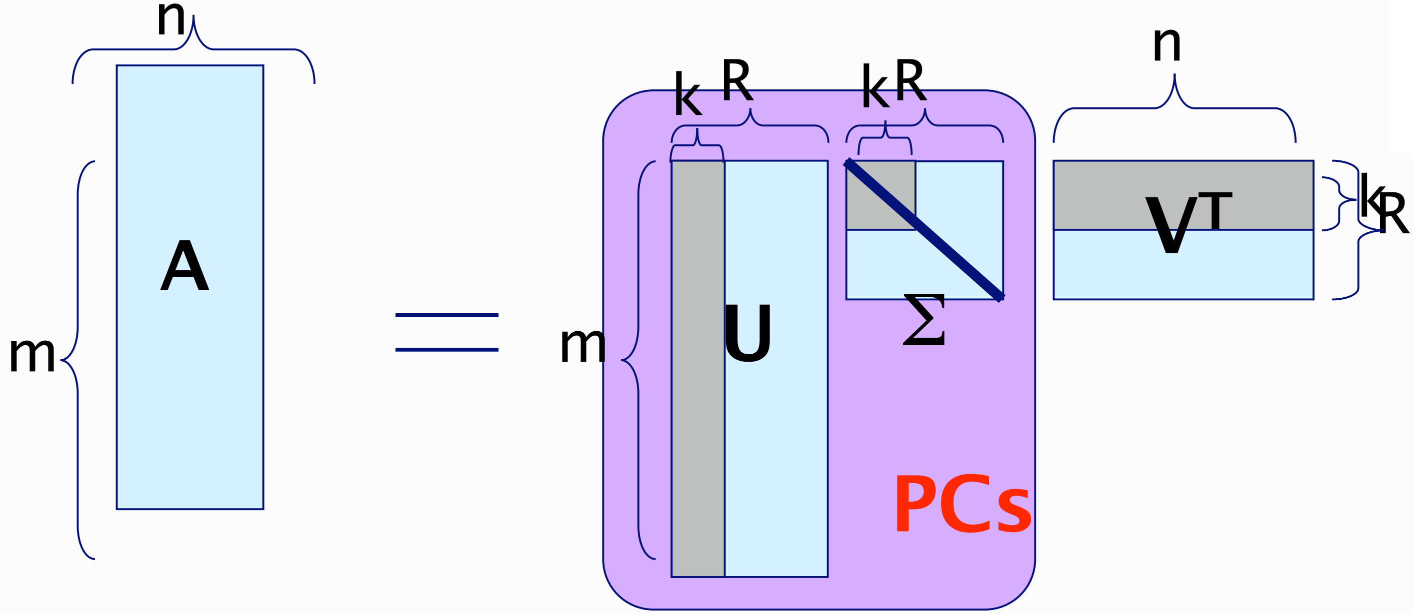
Preliminaries- PCA

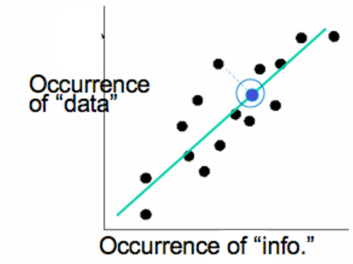
- Principal Component Analysis is a method of dimensionality reduction, based on SVD.



Principal Component Analysis (PCA)

- SVD $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$





  - PCA is an important application of SVD
  - Note that U and V are dense and may have negative entries

Outline for Part 2

- Matrix decomposition
- Principal Component Analysis
- Random walks and ranking algorithms
  - HITS, TOPHITS
  - Pagerank
- Co-clustering and cross-association
- Self-similarity
- Entropy plots

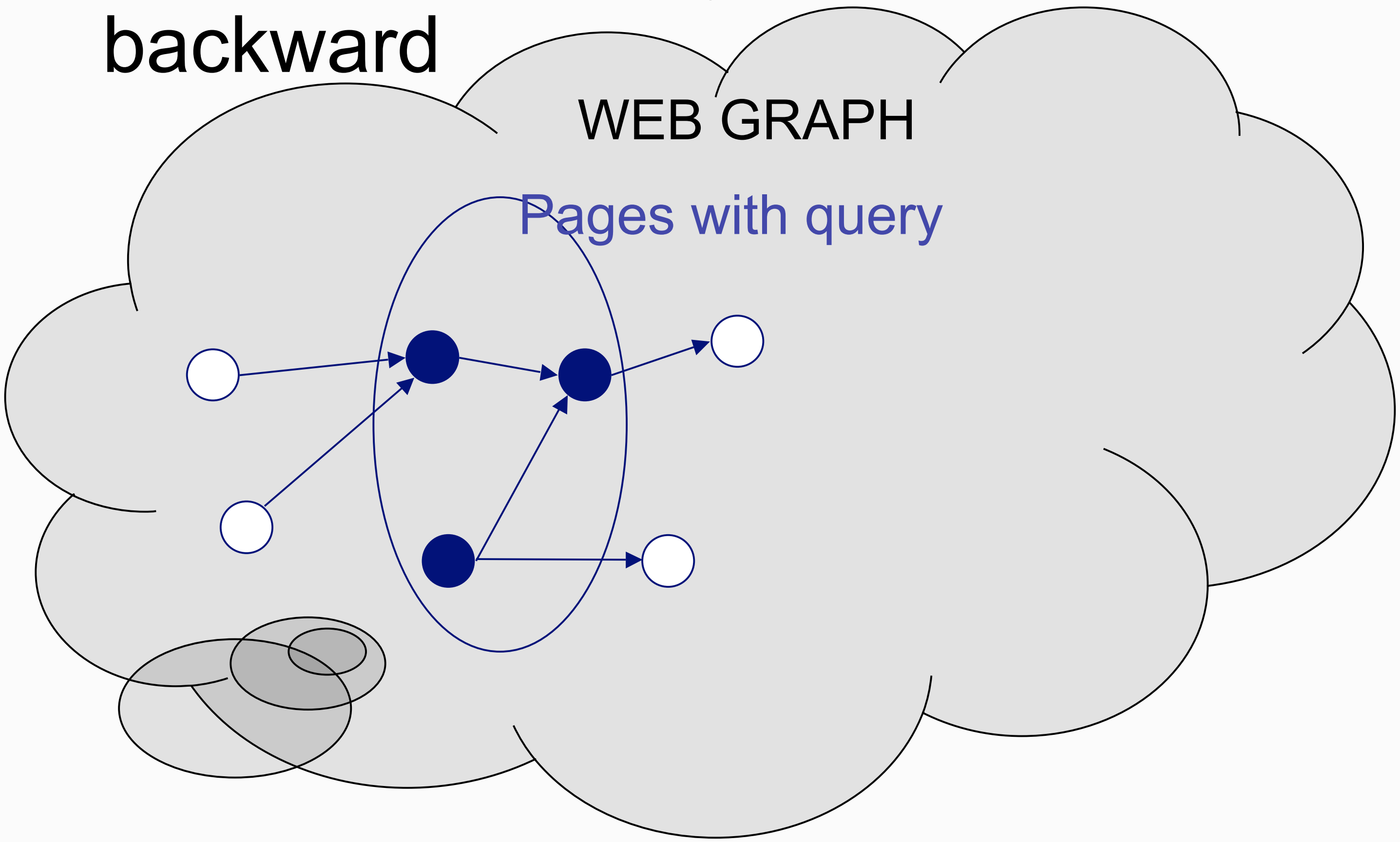
Kleinberg's algorithm HITS

- Problem def: given the web and a query
- Find the most 'authoritative' web pages for this query

Details:
J. Kleinberg. Authoritative sources in a hyperlinked environment. SODA 1998

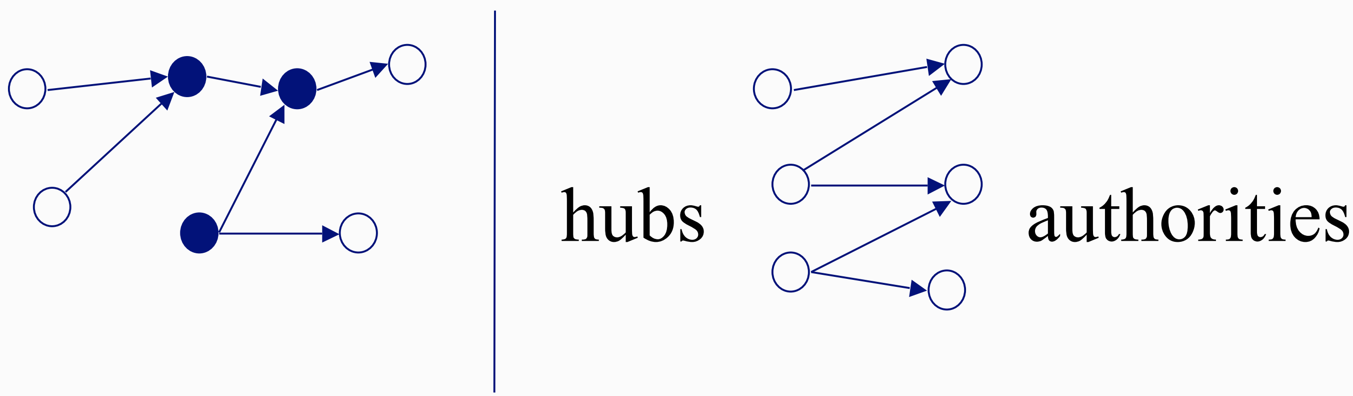
Kleinberg's algorithm HITS

- Step 0: find all pages containing the query terms
- Step 1: expand by one move forward and backward



Kleinberg's algorithm HITS

- On the resulting graph, give high score (= 'authorities') to nodes that many important nodes point to
- Give high importance score ('hubs') to nodes that point to good 'authorities'



Kleinberg's Algorithm: HITS

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: HITS

Details

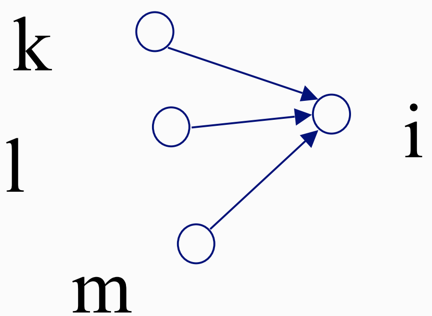
- Let $\mathbf{A}$ be the adjacency matrix:
  - the (i,j) entry is 1 if the edge from i to j exists
- Let $\mathbf{h}$ and $\mathbf{a}$ be [n x 1] vectors with the ‘hubness’ and ‘authoritativiness’ scores.
- Then:

Kleinberg’s algorithm: H

Details

Then:

$$a_i = h_k + h_l + h_m$$



that is

a_i = Sum (h_j) over all j that (j,i) edge exists

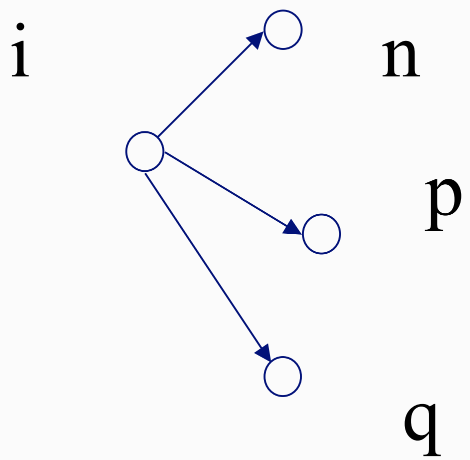
or

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

Kleinberg's algorithm: H

Details

symmetrically, for the 'hubness':



$$h_i = a_n + a_p + a_q$$

that is

h_i = Sum (q_j) over all j that (i,j) edge exists

or

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

Kleinberg's algorithm: HITS

Details

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: HITS

Details

a is a <u>right singular vector</u> of the adjacency matrix **A** (by dfn!), a.k.a the <u>eigenvector</u> of $\mathbf{A}^T\mathbf{A}$

h, then, is the left singular vector.

HITS results

Authority scores for query ‘java’:

0.328 www.gamelan.com

0.251 java.sun.com

0.190 www.digitalfocus.com (“the java developer”)

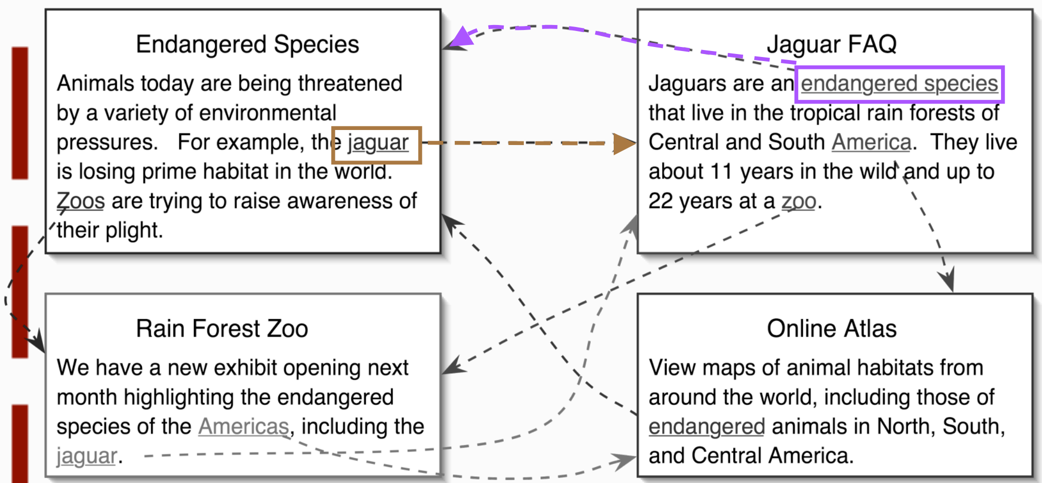
Carnegie Mellon

Outline for Part 2

How to exploit anchortext?

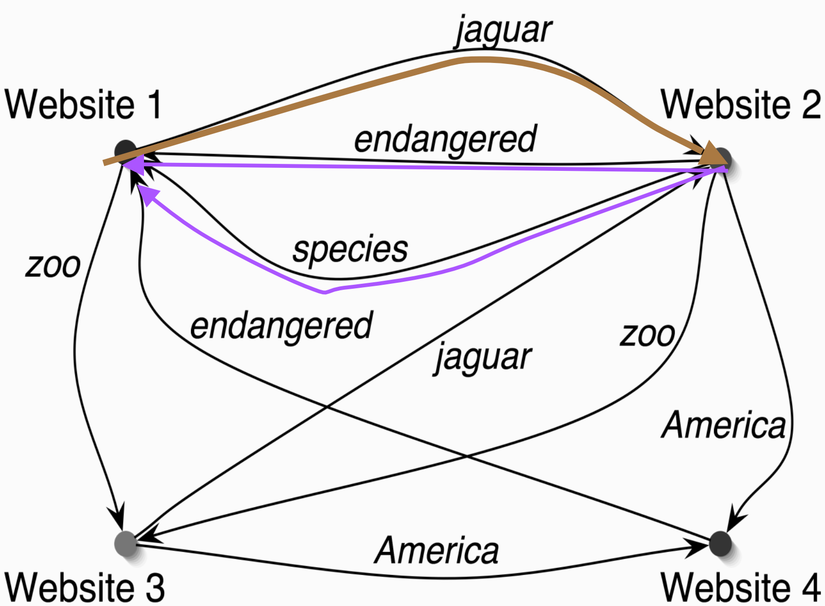
- Matrix decompos
- Principal Component
- Random walks and ranking algorithms
  - HITS, TOPHITS
  - Pagerank
- Co-clustering and cross-association
- Self-similarity
- Entropy plots

Three-Dimensional View of the Web

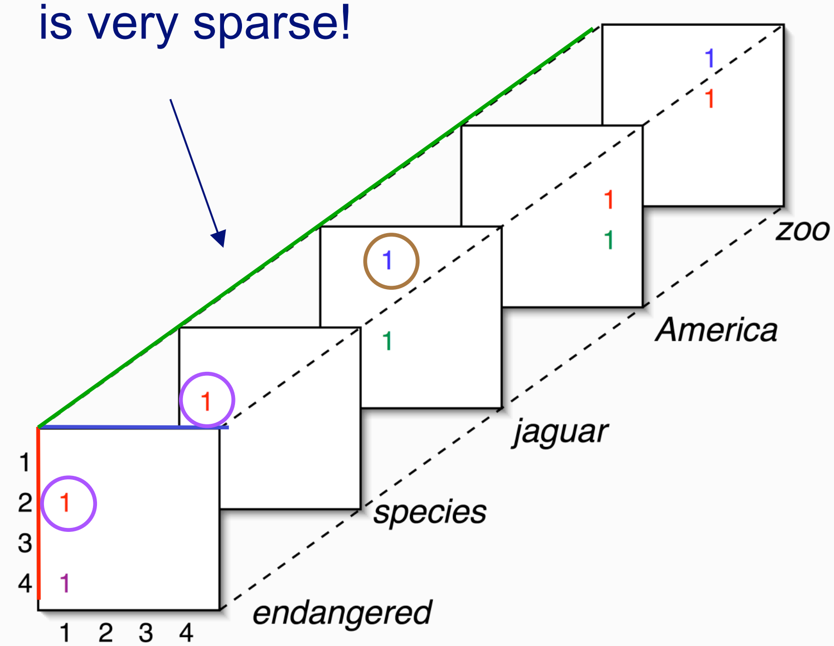


Kolda, Bader, Kenny, ICDM05

$$x_{ijk} = \begin{cases} 1 & \text{if page } i \to \text{page } j \\ & \text{with term } k \\ 0 & \text{otherwise} \end{cases}$$



Observe that this tensor is very sparse!

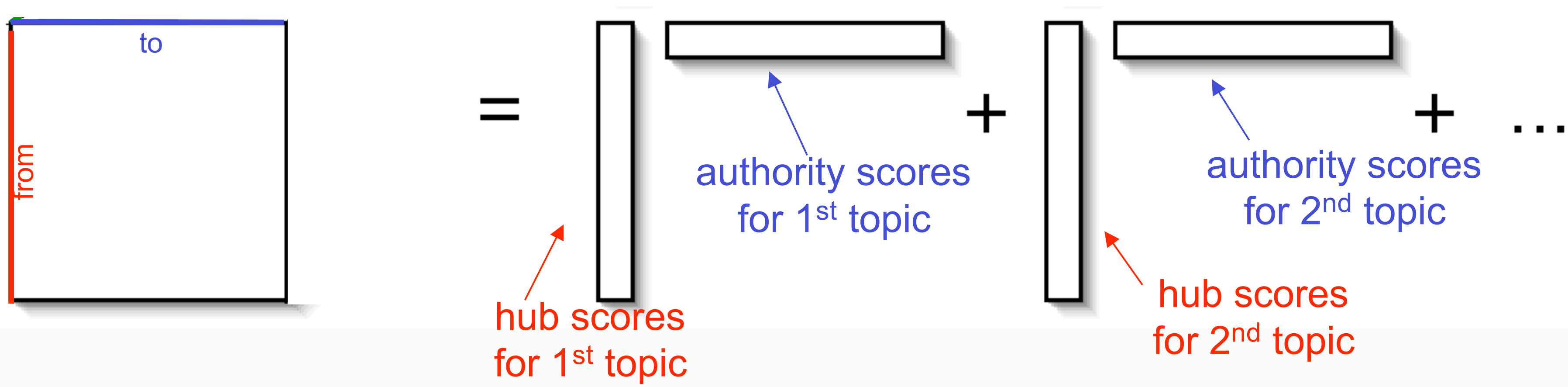


Carnegie Mellon

Topical HITS (TOPHITS)

Main Idea: Extend the idea behind the HITS model to incorporate term (i.e., topical) information.

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \, \mathbf{h}_r \circ \mathbf{a}_r$$

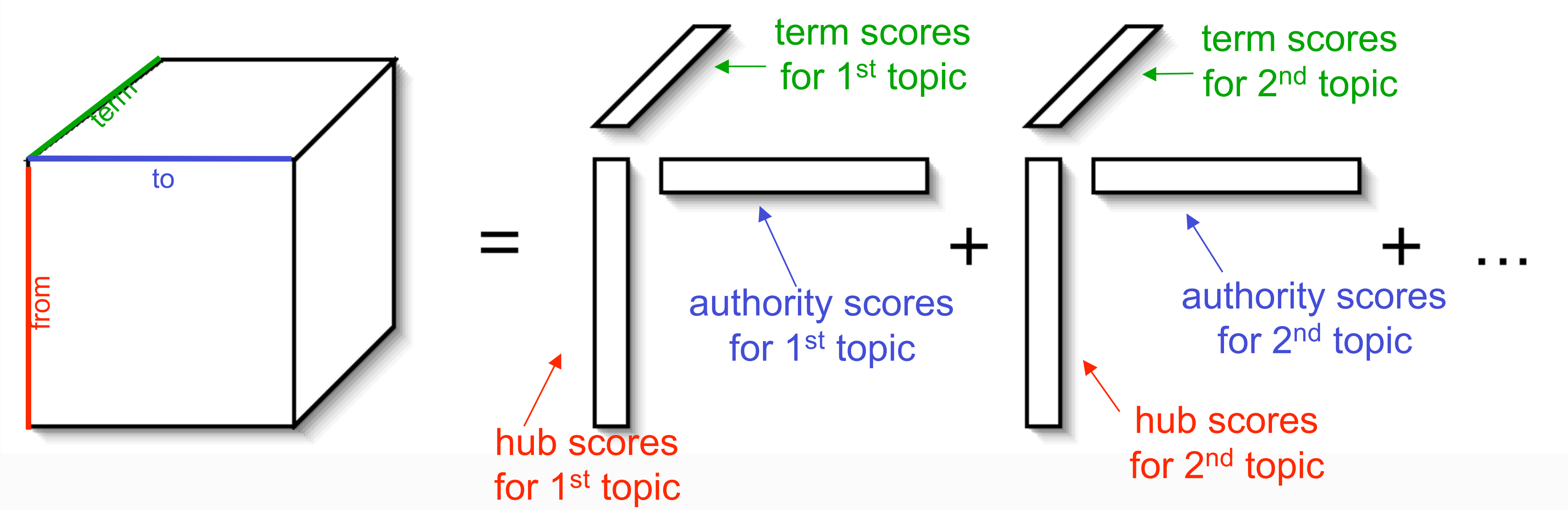


Carnegie Mellon

Topical HITS (TOPHITS)

Main Idea: Extend the idea behind the HITS model to incorporate term (i.e., topical) information.

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \, \mathbf{h}_r \circ \mathbf{a}_r \circ \mathbf{t}_r$$



TOPHITS Terms & Authorities

Carnegie Mellon

1st Principal Factor

.23	JAVA	.86	java.sun.com
.18	SUN		
.17	PLATF		
.16	SOLAR		
.16	DEVEL		
.15	EDITIO		
.15	DOWN		
.14	INFO		
.12	SOFTW		
.12	NO-RE		

2nd Principal Factor

.20	NO-READABLE-TEXT	.99	www.lehigh.edu
.16	FACUL		
.16	SEARC		
.16	NEWS		
.16	LIBRAR		
.16	COMP		
.12	LEHIG		

3rd Principal Factor

.15	NO-READABLE-TEXT	.97	www.ibm.com
.15	IBM	.18	www.alphaworks.ibm.com
.12	SERVI		
.12	WEBS		
.12	WEB		
.11	DEVEL		
.11	LINUX		
.11	RESOU		
.11	TECHN		
.10	DOWN		

4th Principal Factor

.26	INFORMATION	.87	www.pueblo.gsa.gov
.24	FEDERAL	.24	www.irs.gov
.23	CITIZE		
.22	OTHER		
.19	CENTE		
.19	LANGU		
.15	U.S		
.15	PUBLI		
.14	CONSU		
.13	FREE		

6th Principal Factor

.26	PRESIDENT	.87	www.whitehouse.gov
.25	NO-READABLE-TEXT	.18	www.irs.gov
.25	BUSH		
.25	WELC		
.17	WHITE		
.16	U.S		
.15	HOUS		
.13	BUDG		
.13	PRESI		
.11	OFFIC		

12th Principal Factor

.75	OPTIMIZATION	.35	www.palisade.com
.58	SOFTWARE	.35	www.solver.com
.08	DECIS		
.07	NEOS		
.06	TREE		
.05	GUIDE		
.05	SEARC		
.05	ENGIN		
.05	CONT		
.05	ILOG		

13th Principal Factor

.46	ADOBE	.99	www.adobe.com
.45	READER		
.45	ACRO		
.30	FREE		
.30	NO-RE		
.29	HERE		
.29	COPY		
.05	DOWN		

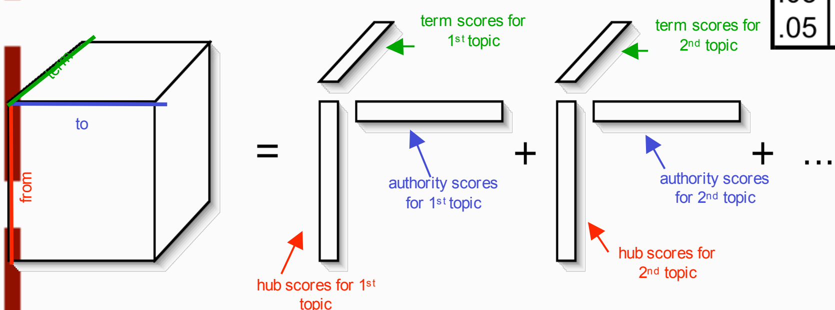
16th Principal Factor

.50	WEATHER	.81	www.weather.gov
.24	OFFICE	.41	www.spc.noaa.gov
.23	CENTER	.30	lwf.ncdc.noaa.gov
.19	NO-RE		
.17	ORGA		
.15	NWS		
.15	SEVER		
.15	FIRE		
.15	POLIC		
.14	CLIMA		

19th Principal Factor

.22	TAX	.73	www.irs.gov
.17	TAXES	.43	travel.state.gov
.15	CHILD	.22	www.ssa.gov
.15	RETIREMENT	.08	www.govbenefits.gov
.14	BENEFITS	.06	www.usdoj.gov
.14	STATE	.03	www.census.gov
.14	INCOME	.03	www.usmint.gov
.13	SERVICE	.02	www.nws.noaa.gov
.13	REVENUE	.02	www.gsa.gov
.12	CREDIT	.01	www.annualcreditreport.com

Tensor PARAFAC

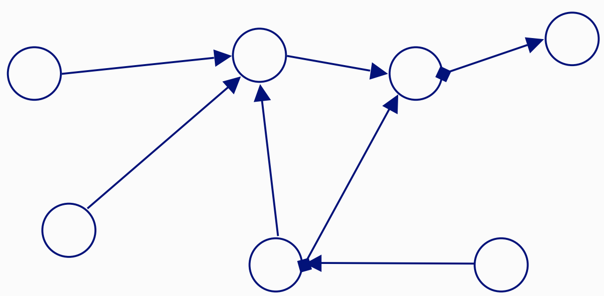


Outline for Part 2

- Matrix decomposition
- Principal Component Analysis
- Random walks and ranking algorithms
  - HITS, TOPHITS
  - Pagerank
- Co-clustering and cross-association
- Self-similarity
- Entropy plots

Pagerank motivation

Given a directed graph, find its most interesting/central node

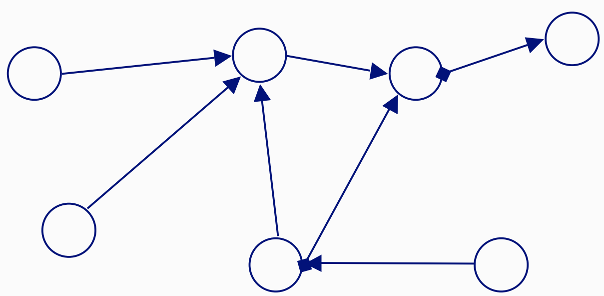


A node is important, if it is connected with important nodes (recursive, but OK!)

Motivating problem – PageRank solution

Given a directed graph, find its most interesting/central node

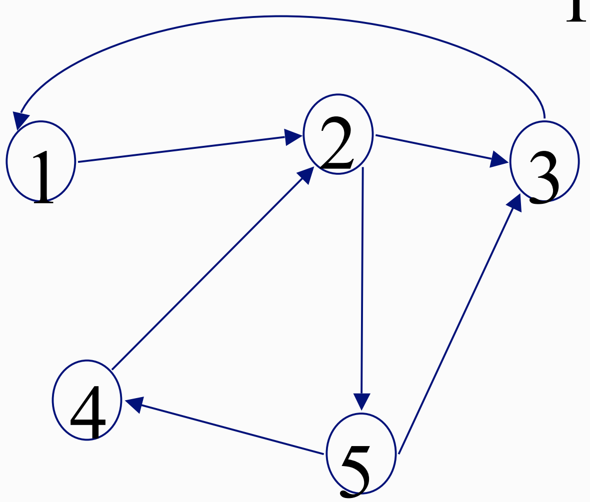
Proposed solution: Random walk; spot most 'popular' node (-> steady state prob. (ssp))



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

(Simplified) PageRank algorithm

- Let **A** be the transition matrix (= adjacency matrix); let **B** be the transpose, column-normalized - then

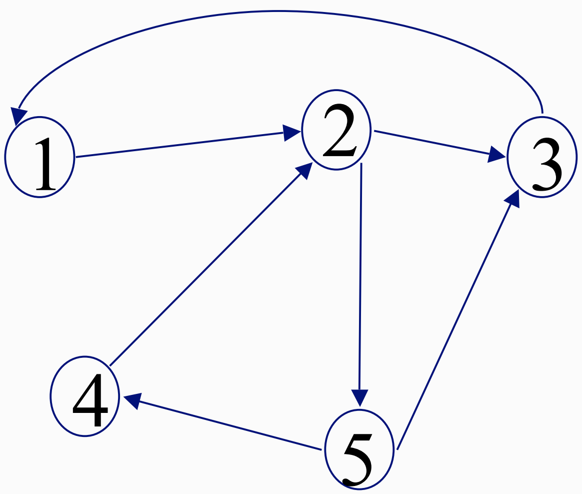


From / To

$$\mathbf{B}\quad \begin{bmatrix} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{bmatrix} \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} = \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}$$

(Simplified) PageRank algorithm

- $\mathbf{B}\ \mathbf{p}_t = \mathbf{p}_{t+1}$



$$\underbrace{\begin{bmatrix} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}}_{\mathbf{p}_t} = \underbrace{\begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}}_{\mathbf{p}_{t+1}}$$

Details

(Simplified) PageRank algorithm

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

(Simplified) PageRank

Details

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

Full version: with occasional random jumps

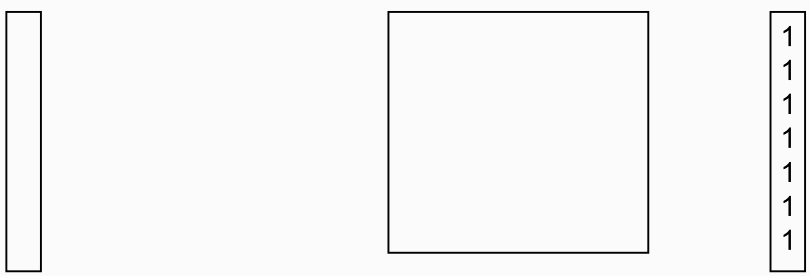
This will make the matrix irreducible

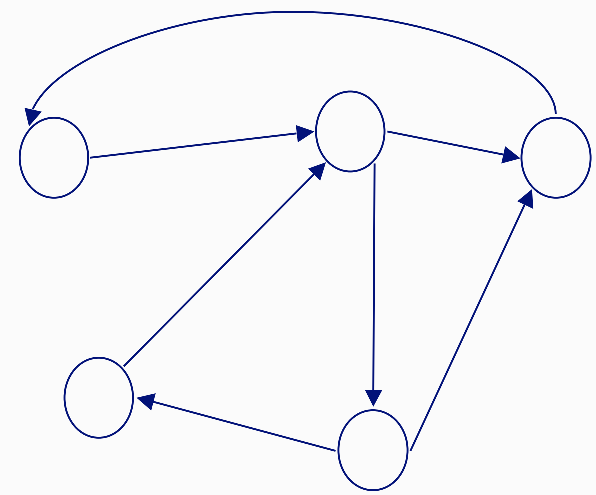
Full Algorithm

- With probability $1\text{-}c$, fly-out to a random node
- Then, we have

$$\mathbf{p} = c\,\mathbf{B}\,\mathbf{p} + (1\text{-}c)/n\ \mathbf{1} \Rightarrow$$

$$\mathbf{p} = (1\text{-}c)/n\ [\mathbf{I} - c\,\mathbf{B}]^{-1}\ \mathbf{1}`$$



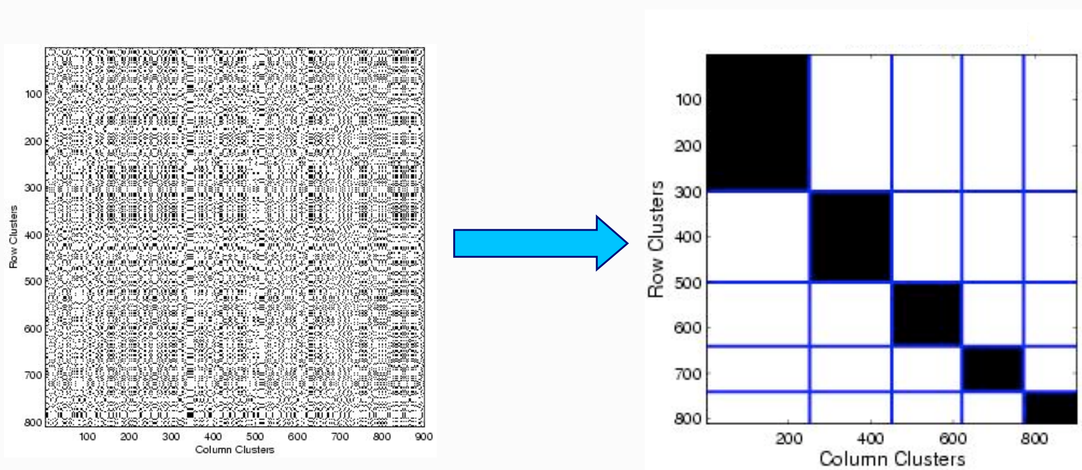


Outline for Part 2

- Matrix decomposition
- Principal Component Analysis
- Random walks and ranking algorithms
- Co-clustering and cross-association
- Self-similarity
- Entropy plots

Co-clustering

- Given data matrix and the number of row and column groups k and l
- Simultaneously
  - Cluster rows of $p(X, Y)$ into k disjoint groups
  - Cluster columns of $p(X, Y)$ into l disjoint groups



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

eg, terms x documents (normalized)

approximation q

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

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 (k x l)

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

doc x doc group

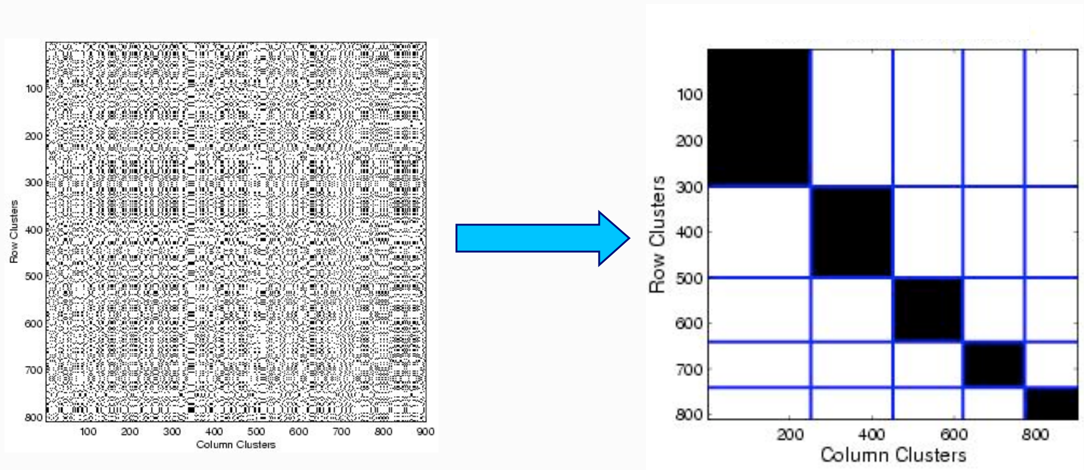
term x term-group

Co-clustering

- Details: Dhillon et. al. Information-Theoretic Co-clustering, KDD 2003.
- Uses KL divergence, instead of L2
- The middle matrix is **not** diagonal

- Must specify k and l (number of row, column groups).

Cross-association

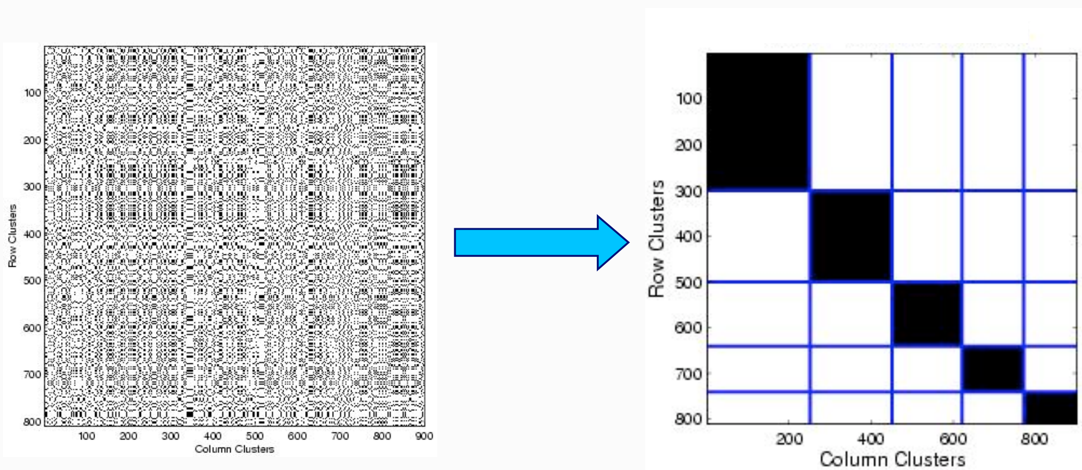


Desiderata:

- ✓ Simultaneously discover row and column groups
- ✓ Fully Automatic: No “magic numbers”
- ✓ Scalable to large matrices

Cross-association

- Main idea:
- Automatically decide *k* and *l* and reorder rows to reach best compression.
- Details: Chakrabarti et. al. Fully automatic cross-associations. KDD04.



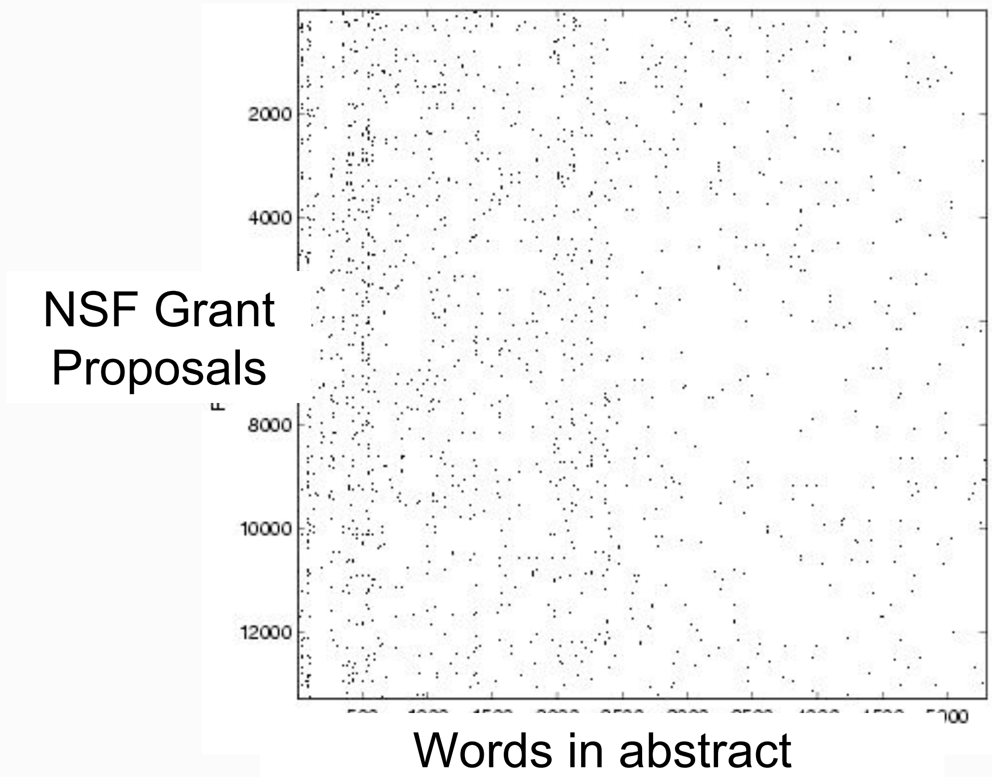
Cross-association

Details

- Start with k=1, l=1
- Shuffle rows and columns
- Split:
  - Pick row group *g* with maximum entropy
  - Pick rows from *g* that maximize the entropy, make new group
  - (Repeat for columns)
- Repeat until total description of matrix (each group description + describing groups) is minimized

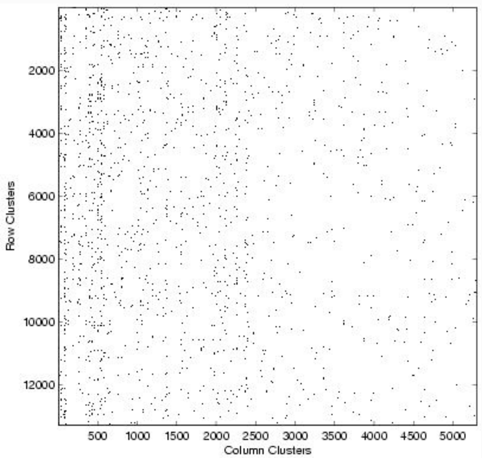
Cross-association Results

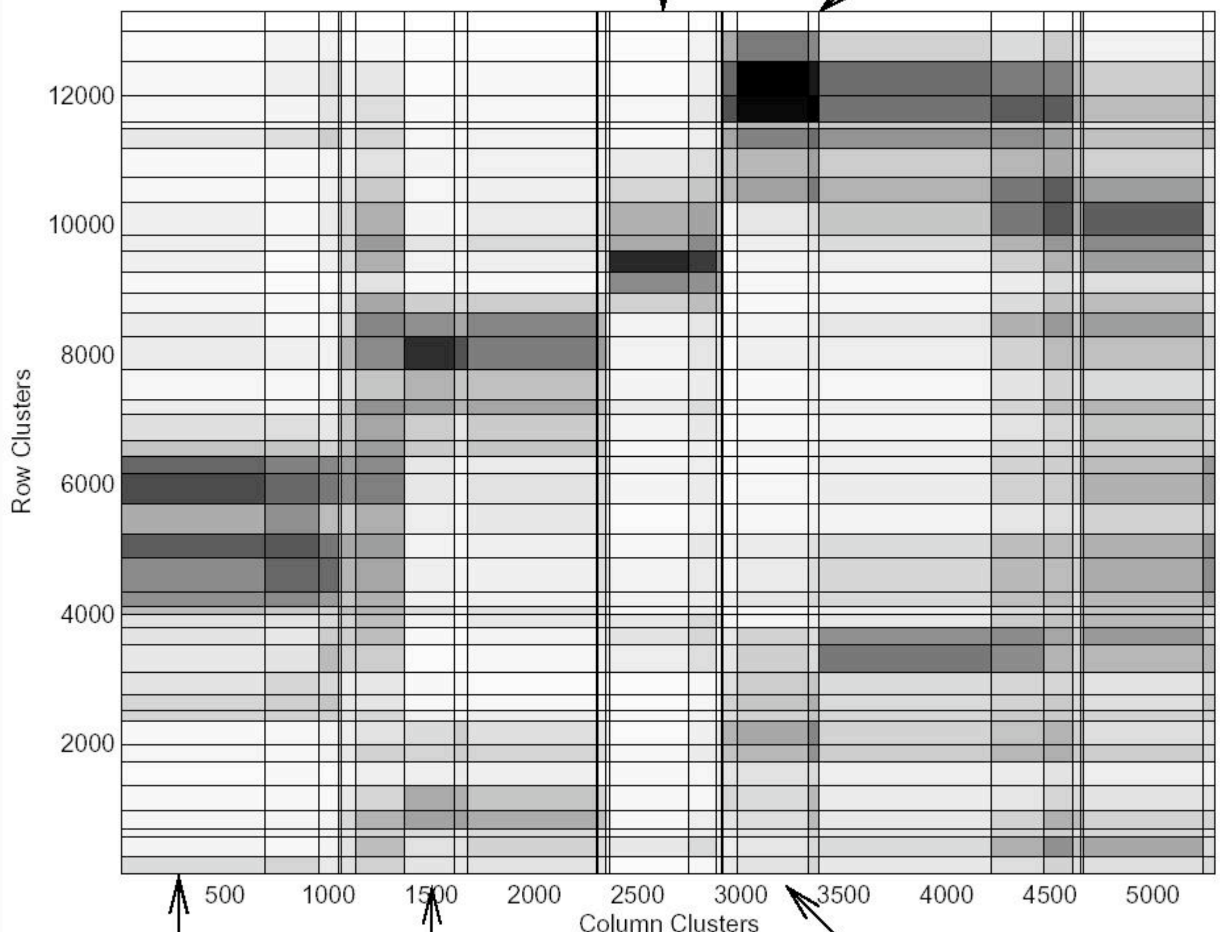
- NSF Grant proposals
- 13,297 documents
- 5,298 words
- 805,063 entries



Cross-association Results

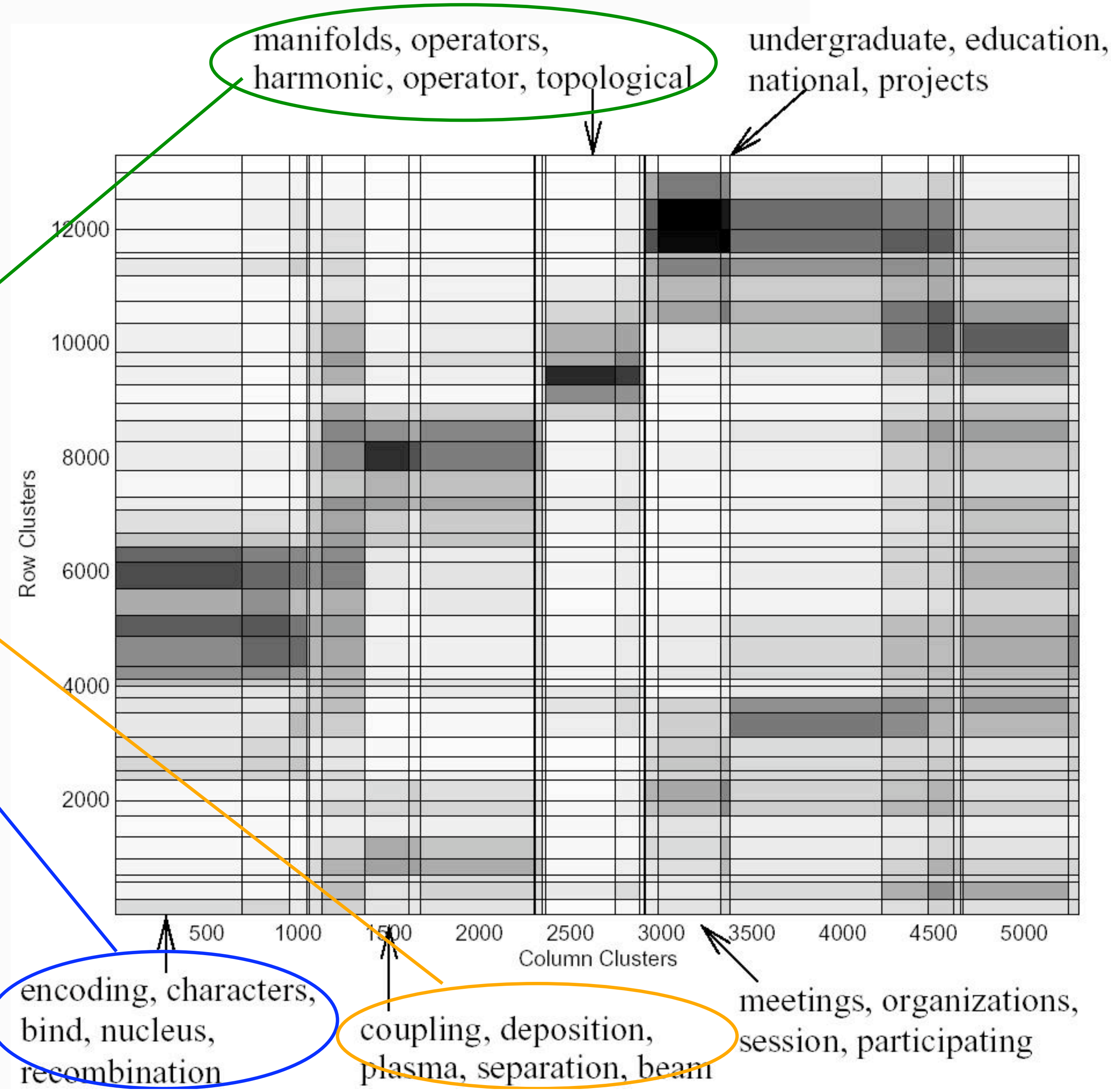
- NSF grants-words
- Found groups: k=41, l=28





Cross-association Results

- Cross-associations refer to topics:
- Mathematics
- Physics
- Genetics



Algorithm

Code for cross-associations (matlab):

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

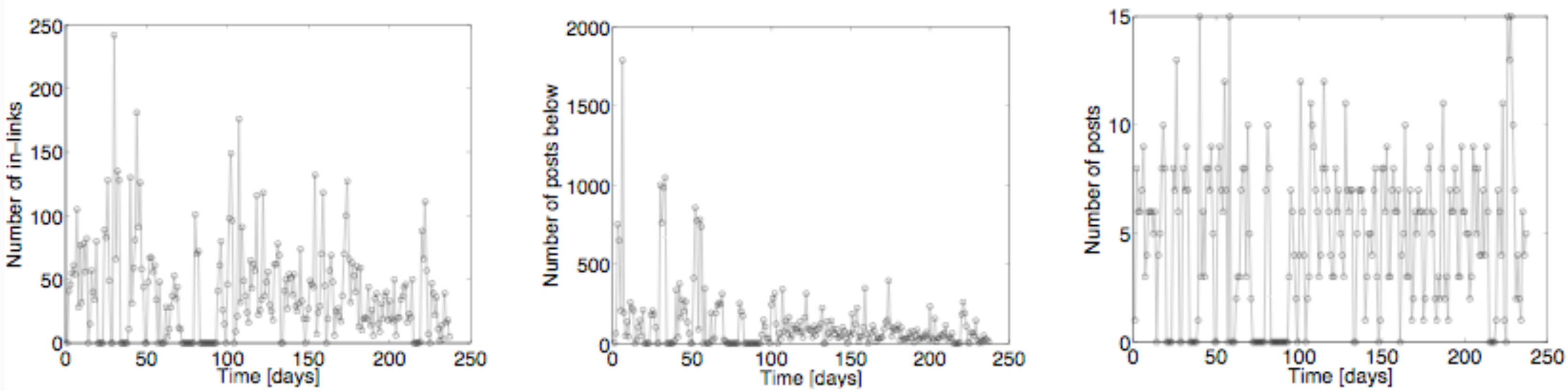
Variations and extensions:

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

Outline for Part 2

- Matrix decomposition
- Principal Component Analysis
- Random walks and ranking algorithms
- Co-clustering and cross-association
- Self-similarity
- Entropy plots

- How to identify less obvious patterns -- for instance, in time series data?



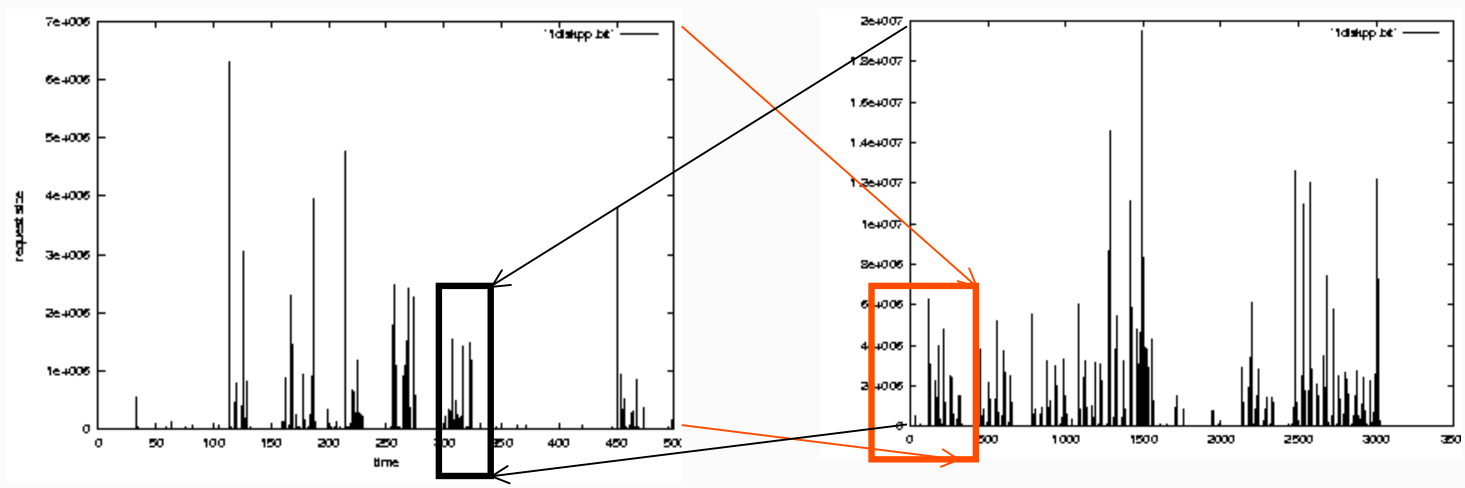
(a) in-links (b) conv. mass (c) num. posts

Carnegie Mellon

Self-similarity

- Self-similarity helps describe patterns.
- Example: disk traces

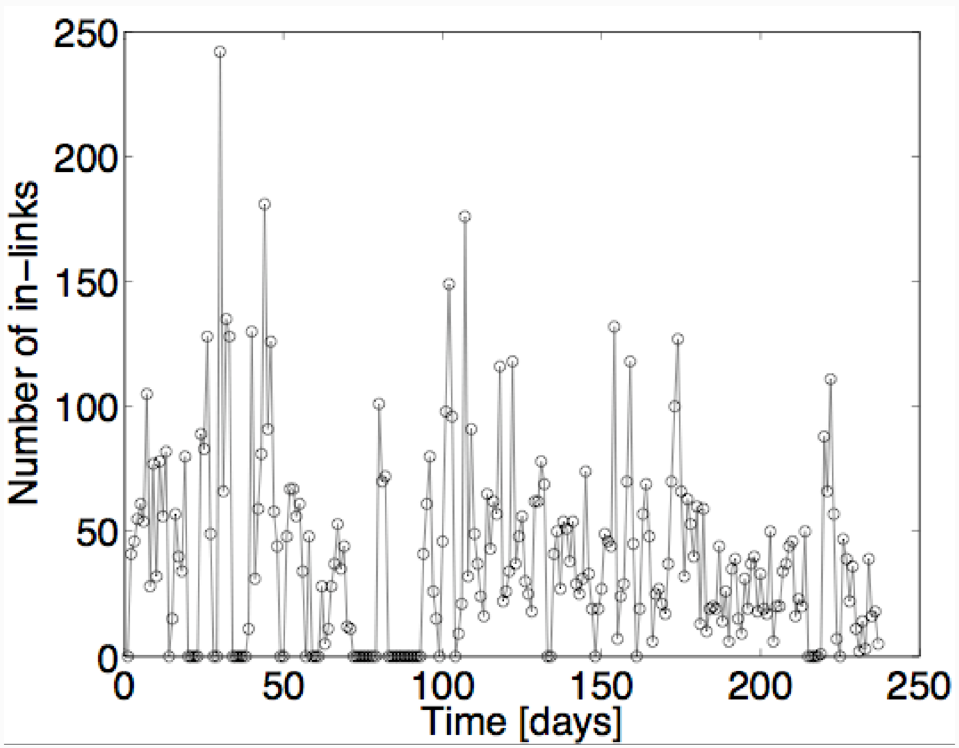
#bytes



time

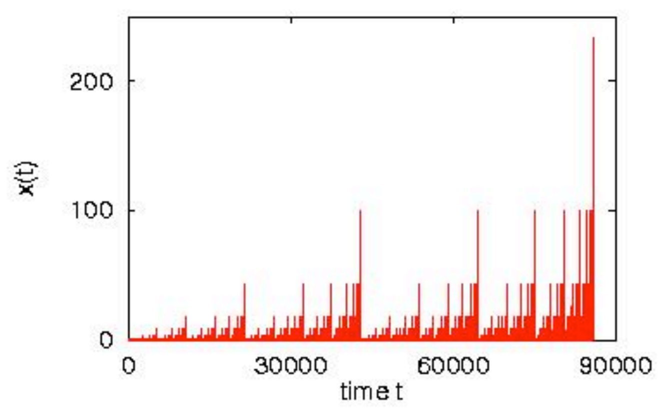
Self-similarity

- Example: blog link traffic
- How can we generate self-similar sequences?



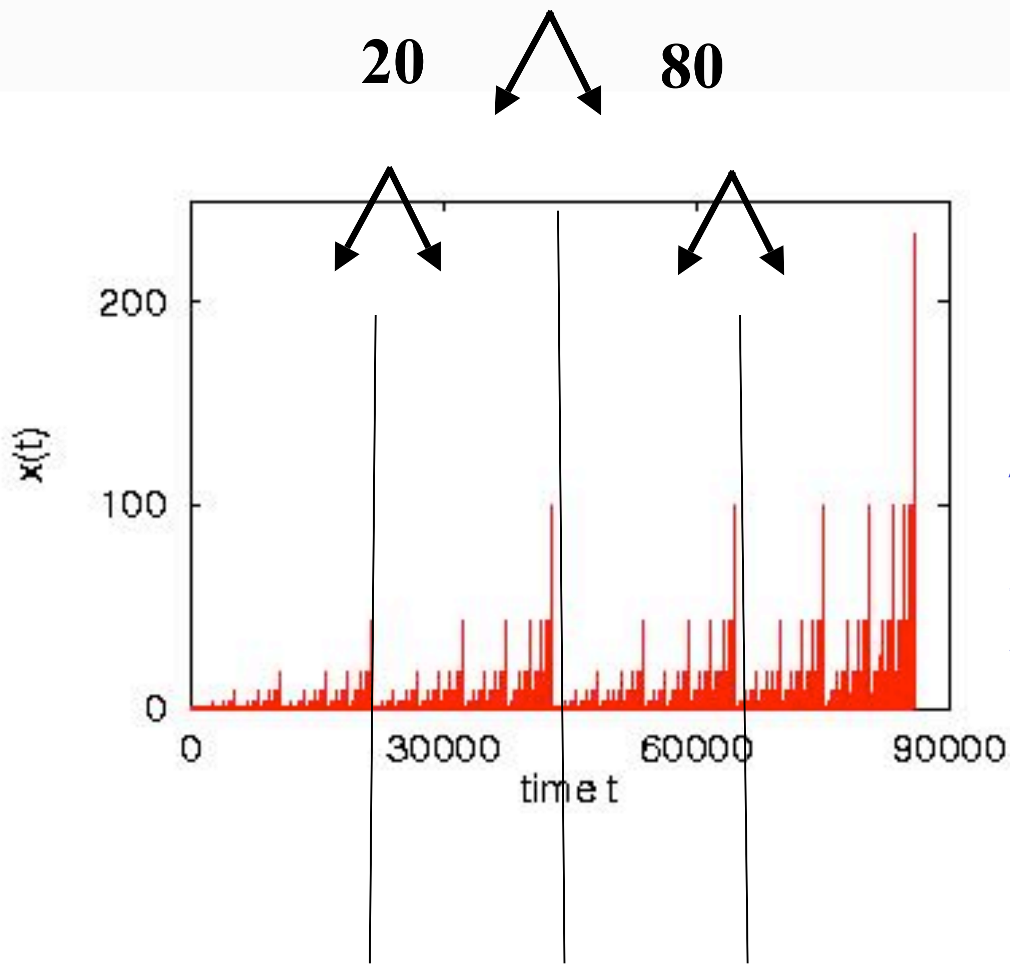
Self-similarity

- The *80-20 law* describes self-similarity.
- For any sequence, we divide it into two equal-length subsequences. 80% of traffic is in one, 20% in the other.
  - Repeat recursively.



Self-similarity

- The *bias factor* for the 80-20 law is *b*=0.8.
- For Poisson arrivals (uniform), bias factor is 0.5.



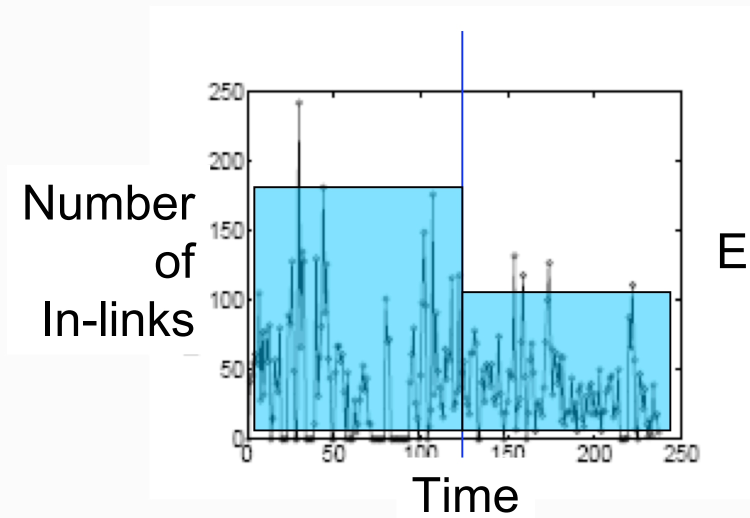
Q: How do we estimate *b*?

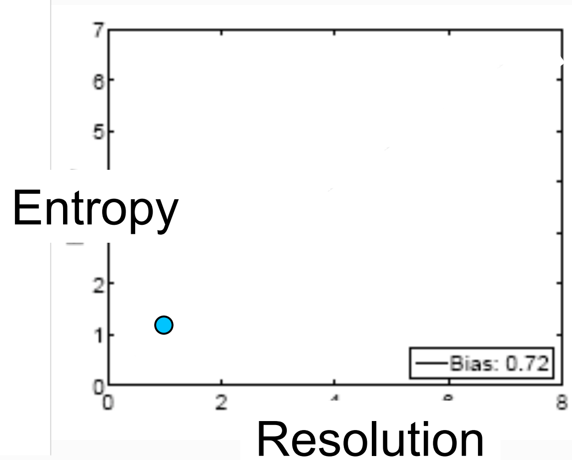
A: Many ways (Hurst exponent, variance plot). We use entropy plots.

Entropy plots

Details

- An entropy plot plots entropy vs. resolution.
- From time series data, begin with resolution R= T/2.
- Record entropy H_R

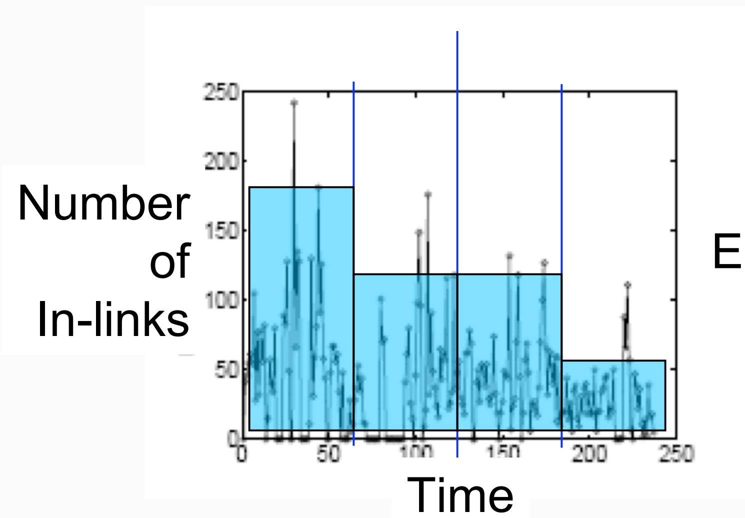


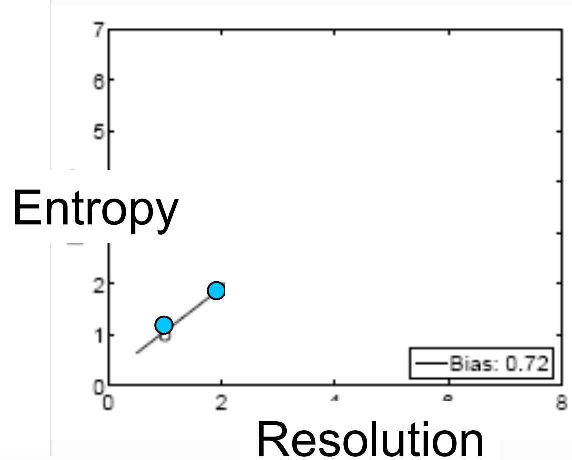


Entropy plots

Details

- An entropy plot plots entropy vs. resolution.
- From time series data, begin with resolution R= T/2.
- Record entropy H_R
- Recursively take finer resolutions.



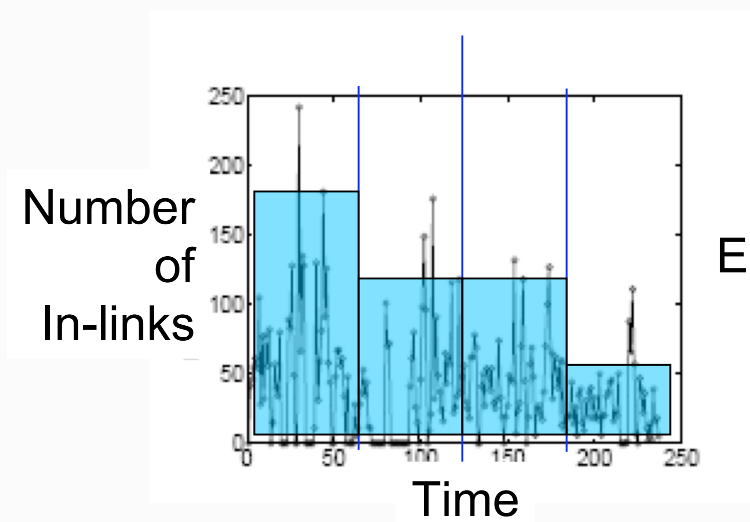


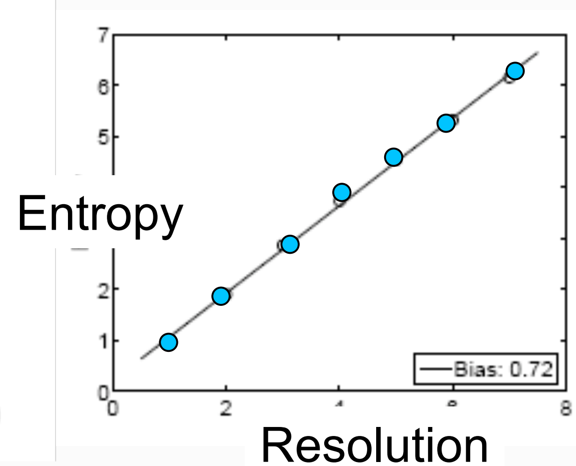
Carnegie Mellon

Entropy plots

Details

- An entropy plot plots entropy vs. resolution.
- From time series data, begin with resolution r= T/2.
- Record entropy H_R
- Recursively take finer resolutions.





Carnegie Mellon

Definitions

Details

- *Entropy* measures the non-uniformity of histogram at a given resolution.
- We define entropy of our sequence at given *R* :

$$H_p = -\sum_{t=1}^{2^R} p(t) log_2 p(t)$$

where *p(t)* is percentage of posts from a blog on interval t, *R* is resolution and 2^R is number of intervals.

b-model

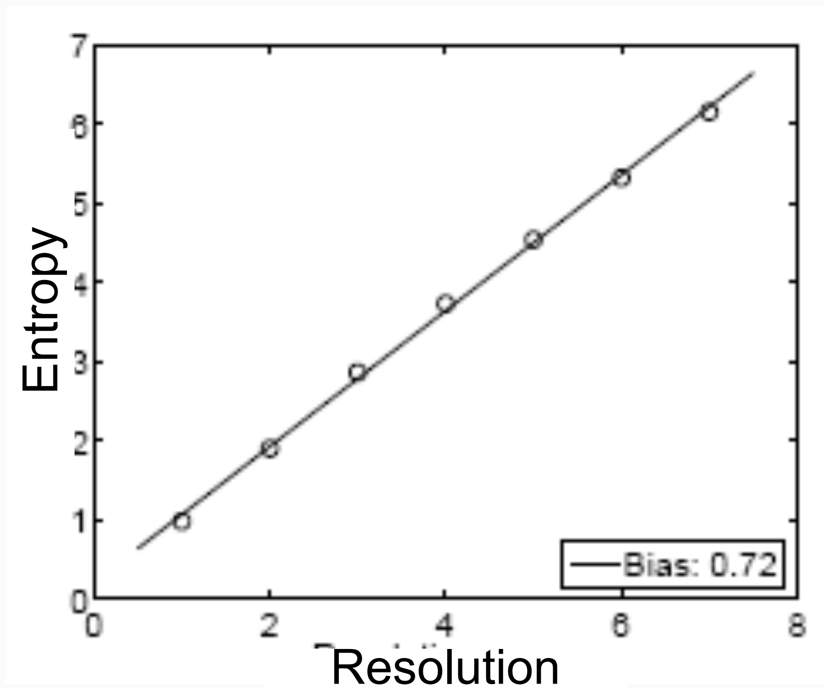
Details

- For a b-model (and self similar cases), entropy plot is linear. The slope *s* will tell us the bias factor.
- Lemma: For traffic generated by a b-model, the bias factor *b* obeys the equation:

$$s = - b \log_2 b - (1-b) \log_2 (1-b)$$

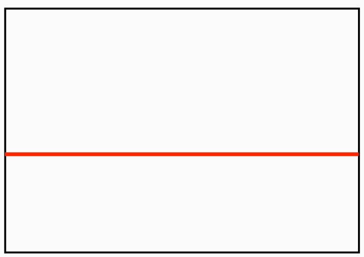
Entropy Plots

- Self-similarity → Linear plot



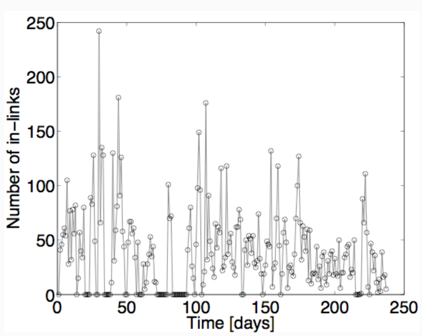
Entropy Plots

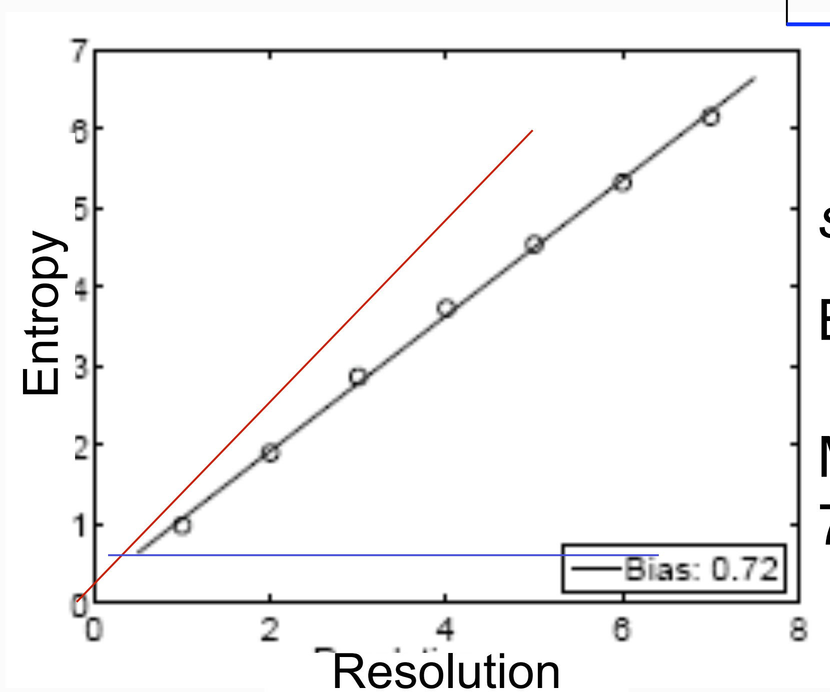
- Self-similarity → Linear plot
- Uniform: slope *s=1. bias=.5*   Point mass*: s=0. bias=1*



time

time





s= 0.85

By Lemma 1, *b*= 0.72

Most blog traffic follows 70-30 law.

Software

- Tensor Toolbox: Matlab add-in for tensors
  - http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/
- NetworkX- Python package to work with graphs easily (graph properties)
  - https://networkx.lanl.gov/
- Proximity: relational knowledge discovery
  - http://kdl.cs.umass.edu/proximity/index.html

Bibliography: Part 2

- Matrices and Tensors
  - Demmel, Applied Numerical Linear Algebra
  - Joliffe, *Principal Component Analysis* (2nd ed) Springer, 2002.
  - Gilbert Strang, *Linear Algebra and its Applications* (4th ed) Brooks & Cole, 2005.
  - Faloutsos, C.; Kolda, T. G. & Sun, J. (2007), Mining large graphs and streams using matrix and tensor tools., *in*.,'SIGMOD Conference', ACM, , pp. 1174.

- Ranking
  - Kleinberg, J. M. (1999), 'Authoritative sources in a hyperlinked environment', *Journal of the ACM* **46**(5), 604--632.
  - Page, L.; Brin, S.; Motwani, R. & Winograd, T. (1998),'The PageRank Citation Ranking: Bringing Order to the Web', Technical report, Stanford Digital Library Technologies Project.
  - T. Kolda and B. Bader. The TOPHITS model for higher-order web link analysis. In: Workshop on Link Analysis, Counterterrorism and Security, 2006.

- Cross-association and co-clustering

  - Chakrabarti, D. (2004), AutoPart: Parameter-Free Graph Partitioning and Outlier Detection., *in* Jean-François Boulicaut; Floriana Esposito; Fosca Giannotti & Dino Pedreschi, ed.,'PKDD', Springer, , pp. 112-124.
  - Dhillon, I.; Mallela, S. & Modha, D. (2003), 'Information-theoretic co-clustering', *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 89--98.

- Self-similarity

  - Crovella, M. & Bestavros, A. (1996), 'Self-Similarity in World Wide Web Traffic, Evidence and Possible Causes', *Sigmetrics*, 160-169.
  - Schroeder, M. (1991), *Fractals, Chaos, Power Laws: Minutes From an Infinite Paradise*, W. H. Freeman.
  - Shannon, C. E. & Weaver, W. (1963), *A Mathematical Theory of Communication*, University of Illinois Press, Champaign, IL, USA.
  - Wang, M.; Madhyastha, T.; Chang, N. H.; Papadimitriou, S. & Faloutsos, C. (2002), 'Data Mining Meets Performance Evaluation: Fast Algorithms for Modeling Bursty Traffic', *ICDE*.

- Stretch break!