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

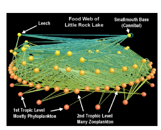
Christos Faloutsos
Gary Miller
Charalampos (Babis) Tsourakakis
CMU

Outline

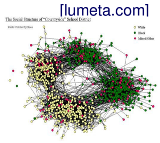
- Introduction – Motivation
- Task 1: Node importance
- Task 2: Community detection
- Task 3: Recommendations
- Task 4: Connection sub-graphs
- Task 5: Mining graphs over time
- Task 6: Virus/influence propagation
- Task 7: Spectral graph theory
- Task 8: Tera/peta graph mining: hadoop
- Observations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

Graphs - why should we care?

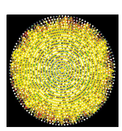
Internet Map
[lumeta.com]



Food Web
[Martinez '91]



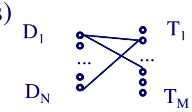
Friendship Network
[Moody '01]



Protein Interactions
[genomebiology.com]

Graphs - why should we care?

- IR: bi-partite graphs (doc-terms)



- web: hyper-text graph
- ... and more:

Graphs - why should we care?

- network of companies & board-of-directors members
- 'viral' marketing
- web-log ('blog') news propagation
- computer network security: email/IP traffic and anomaly detection
- ....
- **Any M:N relationship -> Graph**
- **Any subject-verb-object construct: -> Graph**

Graphs and matrices

- Closely related
- Powerful tools from matrix algebra, for graph mining
- (and, conversely, great problems for CS/Math to solve)

Examples of Matrices:

Graph - social network

	John	Peter	Mary	Nick	...
John	0	11	22	55	...
Peter	5	0	6	7	...
Mary	...	...	...	...	...
Nick	...	...	...	...	...
...	...	...	...	...	...

Examples of Matrices:

Market basket

- **market basket** as in Association Rules

	milk	bread	choc.	wine	...
John	13	11	22	55	...
Peter	5	4	6	7	...
Mary	...	...	...	...	...
Nick	...	...	...	...	...
...	...	...	...	...	...

Examples of Matrices:
Documents and terms

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

Examples of Matrices:
Authors and terms

	data	mining	classif.	tree	...
John	13	11	22	55	...
Peter	5	4	6	7	...
Mary	...	...	...	...	...
Nick	...	...	...	...	...
...	...	...	...	...	...

Examples without networks

- (or, at least, no straightforward network interpretation)

Examples of Matrices:
cloud of n-d points

	chol#	blood#	age	..	...
John	13	11	22	55	...
Peter	5	4	6	7	...
Mary	...	...	...	...	...
Nick	...	...	...	...	...
...	...	...	...	...	...

Examples of Matrices:
sensor-ids and time-ticks

	temp1	temp2	humid.	pressure	...
t1	13	11	22	55	...
t2	5	4	6	7	...
t3	...	...	...	...	...
t4	...	...	...	...	...
...	...	...	...	...	...

Outline

- Introduction – Motivation
- Task 1: Node importance
- Task 2: Community detection
- …
- Conclusions

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

Task 1: Node importance

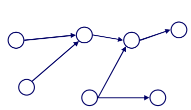
Faloutsos, Miller, Tsourakakis

CMU

Outline

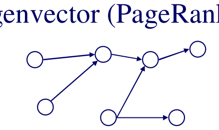
Node importance - Motivation:

- Given a graph (eg., web pages containing the desirable query word)
- Q: Which node is the most important?



Node importance - Motivation:

- Given a graph (eg., web pages containing the desirable query word)
- Q: Which node is the most important?
- A1: HITS (SVD = Singular Value Decomposition)
- A2: eigenvector (PageRank)



CMU SCS

Node importance - motivation

- SVD and eigenvector analysis: very closely related
- See ‘theory Task’, later

CMU SCS

SVD - Detailed outline

- Motivation
- Definition - properties
- Interpretation
- Complexity
- Case studies

CMU SCS

SVD - Motivation

- problem #1: text - LSI: find ‘concepts’
- problem #2: compression / dim. reduction

CMU SCS

SVD - Motivation

- problem #1: text - LSI: find ‘concepts’

term / document	data	information	retrieval	brain	lung
CS-TR1	1	1	1	0	0
CS-TR2	2	2	2	0	0
CS-TR3	1	1	1	0	0
CS-TR4	5	5	5	0	0
MED-TR1	0	0	0	2	2
MED-TR2	0	0	0	3	3
MED-TR3	0	0	0	1	1

CMU SCS

SVD - Motivation

- Customer-product, for recommendation system:

$$\begin{array}{r|ccccc|} & \text{bread} & \text{lettuce} & \text{tomatos} & \text{beef} & \text{chicken} \\ \hline \text{vegetarians} & 1 & 1 & 1 & 0 & 0 \\ & 2 & 2 & 2 & 0 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ & 5 & 5 & 5 & 0 & 0 \\ \text{meat eaters} & 0 & 0 & 0 & 2 & 2 \\ & 0 & 0 & 0 & 3 & 3 \\ & 0 & 0 & 0 & 1 & 1 \end{array}$$

CMU SCS

details

SVD - Motivation

- problem #2: compress / reduce dimensionality

CMU SCS

details

Problem - specs

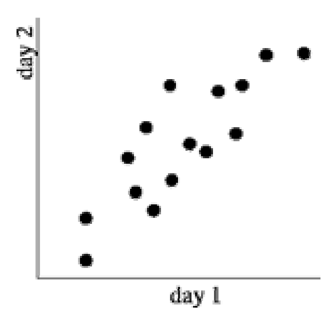
- ~10**6 rows; ~10**3 columns; no updates;
- random access to any cell(s) ; small error: OK

customer \ day	We 7/10/96	Th 7/11/96	Fr 7/12/96	Sa 7/13/96	Su 7/14/96
ABC Inc.	1	1	1	0	0
DEF Ltd.	2	2	2	0	0
GHI Inc.	1	1	1	0	0
KLM Co.	5	5	5	0	0
Smith	0	0	0	2	2
Johnson	0	0	0	3	3
Thompson	0	0	0	1	1

CMU SCS

details

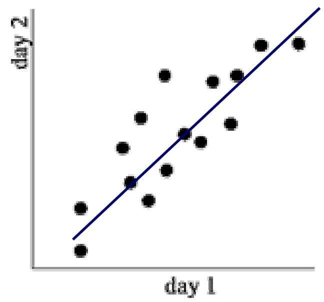
SVD - Motivation



CMU SCS

SVD - Motivation

details



KDD'09 Faloutsos, Miller, Tsourakakis P1-13

CMU SCS

SVD - Detailed outline

- Motivation
- Definition - properties
- Interpretation
- Complexity
- Case studies
- Additional properties

KDD'09 Faloutsos, Miller, Tsourakakis P1-14

CMU SCS

SVD - Definition

(reminder: matrix multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

3×2 $\quad$ 2×1

KDD'09 Faloutsos, Miller, Tsourakakis P1-15

CMU SCS

SVD - Definition

(reminder: matrix multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

3×2 $\quad$ 2×1 $\quad$ 3×1

KDD'09 Faloutsos, Miller, Tsourakakis P1-16

SVD - Definition

(reminder: matrix multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ \\ \\ \end{bmatrix}$$

3 x *2*   *2* x *1*   *3* x *1*

SVD - Definition

(reminder: matrix multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ \\ \end{bmatrix}$$

3 x *2*   *2* x *1*   *3* x *1*

SVD - Definition

(reminder: matrix multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

SVD - Definition

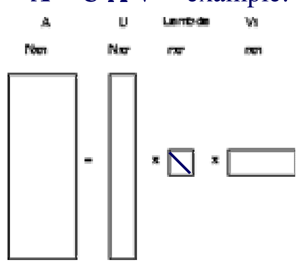
$$\boxed{\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} (\mathbf{V}_{[m \times r]})^T}$$

- **A**: n x m matrix (eg., n documents, m terms)
- **U**: n x r matrix (n documents, r concepts)
- **Λ**: r x r diagonal matrix (strength of each 'concept') (r : rank of the matrix)
- **V**: m x r matrix (m terms, r concepts)

CMU SCS

SVD - Definition

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



KDD'09 Faloutsos, Miller, Tsourakakis P1-21

CMU SCS

SVD - Properties

THEOREM [Press+92]: always possible to decompose matrix **A** into $\mathbf{A} = \mathbf{U}\, \boldsymbol{\Lambda}\, \mathbf{V}^T$, where

- **U, Λ, V**: unique (*)
- **U**, **V**: column orthonormal (ie., columns are unit vectors, orthogonal to each other)
  - $\mathbf{U}^T\mathbf{U} = \mathbf{I}$; $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ (**I:** identity matrix)
- **Λ:** singular are positive, and sorted in decreasing order

KDD'09 Faloutsos, Miller, Tsourakakis P1-22

CMU SCS

SVD - Example

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

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

KDD'09 Faloutsos, Miller, Tsourakakis P1-23

CMU SCS

SVD - Example

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

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

KDD'09 Faloutsos, Miller, Tsourakakis P1-24

SVD - Example

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

doc-to-concept similarity matrix

CS-concept

MD-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 - Example

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

'strength' of 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 - Example

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

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

term-to-concept similarity matrix

CS-concept

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

CMU SCS

SVD - Detailed outline

- Motivation
- Definition - properties
- Interpretation
- Complexity
- Case studies
- Additional properties

CMU SCS

SVD - Interpretation #1

'documents', 'terms' and 'concepts':

- **U**: document-to-concept similarity matrix
- **V**: term-to-concept sim. matrix
- $\mathbf{\Lambda}$: its diagonal elements: 'strength' of each concept

CMU SCS

SVD – Interpretation #1

'documents', 'terms' and 'concepts':

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

A:

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

A:

CMU SCS

SVD – Interpretation #1

'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

SVD properties

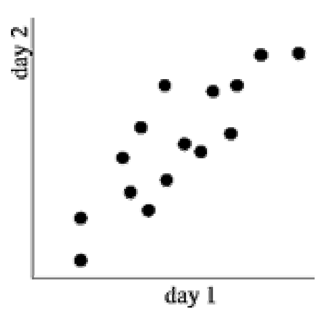
- **V** are the eigenvectors of the *covariance matrix* $\mathbf{A}^T\mathbf{A}$
- **U** are the eigenvectors of the *Gram (inner-product) matrix* $\mathbf{A}\mathbf{A}^T$

Further reading:
1. Ian T. Jolliffe, *Principal Component Analysis* (2nd ed), Springer, 2002.
2. Gilbert Strang, *Linear Algebra and Its Applications* (4th ed), Brooks Cole, 2005.

SVD - Interpretation #2

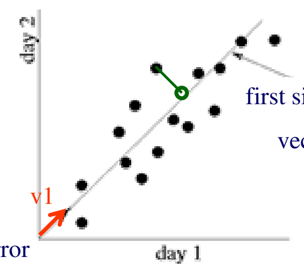
- best axis to project on: ('best' = min sum of squares of projection errors)

SVD - Motivation



SVD - interpretation #2

SVD: gives best axis to project



first singular vector

- minimum RMS error

CMU SCS

details

SVD - Interpretation #2

customer \ day	We 7/10/96	Th 7/11/96	Fr 7/12/96	Sa 7/13/96	Su 7/14/96
ABC Inc.	1	1	1	0	0
DEF Ltd.	2	2	2	0	0
GHI Inc.	1	1	1	0	0
KLM Co.	5	5	5	0	0
Smith	0	0	0	2	2
Johnson	0	0	0	3	3
Thompson	0	0	0	1	1

CMU SCS

details

SVD - Interpretation #2

- **A** = **U Λ V**^{T} - example:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{array}{c} \text{v1} \\ \begin{bmatrix} \boxed{0.58 \;\; 0.58 \;\; 0.58 \;\; 0 \;\; 0} \\ 0 \;\; 0 \;\; 0 \;\; 0.71 \;\; 0.71 \end{bmatrix} \end{array}$$

CMU SCS

details

SVD - Interpretation #2

- **A** = **U Λ V**^{T} - example:

variance ('spread') on the v1 axis

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

CMU SCS

details

SVD - Interpretation #2

- **A** = **U Λ V**^{T} - example:
  - **U Λ** gives the coordinates of the points in the projection axis

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

details

SVD - Interpretation #2

- More details
- Q: how exactly is dim. reduction done?

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

details

SVD - Interpretation #2

- More details
- Q: how exactly is dim. reduction done?
- A: set the smallest singular values to zero:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & \cancel{5.29} \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

details

SVD - Interpretation #2

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & \cancel{0} \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

details

SVD - Interpretation #2

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 & \cancel{0} \\ 0.36 & \cancel{0} \\ 0.18 & \cancel{0} \\ 0.90 & \cancel{0} \\ 0 & \cancel{0.53} \\ 0 & \cancel{0.80} \\ 0 & \cancel{0.27} \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & \cancel{0} \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0.71} & \cancel{0.71} \end{bmatrix}$$

details

SVD - Interpretation #2

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \end{bmatrix}$$

details

SVD - Interpretation #2

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

details

SVD - Interpretation #2

Exactly equivalent:
'spectral decomposition' of the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

details

SVD - Interpretation #2

Exactly equivalent:
'spectral decomposition' of the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \text{x} \begin{bmatrix} \lambda_1 & \varnothing \\ \varnothing & \lambda_2 \end{bmatrix} \text{x} \begin{bmatrix} \text{---} & v_1 & \text{---} \\ \text{---} & v_2 & \text{---} \end{bmatrix}$$

CMU SCS

details

SVD - Interpretation #2

Exactly equivalent:
'spectral decomposition' of the matrix:

$$\begin{array}{cc} & \leftarrow m \rightarrow \\ n & \begin{bmatrix} 1&1&1&0&0\\2&2&2&0&0\\1&1&1&0&0\\5&5&5&0&0\\0&0&0&2&2\\0&0&0&3&3\\0&0&0&1&1 \end{bmatrix} \end{array} = \lambda_1 \; u_1 \; v^T_1 + \lambda_2 \; u_2 \; v^T_2 + \ldots$$

KDD'09 Faloutsos, Miller, Tsourakakis P1-49

CMU SCS

details

SVD - Interpretation #2

Exactly equivalent:
'spectral decomposition' of the matrix:

$$\begin{array}{cc} & \leftarrow m \rightarrow \\ n & \begin{bmatrix} 1&1&1&0&0\\2&2&2&0&0\\1&1&1&0&0\\5&5&5&0&0\\0&0&0&2&2\\0&0&0&3&3\\0&0&0&1&1 \end{bmatrix} \end{array} = \overbrace{\lambda_1 \; \underbrace{u_1}_{n \times 1} \; \underbrace{v^T_1}_{1 \times m} + \lambda_2 \; u_2 \; v^T_2 + \ldots}^{r \text{ terms}}$$

KDD'09 Faloutsos, Miller, Tsourakakis P1-50

CMU SCS

details

SVD - Interpretation #2

approximation / dim. reduction:
by keeping the first few terms (Q: how many?)

$$\begin{array}{cc} & \leftarrow m \rightarrow \\ n & \begin{bmatrix} 1&1&1&0&0\\2&2&2&0&0\\1&1&1&0&0\\5&5&5&0&0\\0&0&0&2&2\\0&0&0&3&3\\0&0&0&1&1 \end{bmatrix} \end{array} = \lambda_1 \; u_1 \; v^T_1 + \lambda_2 \; u_2 \; v^T_2 + \ldots$$

assume: $\lambda_1 >= \lambda_2 >= \ldots$

KDD'09 Faloutsos, Miller, Tsourakakis P1-51

CMU SCS

details

SVD - Interpretation #2

A (heuristic - [Fukunaga]): keep 80-90% of 'energy' (= sum of squares of λ_i 's)

$$\begin{array}{cc} & \leftarrow m \rightarrow \\ n & \begin{bmatrix} 1&1&1&0&0\\2&2&2&0&0\\1&1&1&0&0\\5&5&5&0&0\\0&0&0&2&2\\0&0&0&3&3\\0&0&0&1&1 \end{bmatrix} \end{array} = \lambda_1 \; u_1 \; v^T_1 + \lambda_2 \; u_2 \; v^T_2 + \ldots$$

assume: $\lambda_1 >= \lambda_2 >= \ldots$

KDD'09 Faloutsos, Miller, Tsourakakis P1-52

SVD - Detailed outline

- Motivation
- Definition - properties
- Interpretation
  - #1: documents/terms/concepts
  - #2: dim. reduction
  - #3: picking non-zero, rectangular 'blobs'
- Complexity
- Case studies
- Additional properties

SVD - Interpretation #3

- finds non-zero 'blobs' in a data matrix

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #3

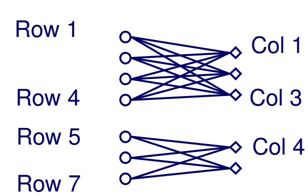
- finds non-zero 'blobs' in a data matrix

$$\left[\begin{array}{ccc|cc} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ \hline 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #3

- finds non-zero 'blobs' in a data matrix =
- 'communities' (bi-partite cores, here)

$$\left[\begin{array}{ccc|cc} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ \hline 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$



CMU SCS

SVD - Detailed outline

- Motivation
- Definition - properties
- Interpretation
- Complexity
- Case studies
- Additional properties

CMU SCS

SVD - Complexity

- O(n * m * m) or O(n * n * m) (whichever is less)
- less work, if we just want singular values
- or if we want first *k* singular vectors
- or if the matrix is sparse [Berry]
- Implemented: in any linear algebra package (LINPACK, matlab, Splus, mathematica ...)

CMU SCS

SVD - conclusions so far

- SVD: $\mathbf{A} = \mathbf{U} \, \mathbf{\Lambda} \, \mathbf{V}^T$: unique (*)
- **U**: document-to-concept similarities
- **V**: term-to-concept similarities
- **Λ**: strength of each concept
- dim. reduction: keep the first few strongest singular values (80-90% of 'energy')
  - SVD: picks up linear correlations
- SVD: picks up non-zero 'blobs'

CMU SCS

details

SVD - Detailed outline

- Motivation
- Definition - properties
- Interpretation
- Complexity
- SVD properties
- Case studies
- Conclusions

CMU SCS

details

SVD - Other properties - summary

- can produce orthogonal basis (obvious) (who cares?)
- can solve over- and under-determined linear problems (see C(1) property)
- can compute 'fixed points' (= 'steady state prob. in Markov chains') (see C(4) property)

KDD'09 Faloutsos, Miller, Tsourakakis P1-61

CMU SCS

details

SVD -outline of properties

- (A): obvious
- (B): less obvious
- (C): least obvious (and most powerful!)

KDD'09 Faloutsos, Miller, Tsourakakis P1-62

CMU SCS

details

Properties - by defn.:

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

A(1): $\mathbf{U}^{T}_{[r \times n]} \mathbf{U}_{[n \times r]} = \mathbf{I}_{[r \times r]}$ (identity matrix)

A(2): $\mathbf{V}^{T}_{[r \times n]} \mathbf{V}_{[n \times r]} = \mathbf{I}_{[r \times r]}$

A(3): $\mathbf{\Lambda}^{k} = \mathrm{diag}(\lambda_1^{k}, \lambda_2^{k}, \ldots \lambda_r^{k})$ (k: ANY real number)

A(4): $\mathbf{A}^{\mathbf{T}} = \mathbf{V}\, \mathbf{\Lambda}\, \mathbf{U}^{\mathbf{T}}$

KDD'09 Faloutsos, Miller, Tsourakakis P1-63

CMU SCS

details

Less obvious properties

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

B(1): $\mathbf{A}_{[n \times m]} (\mathbf{A}^{T})_{[m \times n]} = ??$

KDD'09 Faloutsos, Miller, Tsourakakis P1-64

CMU SCS

details

Less obvious properties

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

B(1): $\mathbf{A}_{[n \times m]} (\mathbf{A}^{T})_{[m \times n]} = \mathbf{U}\, \mathbf{\Lambda}^2\, \mathbf{U}^{T}$

symmetric; Intuition?

CMU SCS

details

Less obvious properties

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

B(1): $\mathbf{A}_{[n \times m]} (\mathbf{A}^{T})_{[m \times n]} = \mathbf{U}\, \mathbf{\Lambda}^2\, \mathbf{U}^{T}$

symmetric; Intuition?

'document-to-document' similarity matrix

B(2): symmetrically, for 'V'

(AT) [m x n] A [n x m] = V L2 VT

Intuition?

CMU SCS

details

Less obvious properties

A: term-to-term similarity matrix

B(3): $((\mathbf{A}^{T})_{[m \times n]} \mathbf{A}_{[n \times m]})^{k} = \mathbf{V}\, \mathbf{\Lambda}^{2k}\, \mathbf{V}^{T}$

and

B(4): $(\mathbf{A}^{T} \mathbf{A})^{k} \sim \mathbf{v}_1\, \lambda_1^{2k}\, \mathbf{v}_1^{T}$ for k>>1

where

$\mathbf{v}_1$: [m x 1] first column (singular-vector) of $\mathbf{V}$

λ_1: strongest singular value

CMU SCS

details

Less obvious properties

B(4): $(\mathbf{A}^{T} \mathbf{A})^{k} \sim \mathbf{v}_1\, \lambda_1^{2k}\, \mathbf{v}_1^{T}$ for k>>1

B(5): $(\mathbf{A}^{T} \mathbf{A})^{k}\, \mathbf{v'} \sim$ (constant) $\mathbf{v}_1$

ie., for (almost) any **v'**, it converges to a vector parallel to $\mathbf{v}_1$

Thus, useful to compute first singular vector/value (as well as the next ones, too...)

details

Less obvious properties - repeated:

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^T_{[r \times m]}$

B(1): $\mathbf{A}_{[n \times m]} (\mathbf{A}^T)_{[m \times n]} = \mathbf{U} \mathbf{\Lambda}^2 \mathbf{U}^T$

B(2): $(\mathbf{A}^T)_{[m \times n]} \mathbf{A}_{[n \times m]} = \mathbf{V} \mathbf{\Lambda}^2 \mathbf{V}^T$

B(3): $((\mathbf{A}^T)_{[m \times n]} \mathbf{A}_{[n \times m]})^k = \mathbf{V} \mathbf{\Lambda}^{2k} \mathbf{V}^T$

B(4): $(\mathbf{A}^T \mathbf{A})^k \sim v_1 \lambda_1^{2k} v_1^T$

B(5): $(\mathbf{A}^T \mathbf{A})^k \mathbf{v'} \sim$ (constant) $\mathbf{v}_1$

details

Least obvious properties - cont'd

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^T_{[r \times m]}$

C(2): $\mathbf{A}_{[n \times m]} \mathbf{v}_{1\,[m \times 1]} = \lambda_1 \mathbf{u}_{1\,[n \times 1]}$

where $\mathbf{v}_1$, $\mathbf{u}_1$ the first (column) vectors of **V**, **U**. ($\mathbf{v}_1$ == right-singular-vector)

C(3): symmetrically: $\mathbf{u}_1^T \mathbf{A} = \lambda_1 \mathbf{v}_1^T$

$\mathbf{u}_1$ == left-singular-vector

Therefore:

details

Least obvious properties - cont'd

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^T_{[r \times m]}$

$\boxed{\text{C(4): } \mathbf{A}^T \mathbf{A} \mathbf{v}_1 = \lambda_1^2 \mathbf{v}_1}$

(**fixed point** - the dfn of eigenvector for a symmetric matrix)

details

Least obvious properties - altogether

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^T_{[r \times m]}$

C(1): $\mathbf{A}_{[n \times m]} \mathbf{x}_{[m \times 1]} = \mathbf{b}_{[n \times 1]}$

then, $\mathbf{x}_0 = \mathbf{V} \mathbf{\Lambda}^{(-1)} \mathbf{U}^T \mathbf{b}$: shortest, actual or least-squares solution

C(2): $\mathbf{A}_{[n \times m]} \mathbf{v}_{1\,[m \times 1]} = \lambda_1 \mathbf{u}_{1\,[n \times 1]}$

C(3): $\mathbf{u}_1^T \mathbf{A} = \lambda_1 \mathbf{v}_1^T$

C(4): $\mathbf{A}^T \mathbf{A} \mathbf{v}_1 = \lambda_1^2 \mathbf{v}_1$

details

Properties - conclusions

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

B(5): $(\mathbf{A}^{T} \mathbf{A})^{k} \mathbf{v'} \sim$ (constant) $\mathbf{v}_1$

C(1): $\mathbf{A}_{[n \times m]} \mathbf{x}_{[m \times 1]} = \mathbf{b}_{[n \times 1]}$

then, $\mathbf{x}_0 = \mathbf{V} \mathbf{\Lambda}^{(-1)} \mathbf{U}^{T} \mathbf{b}$**:** shortest, actual or least-squares solution

C(4): $\mathbf{A}^{\mathbf{T}} \mathbf{A}\ \mathbf{v}_1 = \boldsymbol{\lambda}_1^2\ \mathbf{v}_1$

SVD - detailed outline

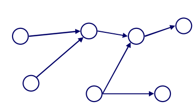
- ...
- SVD properties
- case studies
  - Kleinberg's algorithm
  - Google's algorithm
- Conclusions

Kleinberg's algo (HITS)

Kleinberg, Jon (1998). *Authoritative sources in a hyperlinked environment*. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms.

Recall: problem dfn

- Given a graph (eg., web pages containing the desirable query word)
- Q: Which node is the most important?



CMU SCS

Kleinberg's algorithm

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

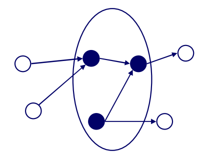
Step 0: find all pages containing the query terms

Step 1: expand by one move forward and backward

CMU SCS

Kleinberg's algorithm

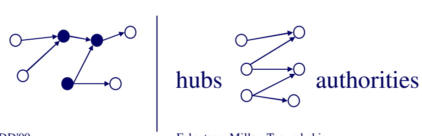
- Step 1: expand by one move forward and backward



CMU SCS

Kleinberg's algorithm

- 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')



CMU SCS

Kleinberg's algorithm

observations

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

CMU SCS

Kleinberg's algorithm

Let E be the set of edges and $\mathbf{A}$ be the adjacency matrix:

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

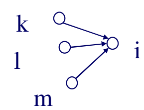
Let h and a be [n x 1] vectors with the 'hubness' and 'authoritativiness' scores.

Then:

KDD'09 Faloutsos, Miller, Tsourakakis P1-81

CMU SCS

Kleinberg's algorithm



Then:

$$a_i = h_k + h_l + h_m$$

that is

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

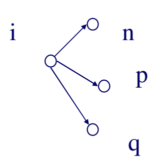
or

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

KDD'09 Faloutsos, Miller, Tsourakakis P1-82

CMU SCS

Kleinberg's algorithm



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

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

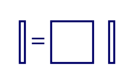
KDD'09 Faloutsos, Miller, Tsourakakis P1-83

CMU SCS

Kleinberg's algorithm

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

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



Recall properties:

C(2): $\mathbf{A}_{[n \times m]}\, \mathbf{v_1}_{[m \times 1]} = \lambda_1\, \mathbf{u_1}_{[n \times 1]}$

C(3): $\mathbf{u_1}^T \mathbf{A} = \lambda_1\, \mathbf{v_1}^T$

KDD'09 Faloutsos, Miller, Tsourakakis P1-84

CMU SCS

Kleinberg's algorithm

In short, the solutions to

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

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

are the left- and right- singular-vectors of the adjacency matrix **A.**

Starting from random **a'** and iterating, we'll eventually converge

(Q: to which of all the singular-vectors? why?)

CMU SCS

Kleinberg's algorithm

(Q: to which of all the singular-vectors? why?)

A: to the ones of the strongest singular-value, because of property B(5):

B(5): $(\mathbf{A}^T\ \mathbf{A}\ )^k\ \mathbf{v'} \sim$ (constant) $\mathbf{v}_1$

CMU SCS

Kleinberg's algorithm - results

Eg., for the query 'java':

0.328 www.gamelan.com

0.251 java.sun.com

0.190 www.digitalfocus.com ("the java developer")

CMU SCS

Kleinberg's algorithm - discussion

- 'authority' score can be used to find 'similar pages' (how?)

CMU SCS

SVD - detailed outline

- ...
- Complexity
- SVD properties
- Case studies
  - Kleinberg’s algorithm (HITS)
  - Google’s algorithm
- Conclusions

KDD'09 Faloutsos, Miller, Tsourakakis P1-89

CMU SCS

PageRank (google)

Larry Page   Sergey Brin

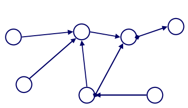
•Brin, Sergey and Lawrence Page (1998). *Anatomy of a Large-Scale Hypertextual Web Search Engine*. 7th Intl World Wide Web Conf.

KDD'09 Faloutsos, Miller, Tsourakakis P1-90

CMU SCS

Problem: PageRank

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



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

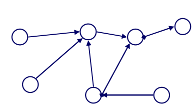
KDD'09 Faloutsos, Miller, Tsourakakis P1-91

CMU SCS

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))

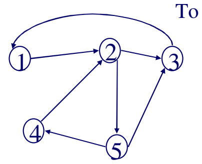


A node has high ssp, if it is connected with high ssp nodes (recursive, but OK!)

KDD'09 Faloutsos, Miller, Tsourakakis P1-92

(Simplified) PageRank algorithm

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

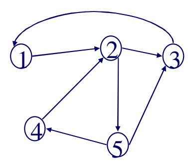


From / To

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

(Simplified) PageRank algorithm

- **B p = p**



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

Definitions

A Adjacency matrix (from-to)

D Degree matrix = (diag (d1, d2, …, dn))

B Transition matrix: to-from, column normalized

$\mathbf{B} = \mathbf{A}^T \mathbf{D}^{-1}$

(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)
- Why does such a **p** exist?
  - **p** exists if **B** is nxn, nonnegative, irreducible [Perron–Frobenius theorem]

(Simplified) PageRank algorithm

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

Full version of algo: with occasional random jumps

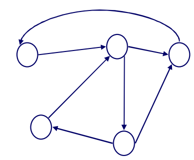
Why? To make the matrix irreducible

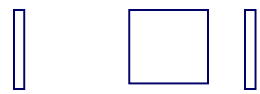
Full Algorithm

- With probability *1-c*, fly-out to a random node
- Then, we have

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

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





Alternative notation

M Modified transition matrix

$\mathbf{M} = \text{c } \mathbf{B} + (1\text{-c})/\text{n } \; \mathbf{1} \; \mathbf{1}^T$

Then

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

That is: the steady state probabilities =

PageRank scores form the *first eigenvector* of the 'modified transition matrix'

Parenthesis: intuition behind eigenvectors

Formal definition

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

$$\boxed{\mathbf{A}\ \mathbf{x} = \lambda\ \mathbf{x}}$$

CLOSELY related to singular values:

Property #1: Eigen- vs singular-values

if

$$\mathbf{B}_{[n \times m]} = \mathbf{U}_{[n \times r]}\ \mathbf{\Lambda}_{[r \times r]}\ (\mathbf{V}_{[m \times r]})^T$$

then $\mathbf{A} = (\mathbf{B}^T\mathbf{B})$ is symmetric and

$$\text{C(4): } \mathbf{B}^T\ \mathbf{B}\ \mathbf{v}_i = \lambda_i^2\ \mathbf{v}_i$$

ie, $\mathbf{v}_1$, $\mathbf{v}_2$, ...: eigenvectors of $\mathbf{A} = (\mathbf{B}^T\mathbf{B})$

Property #2

- If $\mathbf{A}_{[nxn]}$ is a real, symmetric matrix
- Then it has *n* real eigenvalues

(if **A** is not symmetric, some eigenvalues may be complex)

Property #3

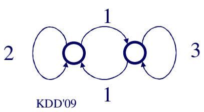
- If $\mathbf{A}_{[nxn]}$ is a real, symmetric matrix
- Then it has *n* real eigenvalues
- And they agree with its *n* singular values, except possibly for the sign

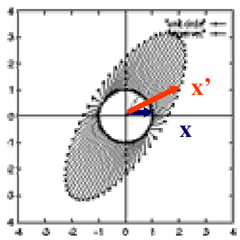
CMU SCS

Intuition

- **A** as vector transformation

$$\begin{array}{ccccc} \mathbf{x'} & & \mathbf{A} & & \mathbf{x} \\ \begin{bmatrix} 2 \\ 1 \end{bmatrix} & = & \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} & & \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{array}$$



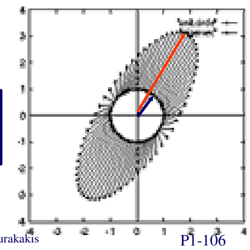


CMU SCS

Intuition

- By defn., eigenvectors remain parallel to themselves ('**fixed points**')

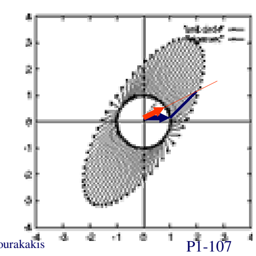
$$\begin{array}{ccccc} \lambda_1 & \mathbf{v_1} & & \mathbf{A} & \mathbf{v_1} \\ 3.62\ * & \begin{bmatrix} 0.52 \\ 0.85 \end{bmatrix} & = & \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} & \begin{bmatrix} 0.52 \\ 0.85 \end{bmatrix} \end{array}$$



CMU SCS

Convergence

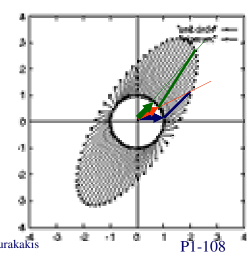
- Usually, fast:



CMU SCS

Convergence

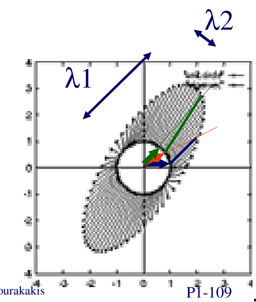
- Usually, fast:



CMU SCS

Convergence

- Usually, fast:
- depends on ratio
  $\lambda 1 : \lambda 2$



CMU SCS

Kleinberg/google - conclusions

SVD helps in graph analysis:

hub/authority scores: strongest left- and right-singular-vectors of the adjacency matrix

random walk on a graph: steady state probabilities are given by the strongest eigenvector of the (modified) transition matrix

CMU SCS

Conclusions

- SVD: a **valuable** tool
- given a document-term matrix, it finds ‘concepts’ (LSI)
- ... and can find fixed-points or steady-state probabilities (google/ Kleinberg/ Markov Chains)

CMU SCS

Conclusions cont’d

(We didn’t discuss/elaborate, but, SVD

- ... can reduce dimensionality (KL)
- ... and can find rules (PCA; RatioRules)
- ... and can solve optimally over- and under-constraint linear systems (least squares / query feedbacks)

References

- Berry, Michael: http://www.cs.utk.edu/~lsi/
- Brin, S. and L. Page (1998). *Anatomy of a Large-Scale Hypertextual Web Search Engine*. 7th Intl World Wide Web Conf.

References

- Christos Faloutsos, *Searching Multimedia Databases by Content*, Springer, 1996. (App. D)
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press.
- I.T. Jolliffe *Principal Component Analysis* Springer, 2002 (2nd ed.)

References cont'd

- Kleinberg, J. (1998). *Authoritative sources in a hyperlinked environment*. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms.
- Press, W. H., S. A. Teukolsky, et al. (1992). *Numerical Recipes in C*, Cambridge University Press. `www.nr.com`

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

Task 2: Community Detection

Faloutsos, Miller, Tsourakakis
CMU

Outline

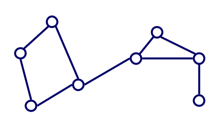
- Introduction – Motivation
- Task 1: Node importance
- Task 2: Community detection
- Task 3: Recommendations
- Task 4: Connection sub-graphs
- Task 5: Mining graphs over time
- Task 6: Virus/influence propagation
- Task 7: Spectral graph theory
- Task 8: Tera/peta graph mining: hadoop
- Observations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

Detailed outline

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

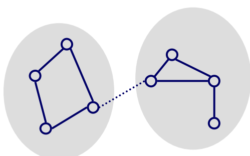
Problem

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



Problem

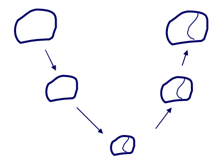
- Given a graph, and *k*
- Break it into *k* (disjoint) communities



$k = 2$

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 2^{nd} smallest eigenvector of the (normalized) Laplacian

See details in 'Task 7', later

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

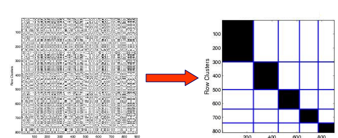
- Motivation
- Hard clustering – *k* pieces
- Hard co-clustering – (*k,l*) pieces
- Hard clustering – optimal # pieces
- Soft clustering – matrix decompositions
- Observations

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



Co-clustering

details

- 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

eg, terms x documents

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

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

med. doc

cs doc

med. terms

cs terms

common terms

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

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
  - 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

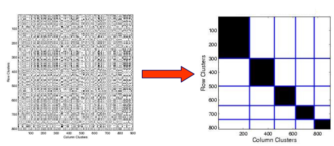
Desiderata:

✓ Simultaneously discover row and column groups

✘ Fully Automatic: No "magic numbers"

✓ Scalable to large graphs

Cross-association



Desiderata:

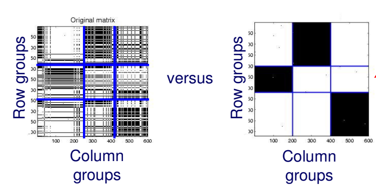
✓ Simultaneously discover row and column groups

✓ Fully Automatic: No "magic numbers"

✓ Scalable to large matrices

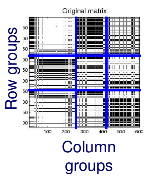
Reference:
1. Chakrabarti et al. Fully Automatic Cross-Associations, KDD'04

What makes a cross-association "good"?

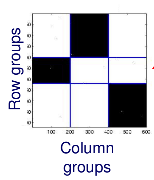


Why is this better?

What makes a cross-association "good"?



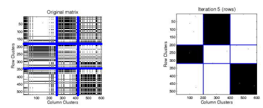
versus



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

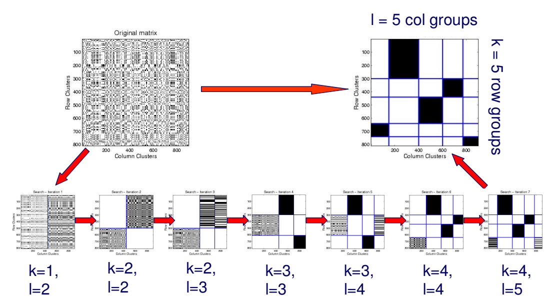
Good Compression → Better Clustering

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

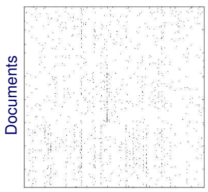
Minimize the total cost (# bits)

for lossless compression

Algorithm



Experiments



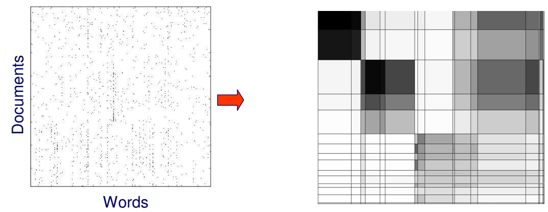
"CLASSIC"

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

Combination of 3 sources:

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

Experiments



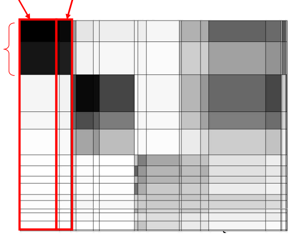
"CLASSIC" graph of documents & words: k=15, l=19

Experiments

insipidus, alveolar, aortic, death, prognosis, intravenous

blood, disease, clinical, cell, tissue, patient

MEDLINE (medical)



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

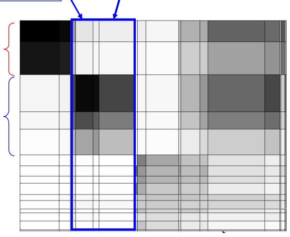
Experiments

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

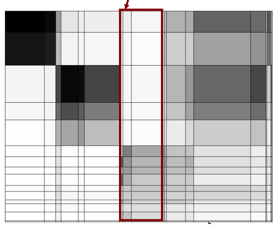
Experiments

shape, nasa, leading, assumed, thin

MEDLINE (medical)

CISI (Information Retrieval)

CRANFIELD (aerodynamics)



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

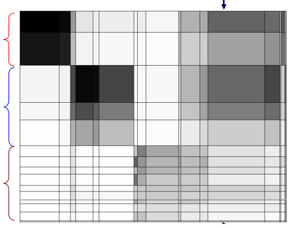
Experiments

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):

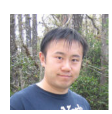
www.cs.cmu.edu/~deepay/mywww/software/CrossAssociations-01-27-2005.tgz

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

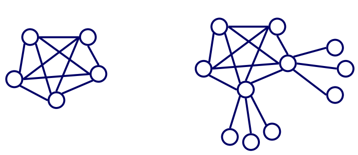
- Motivation
- Hard clustering – *k* pieces
- Hard co-clustering – (*k*,*l*) pieces
- Hard clustering – optimal # pieces
- Observations

Observation #1

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

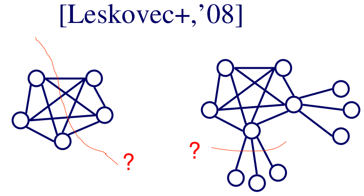
Observation #2

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



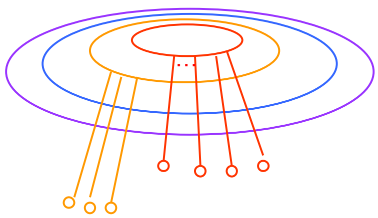
Observation #2

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



CMU SCS

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.

CMU SCS

Strange behavior of min cuts

- 'negative dimensionality' (!)

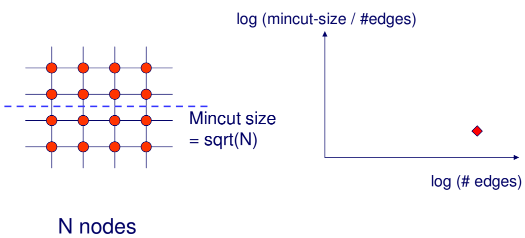
NetMine: New Mining Tools for Large Graphs, by D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos and G. Blelloch, in the SDM 2004 Workshop on Link Analysis, Counter-terrorism and Privacy

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

CMU SCS

"Min-cut" plot

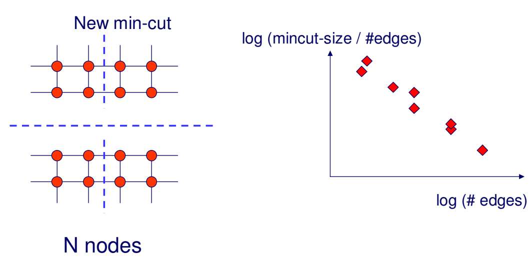
- Do min-cuts recursively.



CMU SCS

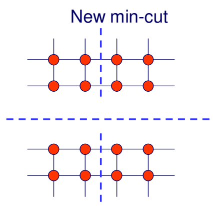
"Min-cut" plot

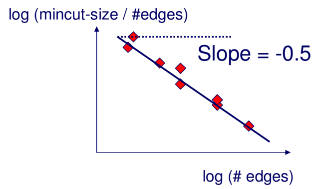
- Do min-cuts recursively.



"Min-cut" plot

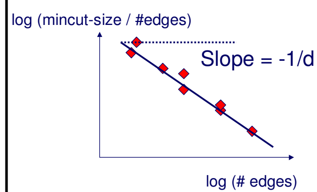
- Do min-cuts recursively.



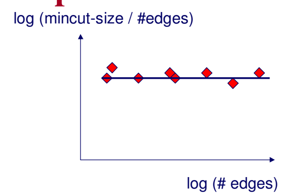


For a d-dimensional grid, the slope is -1/d

"Min-cut" plot



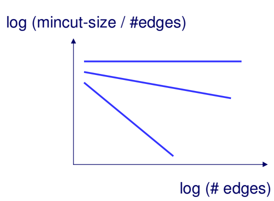
For a d-dimensional grid, the slope is -1/d



For a random graph, the slope is 0

"Min-cut" plot

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



?

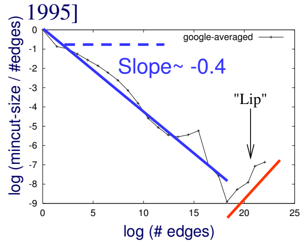
Experiments

- 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

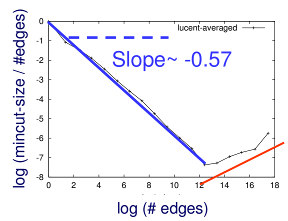
- Used the METIS algorithm [Karypis, Kumar, 1995]



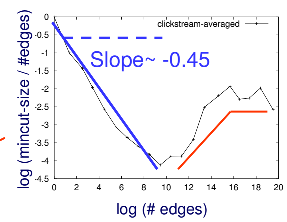
- Google Web graph
- Values along the y-axis are averaged
- We observe a “lip” for large edges
- Slope of -0.4, corresponds to a 2.5-dimensional grid!

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**

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

Task 3: Recommendations & proximity

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

Acknowledgement:

Most of the foils in 'Task 3' are by

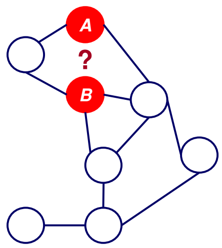
Hanghang TONG

www.cs.cmu.edu/~htong

Detailed outline

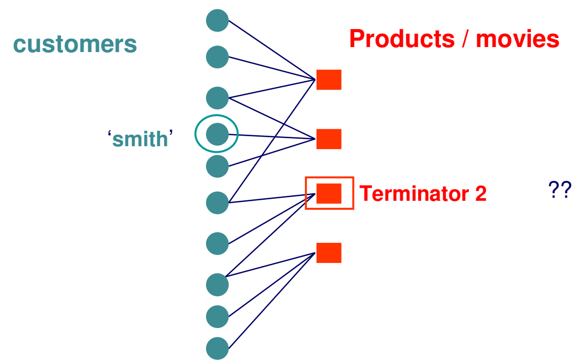
- ➡ Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

Motivation: Link Prediction



Should we introduce
Mr. A to Mr. B?

Motivation - recommendations



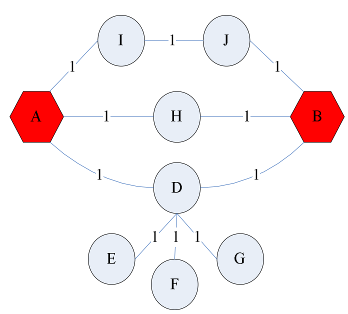
Answer: proximity

- 'yes', if 'A' and 'B' are 'close'
- 'yes', if 'smith' and 'terminator 2' are 'close'

QUESTIONS in this part:

- How to measure 'closeness'/proximity?
- How to do it quickly?
- What else can we do, given proximity scores?

How close is 'A' to 'B'?



a.k.a Relevance, Closeness, 'Similarity'…

Why is it useful?

- Recommendation

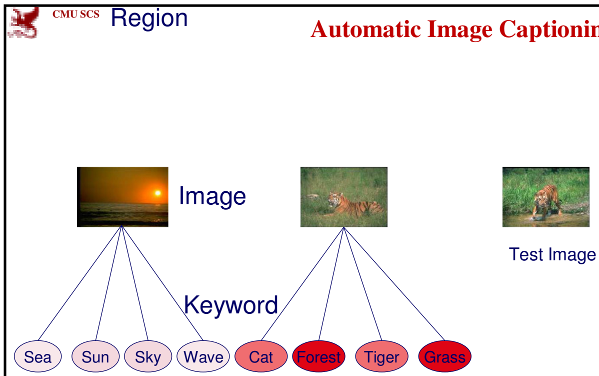
And many more

- **Image captioning** [Pan+]
- **Conn. / CenterPiece subgraphs** [Faloutsos+], [Tong+], [Koren+]

and

- Link prediction [Liben-Nowell+], [Tong+]
- Ranking [Haveliwala], [Chakrabarti+]
- Email Management [Minkov+]
- Neighborhood Formulation [Sun+]
- Pattern matching [Tong+]
- Collaborative Filtering [Fouss+]
- …

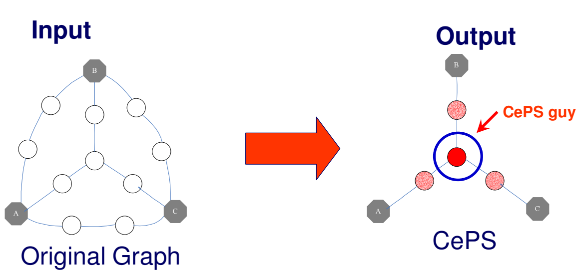
Automatic Image Captioning



Q: How to assign keywords to the test image?

A: Proximity! [Pan+ 2004]

Center-Piece Subgraph(CePS)



Q: How to find hub for the black nodes?

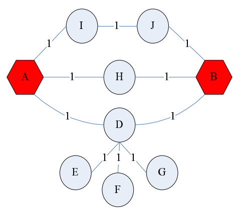
A: Proximity! [Tong+ KDD 2006]

Detailed outline

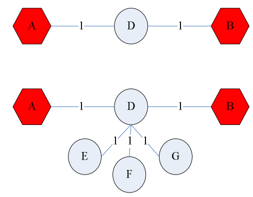
- Problem dfn and motivation
- ➡ Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

How close is 'A' to 'B'?

Should be close, if they have
- many,
- short
- 'heavy' paths

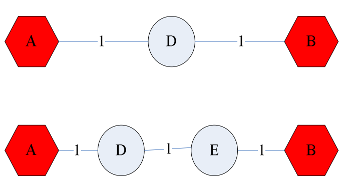


Why not shortest path?



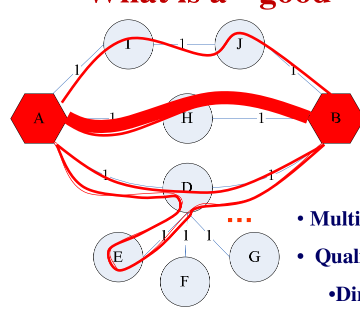
A: 'pizza delivery guy' problem

Why not max. netflow?



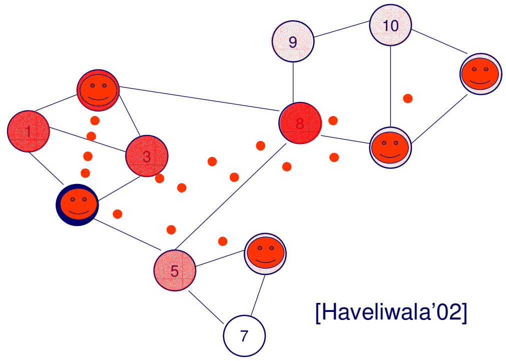
A: No penalty for long paths

What is a ``good'' Proximity?



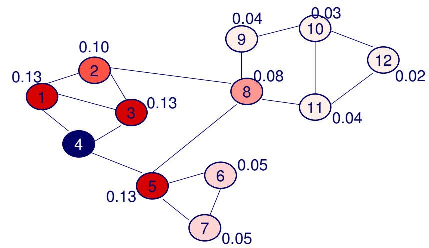
- **Multiple Connections**
- **Quality of connection**
  - **Direct & In-directed Conns**
  - **Length, Degree, Weight…**

Random walk with restart



[Haveliwala'02]

Random walk with restart



	Node 4
Node 1	0.13
Node 2	0.10
Node 3	0.13
Node 4	0.22
Node 5	0.13
Node 6	0.05
Node 7	0.05
Node 8	0.08
Node 9	0.04
Node 10	0.03
Node 11	0.04
Node 12	0.02

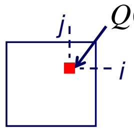
Nearby nodes, higher scores

More red, more relevant

Ranking vector $\vec{r}_4$

Why RWR is a good score?

$Q=(I-c\tilde{W})^{-1}=$



$Q(i,j)\propto r_{i,j}$

$\tilde{W}$: adjacency matrix.
c: damping factor

$$Q = c\,\tilde{W} + c^2\,\tilde{W}^2 + c^3\,\tilde{W}^3 + \cdots$$

all paths from *i* to *j* with length 1

all paths from *i* to *j* with length 2

all paths from *i* to *j* with length 3

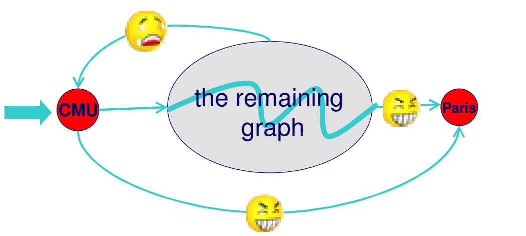
Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
  - variants
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

CMU SCS

Variant: escape probability

- Define Random Walk (RW) on the graph
- Esc_Prob(CMU→Paris)
  - Prob (starting at CMU, reaches Paris before returning to CMU)



Esc_Prob = Pr (smile before cry)

CMU SCS

Other Variants

details

- Other measure by RWs
  - Community Time/Hitting Time [Fouss+]
  - SimRank [Jeh+]
- Equivalence of Random Walks
  - Electric Networks:
    - EC [Doyle+]; SAEC[Faloutsos+]; CFEC[Koren+]
  - Spring Systems
- Katz [Katz], [Huang+], [Scholkopf+]
- Matrix-Forest-based Alg [Chobotarev+]

CMU SCS

Other Variants

details

- Other measure by RWs
  - Community Time/Hitting Time [Fouss+]
  - SimRank [Jeh+]

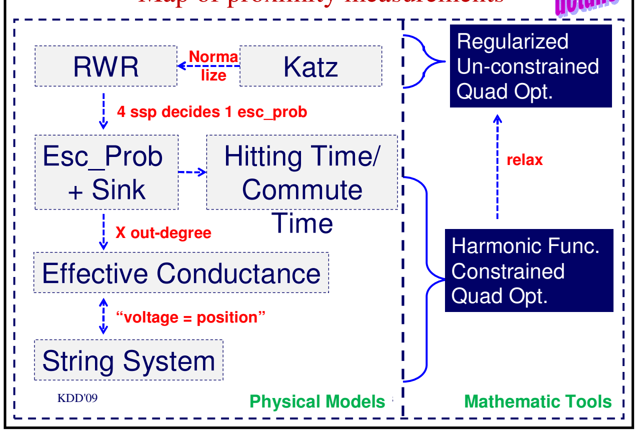
All are "related to" or "similar to" random walk with restart!

- Katz [Katz], [Huang+], [Scholkopf+]
- Matrix-Forest-based Alg [Chobotarev+]

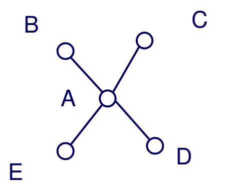
CMU SCS

Map of proximity measurements

details



Notice: Asymmetry (even in undirected graphs)



C-> A : high
A-> C: low

Summary of Proximity Definitions

- Goal: Summarize multiple relationships
- Solutions
  - Basic: Random Walk with Restarts
    - [Haweliwala'02] [Pan+ 2004][Sun+ 2006][Tong+ 2006]
  - Properties: Asymmetry
    - [Koren+ 2006][Tong+ 2007] [Tong+ 2008]
  - Variants: Esc_Prob and many others.
    - [Faloutsos+ 2004] [Koren+ 2006][Tong+ 2007]

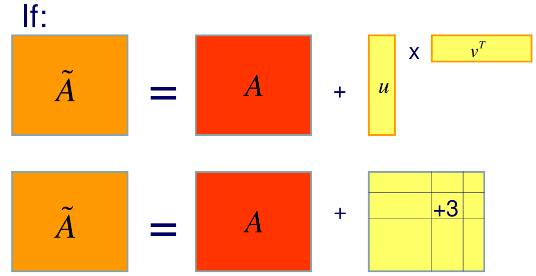
Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
- **Efficient computation**
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

Preliminary: Sherman–Morrison Lemma

details

If:



Preliminary: Sherman–Morrison Lemma

details

If:

$$\tilde{A} = A + u \times v^T$$

Then:

$$\tilde{A}^{-1} = (A + u \cdot v^T)^{-1} = A^{-1} - \frac{A^{-1} \cdot u \cdot v^T A^{-1}}{1 + v^T \cdot A^{-1} \cdot u}$$

Sherman – Morrison Lemma – intuition:

- Given a small perturbation on a matrix
  A -> A'
- We can quickly update its inverse

SM: The block-form

details

A	B
C	D

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

Or…

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

And many other variants…
Also known as Woodbury Identity

SM Lemma: Applications

details

- RLS (Recursive least squares)
  - and almost any algorithm in time series!
- Kalman filtering
- Incremental matrix decomposition
- …
- … and all the fast solutions we will introduce!

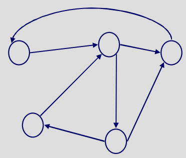
CMU SCS

Reminder: PageRank

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

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

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



CMU SCS

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

The only difference

$$\vec{r}_i = c\tilde{W}\vec{r}_i + (1-c)\vec{e}_i$$

Ranking vector | Adjacency matrix | Restart p | Starting vector

CMU SCS

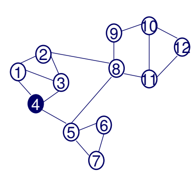
$$\mathbf{p} = c\ \mathbf{B}\ \mathbf{p} + (1\text{-}c)/n\ \mathbf{1}$$

Computing RWR

$$\vec{r}_i = c\tilde{W}\vec{r}_i + (1-c)\vec{e}_i$$

Ranking vector | Adjacency matrix | Restart p | Starting vector

$$\underbrace{\begin{pmatrix}0.13\\0.10\\0.13\\0.22\\0.13\\0.05\\0.05\\0.08\\0.04\\0.03\\0.04\\0.02\end{pmatrix}}_{n \times 1} = 0.9 \times \underbrace{\begin{pmatrix}0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0\\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0\end{pmatrix}}_{n \times n} \begin{pmatrix}0.13\\0.10\\0.13\\0.22\\0.13\\0.05\\0.05\\0.08\\0.04\\0.03\\0.04\\0.02\end{pmatrix} + 0.1 \times \underbrace{\begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}}_{n \times 1}$$



CMU SCS

Q: Given query *i*, how to solve it?

$$\Bigg\{\ ?\ \Bigg\} = 0.9 \times \begin{pmatrix}0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0\\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0\end{pmatrix} \Bigg\{\ ?\ \Bigg\} + 0.1 \times \begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}$$

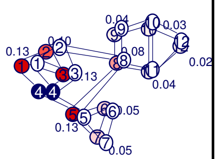
Query

Ranking vector | Adjacency matrix | Ranking vector | Starting vector

CMU SCS

OntheFly: $\vec{r}_i[t+1] = c\tilde{W}\vec{r}_i[t] + (1-c)\vec{e}_i$

$$\vec{r}_i = 0.9 \times \underbrace{\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \end{pmatrix}}_{\tilde{W}} \underbrace{\begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix}}_{\vec{r}_i} + 0.1 \times \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{\vec{e}_i}$$



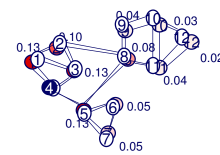
No pre-computation/ light storage

Slow on-line response O(mE)

CMU SCS

PreCompute

R:



$R = c \times Q$

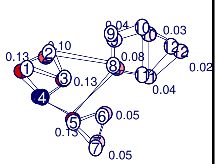
$Q = (I - c\tilde{W})^{-1}$

CMU SCS

PreCompute: $Q = (I - c\tilde{W})^{-1}$

$Q = (I - c\tilde{W})^{-1}$

$$\begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix} \leftarrow 0.1 \times \begin{matrix} 1.29 \\ 0.96 \\ 1.29 \\ 2.06 \\ 1.27 \\ 0.52 \\ 0.52 \\ 0.82 \\ 0.28 \\ 0.34 \\ 0.38 \\ 0.21 \end{matrix}$$



Fast on-line response

Heavy pre-computation/storage cost

O(n^3) O(n^2)

CMU SCS

Q: How to Balance?

How to balance?

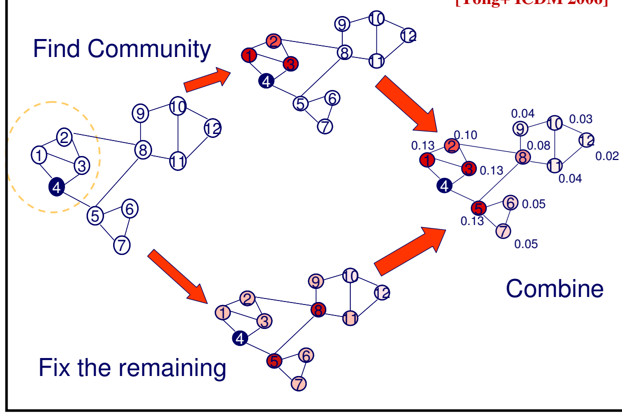
Idea ('B-Lin')

- Break into communities
- Pre-compute all, within a community
- Adjust (with S.M.) for 'bridge edges'

H. Tong, C. Faloutsos, & J.Y. Pan. *Fast Random Walk with Restart and Its Applications*. ICDM, 613-622, 2006.

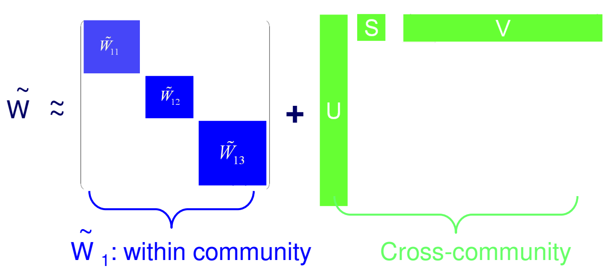
B_Lin: Basic Idea

[Tong+ ICDM 2006]



details

B_Lin: details



details

B_Lin: details

$$\left(I - c\tilde{W}\right)^{-1} \approx \left(\boxed{I - c\tilde{W}_1} - \boxed{cUSV}\right)^{-1}$$

Easy to be inverted

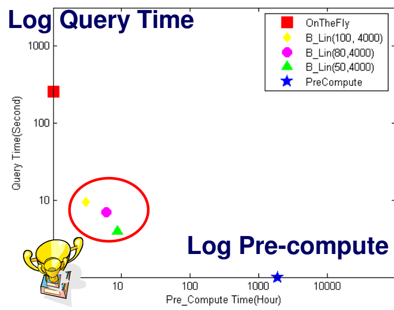
LRA difference

SM Lemma!

B_Lin: summary

- Pre-Computational Stage
  - Q: Efficiently compute and store Q
  - A: A few small, instead of ONE BIG, matrices inversions
- On-Line Stage
  - Q: Efficiently recover one column of Q
  - A: A few, instead of MANY, matrix-vector multiplications

Query Time vs. Pre-Compute Time



- Quality: 90%+
- On-line:
  - Up to 150x speedup
- Pre-computation:
  - Two orders saving

Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

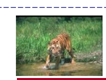
gCaP: Automatic Image Caption

- Q

{Sea Sun Sky Wave}

…

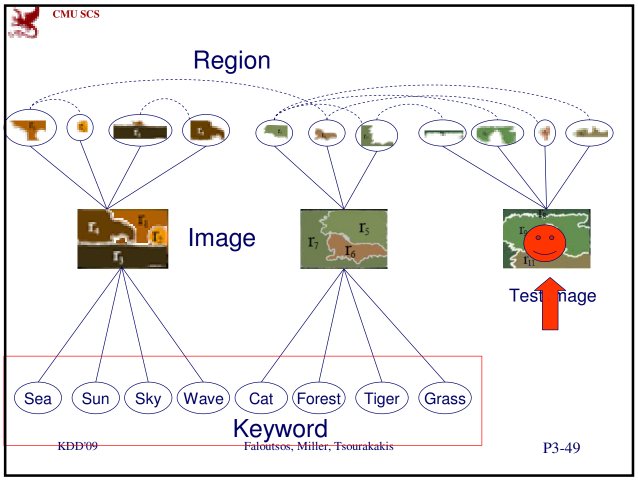
{Cat Forest Grass Tiger}

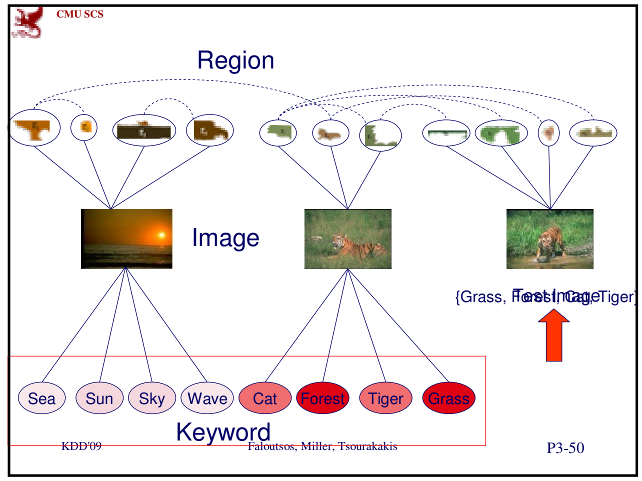


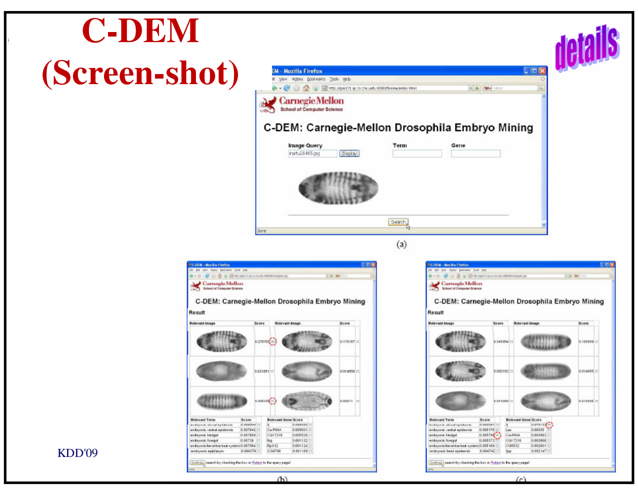
{?, ?, ?,}

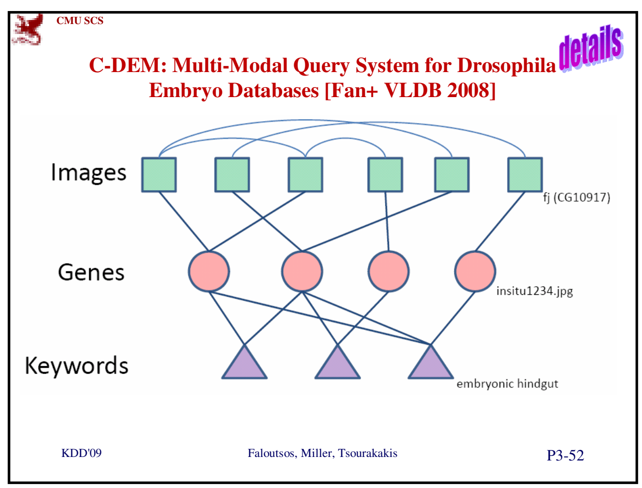
A: Proximity!

[Pan+ KDD2004]


CMU SCS
Region
Image
Test Image
Sea
Sun
Sky
Wave
Cat
Forest
Tiger
Grass
Keyword
KDD'09
Faloutsos, Miller, Tsourakakis
P3-49


CMU SCS
Region
Image
Test Image
{Grass, Forest, Cat, Tiger}
Sea
Sun
Sky
Wave
Cat
Forest
Tiger
Grass
Keyword
KDD'09
Faloutsos, Miller, Tsourakakis
P3-50

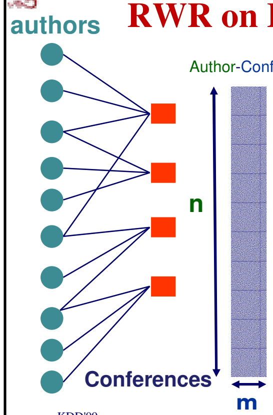

C-DEM
(Screen-shot)
details
Carnegie Mellon
School of Computer Science
C-DEM: Carnegie-Mellon Drosophila Embryo Mining
Image Query
Term
Gene
Search
(a)
C-DEM: Carnegie-Mellon Drosophila Embryo Mining
Result
C-DEM: Carnegie-Mellon Drosophila Embryo Mining
Result
KDD'09


CMU SCS
details
C-DEM: Multi-Modal Query System for Drosophila Embryo Databases [Fan+ VLDB 2008]
Images
Genes
Keywords
fj (CG10917)
insitu1234.jpg
embryonic hindgut
KDD'09
Faloutsos, Miller, Tsourakakis
P3-52

Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: **bi-partite graphs**; tracking
- Conclusions

RWR on Bipartite Graph



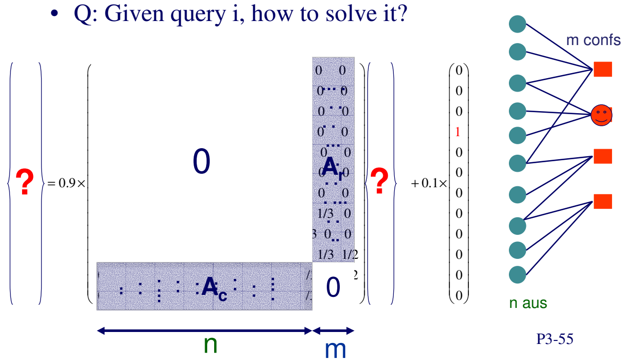
Observation: n >> m!

Examples:

1. DBLP: 400k aus, 3.5k confs
2. NetFlix: 2.7M usrs,18k mvs

RWR on Skewed bipartite graphs

- Q: Given query i, how to solve it?



Idea:

- Pre-compute the smallest, m x m matrix
- Use it to compute the rest proximities, on the fly

H. Tong, S. Papadimitriou, P.S. Yu & C. Faloutsos. *Proximity Tracking on Time-Evolving Bipartite Graphs.* SDM 2008.

BB_Lin: Examples

Dataset	Off-Line Cost	On-Line Cost
DBLP	a few minutes	frac. of sec.
NetFlix	1.5 hours	<0.01 sec.

400k authors
x 3.5k conf.s

2.7m user
x 18k movies

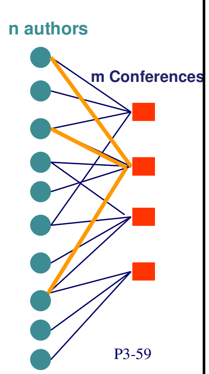
Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; **tracking**
- Conclusions

Problem: update

E' edges changed

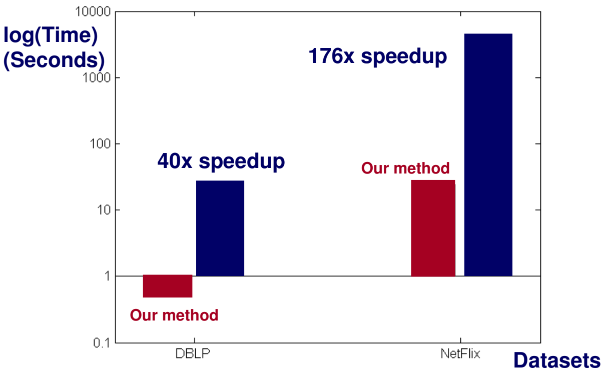
Involves n' authors, m' confs.



Solution:

- Use Sherman-Morrison to quickly update the inverse matrix

Fast-Single-Update



pTrack: Philip S. Yu's Top-5 conferences up to each year

ICDE ICDCS SIGMETRICS PDIS VLDB	CIKM ICDCS ICDE SIGMETRICS ICMCS	KDD SIGMOD ICDM CIKM ICDCS	ICDM KDD ICDE SDM VLDB
1992	1997	2002	2007

DBLP: (Au. x Conf.)
- 400k aus,
- 3.5k confs
- 20 yrs

pTrack: Philip S. Yu's Top-5 conferences up to each year

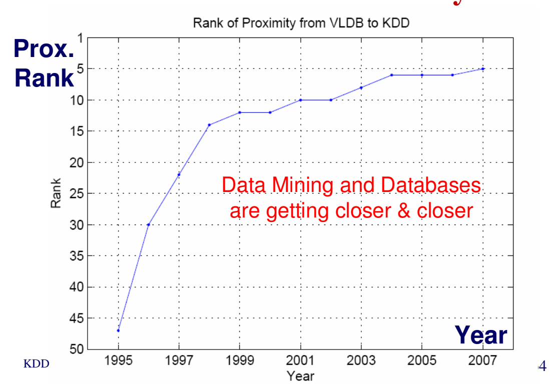
ICDE ICDCS SIGMETRICS PDIS VLDB	CIKM ICDCS ICDE SIGMETRICS ICMCS	KDD SIGMOD ICDM CIKM ICDCS	ICDM KDD ICDE SDM VLDB
1992	1997	2002	2007

Databases
Performance
Distributed Sys.

DBLP: (Au. x Conf.)
- 400k aus,
- 3.5k confs
- 20 yrs

Databases
Data Mining

KDD's Rank wrt. VLDB over years



cTrack:10 most influential authors in NIPS community up to each year

T. Sejnowski

1987	1989	1991	1993	1995	1997	1999
'Abbott_L'	'Bower_J'	'Hinton_G'	'Sejnowski_T'	'Sejnowski_T'	'Sejnowski_T'	'Sejnowski_T'
'Burr_D'	'Hinton_G'	'Koch_C'	'Koch_C'	'Jordan_M'	'Jordan_M'	'Koch_C'
'Denker_J'	'Tesauro_G'	'Bower_J'	'Hinton_G'	'Hinton_G'	'Koch_C'	'Jordan_M'
'Scofield_C	'Denker_J'	'Sejnowski_T'	'Mozer_M'	'Koch_C'	'Hinton_G'	'Hinton_G'
'Bower_J'	'Mead_C'	'LeCun_Y'	'LeCun_Y'	'Mozer_M'	'Mozer_M'	'Mozer_M'
'Brown_N'	'Tenorio_M'	'Mozer_M'	'Denker_J'	'Bengio_Y'	'Dayan_P'	'Dayan_P'
'Carley_L'	'Sejnowski_T'	'Denker_J'	'Bower_J'	'Lippmann_R'	'Bengio_Y'	'Singh_S'
'Chou_P'	'Lippmann_R'	'Waibel_A'	'Kawato_M'	'LeCun_Y'	'Barto_A'	'Bengio_Y'
'Chover_J'	'Touretzky_D'	'Moody_J'	'Waibel_A'	'Waibel_A'	'Tresp_V'	'Tresp_V'
'Eeckman_F'	'Koch_C'	'Lippmann_R'	'Simard_P'	'Simard_P'	'Moody_J'	'Moody_J'

M. Jordan

Author-paper bipartite graph from NIPS 1987-1999.
3k. 1740 papers, 2037 authors, spreading over 13 years

Conclusions - Take-home messages

- **Proximity Definitions**
  - RWR $\vec{r}_i = c\tilde{W}\vec{r}_i + (1-c)\vec{e}_i$
  - and a lot of variants
- **Computation**
  - Sherman–Morrison Lemma
  - Fast Incremental Computation
- **Applications**
  - Recommendations; auto-captioning; tracking
  - Center-piece Subgraphs (next)
  - E-mail management; anomaly detection, …

References

- L. Page, S. Brin, R. Motwani, & T. Winograd. (1998), The PageRank Citation Ranking: Bringing Order to the Web, Technical report, Stanford Library.
- T.H. Haveliwala. (2002) Topic-Sensitive PageRank. In WWW, 517-526, 2002
- J.Y. Pan, H.J. Yang, C. Faloutsos & P. Duygulu. (2004) Automatic multimedia cross-modal correlation discovery. In KDD, 653-658, 2004.

References

- C. Faloutsos, K. S. McCurley & A. Tomkins. (2002) Fast discovery of connection subgraphs. In KDD, 118-127, 2004.
- J. Sun, H. Qu, D. Chakrabarti & C. Faloutsos. (2005) Neighborhood Formation and Anomaly Detection in Bipartite Graphs. In ICDM, 418-425, 2005.
- W. Cohen. (2007) Graph Walks and Graphical Models. Draft.

References

- P. Doyle & J. Snell. (1984) Random walks and electric networks, volume 22. Mathematical Association America, New York.
- Y. Koren, S. C. North, and C. Volinsky. (2006) Measuring and extracting proximity in networks. In KDD, 245–255, 2006.
- A. Agarwal, S. Chakrabarti & S. Aggarwal. (2006) Learning to rank networked entities. In KDD, 14-23, 2006.

References

- S. Chakrabarti. (2007) Dynamic personalized pagerank in entity-relation graphs. In WWW, 571-580, 2007.
- F. Fouss, A. Pirotte, J.-M. Renders, & M. Saerens. (2007) Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation. IEEE Trans. Knowl. Data Eng. 19(3), 355-369 2007.

References

- H. Tong & C. Faloutsos. (2006) Center-piece subgraphs: problem definition and fast solutions. In KDD, 404-413, 2006.
- H. Tong, C. Faloutsos, & J.Y. Pan. (2006) Fast Random Walk with Restart and Its Applications. In ICDM, 613-622, 2006.
- H. Tong, Y. Koren, & C. Faloutsos. (2007) Fast direction-aware proximity for graph mining. In KDD, 747-756, 2007.

References

- H. Tong, B. Gallagher, C. Faloutsos, & T. Eliassi-Rad. (2007) Fast best-effort pattern matching in large attributed graphs. In KDD, 737-746, 2007.
- H. Tong, S. Papadimitriou, P.S. Yu & C. Faloutsos. (2008) Proximity Tracking on Time-Evolving Bipartite Graphs. SDM 2008.

References

- B. Gallagher, H. Tong, T. Eliassi-Rad, C. Faloutsos. Using Ghost Edges for Classification in Sparsely Labeled Networks. KDD 2008
- H. Tong, Y. Sakurai, T. Eliassi-Rad, and C. Faloutsos. Fast Mining of Complex Time-Stamped Events CIKM 08
- H. Tong, H. Qu, and H. Jamjoom. Measuring Proximity on Graphs with Side Information. ICDM 2008

Resources

- www.cs.cmu.edu/~htong/soft.htm
  For software, papers, and ppt of presentations
- www.cs.cmu.edu/~htong/tut/cikm2008/cikm_tutorial.html
  For the CIKM'08 tutorial on graphs and proximity

Again, thanks to **Hanghang TONG** for permission to use his foils in this part

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

Task 4: Center-piece Subgraphs

Faloutsos, Miller and 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
- …
- Conclusions

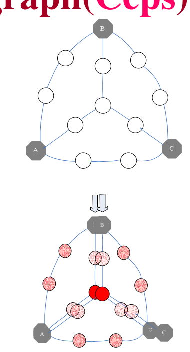
Detailed outline

- Problem definition
- Solution
- Results

H. Tong & C. Faloutsos *Center-piece subgraphs: problem definition and fast solutions*. In KDD, 404-413, 2006.

Center-Piece Subgraph(Ceps)

- Given Q query nodes
- Find Center-piece ($\leq b$)
- Input of Ceps
  - Q Query nodes
  - Budget b
  - k softAnd number
- App.
  - Social Network
  - Law Inforcement
  - Gene Network
  - …



Challenges in Ceps

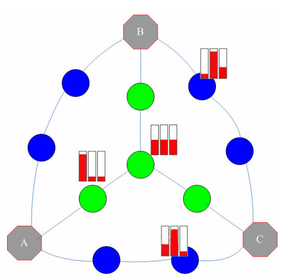
- **Q1: How to measure importance?**
- (Q2: How to extract connection subgraph?
- Q3: How to do it efficiently?)

Challenges in Ceps

- **Q1: How to measure importance?**
- A: “proximity” – but how to combine scores?
- (Q2: How to extract connection subgraph?
- Q3: How to do it efficiently?)

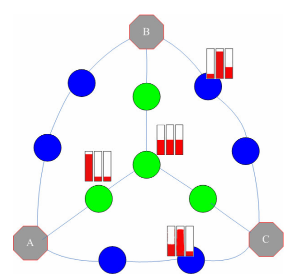
AND: Combine Scores

- Q: How to combine scores?



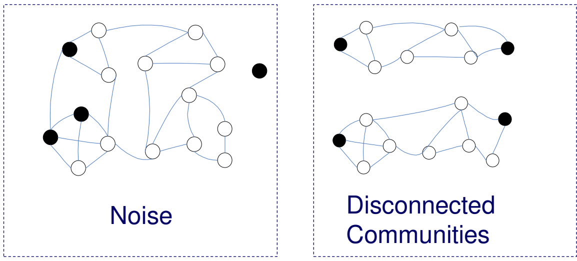
AND: Combine Scores

- Q: How to combine scores?
- A: Multiply
- …= prob. 3 random particles coincide on node *j*



CMU SCS

K_SoftAnd: Relaxation of AND



What if AND query ➔ No Answer?

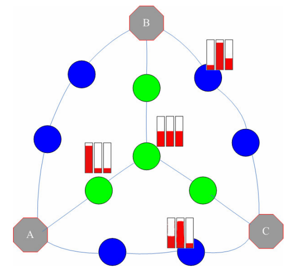
CMU SCS

K_SoftAnd: Combine Scores

Generalization – SoftAND:

We want nodes close to *k* of *Q* (*k*<*Q*) query nodes.

Q: How to do that?



CMU SCS

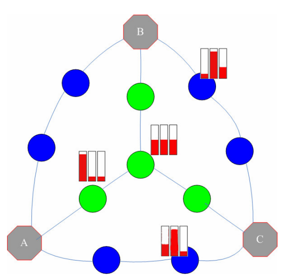
K_SoftAnd: Combine Scores

Generalization – softAND:

We want nodes close to *k* of *Q* (*k*<*Q*) query nodes.

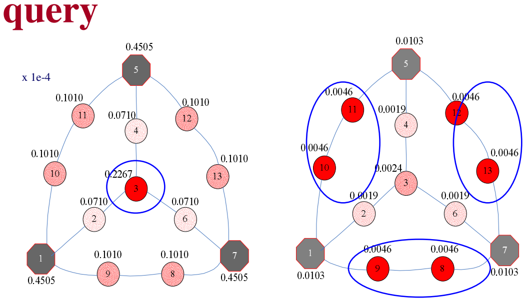
Q: How to do that?

A: Prob(at least *k*-out-of-*Q* will meet each other at j)



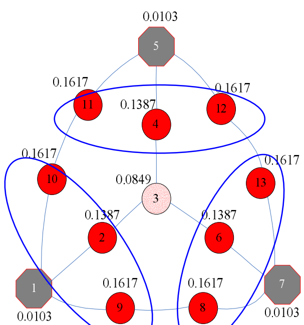
CMU SCS

AND query vs. K_SoftAnd query



And Query 2_SoftAnd Query

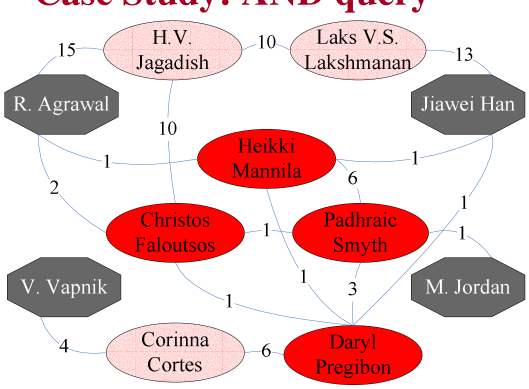
1_SoftAnd query = OR query



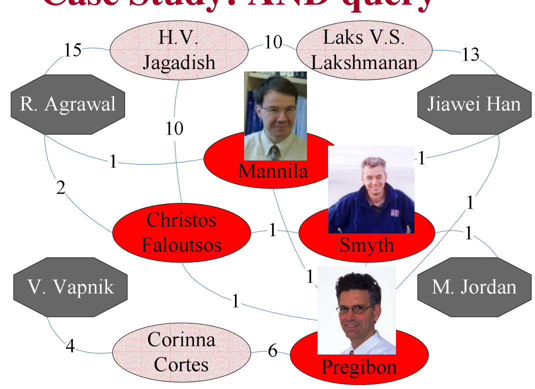
Detailed outline

- Problem definition
- Solution
- Results

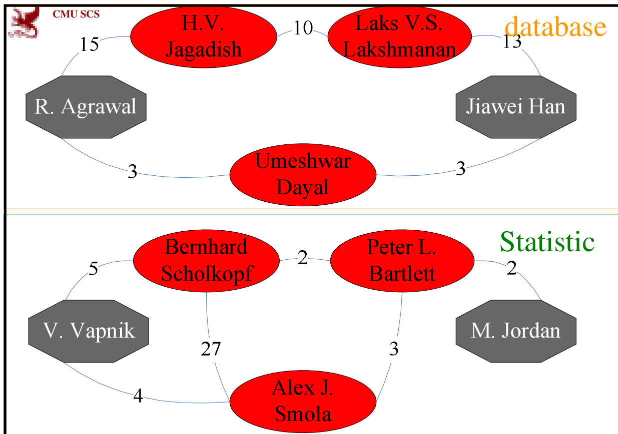
Case Study: AND query



Case Study: AND query



CMU SCS



2_SoftAnd query

Conclusions

Proximity (e.g., w/ RWR) helps answer 'AND' and 'k_softAnd' queries

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

Task 5: Graphs over time & tensors

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

Thanks to

- Tamara Kolda (Sandia)

for the foils on tensor definitions, and on TOPHITS

Detailed outline

- Motivation
- Definitions: PARAFAC and Tucker
- Case study: web mining

Examples of Matrices:
Authors and terms

	data	mining	classif.	tree	...
John	13	11	22	55	...
Peter	5	4	6	7	...
Mary	...	...	...	...	...
Nick	...	...	...	...	...
...	...	...	...	...	...

Motivation: Why tensors?

- Q: what is a tensor?

Motivation: Why tensors?

- A: N-D generalization of matrix:

KDD'09

	data	mining	classif.	tree	...
John	13	11	22	55	...
Peter	5	4	6	7	...
Mary	...	...	...	...	...
Nick	...	...	...	...	...
...	...	...	...	...	...

Motivation: Why tensors?

- A: N-D generalization of matrix:

KDD'07

KDD'08

KDD'09

	data	mining	classif.	tree	...
John	13	11	22	55	...
Peter	5	4	6	7	...
Mary	...	...	...	...	...
Nick	...	...	...	...	...
...	...	...	...	...	...

CMU SCS

Tensors are useful for 3 or more modes

Terminology: 'mode' (or 'aspect'):

Mode#3

data	mining	classif.	tree	...
13	11	22	55	...
5	4	6	7	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

Mode#2

Mode (== aspect) #1

CMU SCS

Notice

- 3rd mode does not need to be time
- we can have more than 3 modes

Dest. port

125

80

				...
13	11	22	55	...
5	4	6	7	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

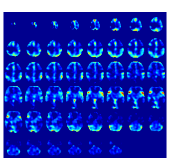
IP source

IP destination

CMU SCS

Notice

- 3rd mode does not need to be time
- we can have more than 3 modes
  - Eg, fFMRI: x,y,z, time, person-id, task-id



From DENLAB, Temple U.
(Prof. V. Megalooikonomou +)

http://denlab.temple.edu/bidms/cgi-bin/browse.cgi

CMU SCS

Motivating Applications

- Why tensors are useful?
  - web mining (TOPHITS)
  - environmental sensors
  - Intrusion detection (src, dst, time, dest-port)
  - Social networks (src, dst, time, type-of-contact)
  - face recognition
  - etc …

Detailed outline

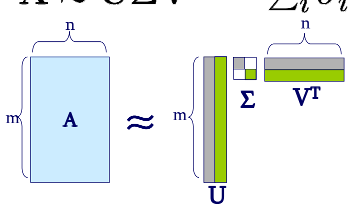
- Motivation
- Definitions: PARAFAC and Tucker
- Case study: web mining

Tensor basics

- Multi-mode extensions of SVD – recall that:

Reminder: SVD

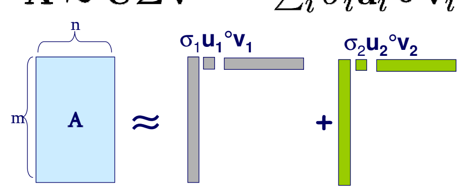
$$\mathbf{A} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i$$



– Best rank-k approximation in L2

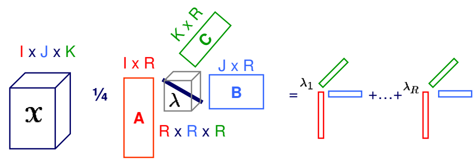
Reminder: SVD

$$\mathbf{A} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i$$



– Best rank-k approximation in L2

Goal: extension to >=3 modes



$$\mathfrak{X} \approx [\![\lambda\,;\mathbf{A},\mathbf{B},\mathbf{C}]\!] = \sum_r \lambda_r\ \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

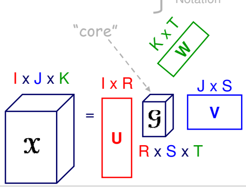
Main points:

- 2 major types of tensor decompositions: PARAFAC and Tucker
- both can be solved with ``alternating least squares'' (ALS)

Specially Structured Tensors

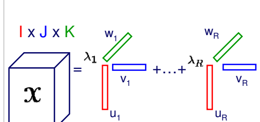
- Tucker Tensor

$$\mathfrak{X} = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$
$$= \sum_r \sum_s \sum_t g_{rst}\ \mathbf{u}_r \circ \mathbf{v}_s \circ \mathbf{w}_t$$
$$\equiv [\![\mathcal{G}\,;\mathbf{U},\mathbf{V},\mathbf{W}]\!] \quad \text{Our Notation}$$

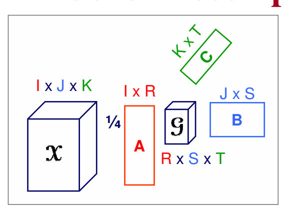


- Kruskal Tensor

$$\mathfrak{X} = \sum_r \lambda_r\ \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r$$
$$\equiv [\![\lambda\,;\mathbf{U},\mathbf{V},\mathbf{W}]\!] \quad \text{Our Notation}$$



Tucker Decomposition - intuition



- author x keyword x conference
- A: author x author-group
- B: keyword x keyword-group
- C: conf. x conf-group
- **G**: how groups relate to each other

Intuition behind core tensor

- 2-d case: co-clustering
- [Dhillon et al. Information-Theoretic Co-clustering, KDD'03]

eg, terms x documents

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

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

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

Tensor tools - summary

- Two main tools
  - PARAFAC
  - Tucker
- Both find row-, column-, tube-groups
  - but in PARAFAC the three groups are identical
- (To solve: Alternating Least Squares)

CMU SCS

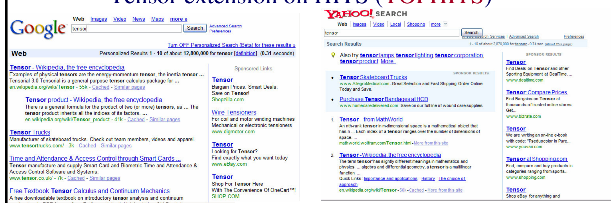
Detailed outline

- Motivation
- Definitions: PARAFAC and Tucker
- Case study: web mining

CMU SCS

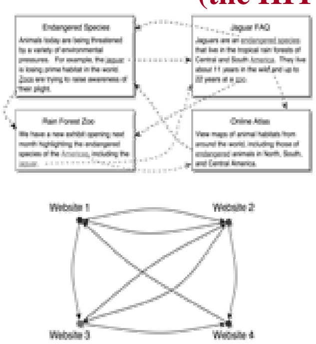
Web graph mining

- How to order the importance of web pages?
  - Kleinberg's algorithm HITS
  - PageRank
  - Tensor extension on HITS (TOPHITS)



CMU SCS

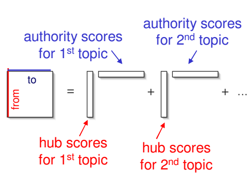
Kleinberg's Hubs and Authorities (the HITS method)



Sparse adjacency matrix and its SVD:

$$x_{ij} = \begin{cases} 1 & \text{if page } i \text{ links to page } j \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{X} \approx \sum_r \sigma_r \, \mathbf{h}_r \circ \mathbf{a}_r$$



Kleinberg, JACM, 1999

CMU SCS

HITS Authorities on Sample Data

1st Principal Factor	
.97	www.ibm.com
.24	www.alphawo
.08	www-128.ibm
.05	www.develop
.02	www.researc
.01	www.redbook
.01	news.com.co

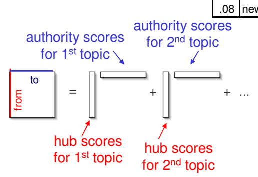
2nd Principal Factor	
.99	www.lehigh.edu
.11	www2.lehigh.edu
.06	www.lehighal
.06	www.lehighs
.02	www.bethleh
.02	www.adobe.c
.02	lewisweb.cc.
.02	www.leo.lehi
.02	www.distanc
.02	fp1.cc.lehigh

3rd Principal Factor	
.75	java.sun.com
.38	www.sun.com
.36	developers.sun.
.24	see.sun.com
.16	www.samag.co
.13	docs.sun.com
.12	blogs.sun.com
.08	sunsolve.sun.co
.08	www.sun-catalo
.08	news.com.com

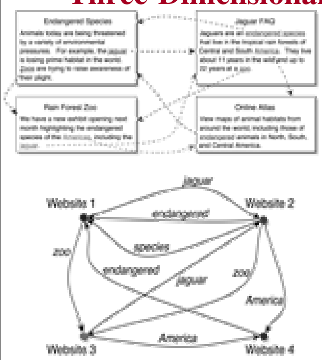
4th Principal Factor	
.60	www.pueblo.gsa.gov
.45	www.whitehouse.gov
.35	www.irs.gov
.31	travel.state
.22	www.gsa.g
.20	www.ssa.g
.16	www.censu
.14	www.govbe
.13	www.kids.g
.13	www.usdoj.

6th Principal Factor	
.97	mathpost.asu.edu
.18	math.la.asu.edu
.17	www.asu.edu
.04	www.act.org
.03	www.eas.asu.edu
.02	archives.math.utk.edu
.02	www.geom.uiuc.edu
.02	www.fulton.asu.edu
.02	www.amstat.org
.02	www.maa.org

We started our crawl from http://www-neos.mcs.anl.gov/neos, and crawled 4700 pages, resulting in 560 cross-linked hosts.

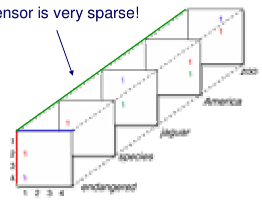


Three-Dimensional View of the Web



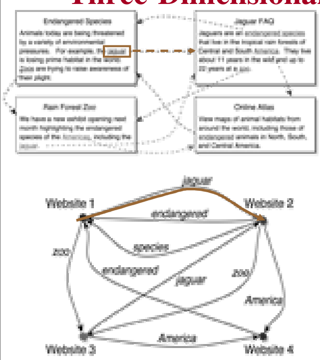
$$x_{ijk} = \begin{cases} 1 & \text{if page } i \to \text{page } j \text{ with term } k \\ 0 & \text{otherwise} \end{cases}$$

Observe that this tensor is very sparse!



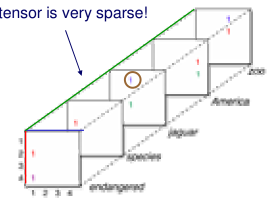
Kolda, Bader, Kenny, ICDM05

Three-Dimensional View of the Web



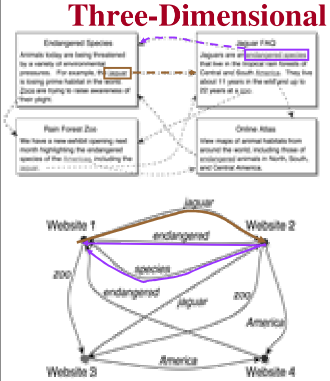
$$x_{ijk} = \begin{cases} 1 & \text{if page } i \to \text{page } j \text{ with term } k \\ 0 & \text{otherwise} \end{cases}$$

Observe that this tensor is very sparse!



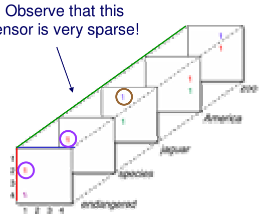
Kolda, Bader, Kenny, ICDM05

Three-Dimensional View of the Web



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

Observe that this tensor is very sparse!

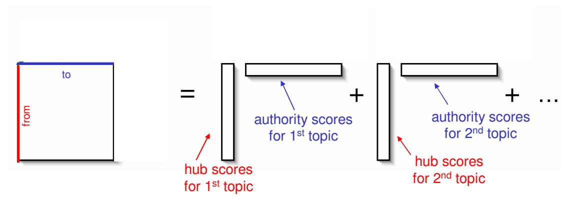


Kolda, Bader, Kenny, ICDM05

Topical HITS (TOPHITS)

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

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

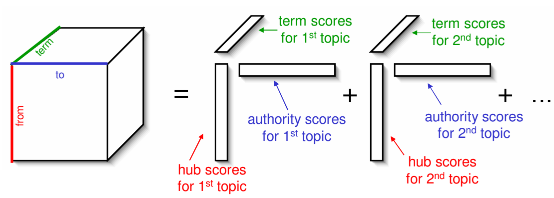


CMU SCS

Topical HITS (TOPHITS)

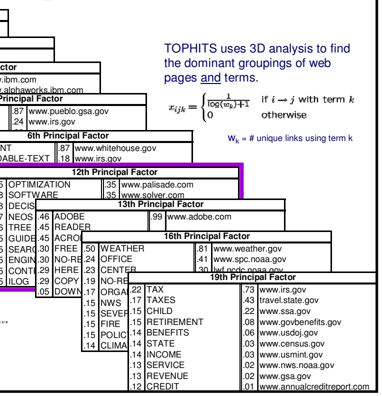
Main Idea: Extend the idea behind the HITS model to incorporate term (i.e., topical) information.

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



CMU SCS

TOPHITS Terms & Authorities on Sample Data

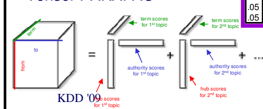


TOPHITS uses 3D analysis to find the dominant groupings of web pages and terms.

$$x_{ijk} = \begin{cases} \frac{1}{\log(w_k)+1} & \text{if } i \to j \text{ with term } k \\ 0 & \text{otherwise} \end{cases}$$

w_k = # unique links using term k

Tensor PARAFAC



CMU SCS

Conclusions

- Real data are often in high dimensions with multiple aspects (modes)
- Tensors provide elegant theory and algorithms
  - PARAFAC and Tucker: discover groups

CMU SCS

References

- T. G. Kolda, B. W. Bader and J. P. Kenny. *Higher-Order Web Link Analysis Using Multilinear Algebra*. In: ICDM 2005, Pages 242-249, November 2005.
- Jimeng Sun, Spiros Papadimitriou, Philip Yu. *Window-based Tensor Analysis on High-dimensional and Multi-aspect Streams*, Proc. of the Int. Conf. on Data Mining (ICDM), Hong Kong, China, Dec 2006

Resources

- See tutorial on tensors, KDD'07 (w/ Tamara Kolda and Jimeng Sun):

www.cs.cmu.edu/~christos/TALKS/KDD-07-tutorial

Tensor tools - resources

- Toolbox: from Tamara Kolda:

csmr.ca.sandia.gov/~tgkolda/TensorToolbox

- T. G. Kolda and B. W. Bader. ***Tensor Decompositions and Applications***. SIAM Review, Volume 51, Number 3, September 2009 csmr.ca.sandia.gov/~tgkolda/pubs/bibtgkfiles/TensorReview-preprint.pdf

- T. Kolda and J. Sun: Scalable Tensor Decomposition for Multi-Aspect Data Mining (ICDM 2008)

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

Task 6: Virus/Influence Propagation

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

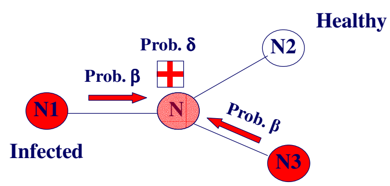
- Epidemic threshold
  - Problem definition
  - Analysis
  - Experiments
- Fraud detection in e-bay

Virus propagation

- How do viruses/rumors propagate?
- Blog influence?
- Will a flu-like virus linger, or will it become extinct soon?

The model: SIS

- 'Flu' like: Susceptible-Infected-Susceptible
- Virus 'strength' s= β/δ



Epidemic threshold τ

of a graph: the value of τ, such that

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

an epidemic can not happen

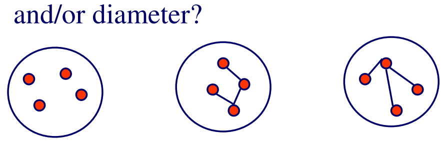
Thus,

- given a graph
- compute its epidemic threshold

Epidemic threshold τ

What should τ depend on?

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



Epidemic threshold

- [Theorem 1] We have no epidemic, if

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

Epidemic threshold

- [Theorem 1] We have no epidemic (*), if

recovery prob.

epidemic threshold

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

attack prob.

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

Proof: [Wang+03]

(*) under mild, conditional-independence assumptions

Beginning of proof

details

Healthy @ t+1:
- (healthy or healed)
- and not attacked @ t

Let: p(i , t) = Prob node i is sick @ t+1

$$1 - p(i, t+1) = (1 - p(i, t) + p(i, t) * \delta) * \Pi_j (1 - \beta\, aji * p(j, t))$$

Below threshold, if the above *non-linear dynamical system* above is 'stable' (eigenvalue of Hessian < 1)

Epidemic threshold for various networks

Formula includes older results as special cases:

- Homogeneous networks [Kephart+White]
  - $\lambda_{1,A} = <k>; \tau = 1/<k>$ (<k> : avg degree)
- Star networks (d = degree of center)
  - $\lambda_{1,A} = sqrt(d); \tau = 1/ sqrt(d)$
- Infinite power-law networks
  - $\lambda_{1,A} = \infty; \tau = 0$; [Barabasi]

Epidemic threshold

- [Theorem 2] Below the epidemic threshold, the epidemic dies out exponentially

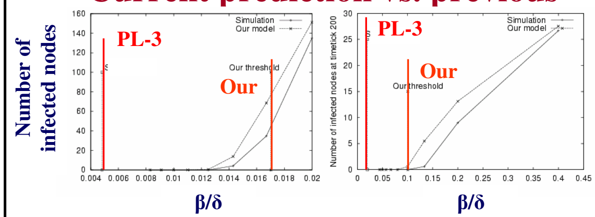
CMU SCS

Detailed outline

- Epidemic threshold
  – Problem definition
  – Analysis
  – Experiments
- Fraud detection in e-bay

CMU SCS

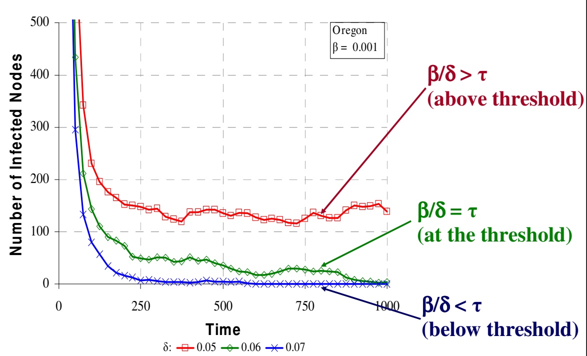
Current prediction vs. previous



The formula's predictions are more accurate

CMU SCS

Experiments (Oregon)

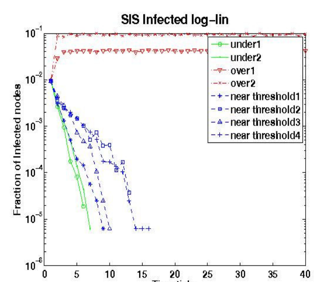


CMU SCS

SIS simulation - # infected nodes vs time

Log - Lin

#inf.
(log scale)



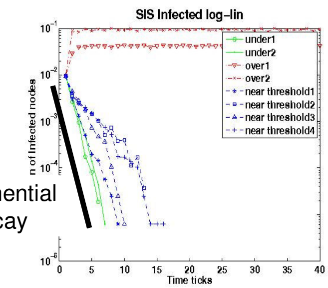
Time (linear scale)

above
at
below

SIS simulation - # infected nodes vs time

Log - Lin

#inf.
(log scale)



Exponential decay

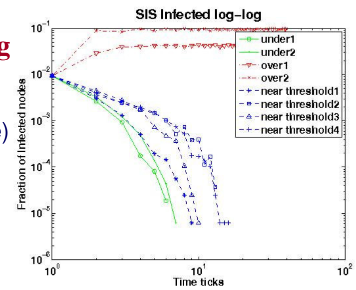
Time (linear scale)

above
at
below

SIS simulation - # infected nodes vs time

Log - Log

#inf.
(log scale)



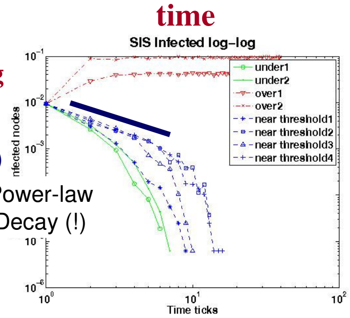
Time (log scale)

above
at
below

SIS simulation - # infected nodes vs time

Log - Log

#inf.
(log scale)



Power-law Decay (!)

Time (log scale)

above
at
below

Detailed outline

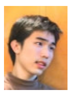
extra

- Epidemic threshold
- Fraud detection in e-bay

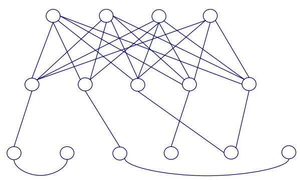
CMU SCS

E-bay Fraud detection

extra



w/ Polo Chau &
Shashank Pandit, CMU



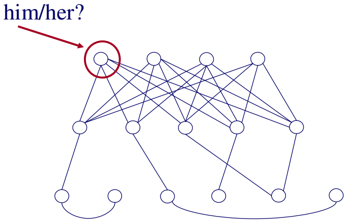
NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks, S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos *(WWW'07)*, pp. 201-210

CMU SCS

E-bay Fraud detection

extra

- lines: positive feedbacks
- would you buy from him/her?

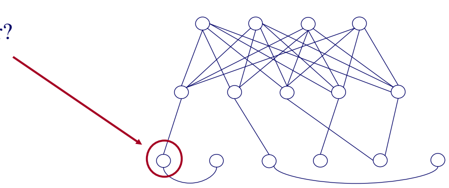


CMU SCS

E-bay Fraud detection

extra

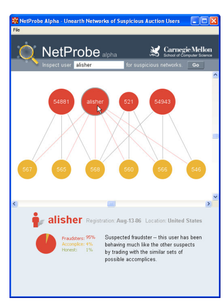
- lines: positive feedbacks
- would you buy from him/her?
- or him/her?



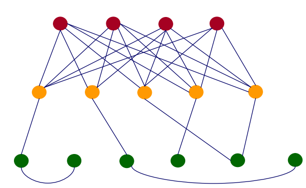
CMU SCS

E-bay Fraud detection - NetProbe

extra



Belief Propagation gives:



CMU SCS

Conclusions

- $\lambda_{1,A}$: Eigenvalue of adjacency matrix determines the survival of a flu-like virus
  - It gives a measure of how well connected is the graph (~ # paths – see Task 7, later)
  - May guide immunization policies
- [Belief Propagation: a powerful algo]

CMU SCS

References

- D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos, *Epidemic Thresholds in Real Networks*, in ACM TISSEC, 10(4), 2008
- Ganesh, A., Massoulie, L., and Towsley, D., 2005. The effect of network topology on the spread of epidemics. In *INFOCOM*.

CMU SCS

References (cont'd)

- Hethcote, H. W. 2000. The mathematics of infectious diseases. *SIAM Review 42*, 599–653.
- Hethcote, H. W. AND Yorke, J. A. 1984. *Gonorrhea Transmission Dynamics and Control*. Vol. 56. Springer. Lecture Notes in Biomathematics.

CMU SCS

References (cont'd)

- Y. Wang, D. Chakrabarti, C. Wang and C. Faloutsos, *Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint*, in SRDS 2003 (pages 25-34), Florence, Italy

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

Christos Faloutsos
Gary Miller
Charalampos (Babis) Tsourakakis
CMU

Outline

- **Reminders**
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length k
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Matrix Representations of G(V,E)

Associate a matrix to a graph:

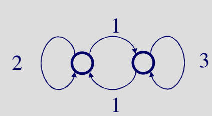
- Adjacency matrix
- Laplacian
- Normalized Laplacian

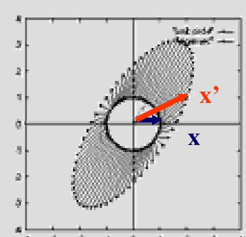
$\left.\begin{array}{l}\text{Adjacency matrix}\\ \text{Laplacian}\end{array}\right\}$ Main focus

Recall: Intuition

- **A** as vector transformation

$$\underset{\mathbf{x'}}{\begin{bmatrix} 2 \\ 1 \end{bmatrix}} = \underset{\mathbf{A}}{\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}} \underset{\mathbf{x}}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}$$

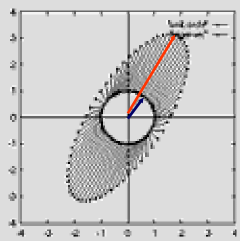




Intuition

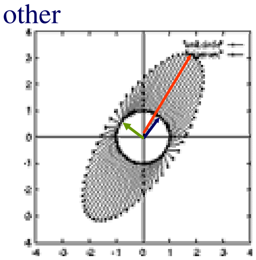
- By defn., eigenvectors remain parallel to themselves (‘**fixed points**’)

$$\begin{array}{cccc} \lambda_1 & \mathbf{v_1} & \mathbf{A} & \mathbf{v_1} \\ 3.62\ * & \begin{bmatrix} 0.52 \\ 0.85 \end{bmatrix} = & \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} & \begin{bmatrix} 0.52 \\ 0.85 \end{bmatrix} \end{array}$$



Intuition

- By defn., eigenvectors remain parallel to themselves (‘**fixed points**’)
- And orthogonal to each other



Keep in mind!

- For the rest of slides we will be talking for square nxn matrices

$$M = \begin{bmatrix} m_{11} & & m_{1n} \\ & \ldots & \\ m_{n1} & & m_{nn} \end{bmatrix}$$

and symmetric ones, i.e,

$$M = M^T$$

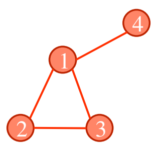
Outline

- Reminders
- **Adjacency matrix**
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length k
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Adjacency matrix

Undirected

$$A_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

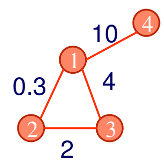


$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Adjacency matrix

Undirected Weighted

$$A_{uv} = \begin{cases} w_{uv} & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

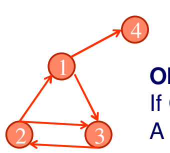


$$A = \begin{pmatrix} 0 & 0.3 & 4 & 10 \\ 0.3 & 0 & 2 & 0 \\ 4 & 2 & 0 & 0 \\ 10 & 0 & 0 & 0 \end{pmatrix}$$

Adjacency matrix

Directed

$$A_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$



Observation
If G is undirected,
$A = A^T$

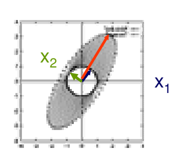
$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Spectral Theorem

Theorem [Spectral Theorem]

- If $M=M^T$, then

$$M = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \begin{bmatrix} — x_1^T — \\ \cdots \\ — x_n^T — \end{bmatrix} = \lambda_1 x_1 x_1^T + \cdots + \lambda_n x_n x_n^T$$



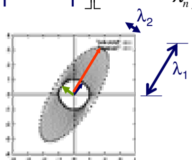
Reminder 1:
x_i, x_j orthogonal

Spectral Theorem

Theorem [Spectral Theorem]

- If $M=M^T$, then

$$M = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \begin{bmatrix} — x_1^T — \\ \cdots \\ — x_n^T — \end{bmatrix} = \lambda_1 x_1 x_1^T + \cdots + \lambda_n x_n x_n^T$$



Reminder 2:
x_i i-th principal axis
λ_i length of i-th principal axis

Outline

- Reminders
- Adjacency matrix
  - **Intuition behind eigenvectors**: Eg., Bipartite Graphs
  - Walks of length k
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

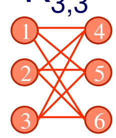
Eigenvectors:

- Give groups
- Specifically for bi-partite graphs, we get each of the two sets of nodes
- Details:

Bipartite Graphs

Any graph with no cycles of odd length is bipartite

$K_{3,3}$



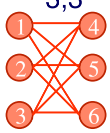
$$A = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$$

Q1: Can we check if a graph is bipartite via its spectrum?
Q2: Can we get the partition of the vertices in the two sets of nodes?

Bipartite Graphs

Adjacency matrix $A = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$

$K_{3,3}$



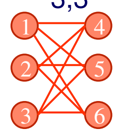
where $B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Eigenvalues: **Λ**=[3,-3,0,0,0,0]

Bipartite Graphs

Adjacency matrix $A = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$

$K_{3,3}$

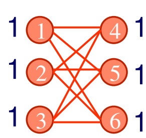


where $B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

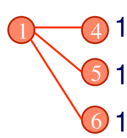
Why $\lambda_1 = -\lambda_2 = 3$?

Recall: A**x**=λ**x**, (λ,**x**) eigenvalue-eigenvector

Bipartite Graphs



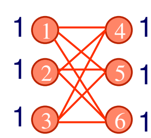
3=3x1



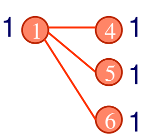
$$\lambda_1 = 3, u_1 = \mathbf{1} = [1,\ 1,\ 1,\ 1,\ 1,\ 1]^T$$

Value @ each node: eg., enthusiasm about a product

Bipartite Graphs



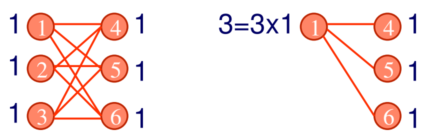
3=3x1



$$\lambda_1 = 3, u_1 = \mathbf{1} = [1,\ 1,\ 1,\ 1,\ 1,\ 1]^T$$

1-vector remains unchanged (just grows by '3' = λ_1)

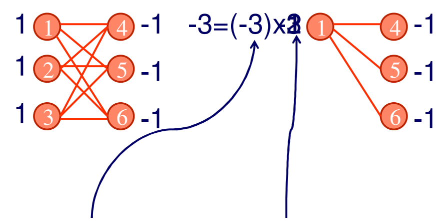
Bipartite Graphs



$$\lambda_1 = 3, u_1 = \mathbf{1} = [1,\ 1,\ 1,\ 1,\ 1,\ 1]^T$$

Which other vector remains unchanged?

Bipartite Graphs



$$\lambda_2 = -3, u_2 = \mathbf{1} = [1,\ 1,\ 1,\ \text{-}1,\ \text{-}1,\ \text{-}1]^T$$

Bipartite Graphs

- Observation
  u_2 gives the partition of the nodes in the two sets S, V-S!

1 2 3 4 5 6

$$\lambda_2 = -3, u_2 = \mathbf{1} = [\underbrace{1,\ 1,\ 1,}_{S}\ \underbrace{\text{-}1,\ \text{-}1,\ \text{-}1}_{V\text{-}S}]^T$$

Question: Were we just "lucky"? Answer: No

Theorem: $\lambda_2 = -\lambda_1$ iff G bipartite. u_2 gives the partition.

Outline

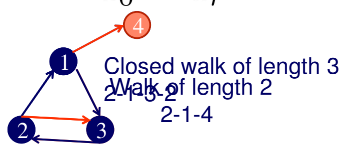
- Reminders
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - **Walks of length *k***
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Walks

- A walk of length r in a directed graph:

$$u_0 \rightarrow u_1 \rightarrow \ldots \rightarrow u_r$$

  where a node can be used more than once.
- Closed walk when: $u_0 = u_r$



Walks

Theorem: G(V,E) directed graph, adjacency matrix A. The number of walks from node u to node v in G with length r is $(A^r)_{uv}$

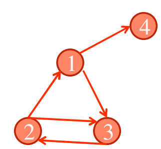
Proof: Induction on k. See Doyle-Snell, p.165

Walks

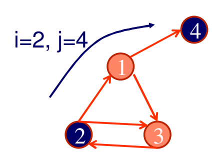
Theorem: G(V,E) directed graph, adjacency matrix A. The number of walks from node u to node v in G with length r is $(A^r)_{uv}$

$$A = \begin{bmatrix} \overset{(i,j)}{a_{ij}^1} \end{bmatrix}, \quad A^2 = \begin{bmatrix} \overset{(i,\, i_1),(i_1,j)}{a_{ij}^2} \end{bmatrix}, \ldots, A^r = \begin{bmatrix} \overset{(i,i_1),..,(i_{r-1},j)}{a_{ij}^r} \end{bmatrix}$$

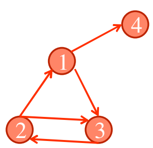
Walks



$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

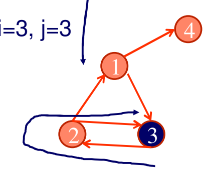


Walks

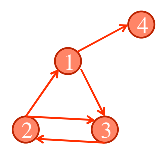


$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

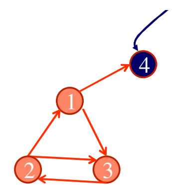
i=3, j=3



Walks



$$A^6 = \begin{pmatrix} 1 & 1 & 2 & 1 \\ 2 & 2 & 3 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

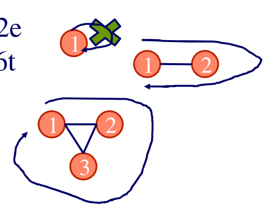


Always 0,
node 4 is a sink

Walks

Corollary: If A is the adjacency matrix of undirected G(V,E) (no self loops), e edges and t triangles. Then the following hold:

a) trace(A) = 0
b) trace(A^2) = 2e
c) trace(A^3) = 6t



Walks

Corollary: If A is the adjacency matrix of undirected G(V,E) (no self loops), e edges and t triangles. Then the following hold:

a) trace(A) = 0
b) trace(A^2) = 2e
c) trace(A^3) = 6t

Computing A^r may be expensive!

Remark: virus propagation

The earlier result makes sense now:

- The higher the first eigenvalue, the more paths available ->
- Easier for a virus to survive

Outline

Main upcoming result

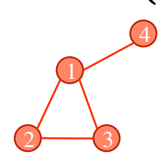
the second eigenvector of the Laplacian (u_2) gives a good cut:

Nodes with positive scores should go to one group

And the rest to the other

Laplacian

$$L_{uv} = \begin{cases} \mathrm{d}_u & \text{if } u = v \\ -1 & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

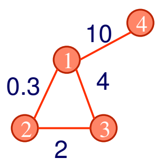


$$\text{L= D-A=} \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

Diagonal matrix, $d_{ii}=d_i$

Weighted Laplacian

$$L_{uv} = \begin{cases} d_u = \sum_v w_{uv} & \text{if } u = v \\ -w_{uv} & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$



$$L = \begin{pmatrix} 14.3 & -0.3 & -4 & -10 \\ -0.3 & 2.3 & -2 & 0 \\ -4 & -2 & 6 & 0 \\ -10 & 0 & 0 & 10 \end{pmatrix}$$

Outline

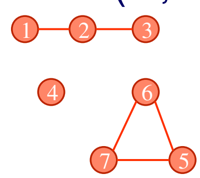
- **Reminders**
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length k
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Connected Components

- **Lemma**: Let G be a graph with n vertices and c connected components. If L is the Laplacian of G, then rank(L)= *n-c*.
- **Proof**: see p.279, Godsil-Royle

Connected Components

G(V,E)



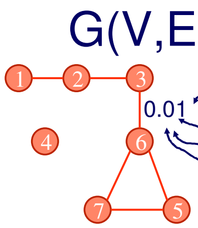
$$L=\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

$$\text{eig(L)}=\begin{pmatrix} \overbrace{0 \quad 0 \quad 0}^{\#\text{zeros} = \#\text{components}} & 1 & 3 & 3 & 3 \end{pmatrix}$$

CMU SCS

Connected Components

G(V,E)



$$L=\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1.01 & 0 & 0 & -0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & -0.1 & 0 & -1 & 2.01 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

#zeros = #components

Indicates a "good cut"

$$\text{eig}(L)=\begin{pmatrix} 0 & 0 & 0.0066 & 1.005 & 3 & 3 & 3.0084 \end{pmatrix}$$

CMU SCS

Outline

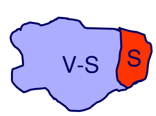
- Reminders
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length *k*
- Laplacian
  - Connected Components
  - **Intuition: Adjacency vs. Laplacian**
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

CMU SCS

Adjacency vs. Laplacian Intuition

details

Let **x** be an indicator vector:



$x_i = 1$, if $i \in S$

$x_i = 0$, if $i \notin S$

Consider now y=Lx

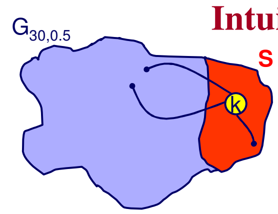
k-th coordinate

$$y_k = (Lx)_k = d_k x_k - \sum_{j:(j,k)\in E(G)} x_j$$

CMU SCS

Adjacency vs. Laplacian Intuition

details

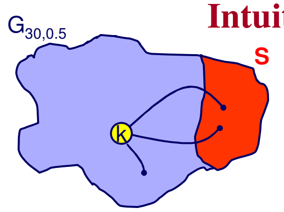


Consider now y=Lx

$$y_k > 0$$

$$y_k = (Lx)_k = d_k x_k - \sum_{j:(j,k)\in E(G)} x_j$$

Adjacency vs. Laplacian Intuition

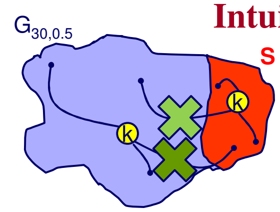


Consider now y=Lx

$$y_k < 0$$

$$y_k = (Lx)_k = d_k x_k - \sum_{j:(j,k) \in E(G)} x_j$$

Adjacency vs. Laplacian Intuition



Consider now y=Lx

$$y_k = 0$$

$$y_k = (Lx)_k = d_k x_k - \sum_{j:(j,k) \in E(G)} x_j$$

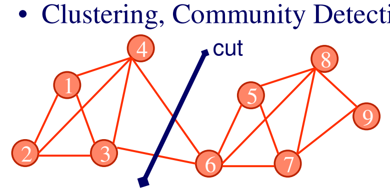
Laplacian: connectivity,
Adjacency: #paths

Outline

- Reminders
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length *k*
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - **Sparsest Cut and Cheeger inequality**:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Why Sparse Cuts?

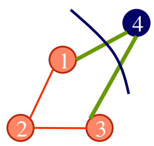
- Clustering, Community Detection



- And more: Telephone Network Design, VLSI layout, Sparse Gaussian Elimination, Parallel Computation

Quality of a Cut

- Isoperimetric number φ of a cut *S:*



#edges across

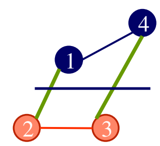
#nodes in smallest partition

$$\phi(S) = \frac{e(S, V-S)}{\min(|S|, |V-S|)}$$

$$\phi(\{4\}) = \frac{2}{min(1,3)} = 2$$

Quality of a Cut

- Isoperimetric number φ of a **graph** = score of best cut:



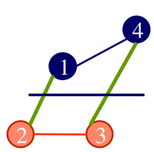
$$\phi(G) = \min_{S \subseteq V} \phi(S)$$

$$\phi(\{1,4\}) = \frac{2}{min(2,2)} = 1$$

and thus $\phi(G) = 1$

Quality of a Cut

- Isoperimetric number φ of a **graph** = score of best cut:



Best cut: hard to find

BUT: Cheeger's inequality gives bounds

λ_2: Plays major role

Let's see the intuition behind λ_2

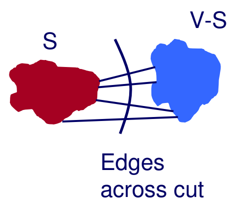
Laplacian and cuts - overview

- A cut corresponds to an indicator vector (ie., 0/1 scores to each node)
- Relaxing the 0/1 scores to real numbers, gives eventually an alternative definition of the eigenvalues and eigenvectors

Why λ_2?

Characteristic Vector **x**

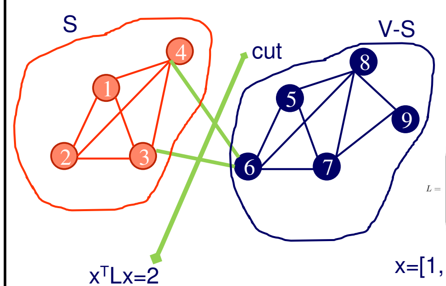
- $x_i = 1, \text{if } i \in S$
- $x_i = 0, \text{if } i \notin S$



Then:

$$x^T L x = \sum_{(i,j)\in E(G)} (x_i - x_j)^2 = e(S, V-S)$$

Why λ_2?



$$L = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 4 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & -1 & -1 & -1 & 0 \\ 0 & 0 & -1 & -1 & -1 & 5 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

$x=[1,1,1,1,0,0,0,0,0]^T$

details

Why λ_2?

$$r(S) = \frac{e(S,V-S)}{|S||V-S|} \Rightarrow \frac{\phi(S)}{n} \leq r(S) \leq \frac{\phi(S)}{\frac{n}{2}}$$

Ratio cut

Sparsest ratio cut $\quad r(G) = \min_{S\subset V} r(S) = \min_{x\in\{0,1\}^n} \frac{1}{n}\frac{x^T L x}{x^T x}$

NP-hard

Relax the constraint: $x \in \{0,1\}^n \rightarrow x \in \mathbb{R}^n$

Normalize: $\sum_i x_i = 0$

$\Rightarrow$?

details

Why λ_2?

Sparsest ratio cut $\quad r(G) = \min_{S\subset V} r(S) = \min_{x\in\{0,1\}^n} \frac{1}{n}\frac{x^T L x}{x^T x}$

NP-hard

Relax the constraint: $x \in \{0,1\}^n \rightarrow x \in \mathbb{R}^n$

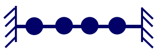
Normalize: $\sum_i x_i = 0$

$\Rightarrow \lambda_2$

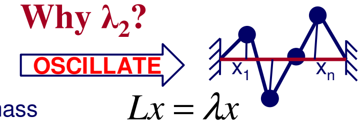
because of the Courant-Fisher theorem (applied to *L*)

$$\lambda_2 = \min_{\sum_i u_i = 0, u\neq 0} \frac{u^T L u}{u^T u} = \min_{\sum_i u_i = 0, u \neq 0} \frac{\sum_{(i,j)\in E(G)} (u_i - u_j)^2}{\sum_i u_i^2}$$

Why λ_2?



Each ball 1 unit of mass

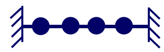


$$Lx = \lambda x$$

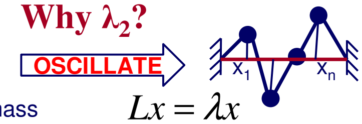
Dfn of eigenvector

Matrix viewpoint:

Why λ_2?



Each ball 1 unit of mass



$$Lx = \lambda x$$

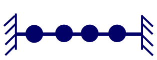
Force due to neighbors

displacement

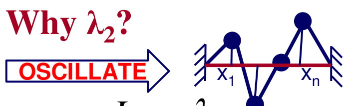
Hooke's constant

Physics viewpoint:

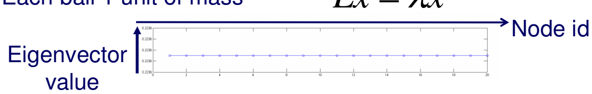
Why λ_2?



Each ball 1 unit of mass

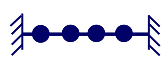


$$Lx = \lambda x$$

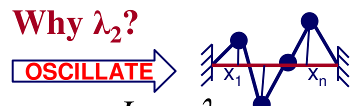


For the first eigenvector:
All nodes: same displacement (= value)

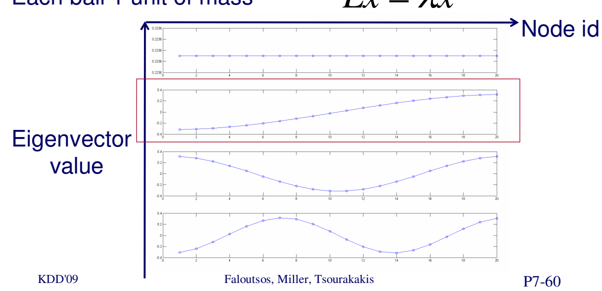
Why λ_2?



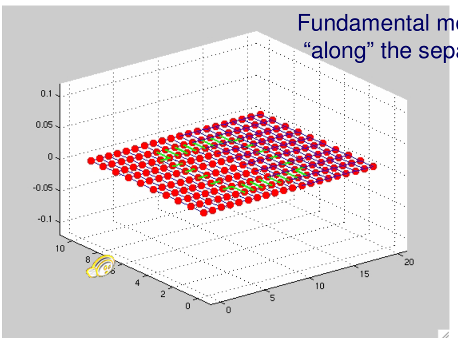
Each ball 1 unit of mass



$$Lx = \lambda x$$



Why λ_2?



Fundamental mode of vibration:
"along" the separator

Cheeger Inequality

Score of best cut
(**hard** to compute)

$$\frac{\phi^2}{2d_{max}} \leq \lambda_2 \leq 2\phi(G)$$

Max degree

2nd smallest eigenvalue
(**easy** to compute)

Cheeger Inequality and graph partitioning heuristic:

$$\frac{\phi^2}{2d_{max}} \leq \lambda_2 \leq 2\phi(G)$$

- Step 1: Sort vertices in non-decreasing order according to their score of the second eigenvector
- Step 2: Decide where to cut.
  - Bisection
  - **Best ratio cut**

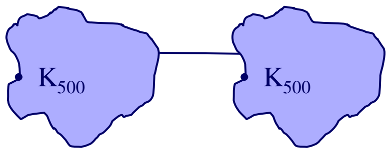
Two common heuristics

Outline

- Reminders
- Adjacency matrix
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Sparsest Cut and Cheeger inequality:
    - Derivation, intuition
    - **Example**
- Normalized Laplacian

CMU SCS

Example: Spectral Partitioning



dumbbell graph

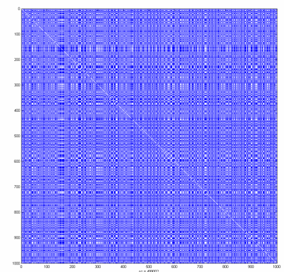
In social network analysis, such clusters are called communities

```
A = zeros(1000);
A(1:500,1:500)=ones(500)-eye(500);
A(501:1000,501:1000)= ones(500)-eye(500);
myrandperm = randperm(1000);
B = A(myrandperm,myrandperm);
```


CMU SCS

Example: Spectral Partitioning

- This is how adjacency matrix of B looks



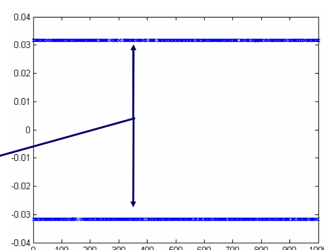
spy(B)

CMU SCS

Example: Spectral Partitioning

- This is how the 2nd eigenvector of B looks like.

```
L = diag(sum(B))-B;
[u v] = eigs(L,2,'SM');
plot(u(:,1),'x')
```



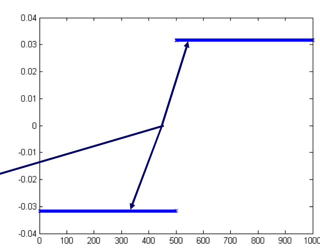
Not so much information yet…

CMU SCS

Example: Spectral Partitioning

- This is how the 2nd eigenvector looks if we sort it.

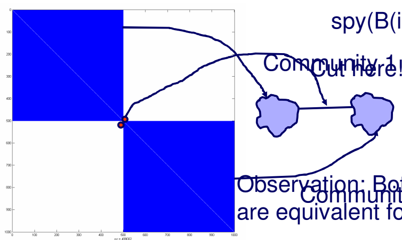
```
[ign ind] = sort(u(:,1));
plot(u(ind),'x')
```



But now we see the two communities!

Example: Spectral Partitioning

- This is how adjacency matrix of B looks now

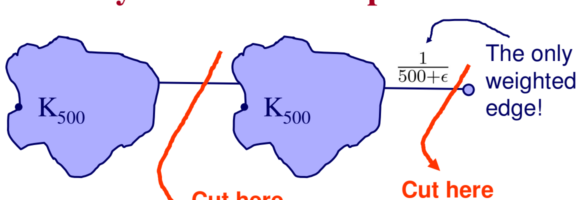


Observation: Both heuristics are equivalent for the dumbbell

Outline

- Reminders
- Adjacency matrix
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Sparsest Cut and Cheeger inequality:
- ➡ Normalized Laplacian

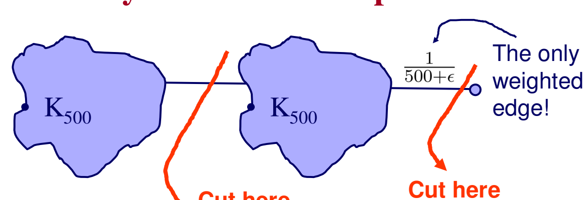
Why Normalized Laplacian



φ= $\frac{1}{500}$ > φ= $\frac{1}{500+\epsilon}$

So, φ is not good here…

Why Normalized Laplacian



Optimize Cheeger constant h(G), balanced cuts

φ= $\frac{1}{500}$ > φ= $\frac{1}{500+\epsilon}$

$$\boxed{h_G = \min_S h_G(S)}$$ where

$$h(S) = \frac{e(S, V-S)}{\min(vol(S), vol(V-S))}$$

$$vol(S) = \sum_{v \in S} d_v$$

Extensions

- Normalized Laplacian
  - – Ng, Jordan, Weiss Spectral Clustering
  - – Laplacian Eigenmaps for Manifold Learning
  - – Computer Vision and many more applications…

Standard reference: Spectral Graph Theory Monograph by Fan Chung Graham

Conclusions

Spectrum tells us a lot about the graph:

- Adjacency: #Paths
- Laplacian: Sparse Cut
- Normalized Laplacian: Normalized cuts, tend to avoid unbalanced cuts

References

- Fan R. K. Chung: *Spectral Graph Theory* (AMS)
- Chris Godsil and Gordon Royle: *Algebraic Graph Theory* (Springer)
- Bojan Mohar and Svatopluk Poljak: *Eigenvalues in Combinatorial Optimization*, IMA Preprint Series #939
- Gilbert Strang: *Introduction to Applied Mathematics* (Wellesley-Cambridge Press)

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

Task 8: hadoop and Tera/Peta byte graphs

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

Scalability

- How about if graph/tensor does not fit in core?
- How about handling huge graphs?

Scalability

- How about if graph/tensor does not fit in core?
- ['MET': Kolda, Sun, ICMD'08, best paper award]
- How about handling huge graphs?

Scalability

- Google: > 450,000 processors in clusters of ~2000 processors each [Barroso, Dean, Hölzle, *“Web Search for a Planet: The Google Cluster Architecture”* IEEE Micro 2003]
- Yahoo: 5Pb of data [Fayyad, KDD’07]
- Problem: machine failures, on a daily basis
- How to parallelize data mining tasks, then?

Scalability

- Google: > 450,000 processors in clusters of ~2000 processors each [Barroso, Dean, Hölzle, *“Web Search for a Planet: The Google Cluster Architecture”* IEEE Micro 2003]
- Yahoo: 5Pb of data [Fayyad, KDD’07]
- Problem: machine failures, on a daily basis
- How to parallelize data mining tasks, then?
- A: map/reduce – hadoop (open-source clone) http://hadoop.apache.org/

2’ intro to hadoop

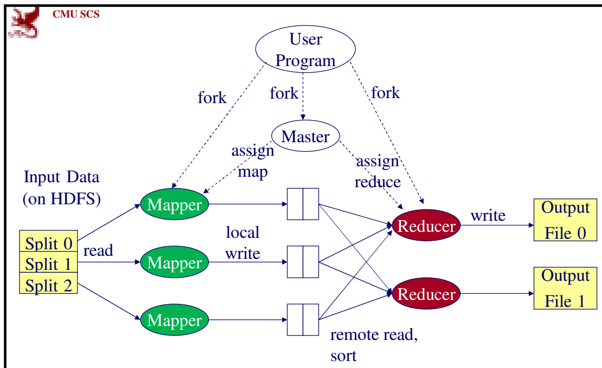
- master-slave architecture; n-way replication (default n=3)
- ‘group by’ of SQL (in parallel, fault-tolerant way)
- e.g, find histogram of word frequency
  - compute local histograms
  - then merge into global histogram

select course-id, count(*)
from ENROLLMENT
group by course-id

2’ intro to hadoop

- master-slave architecture; n-way replication (default n=3)
- ‘group by’ of SQL (in parallel, fault-tolerant way)
- e.g, find histogram of word frequency
  - compute local histograms
  - then merge into global histogram

select course-id, count(*) reduce
from ENROLLMENT
group by course-id map



By default: 3-way replication;
Late/dead machines: ignored, **transparently** (!)

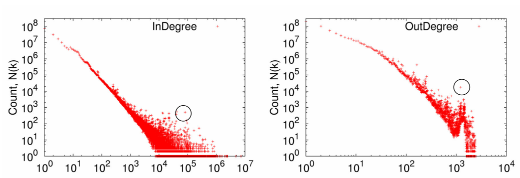
D.I.S.C.

- 'Data Intensive Scientific Computing' [R. Bryant, CMU]
  - 'big data'
  - www.cs.cmu.edu/~bryant/pubdir/cmu-cs-07-128.pdf

Analysis of a large graph

~200Gb (Yahoo crawl) - Degree Distribution:

- in 12 minutes with 50 machines
- Many link spams at out-degree 1200



Conclusions

- Hadoop: promising architecture for Tera/Peta scale graph mining

Resources:

- http://hadoop.apache.org/core/
- http://hadoop.apache.org/pig/

  Higher-level language for data processing

References

- Jeffrey Dean and Sanjay Ghemawat, *MapReduce: Simplified Data Processing on Large Clusters,* OSDI'04
- Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, Andrew Tomkins: *Pig latin: a not-so-foreign language for data processing*. SIGMOD 2008: 1099-1110

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

Christos Faloutsos
Gary Miller
Charalampos (Babis) Tsourakakis
CMU

Outline

Observations – 'laws' of real graphs

- Observation #1: small and SHRINKING diameter
- Observation #2: power law / skewed degree distributions
- Observation #3: power laws in several aspects
- Observation #4: communities

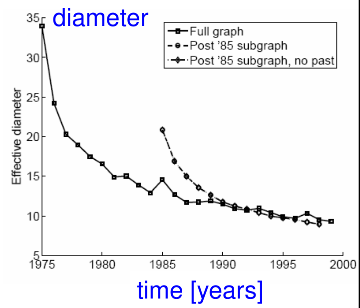
Observation 1 – diameter

- Small diameter – 'six degrees'
- … and the diameter SHRINKS as the graph grows (!)

CMU SCS

Diameter – "Patents"

- Patent citation network
- 25 years of data



CMU SCS

Observation 1 – diameter

- Small diameter – 'six degrees'
- … and the diameter SHRINKS as the graph grows (!)

Practical implication: BFS may die:

– 3-step-away neighbors => half of the graph!

CMU SCS

Observations 2 – degree distribution

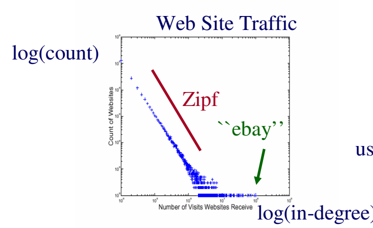
Skewed degree distribution

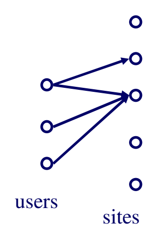
- Most nodes have degree 1 or 2
- … but they probably have a neighbor with degree 100,000 or so (!)

CMU SCS

Degree distributions

- web hit counts [w/ A. Montgomery]

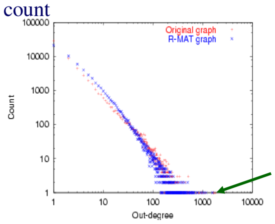




epinions.com

- who-trusts-whom [Richardson + Domingos, KDD 2001]

count



trusts-2000-people user

(out) degree

Observation 2 – degree distributions

Skewed degree distribution

- Most nodes have degree 1 or 2
- … but they probably have a neighbor with degree 100,000 or so (!)

Practical implications:

- May need to delete/ignore those high degree nodes
- Could probably also trim the 1-degree nodes, saving significant space and time

Observation 3 – power laws

Power-laws / skewed distributions in everything:

- Most pairs: within 2-3 steps; but, some pair: ~20 or more steps away
- Triangles: power laws[Tsourakakis'08]
- # of cliques: ditto [Du+'09]
- Weight vs degree: ditto [McGlohon+'08]

Observation 4 – communities

- 'Negative dimensionality' paradox [Chakrabarti+'04]

Practical implication:

- Graphs may have no good cuts

Conclusions

0) Graphs appear in numerous settings

1) Singular / eigenvalue analysis: valuable
   - Fixed points – random walks – importance
   - Eigenvalue and epidemic threshold
   - Laplacians -> communities

Conclusions – cont'd

2) Random walks -> proximity
   - Recommendations, auto-captioning, etc
   - Fast algo's, through Sherman-Morrison

3) Tera-byte scale graphs: hadoop

4) Beware: counter-intuitive properties
   - small diameters; power-laws; possible lack of good cuts

Acknowledgements

Funding:

National Science Foundation
WHERE DISCOVERIES BEGIN

IIS-0705359, IIS-0534205, DBI-0640543, CNS-0721736

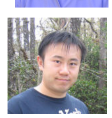
LAWRENCE LIVERMORE NATIONAL LABORATORY IBM Microsoft

YAHOO! Sprint PITA (PA Inf. Tech. Alliance)

hp intel

Acknowledgements - foils




- Chakrabarti, Deepay (cross-associations)
- Kolda, Tamara (tensors)
- Papadimitriou, Spiros (cross-associations)
- Sun, Jimeng (tensors)
- Tong, Hanghang (proximity)

THANK YOU!

Christos Faloutsos
www.cs.cmu.edu/~christos

Gary Miller
www.cs.cmu.edu/~glmiller

Charalampos (Babis) Tsourakakis
www.cs.cmu.edu/~ctsourak

www.cs.cmu.edu/~christos/TALKS/09-KDD-tutorial/