Carnegie Mellon

Talk 2: Graph Mining Tools - SVD, ranking, proximity

Christos Faloutsos

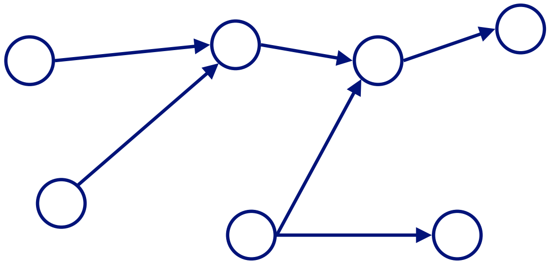
CMU

Outline

- Introduction – Motivation
- **Task 1: Node importance**
- Task 2: Recommendations
- Task 3: Connection sub-graphs
- Conclusions

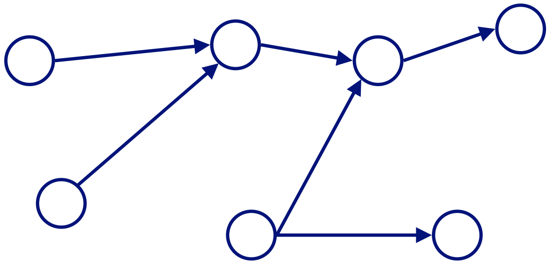
Node importance - Motivation:

- 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)



Node importance - motivation

- SVD and eigenvector analysis: very closely related

SVD - Detailed outline

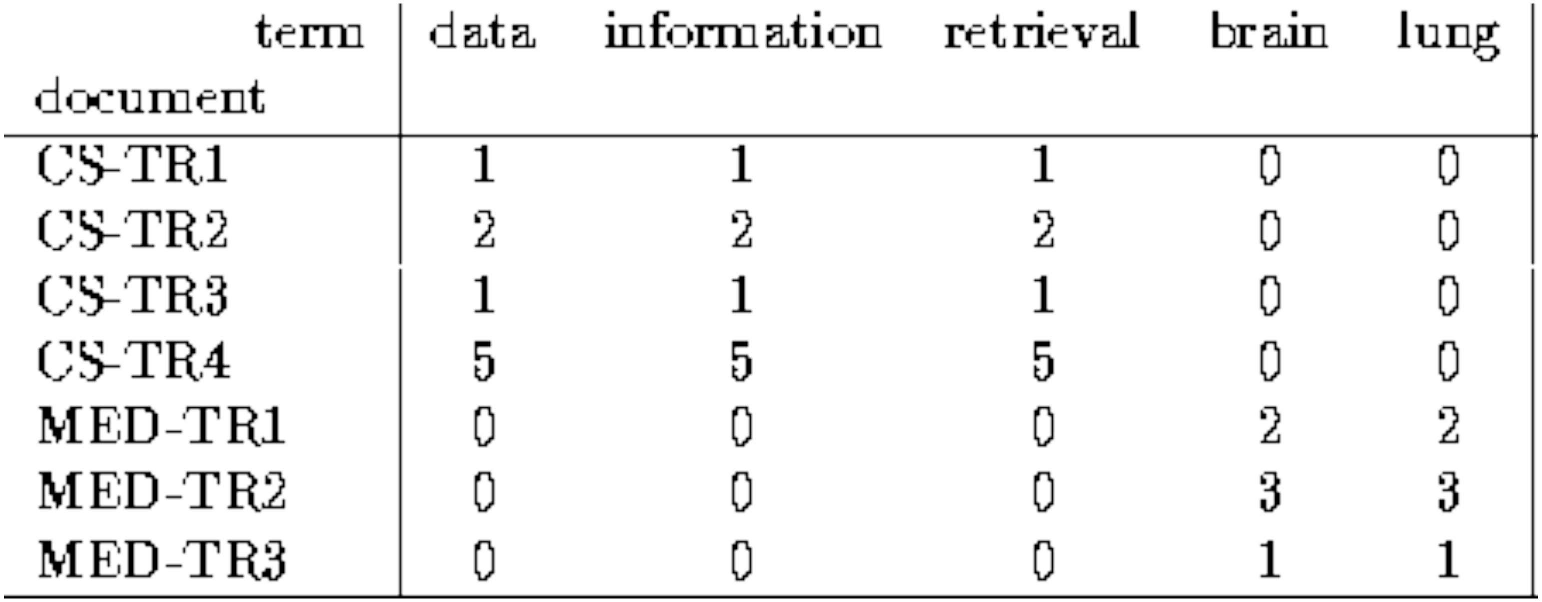
- Motivation
- Definition - properties
- Interpretation
- Complexity
- Case studies

SVD - Motivation

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

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

SVD - Motivation

- Customer-product, for recommendation system:

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

details

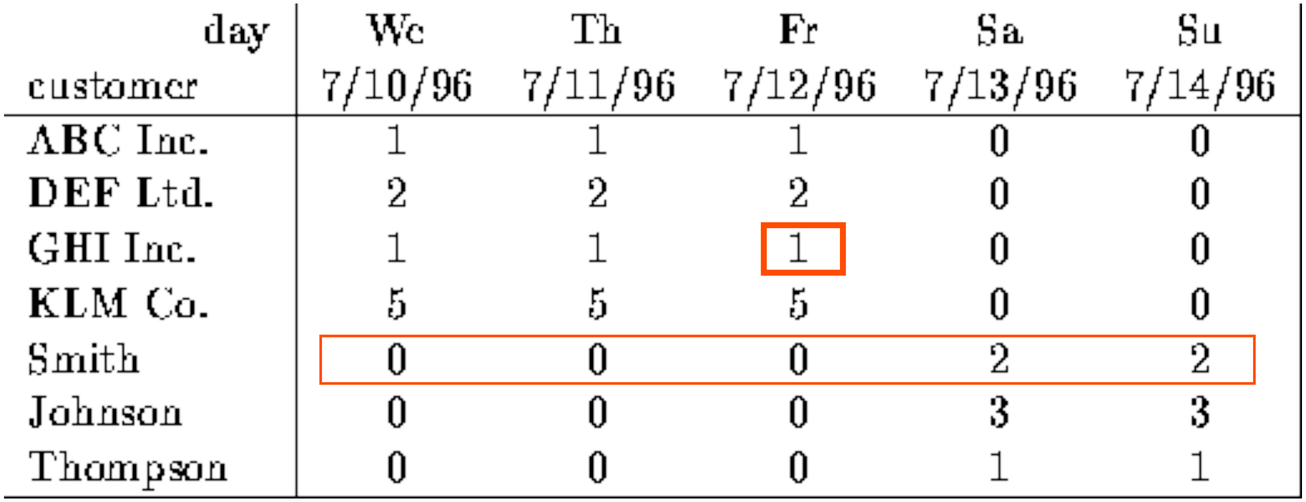
SVD - Motivation

- problem #2: compress / reduce dimensionality

details

Problem - specs

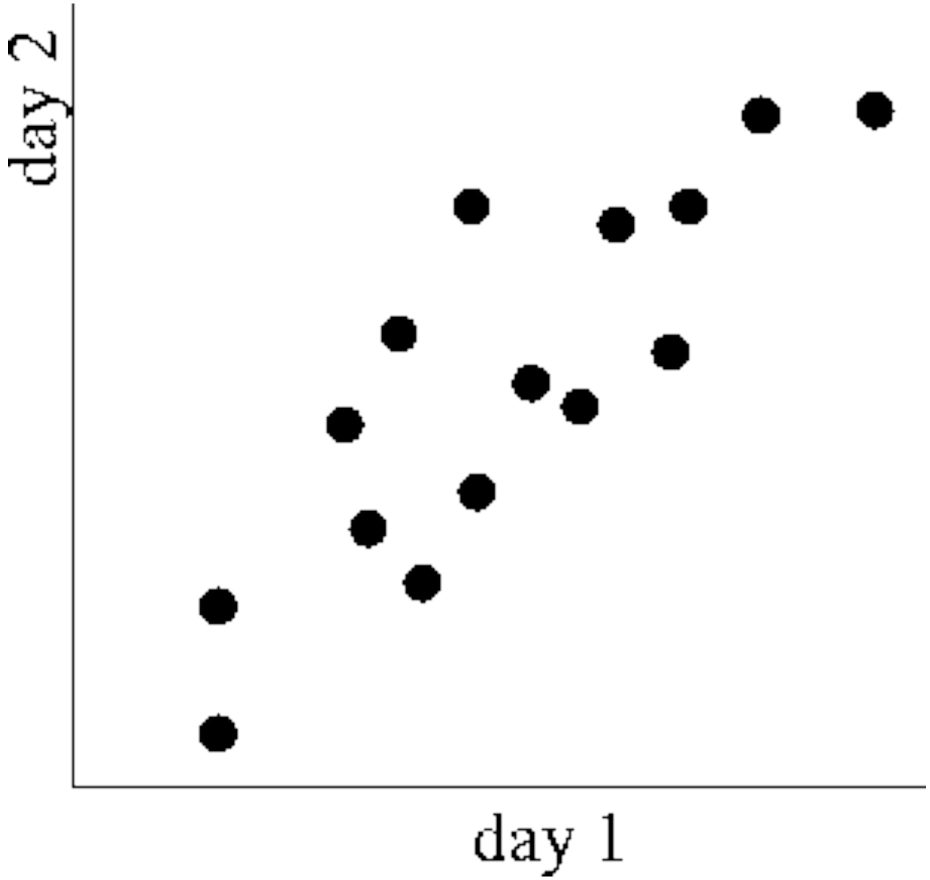
- ~10**6 rows; ~10**3 columns; no updates;
- random access to any cell(s) ; small error: OK



day	We	Th	Fr	Sa	Su
customer	7/10/96	7/11/96	7/12/96	7/13/96	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

details

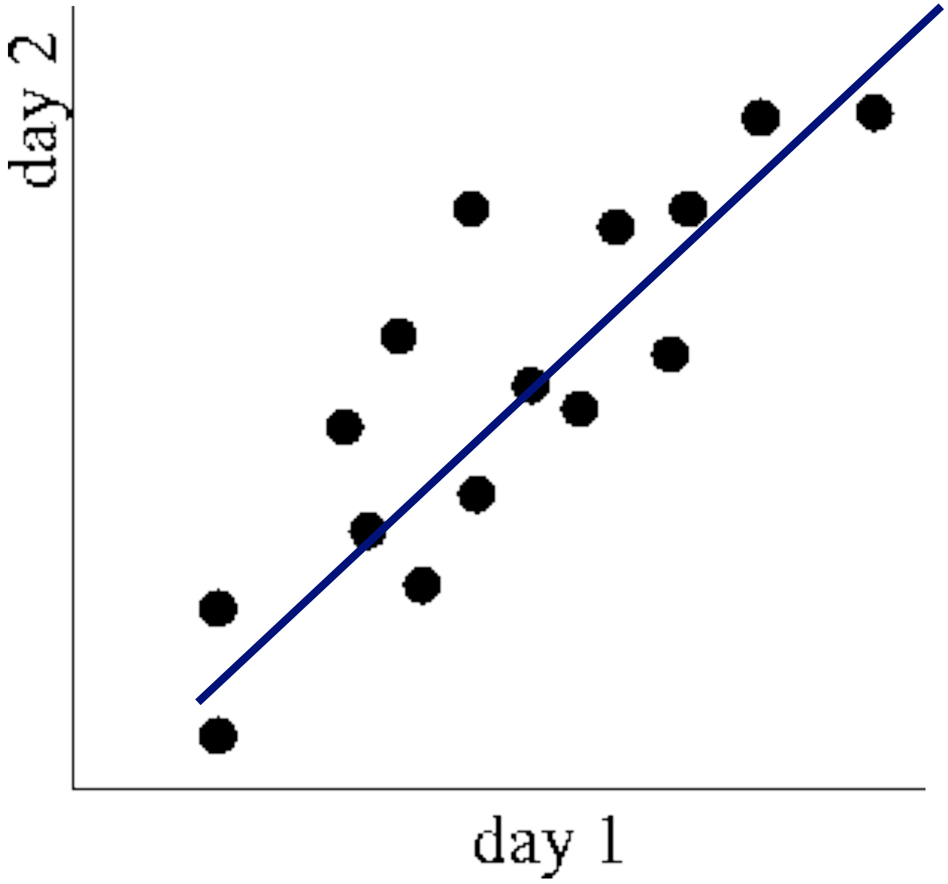
SVD - Motivation



Carnegie Mellon

details

SVD - Motivation



SVD - Detailed outline

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

SVD - Definition

(reminder: matrix multiplication

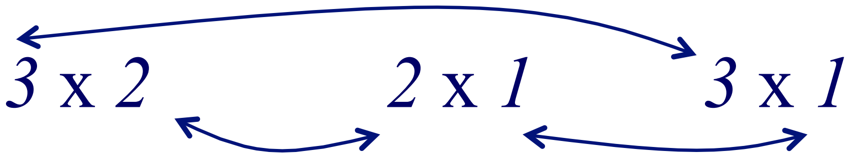
$$\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix} \text{ x } \begin{bmatrix}1\\ -1\end{bmatrix} = \begin{bmatrix} \\ \\ \end{bmatrix}$$

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

SVD - Definition

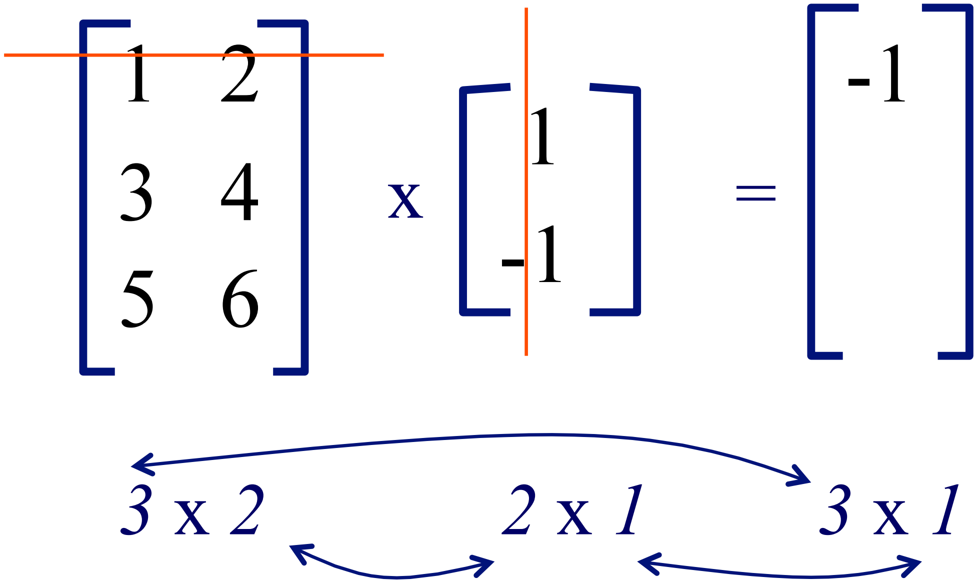
(reminder: matrix multiplication

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


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

SVD - Definition

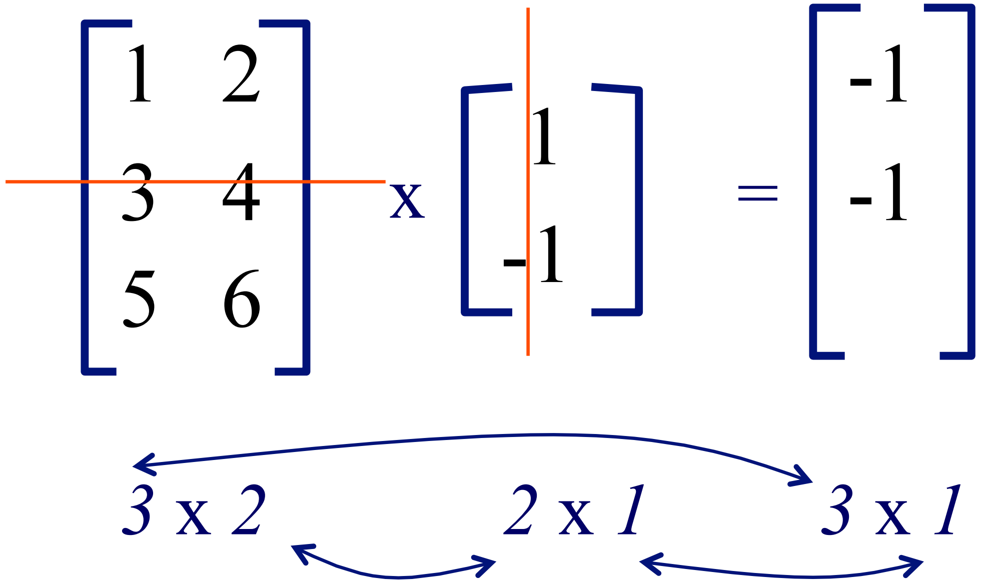
(reminder: matrix multiplication



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

SVD - Definition

(reminder: matrix multiplication



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

SVD - Definition

(reminder: matrix multiplication

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

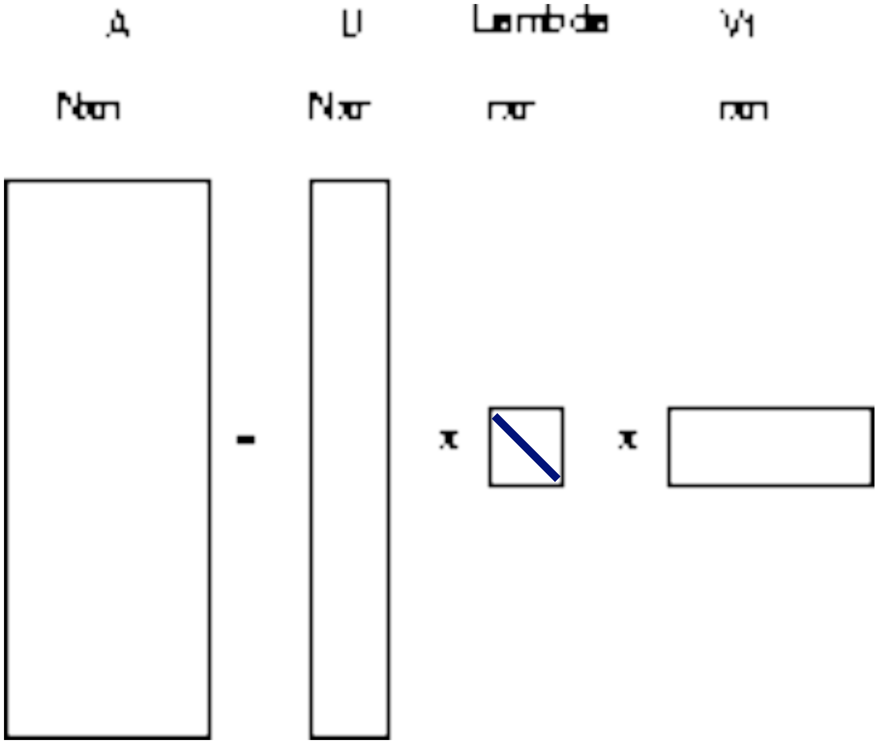
SVD - Definition

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

- **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)

SVD - Definition

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



SVD - Properties

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

- $\mathbf{U}, \mathbf{\Lambda}, \mathbf{V}$: 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)
- $\mathbf{\Lambda}$**:** singular are positive, and sorted in decreasing order

SVD - Example

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

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

SVD - Example

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

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

SVD - Example

- **A** = **U Λ V**^{T} - example:

doc-to-concept similarity matrix

CS-concept

MD-concept

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

SVD - Example

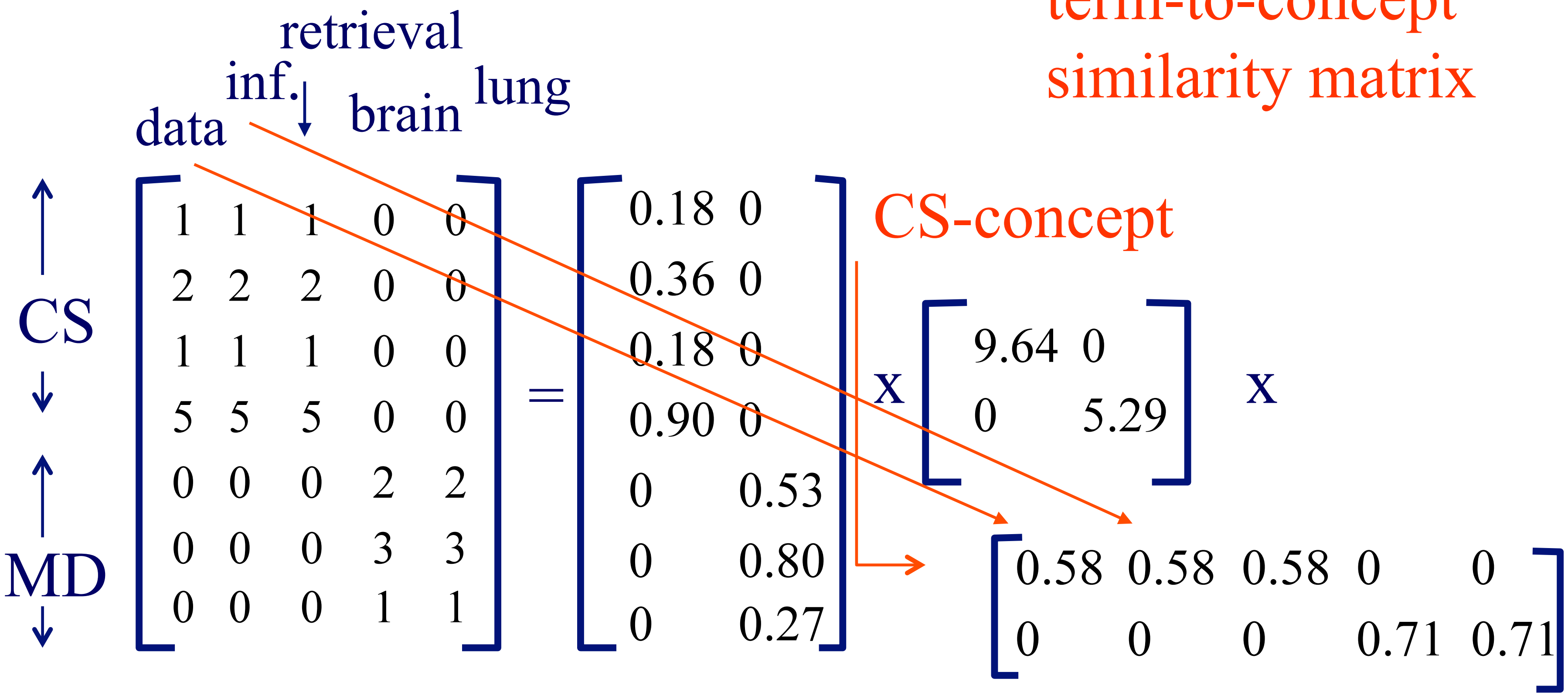
- $\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{V}^T$ - example:

'strength' of CS-concept

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

SVD - Example

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



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

term-to-concept similarity matrix

CS-concept

SVD - Example

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

term-to-concept similarity matrix

CS-concept

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

SVD - Detailed outline

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

SVD - Interpretation #1

‘documents’, ‘terms’ and ‘concepts’:

- **U**: document-to-concept similarity matrix
- **V**: term-to-concept sim. matrix
- **Λ**: its diagonal elements: ‘strength’ of each concept

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:

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.

details

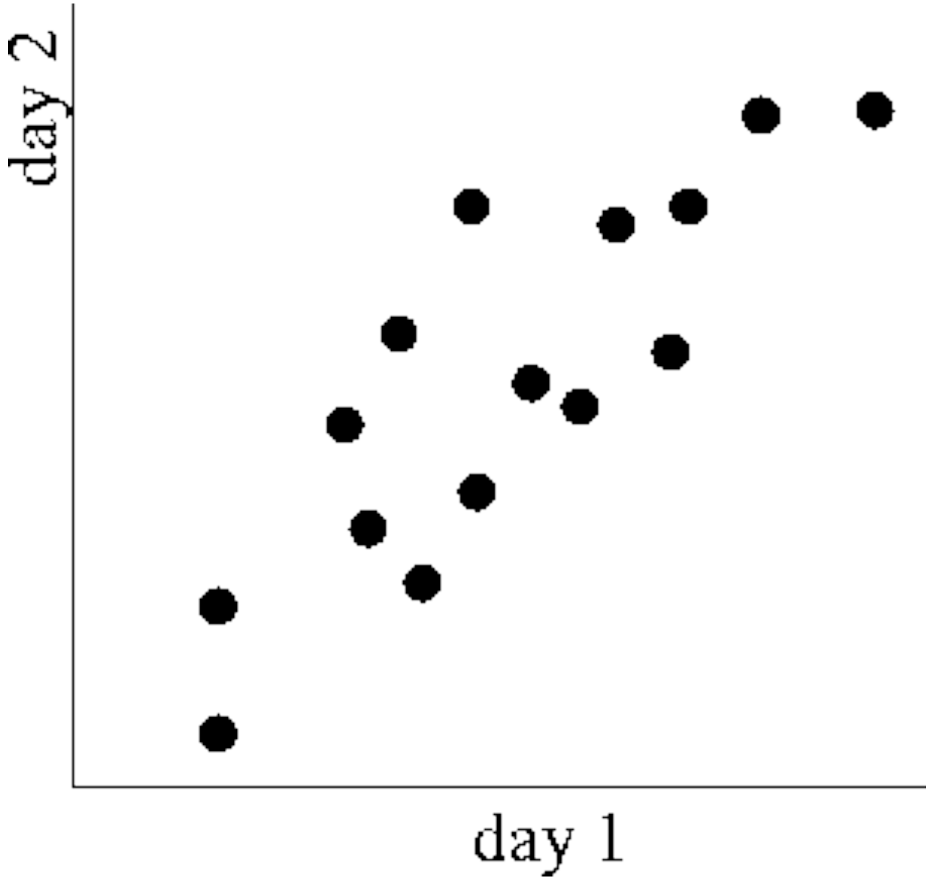
SVD - Interpretation #2

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

Carnegie Mellon

details

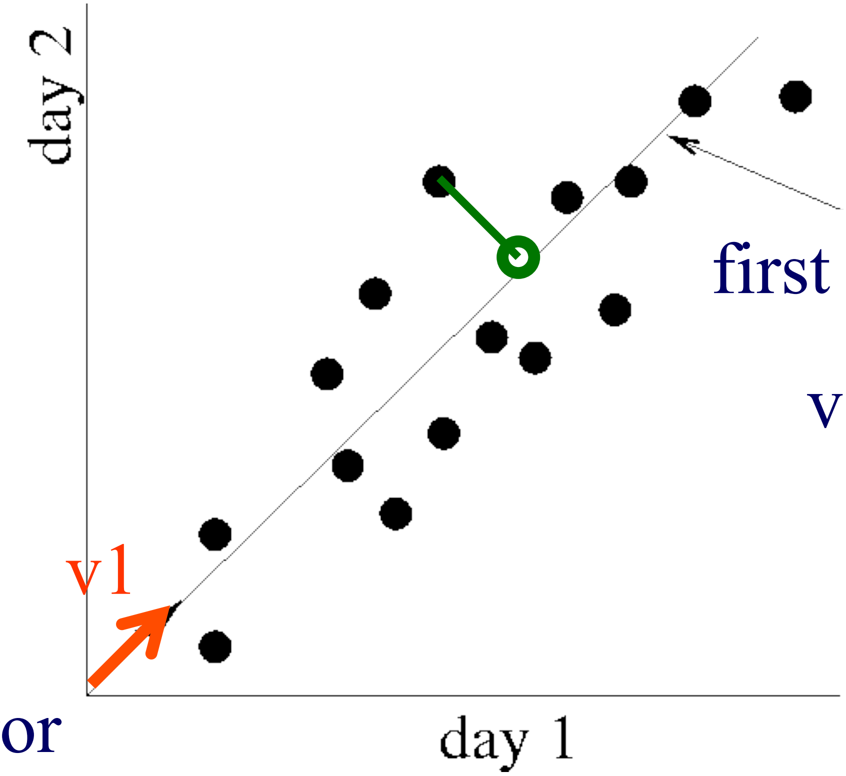
SVD - Motivation



details

SVD - interpretation #2

SVD: gives
best axis to project

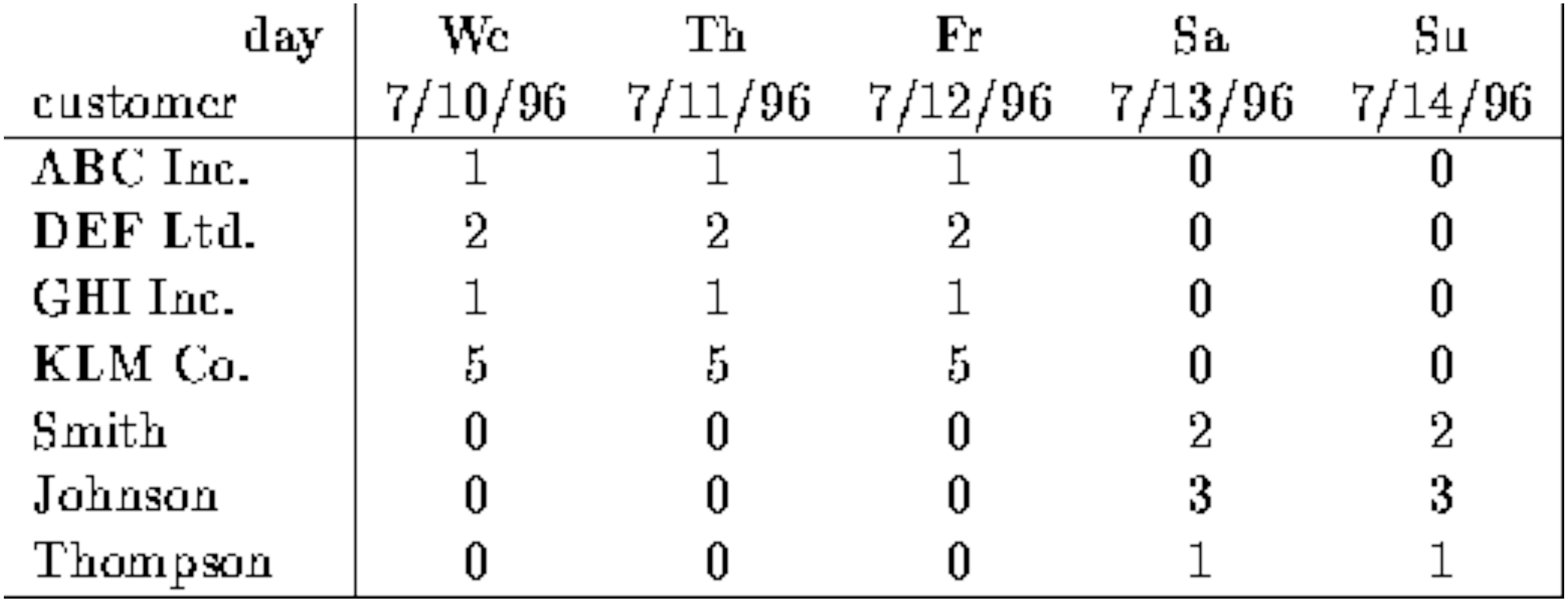


first singular vector

- minimum RMS error

details

SVD - Interpretation #2



day	We	Th	Fr	Sa	Su
customer	7/10/96	7/11/96	7/12/96	7/13/96	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

details

SVD - Interpretation #2

- **A** = **U Λ V**^{T} - example:

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

details

SVD - Interpretation #2

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

variance ('spread') on the v1 axis

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

details

SVD - Interpretation #2

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

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

details

SVD - Interpretation #2

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

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

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

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

details

SVD - Interpretation #2

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

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:

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

details

SVD - Interpretation #2

Exactly equivalent:
'spectral decomposition' of the matrix:

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

details

SVD - Interpretation #2

Exactly equivalent:

'spectral decomposition' of the matrix:

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

details

SVD - Interpretation #2

Exactly equivalent:

'spectral decomposition' of the matrix:

$$n \overset{\longleftarrow \; m \; \longrightarrow}{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}} = \overset{\longleftarrow \; \text{r terms} \; \longrightarrow}{\lambda_1 \quad \underset{\text{n x 1}}{u_1} \quad \underset{\text{1 x m}}{v^T_1} + \quad \lambda_2 \quad u_2 \quad v^T_2 + \ldots}$$

details

SVD - Interpretation #2

approximation / dim. reduction:

by keeping the first few terms (Q: how many?)

$$n \updownarrow \overset{\longleftarrow\; m \;\longrightarrow}{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}} = \lambda_1 \quad u_1 \quad v^T_1 + \quad \lambda_2 \quad u_2 \quad v^T_2 + \ldots$$

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

details

SVD - Interpretation #2

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

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

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

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

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

$$\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Interpretation #3

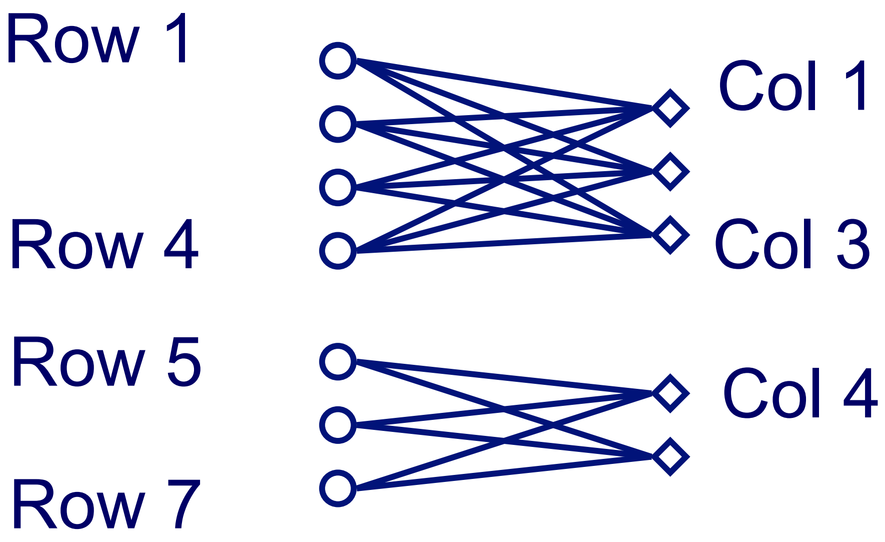
- finds non-zero ‘blobs’ in a data matrix

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

SVD - Interpretation #3

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

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



SVD - Detailed outline

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

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 ...)

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’

details

SVD - Detailed outline

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

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)

details

SVD -outline of properties

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

details

Properties - by defn.:

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

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

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

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

details

Less obvious properties

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

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

symmetric; Intuition?

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

symmetric; Intuition?

‘document-to-document’ similarity matrix

B(2): symmetrically, for ‘V’

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

Intuition?

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 **V**

λ_1: strongest singular value

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

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

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

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

$$\boxed{\text{C(4): } \mathbf{A}^{\mathbf{T}}\, \mathbf{A}\, \mathbf{v}_{\mathbf{1}} = \boldsymbol{\lambda}_{\mathbf{1}}^{\mathbf{2}}\, \mathbf{v}_{\mathbf{1}}}$$

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

Least obvious properties - altogether

details

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

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

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

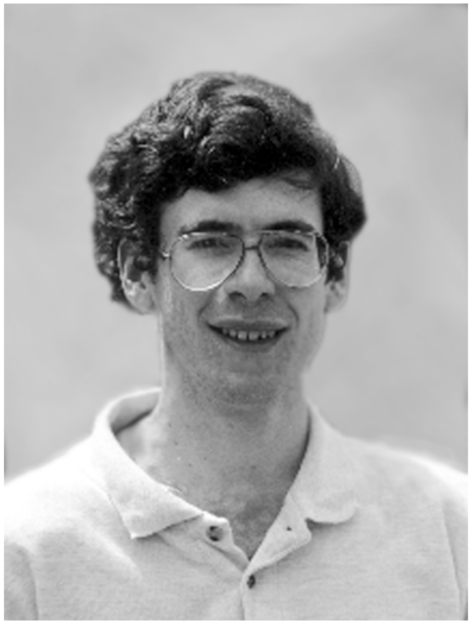
C(4): $\mathbf{A}^{\mathbf{T}}\, \mathbf{A}\, \mathbf{v}_1 = \boldsymbol{\lambda}_1^2\, \mathbf{v}_1$

Carnegie Mellon

SVD - detailed outline

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

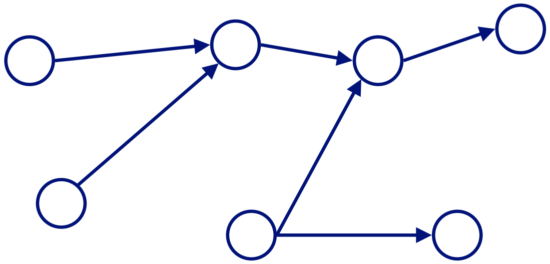
Kleinberg's algo (HITS)



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?



Kleinberg's algorithm

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

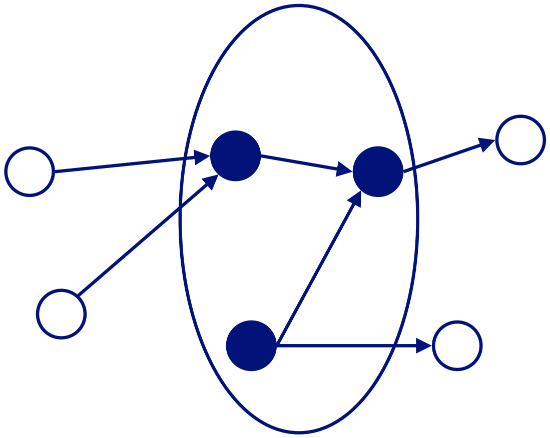
Step 0: find all pages containing the query terms

Step 1: expand by one move forward and backward

Carnegie Mellon

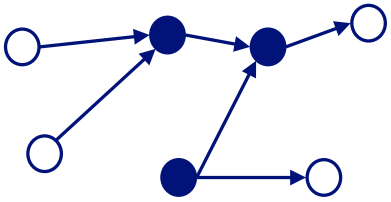
Kleinberg’s algorithm

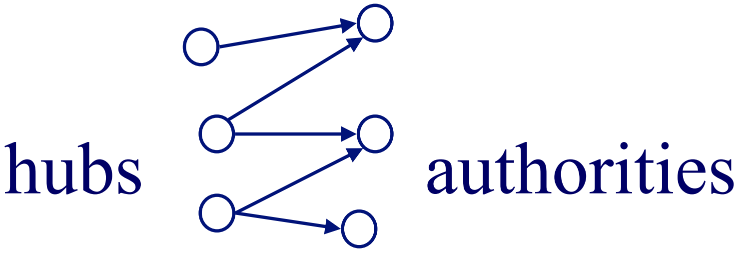
- Step 1: expand by one move forward and backward



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’)





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

Kleinberg's algorithm

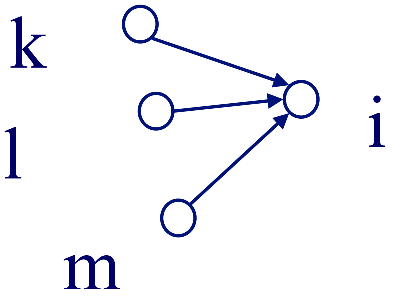
Let E be the set of edges and **A** be the adjacency matrix:

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:

Kleinberg's algorithm



Then:

$$a_i = h_k + h_l + h_m$$

that is

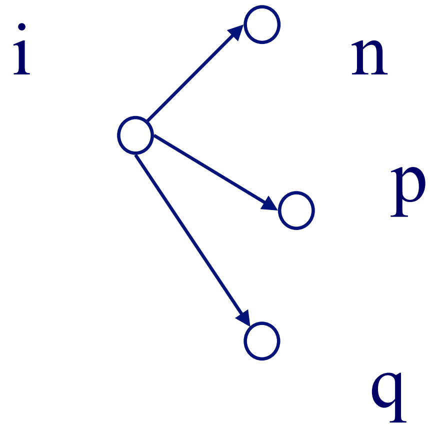
a_i = Sum (h_j)   over all j that (j,i) edge exists

or

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

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

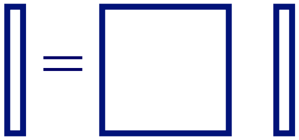
$$\mathbf{h} = \mathbf{A}\ \mathbf{a}$$

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): $\mathbf{A}_{[n\ x\ m]}\ \mathbf{v}_{\mathbf{1}\ [m\ x\ 1]} = \lambda_1\ \mathbf{u}_{\mathbf{1}\ [n\ x\ 1]}$

C(3): $\mathbf{u}_\mathbf{1}^\mathbf{T}\ \mathbf{A} = \lambda_1\ \mathbf{v}_\mathbf{1}^\mathbf{T}$

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?)

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):

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

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")

Kleinberg’s algorithm - discussion

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

SVD - detailed outline

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

PageRank (google)

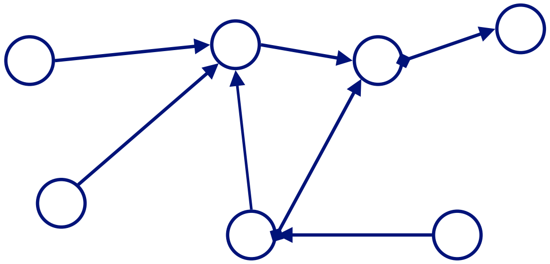
Larry Page

Sergey Brin

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

Problem: PageRank

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

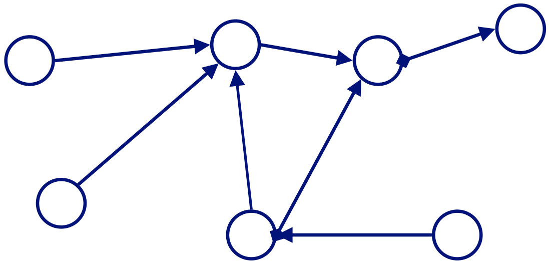


A node is important,
if it is connected
with important nodes
(recursive, but OK!)

Problem: PageRank - solution

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

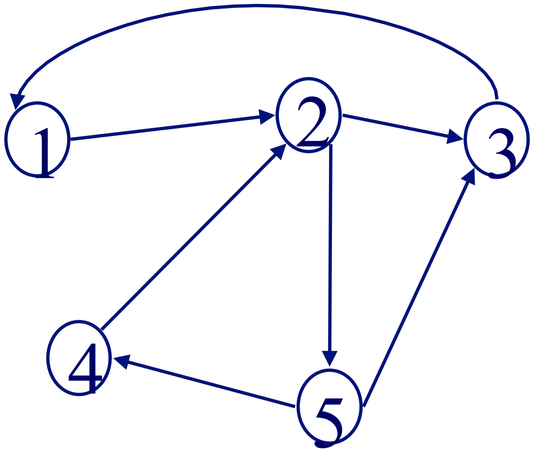
Proposed solution: Random walk; spot most 'popular' node (-> steady state prob. (ssp))



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

(Simplified) PageRank algorithm

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



From

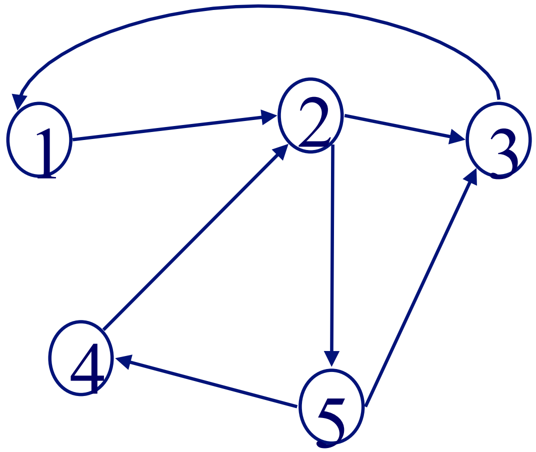
To

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**



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

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

- $\mathbf{B}\,\mathbf{p} = 1 * \mathbf{p}$
- 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

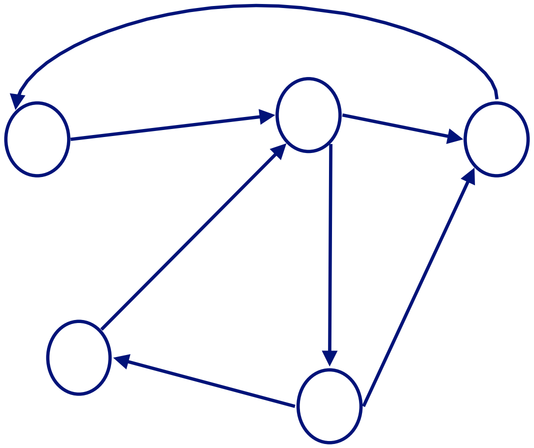
Why? To make the matrix irreducible

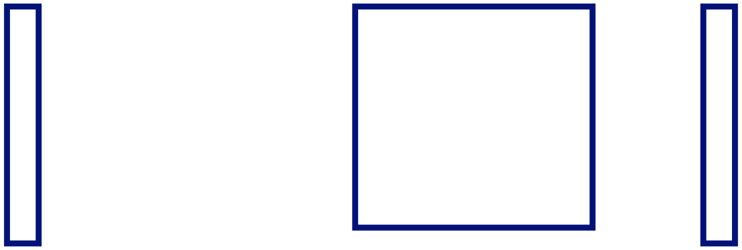
Full Algorithm

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

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

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





Alternative notation

M Modified transition matrix

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

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

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

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

$$C(4)\text{: } \mathbf{B}^T\ \mathbf{B}\ \mathbf{v}_\mathbf{i} = \lambda_\mathbf{i}^2\ \mathbf{v}_\mathbf{i}$$

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

Property #2

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

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

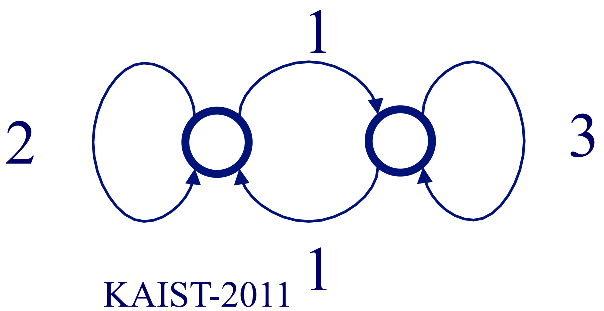
Property #3

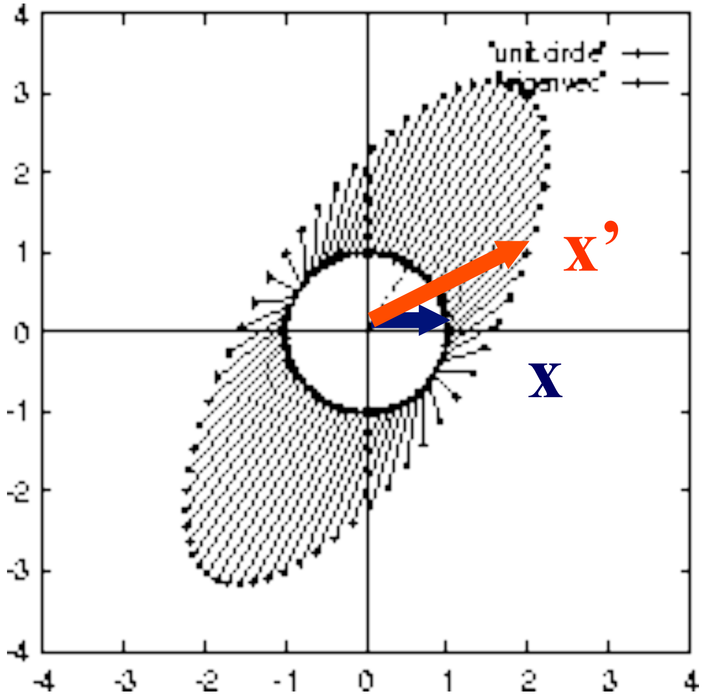
- If $\mathbf{A}_{[\mathbf{nxn}]}$ is a real, symmetric matrix
- Then it has *n* real eigenvalues
- And they agree with its *n* singular values, except possibly for the sign

Intuition

- **A** as vector transformation

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

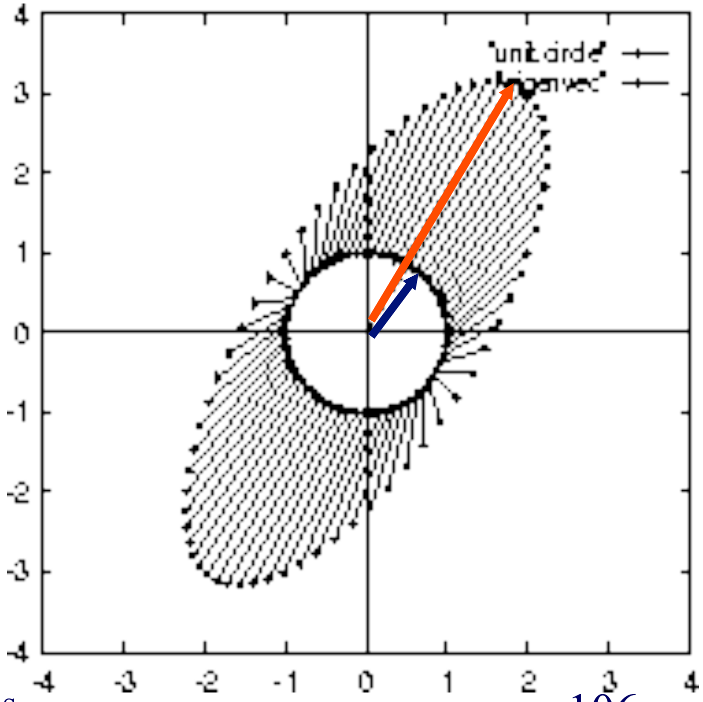




Intuition

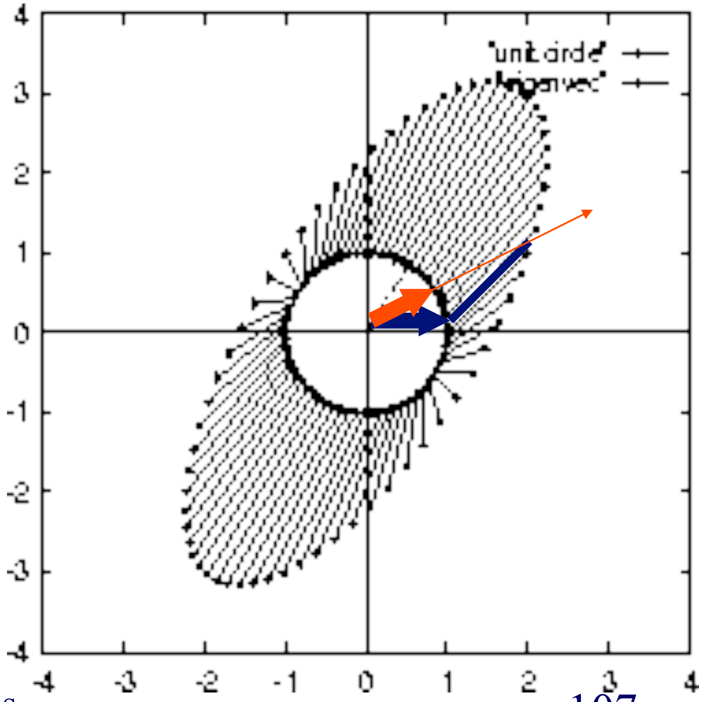
- By defn., eigenvectors remain parallel to themselves (‘**fixed points**’)

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



Convergence

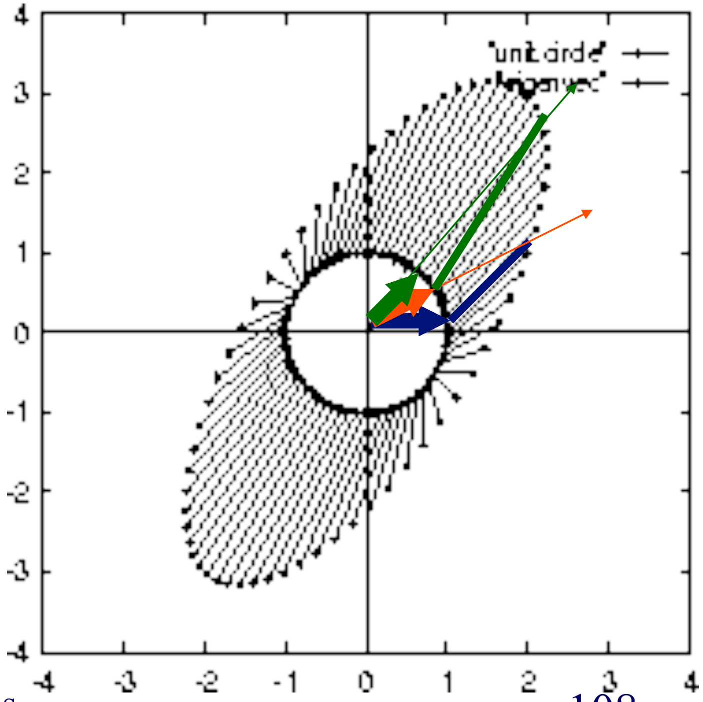
- Usually, fast:



Carnegie Mellon

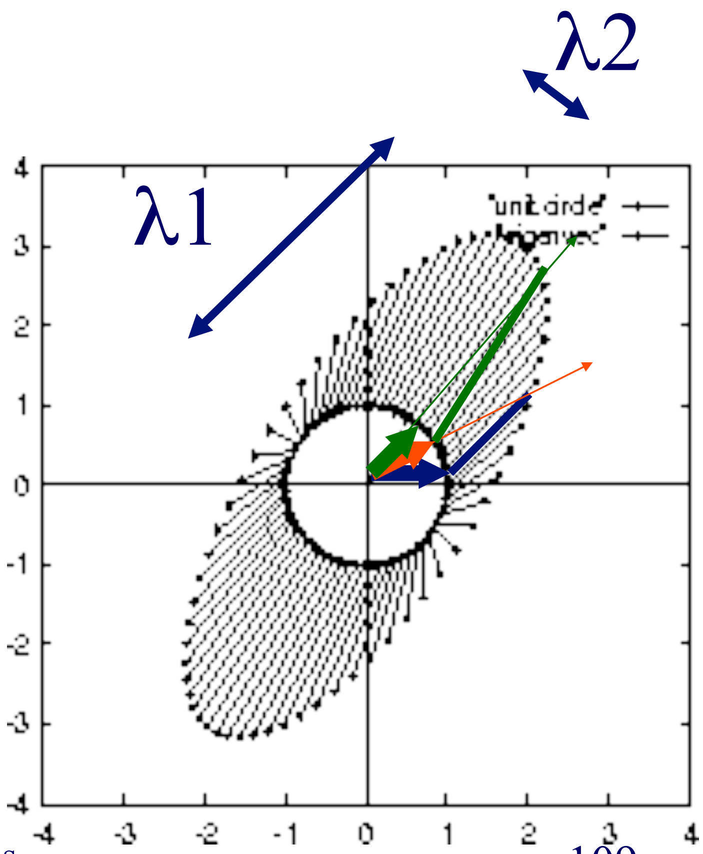
Convergence

- Usually, fast:



Convergence

- Usually, fast:
- depends on ratio
  λ1 : λ2



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

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)

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.

Carnegie Mellon

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`

Outline

- Introduction – Motivation
- Task 1: Node importance
- ➡ **Task 2: Recommendations & proximity**
- Task 3: Connection sub-graphs
- Conclusions

Carnegie Mellon

Acknowledgement:

Most of the foils in ‘Task 2’ are by

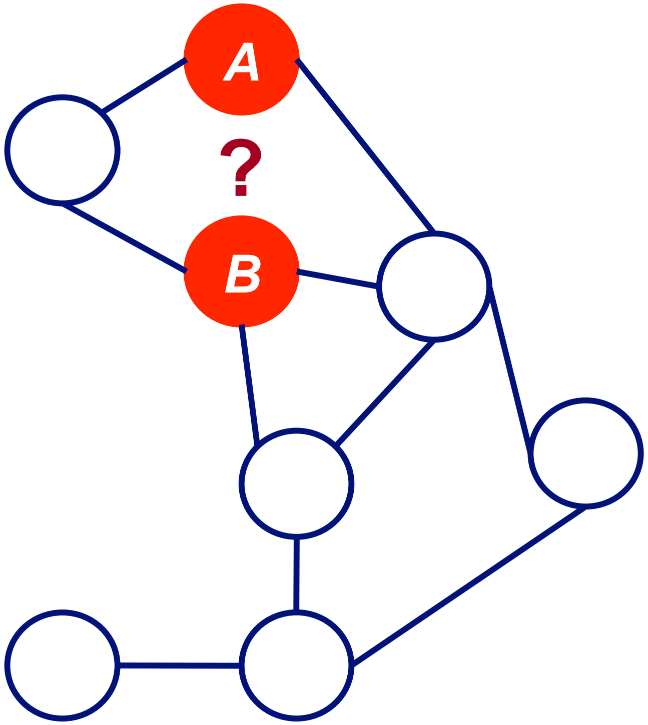
Hanghang TONG

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

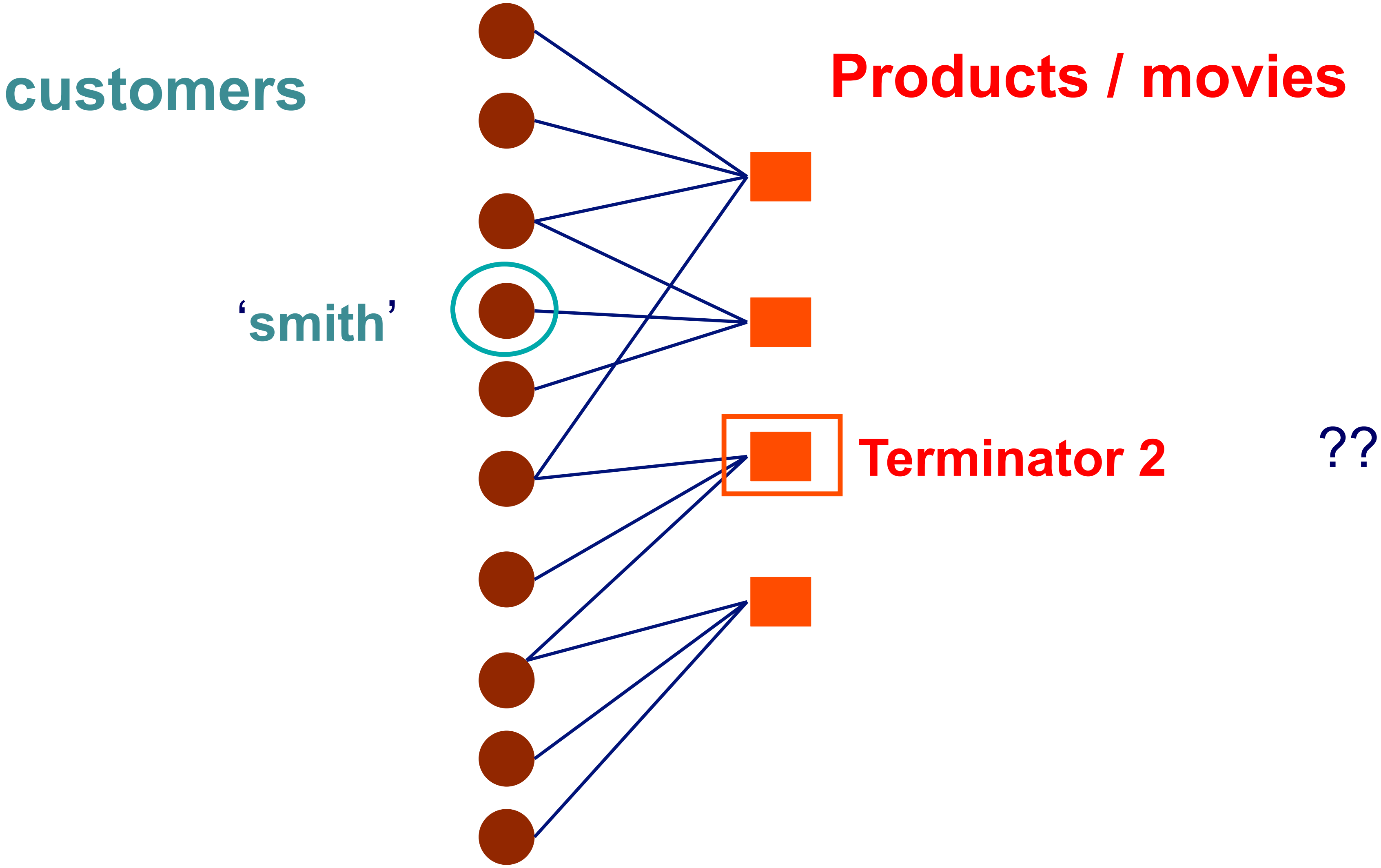
Motivation: Link Prediction



Should we introduce
Mr. A to Mr. B?

Carnegie Mellon

Motivation - recommendations



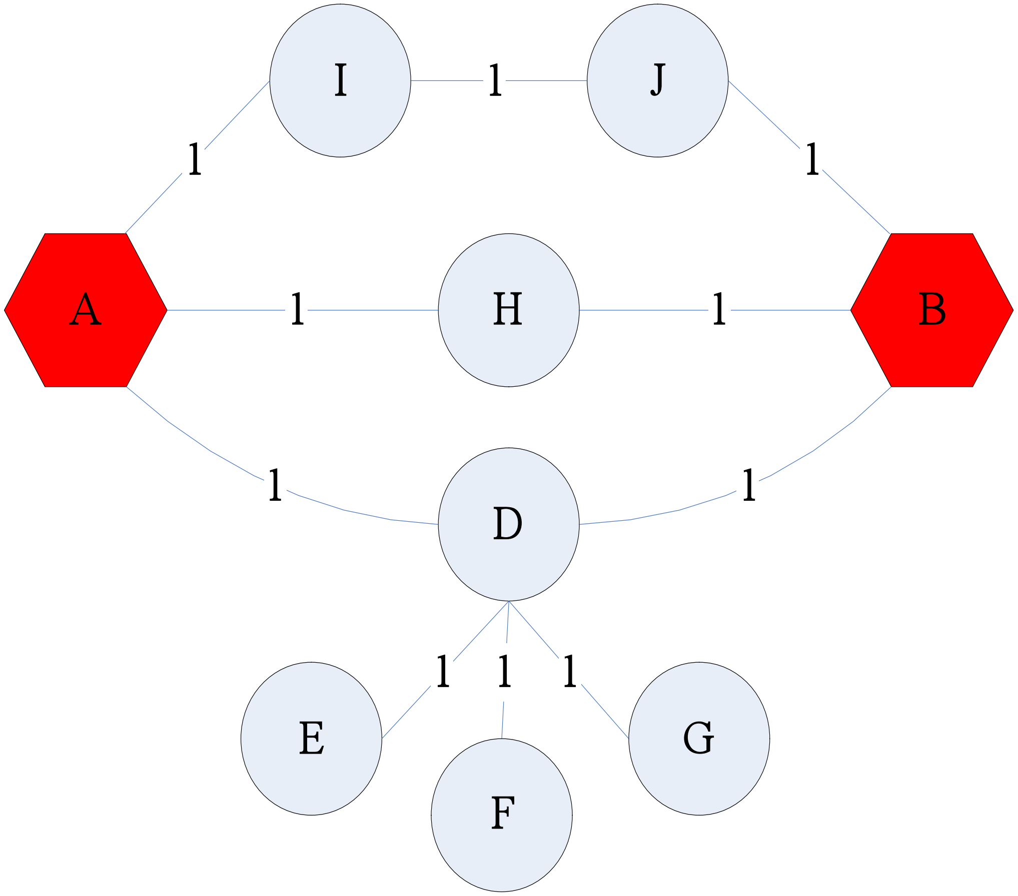
Answer: proximity

- ‘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

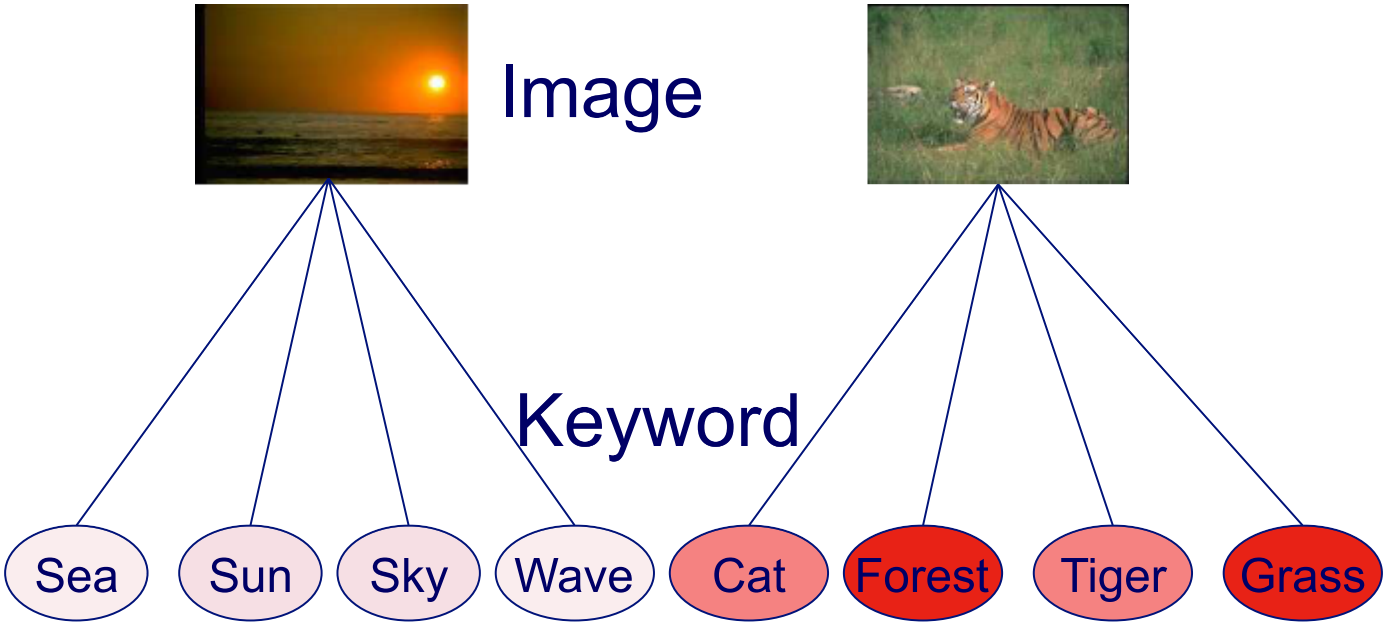
And many more

- **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



Test Image

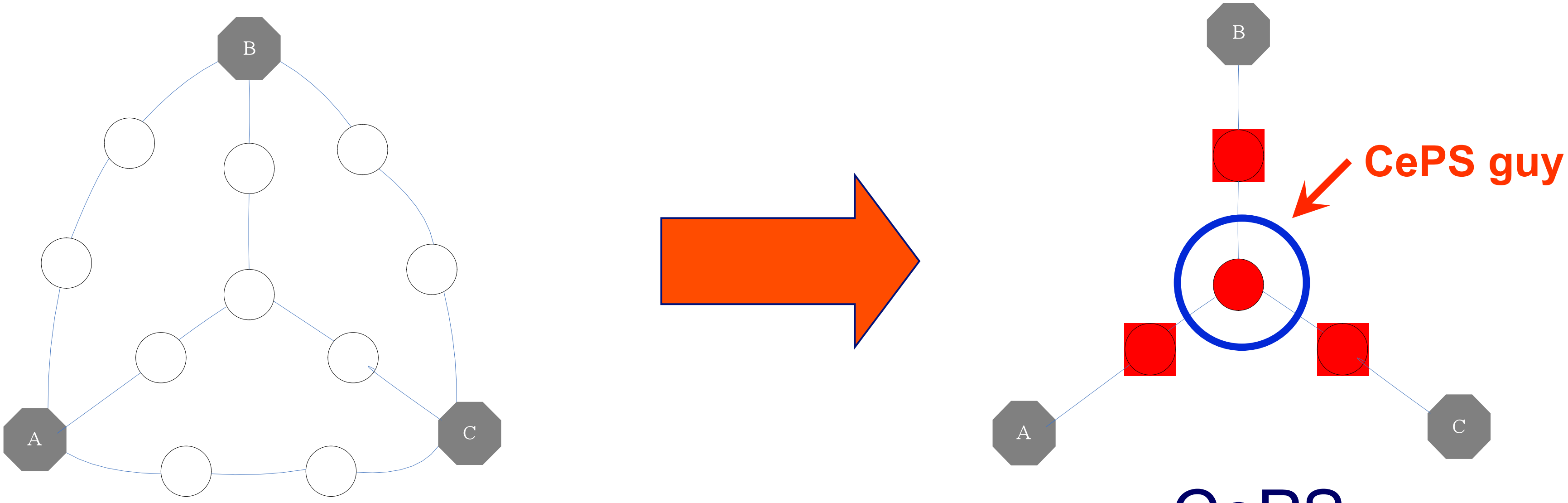
Q: How to assign keywords to the test image?

A: Proximity! [Pan+ 2004]

Center-Piece Subgraph(CePS)

Input

Output



Original Graph

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

Carnegie Mellon

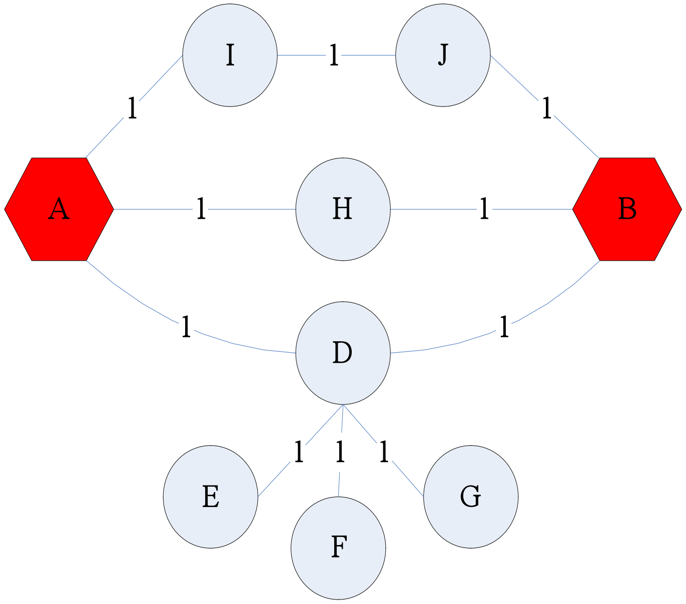
How close is ‘A’ to ‘B’?

Should be close, if they have

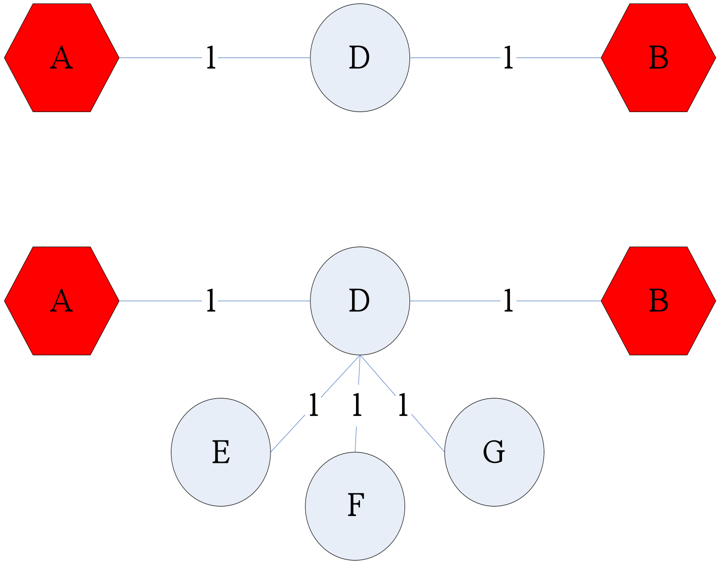
- many,

- short

- ‘heavy’ paths

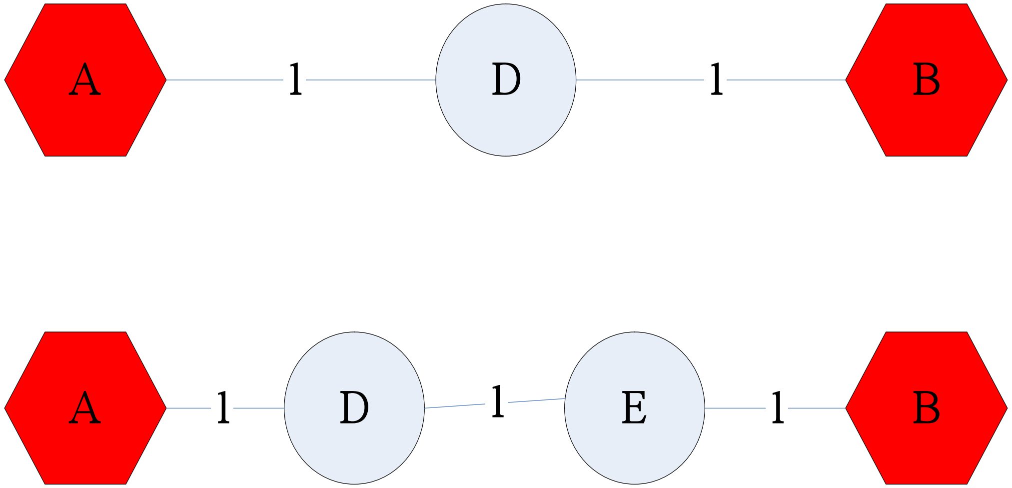


Why not shortest path?



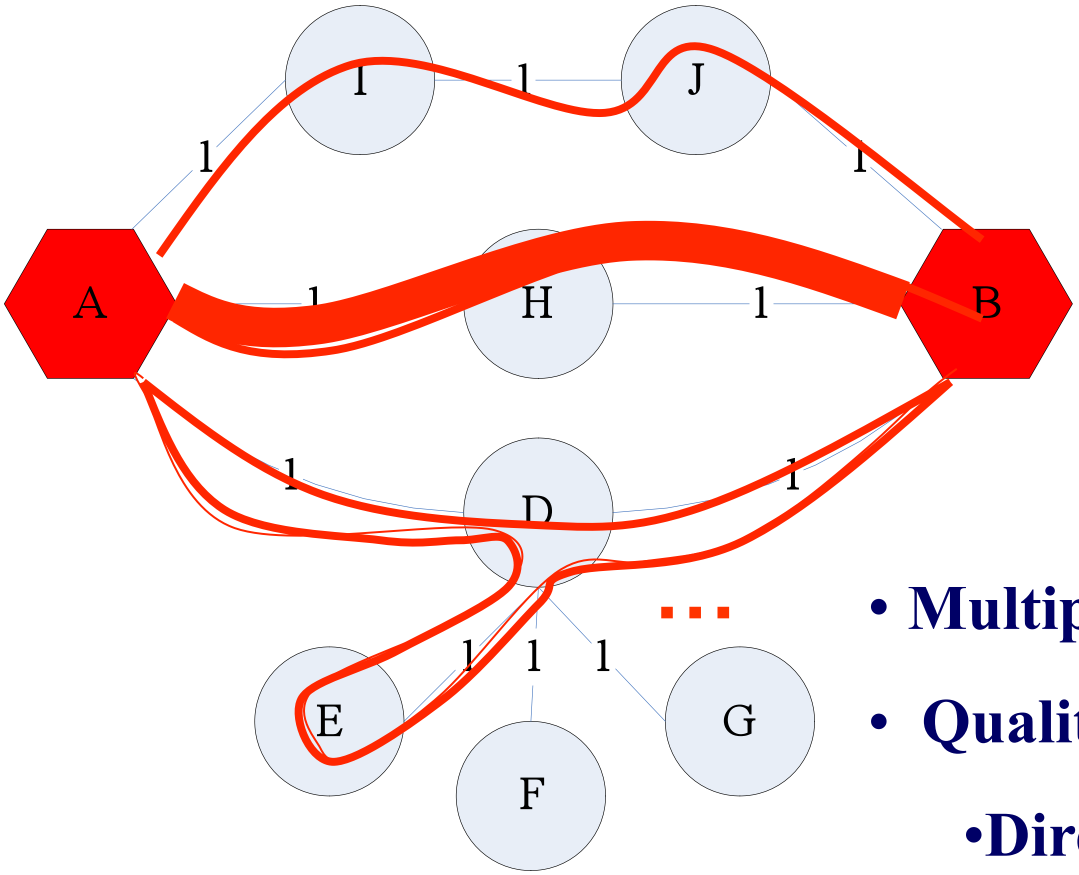
A: ‘pizza delivery guy’ problem

Why not max. netflow?



A: No penalty for long paths

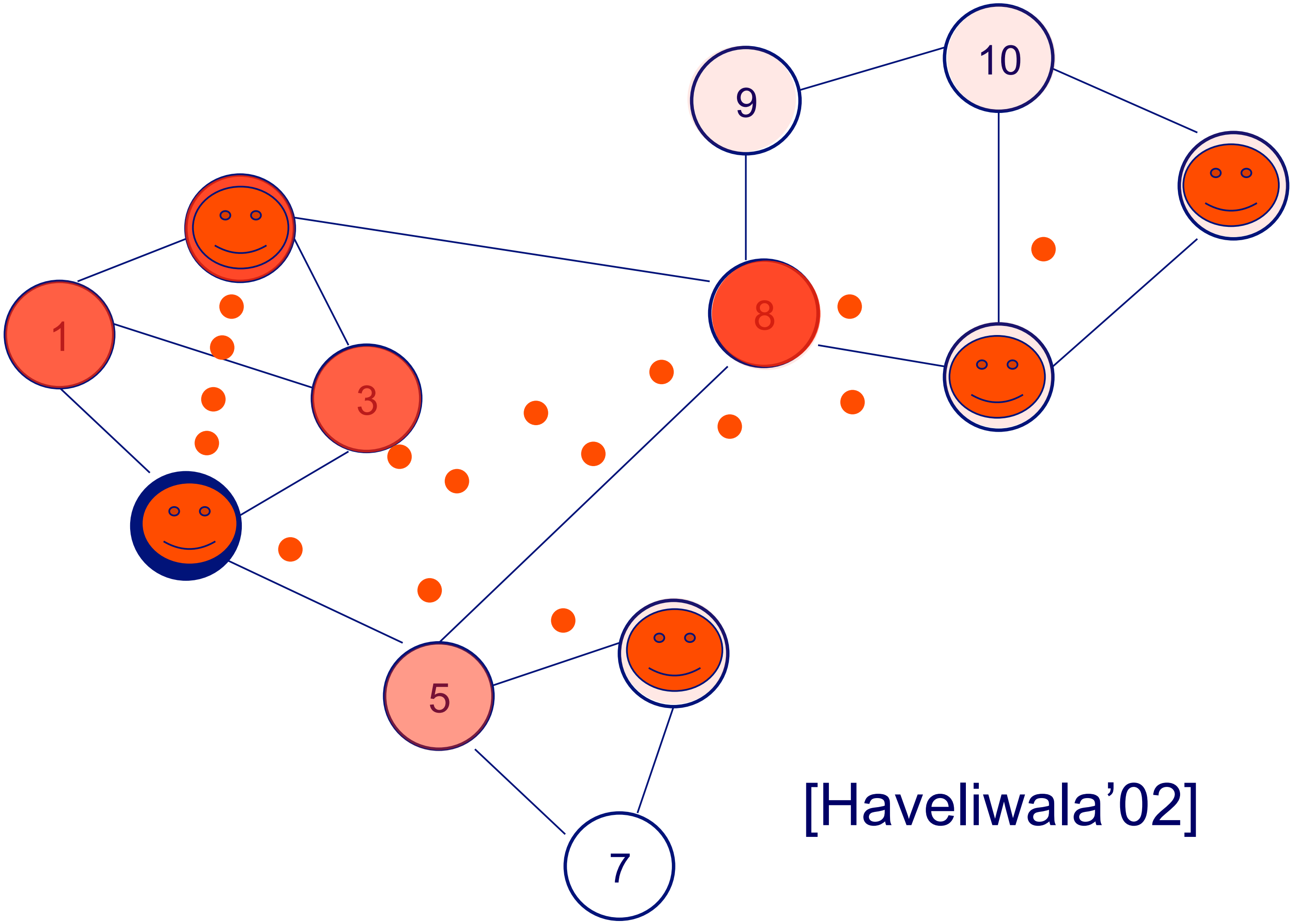
What is a ``good'' Proximity?



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

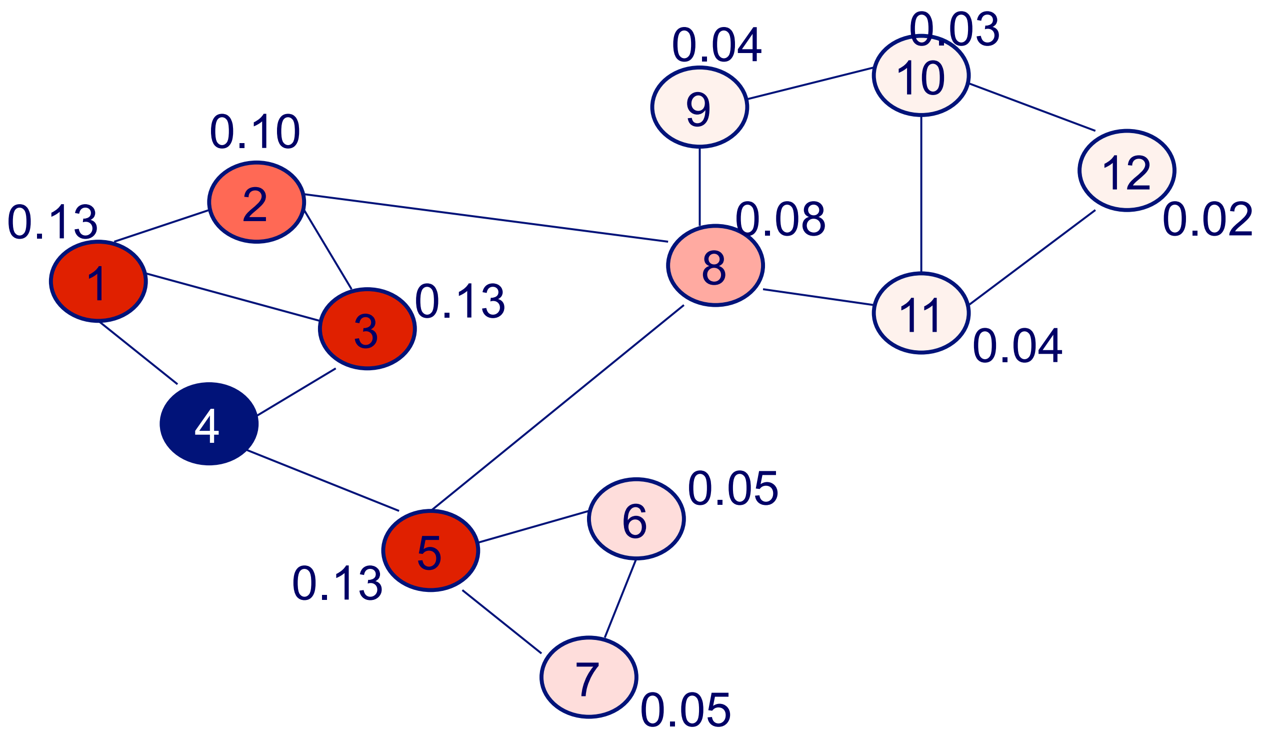
Carnegie Mellon

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

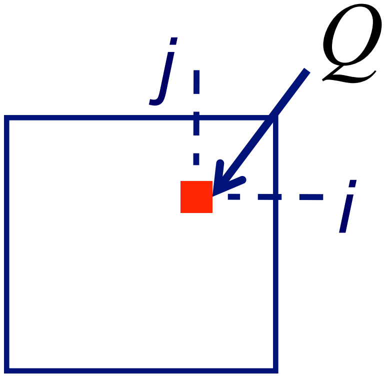
Ranking vector

$\vec{r}_4$

Why RWR is a good score?

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

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



$\tilde{W}$: adjacency matrix.

c: damping factor

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

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

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

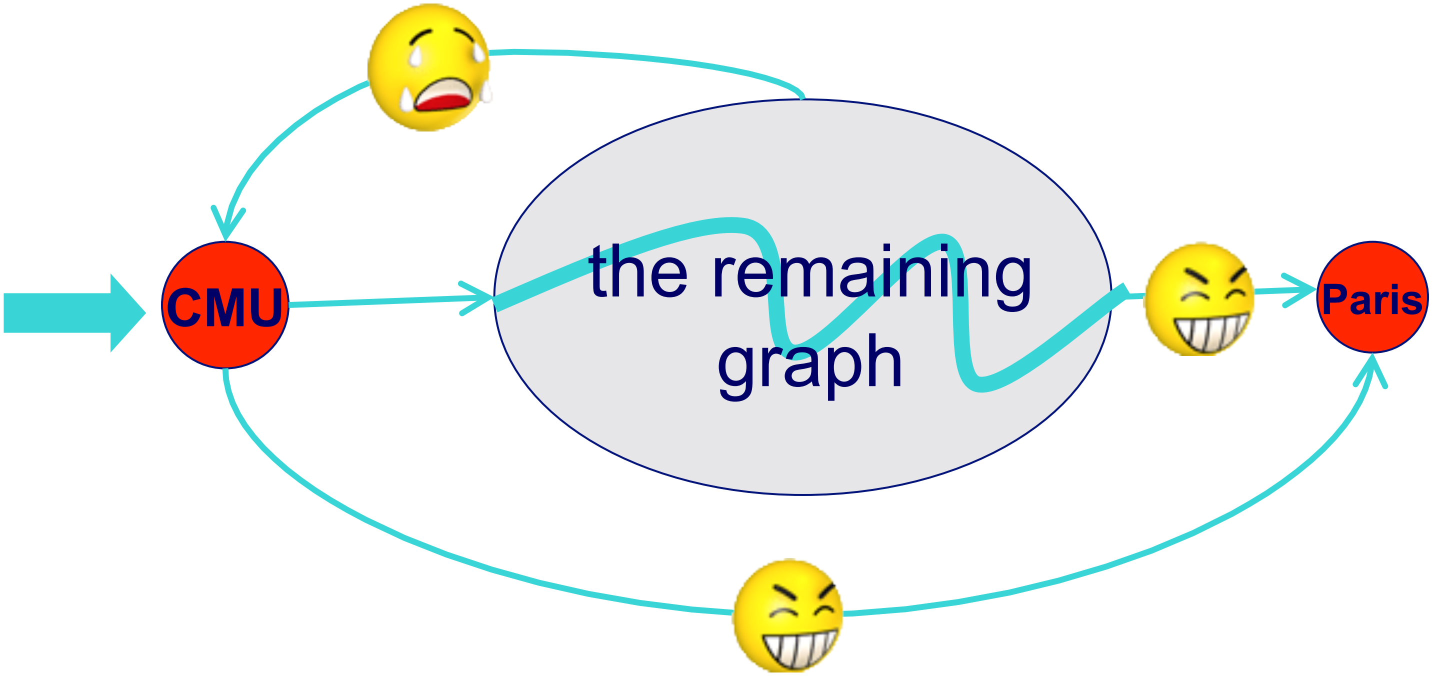
all paths from *i* to *j* with length 3

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

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)

Carnegie Mellon

details

Other Variants

- 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+]

Other Variants

details

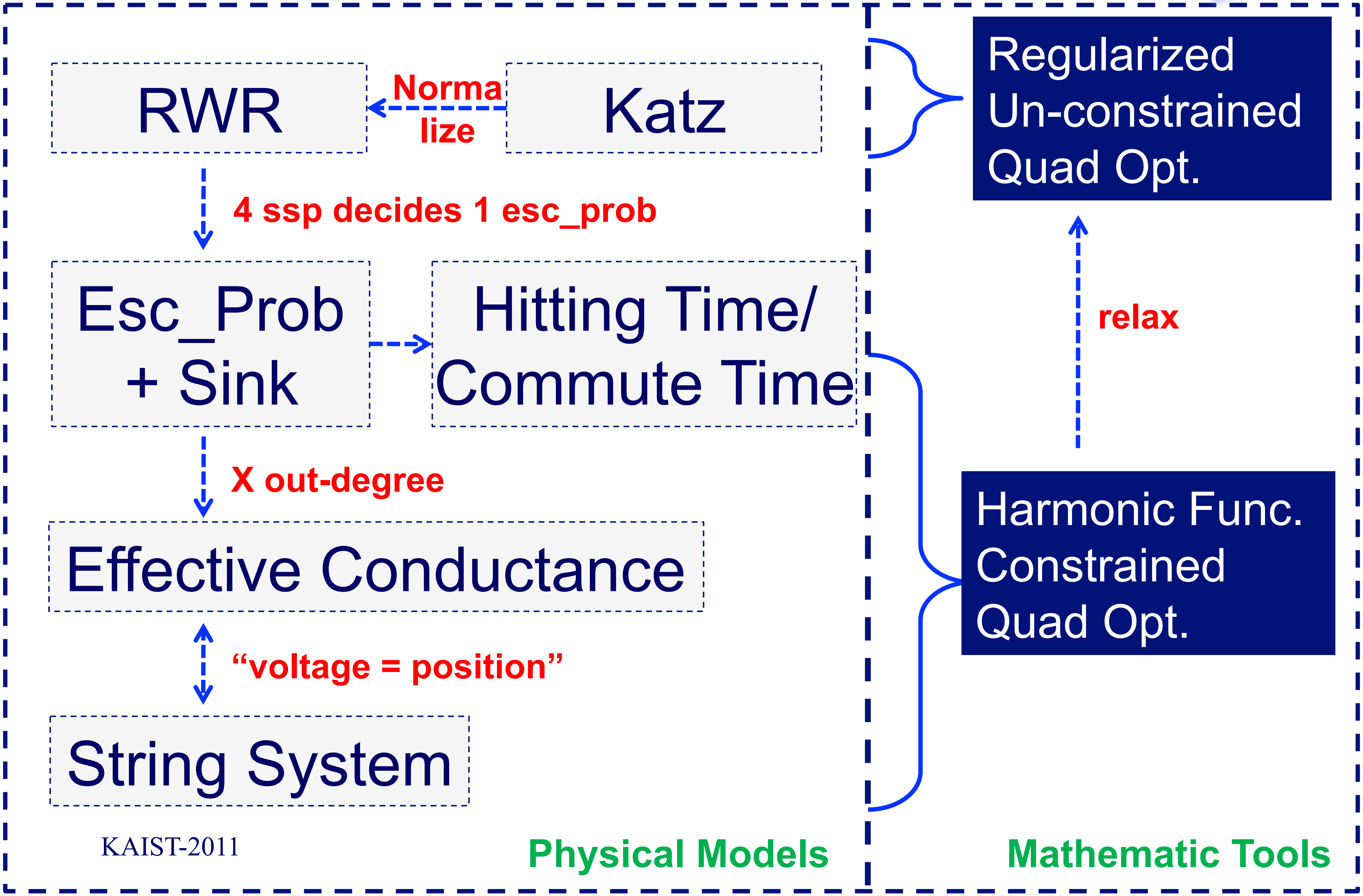
- Other measure by RWs
  - Community Time/Hitting Time [Fouss+]
  - SimRank [Jeh+]

All are “related to” or “similar to” random walk with restart!

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

Map of proximity measurements

details



RWR

Normalize

Katz

4 ssp decides 1 esc_prob

Esc_Prob + Sink

Hitting Time/ Commute Time

X out-degree

Effective Conductance

"voltage = position"

String System

Regularized Un-constrained Quad Opt.

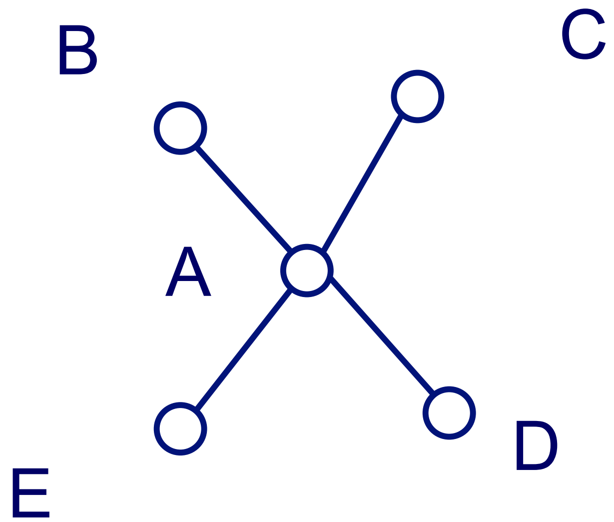
relax

Harmonic Func. Constrained Quad Opt.

Physical Models

Mathematic Tools

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

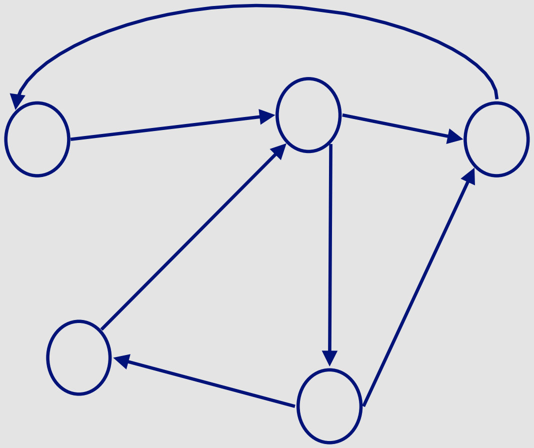
- Problem dfn and motivation
- Solution: Random walk with restarts
- ➡ **Efficient computation**
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

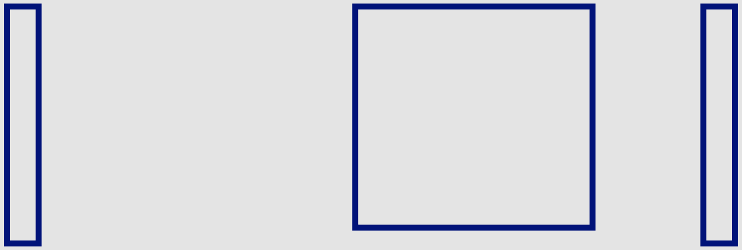
Reminder: PageRank

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

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

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





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

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

Carnegie Mellon

Computing RWR

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

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

Ranking vector

Adjacency matrix

Restart p

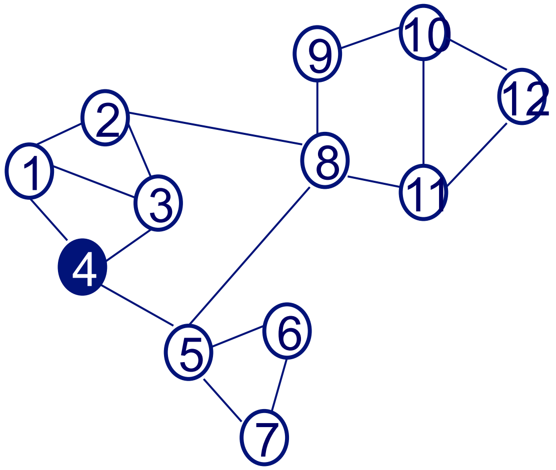
Starting vector

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

n x 1

n x n

n x 1



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

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

Query

Ranking vector

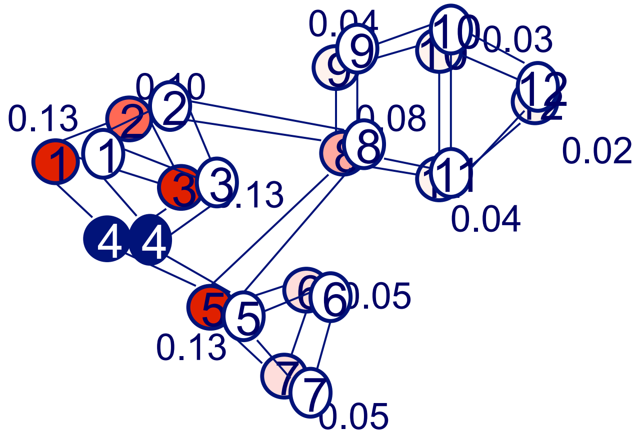
Adjacency matrix

Ranking vector

Starting vector

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

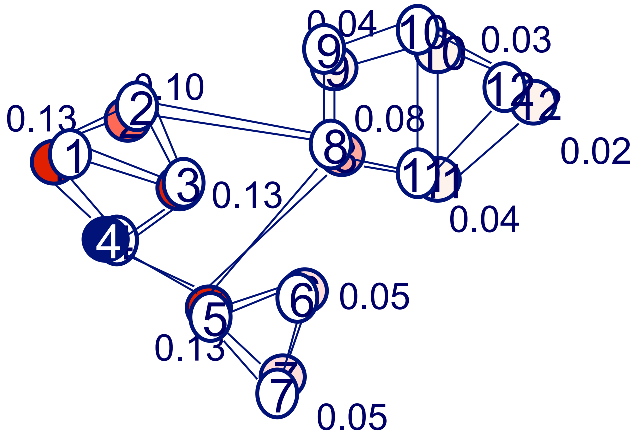
$$\underbrace{\begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix}}_{\vec{r}_i} = 0.9 \times \underbrace{\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\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



- No pre-computation/ light storage
- Slow on-line response O(mE)

PreCompute

R:



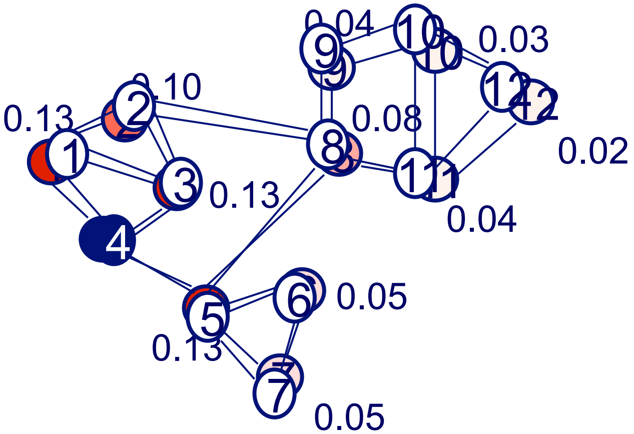
$$R = c \times Q$$

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

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

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

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

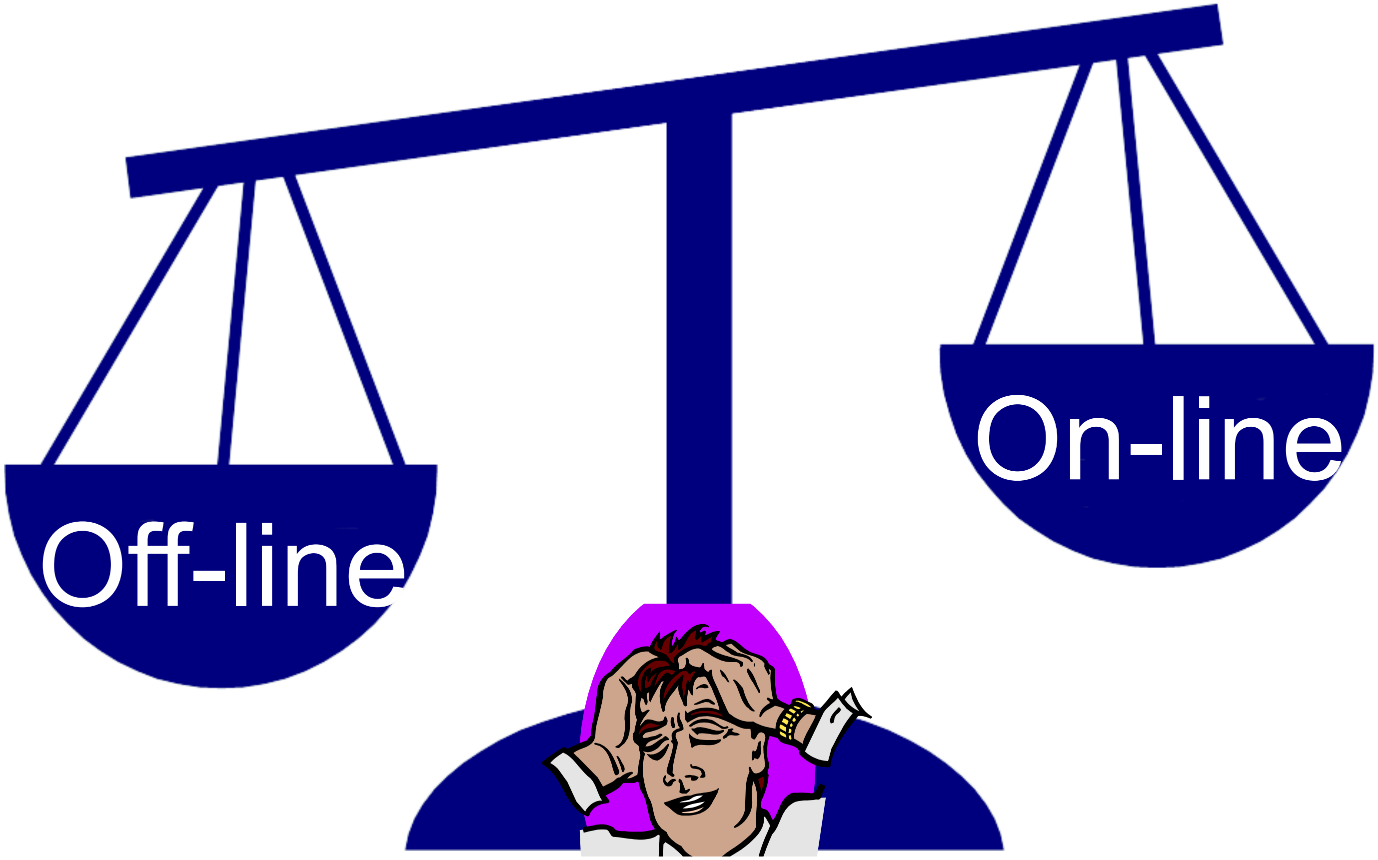


Fast on-line response

Heavy pre-computation/storage cost

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

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.

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}

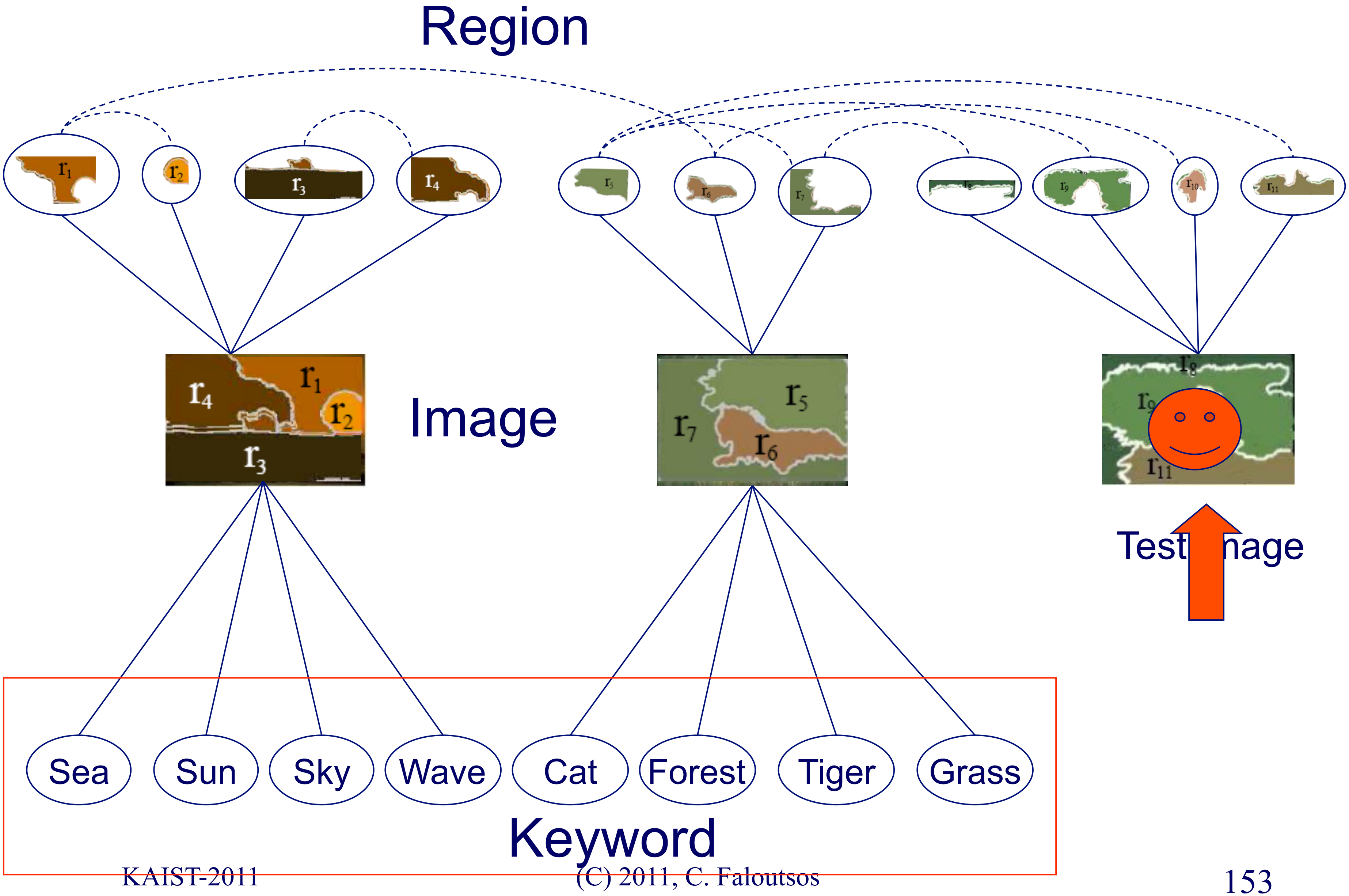
…

{Cat Forest Grass Tiger}

{?, ?, ?,}

A: Proximity!

[Pan+ KDD2004]



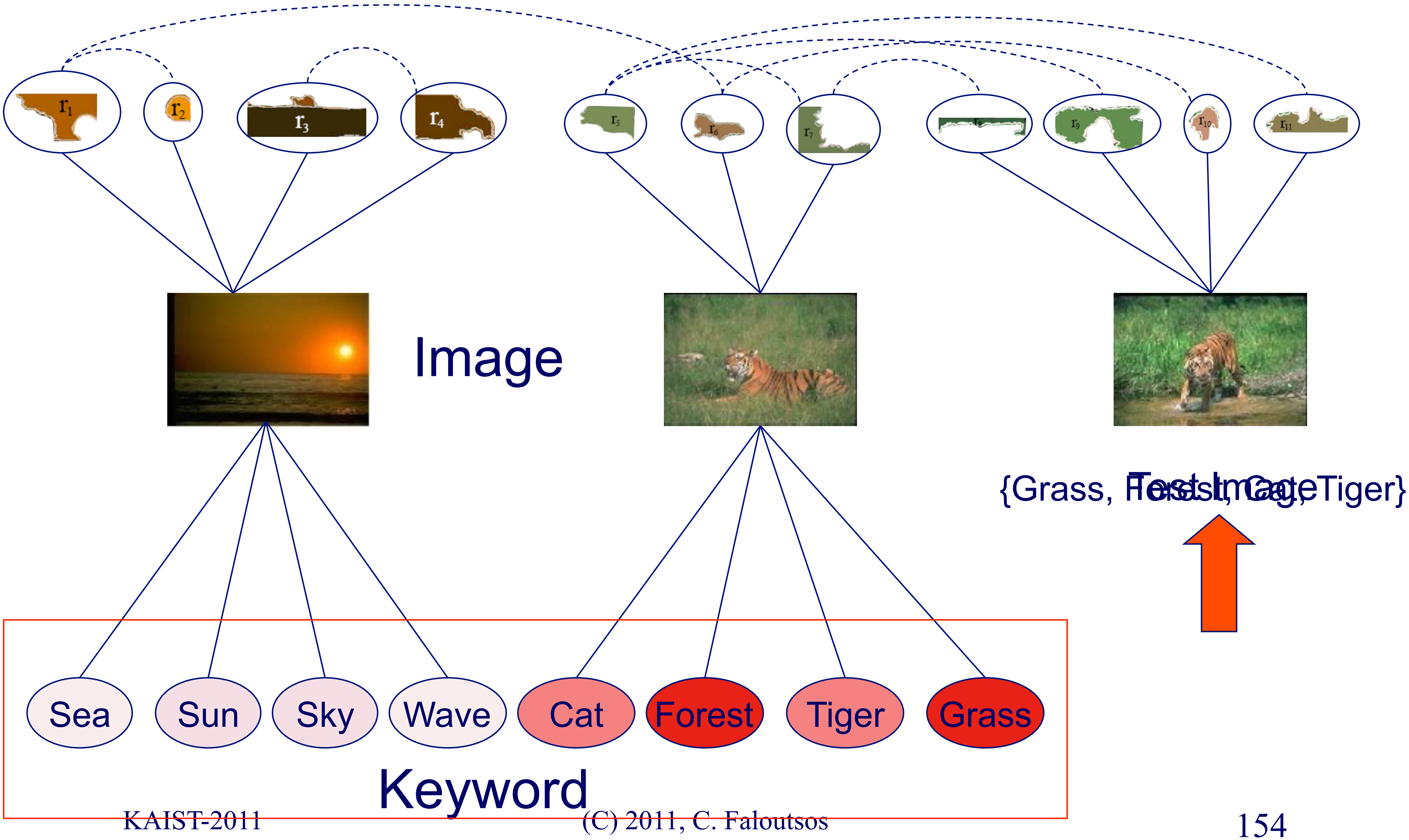
Region

Image

Test Image

Sea
Sun
Sky
Wave
Cat
Forest
Tiger
Grass

Keyword



Region

Image

Test Image

{Grass, Forest, Cat, Tiger}

Sea

Sun

Sky

Wave

Cat

Forest

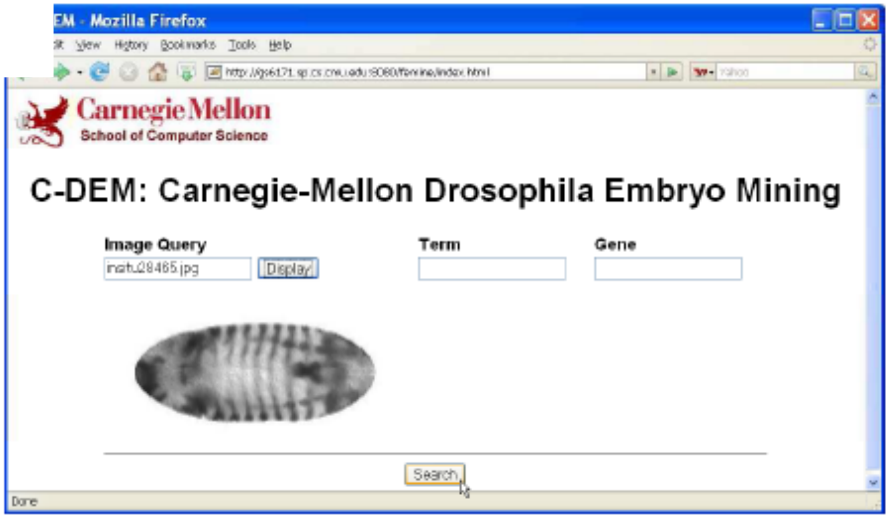
Tiger

Grass

Keyword

C-DEM (Screen-shot)

details



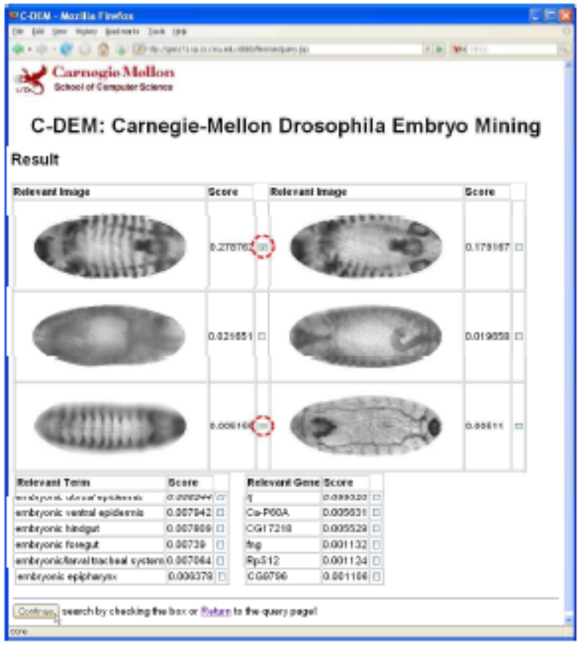
C-DEM: Carnegie-Mellon Drosophila Embryo Mining

Image Query insitu28465.jpg

Term

Gene

(a)



C-DEM: Carnegie-Mellon Drosophila Embryo Mining

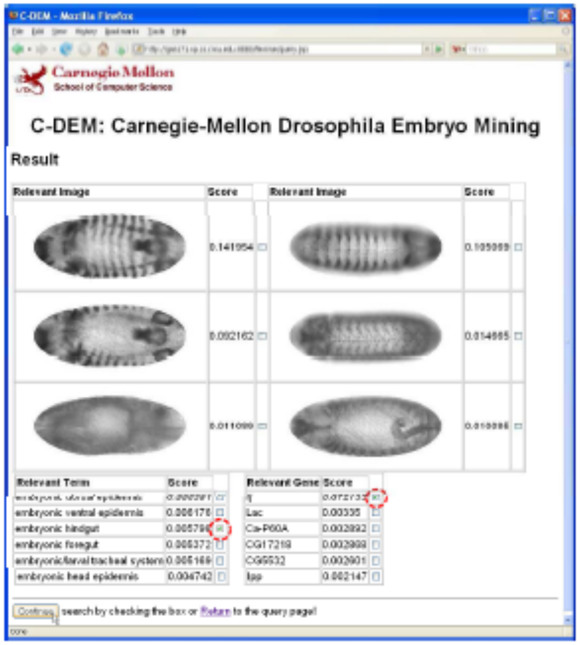
Result

Relevant Term	Score	
[illegible]	[illegible]	☐
embryonic ventral epidermis	0.007942	☐
embryonic hindgut	0.007809	☐
embryonic foregut	0.00739	☐
embryonic/larval tracheal system	0.007064	☐
embryonic epipharynx	0.006379	☐

Relevant Gene	Score	
[illegible]	[illegible]	☐
Ca-P60A	0.005631	☐
CG17218	0.005529	☐
fng	0.001132	☐
RpS12	0.001124	☐
CG8796	0.001106	☐

search by checking the box or Return to the query page!

(b)



C-DEM: Carnegie-Mellon Drosophila Embryo Mining

Result

Relevant Term	Score	
[illegible]	[illegible]	☐
embryonic ventral epidermis	0.006176	☐
embryonic hindgut	0.005796	☑
embryonic foregut	0.005372	☐
embryonic/larval tracheal system	0.005169	☐
embryonic head epidermis	0.004742	☐

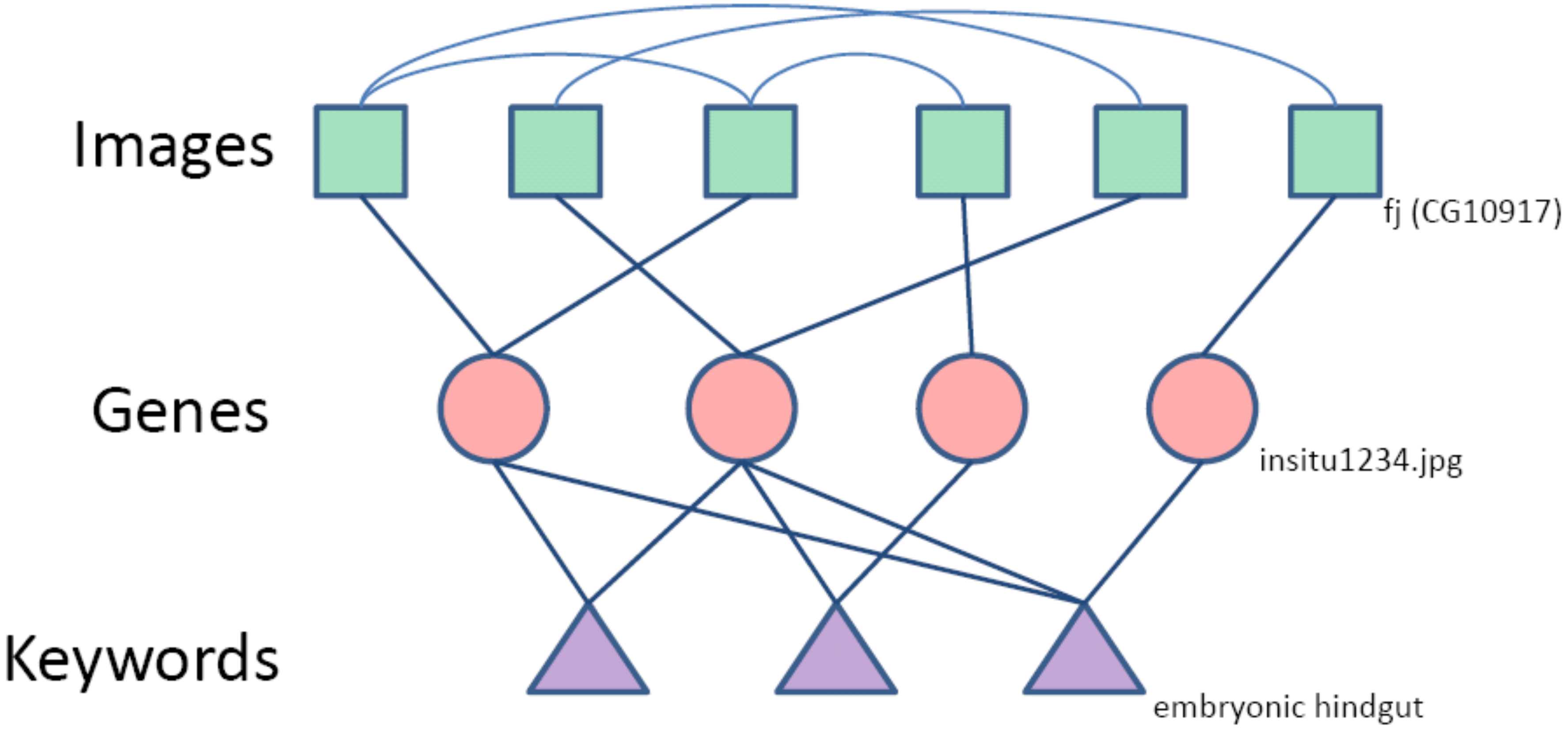
Relevant Gene	Score	
[illegible]	[illegible]	☑
Lac	0.00335	☐
Ca-P60A	0.002892	☐
CG17218	0.002868	☐
CG6532	0.002601	☐
lpp	0.002147	☐

search by checking the box or Return to the query page!

(c)

details

C-DEM: Multi-Modal Query System for Drosophila Embryo Databases [Fan+ VLDB 2008]



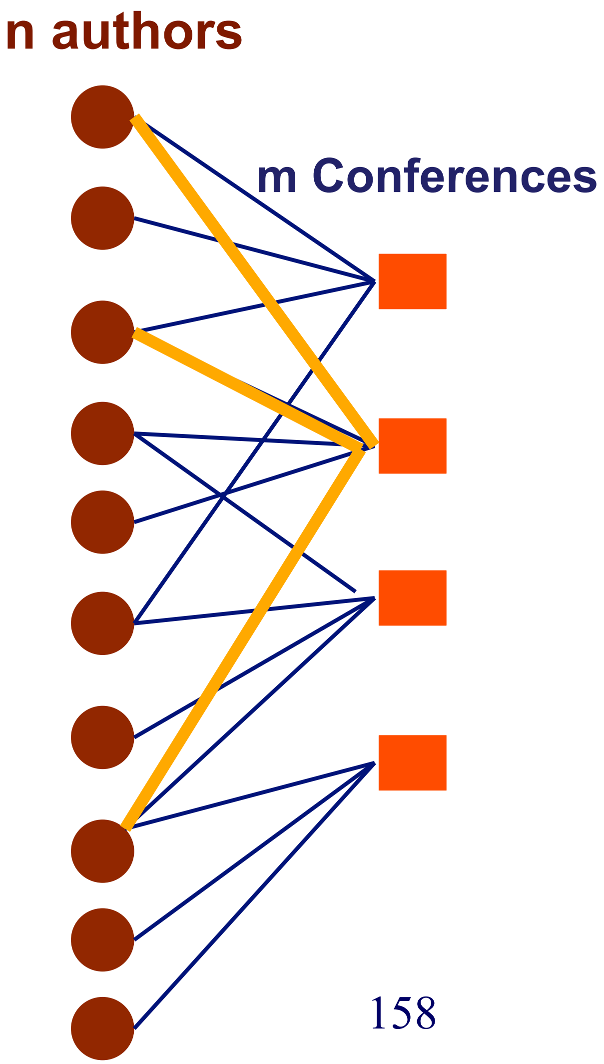
Detailed outline

- 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

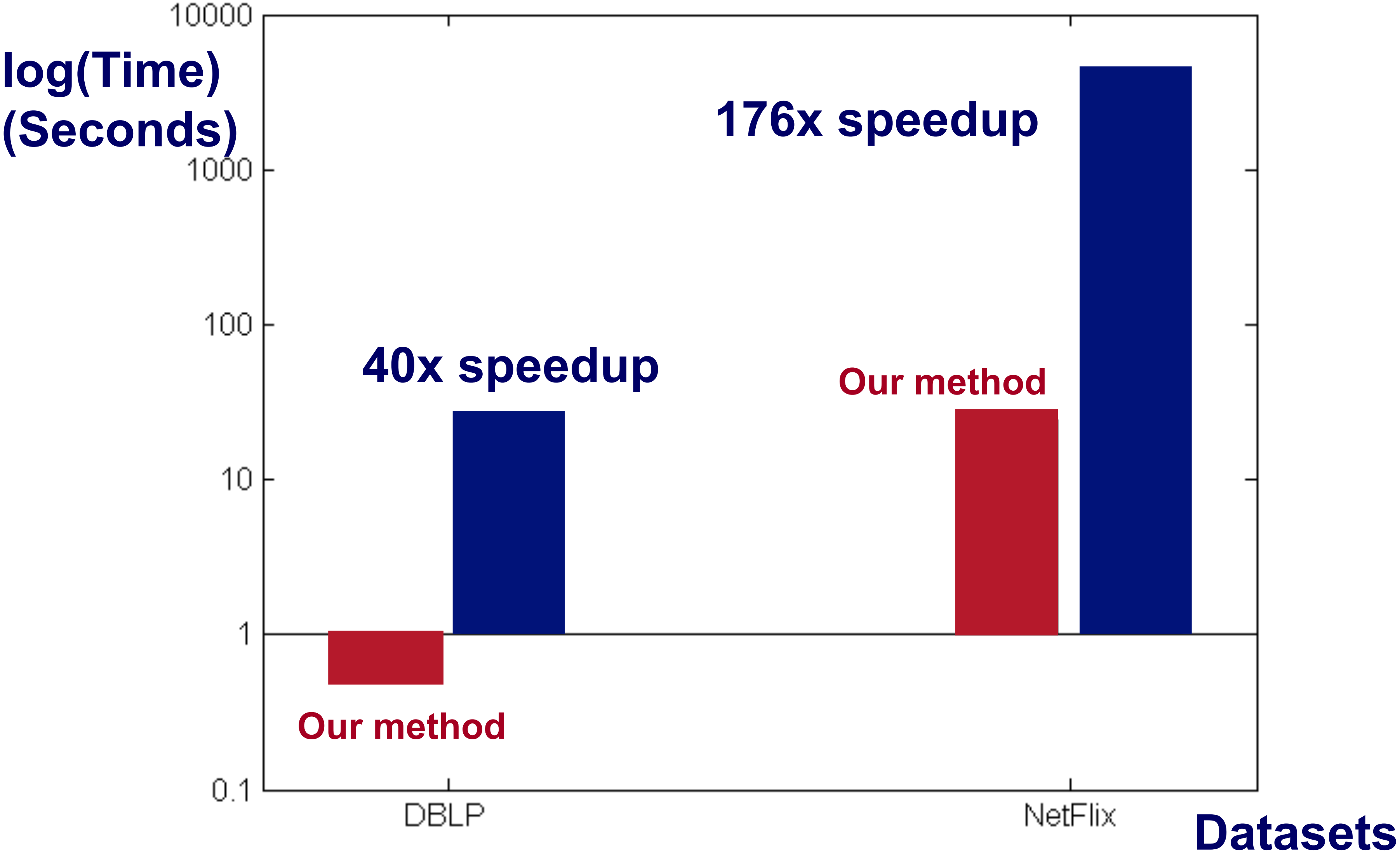
Involves n’ authors, m’ confs.



Solution:

- Use Sherman-Morrison Lemma 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	CIKM	KDD	ICDM
ICDCS	ICDCS	SIGMOD	KDD
SIGMETRICS	ICDE	ICDM	ICDE
PDIS	SIGMETRICS	CIKM	SDM
VLDB	ICMCS	ICDCS	VLDB
1992	1997	2002	2007

↓

Databases
Performance
Distributed Sys.

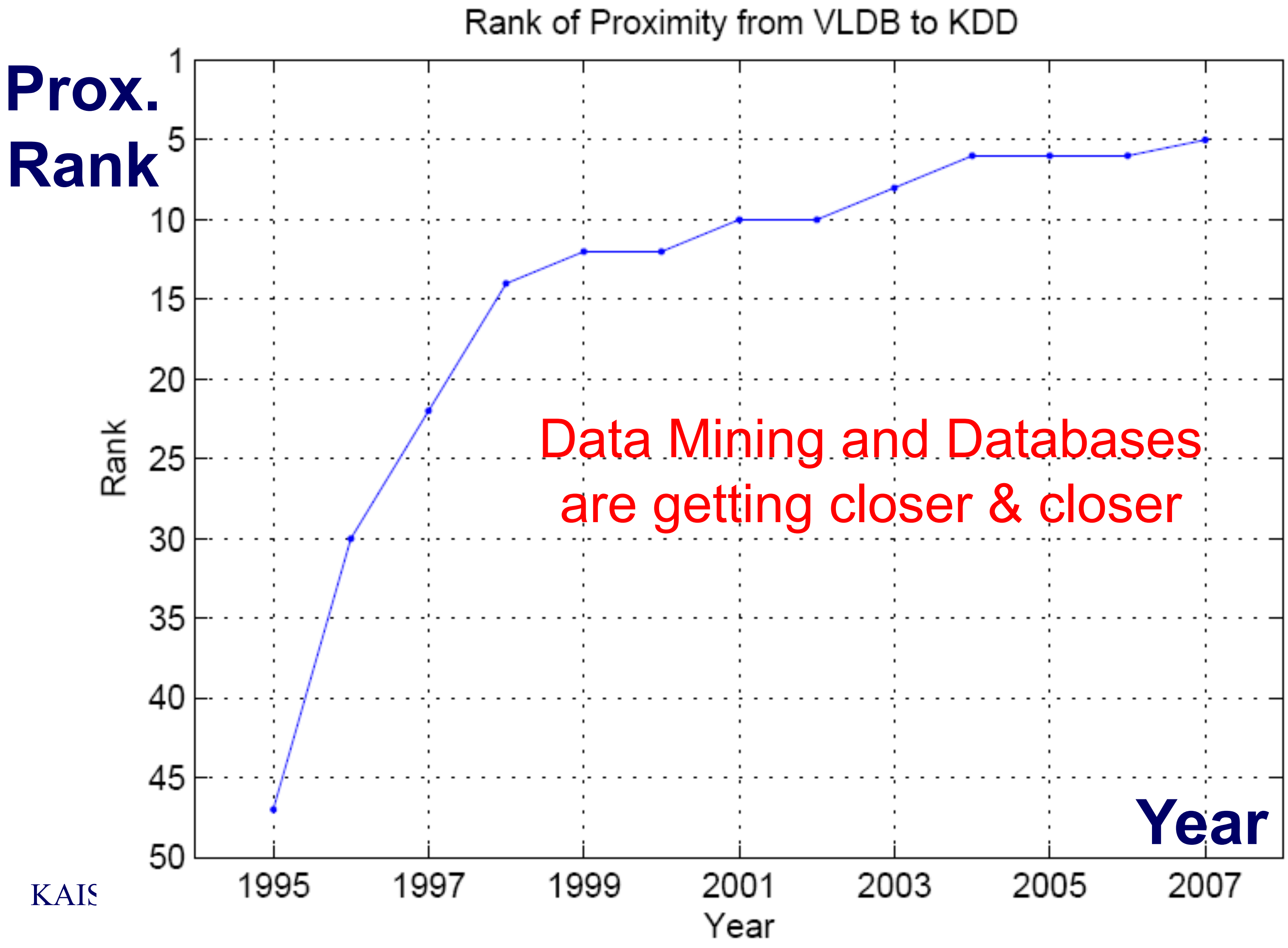
DBLP: (Au. x Conf.)

- 400k aus,
- 3.5k confs
- 20 yrs

↓

Databases
Data Mining

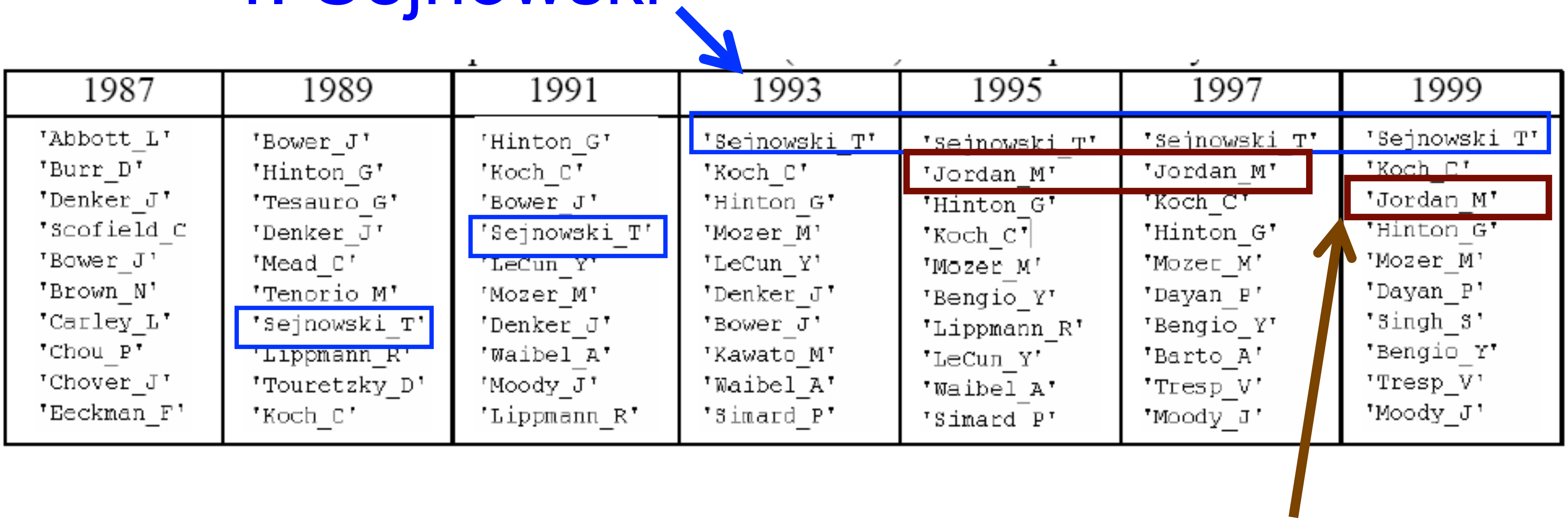
KDD's Rank wrt. VLDB over years



Data Mining and Databases are getting closer & closer

Carnegie Mellon

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 $\vec{r_i} = c * \tilde{\mathbf{W}}\vec{r_i} + (1-c)\vec{e_i}$
  - 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

Outline

- Introduction – Motivation
- Task 1: Node importance
- Task 2: Recommendations & proximity
- **Task 3: Connection sub-graphs**
- Conclusions

Detailed outline

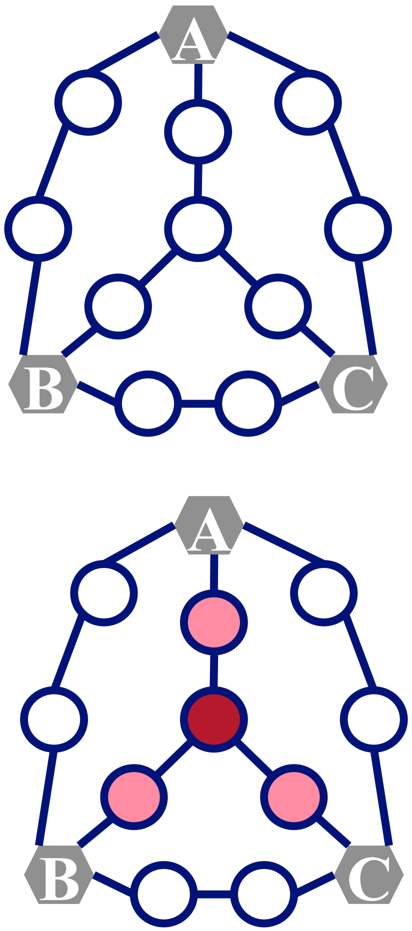
- Problem definition
- Solution
- Results

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?)

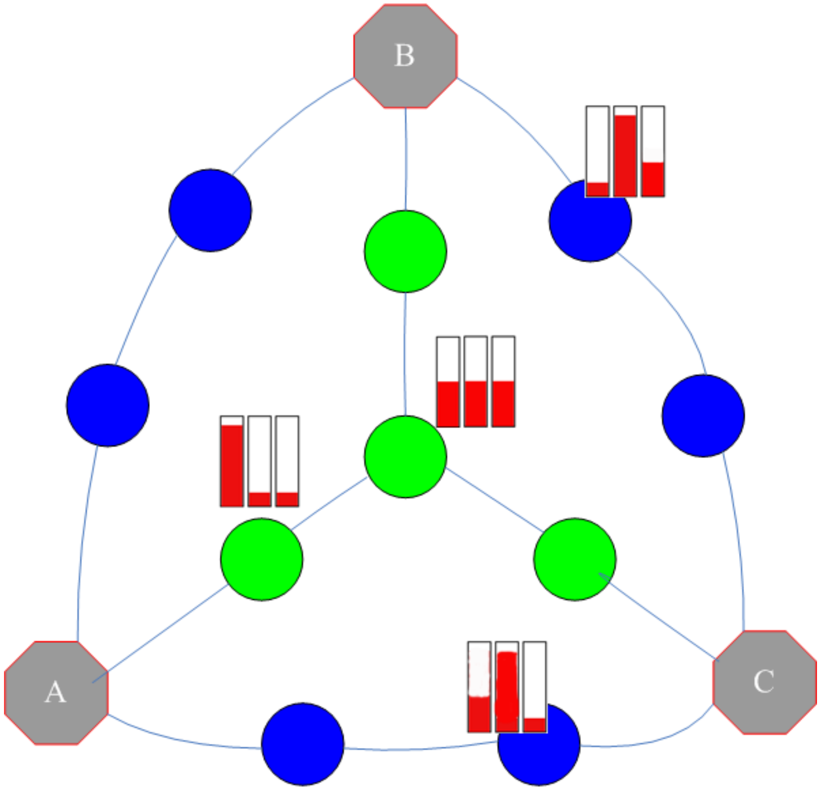
Challenges in Ceps

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

Carnegie Mellon

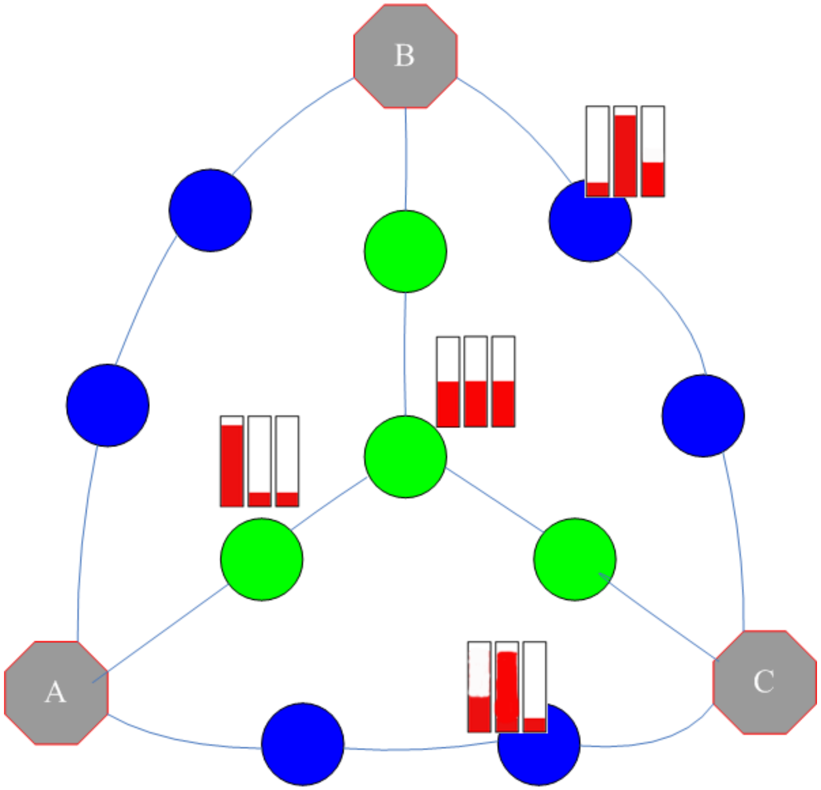
AND: Combine Scores

- Q: How to combine scores?



AND: Combine Scores

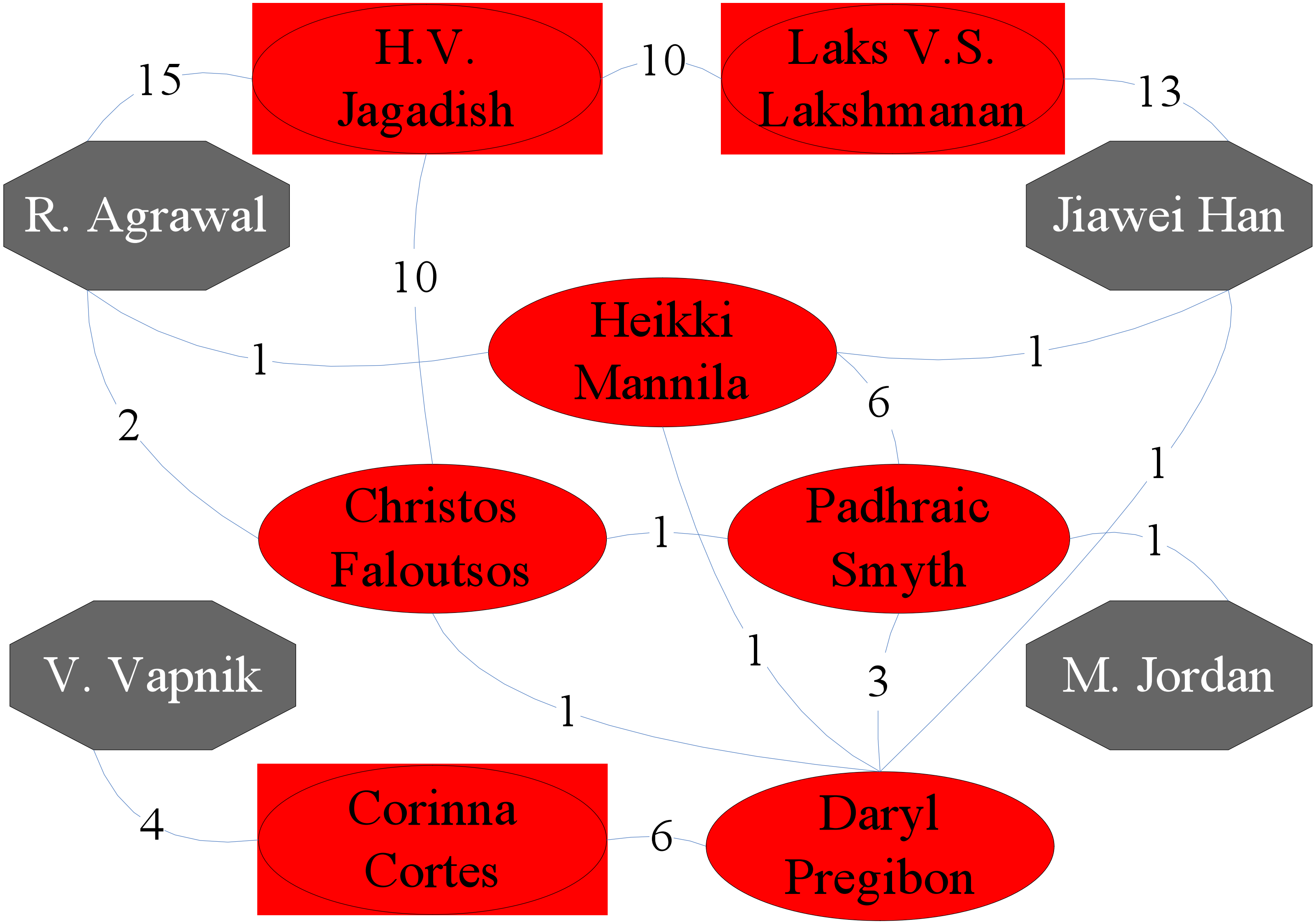
- Q: How to combine scores?
- A: Multiply
- …= prob. 3 random particles coincide on node *j*



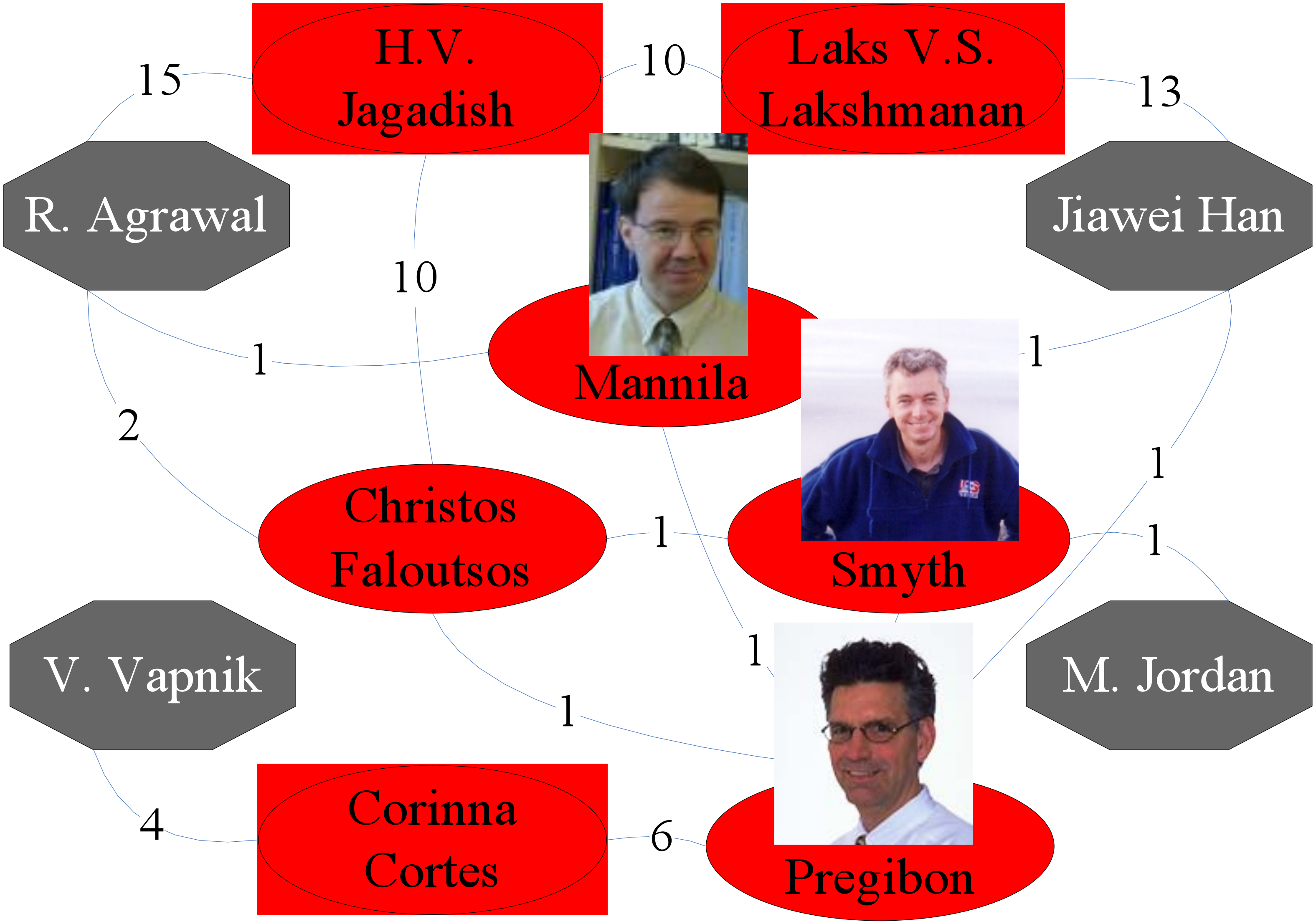
Detailed outline

- Problem definition
- Solution
- ➡ Results

Case Study: AND query



Case Study: AND query



Conclusions

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

Overall conclusions

- SVD: a powerful tool
  - HITS/ pageRank
  - (dimensionality reduction)
- Proximity: Random Walk with Restarts
  - Recommendation systems
  - Auto-captioning
  - Center-Piece Subgraphs