Large Graph Mining: Patterns, Tools and Case Studies

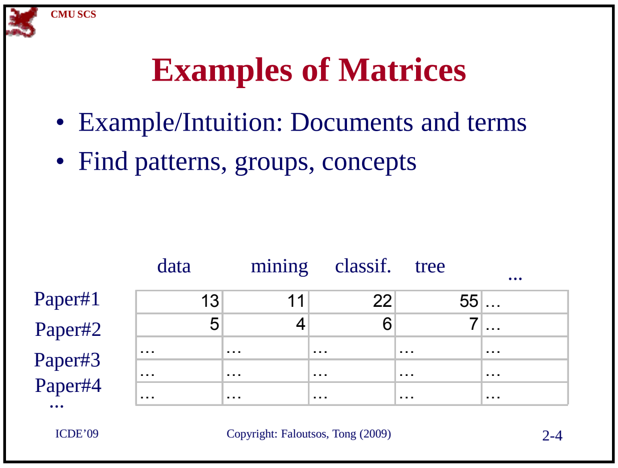
Christos Faloutsos
Hanghang Tong
CMU

Outline

- Part 1: Patterns
- ➡ Part 2: Matrix and Tensor Tools
- Part 3: Proximity
- Part 4: Case Studies

Outline: Part 2

- Matrix Tools
  - ➡ SVD, PCA
  - HITS, PageRank
  - Co-clustering
- Tensor Tools

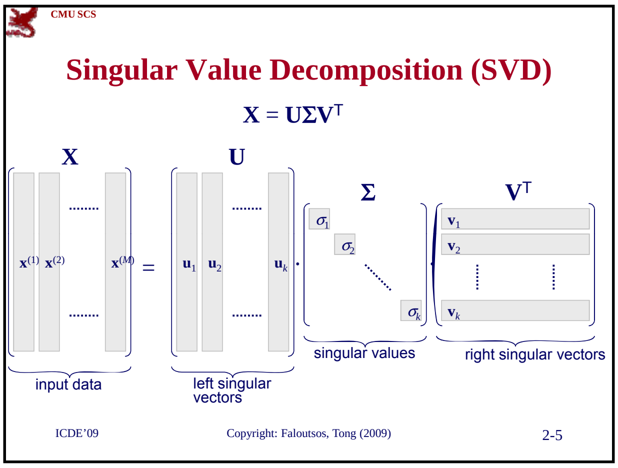


CMU SCS

Examples of Matrices

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

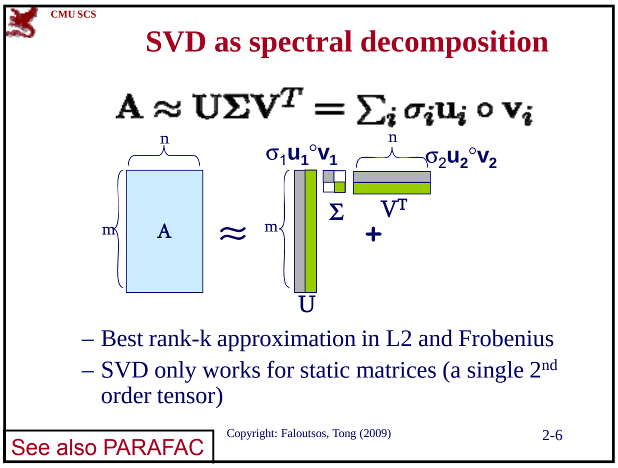
	data	mining	classif.	tree	...
Paper#1	13	11	22	55	…
Paper#2	5	4	6	7	…
Paper#3	…	…	…	…	…
Paper#4	…	…	…	…	…
…	…	…	…	…	…



CMU SCS

Singular Value Decomposition (SVD)

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{T}$$



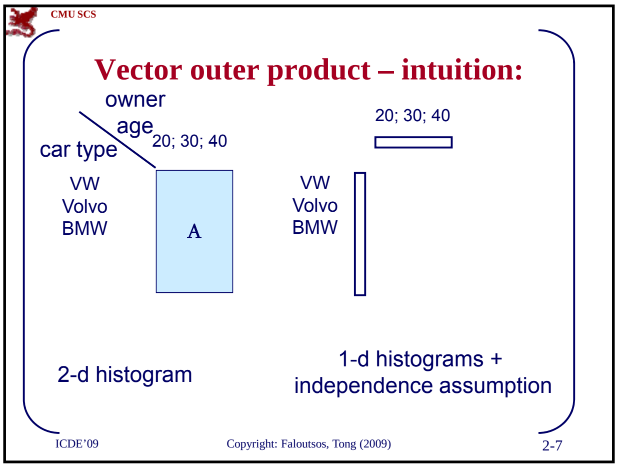
CMU SCS

SVD as spectral decomposition

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

- Best rank-k approximation in L2 and Frobenius
- SVD only works for static matrices (a single 2nd order tensor)

See also PARAFAC

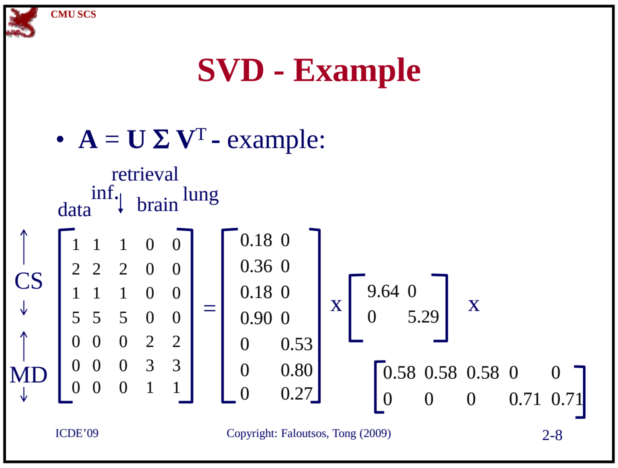


CMU SCS

Vector outer product – intuition:

2-d histogram

1-d histograms +
independence assumption

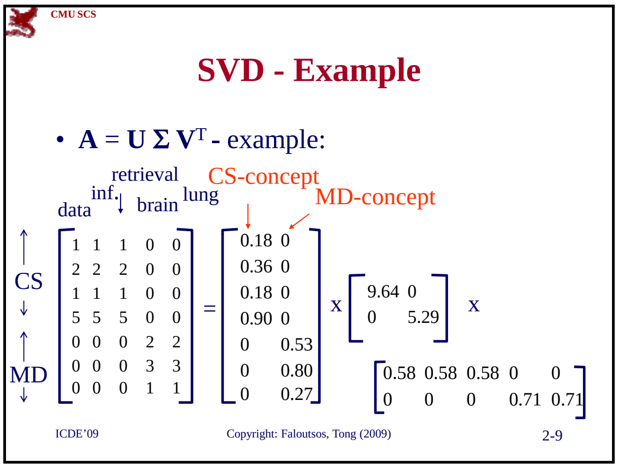


CMU SCS

SVD - Example

- **A** = **U** **Σ** **V**^{T} - example:

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

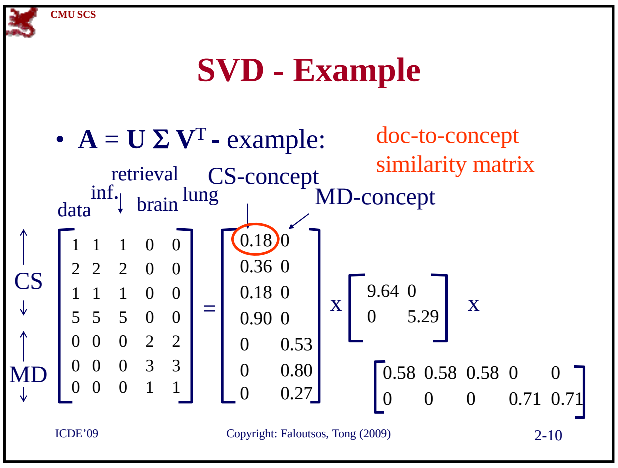


CMU SCS

SVD - Example

- **A** = **U** **Σ** **V**^{T} - example:

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



CMU SCS

SVD - Example

- **A** = **U** **Σ** **V**^{T} - example:

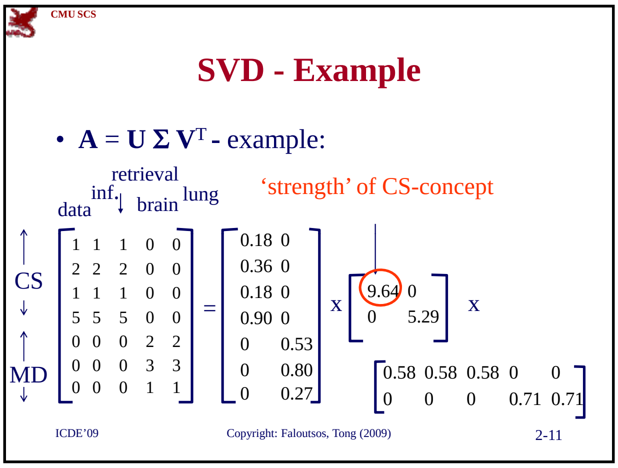
doc-to-concept similarity matrix

CS-concept

MD-concept

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

ICDE'09 Copyright: Faloutsos, Tong (2009) 2-10



CMU SCS

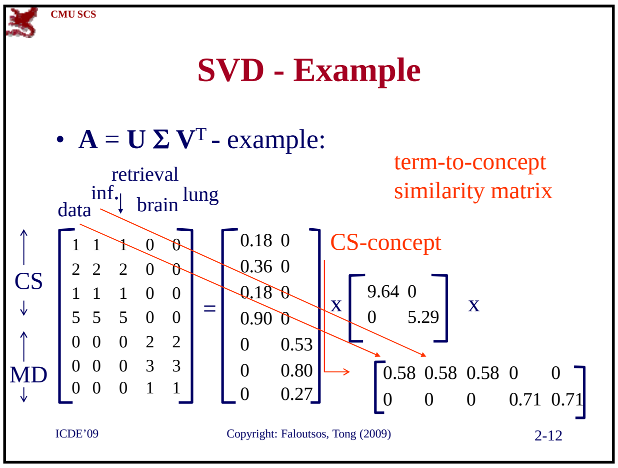
SVD - Example

- **A** = **U** **Σ** **V**^{T} - example:

'strength' of CS-concept

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

ICDE'09 Copyright: Faloutsos, Tong (2009) 2-11



CMU SCS

SVD - Example

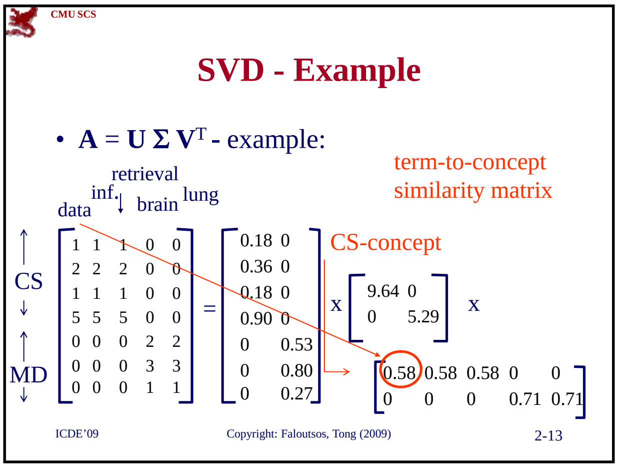
- **A** = **U** **Σ** **V**^{T} - example:

term-to-concept similarity matrix

CS-concept

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

ICDE'09 Copyright: Faloutsos, Tong (2009) 2-12



CMU SCS

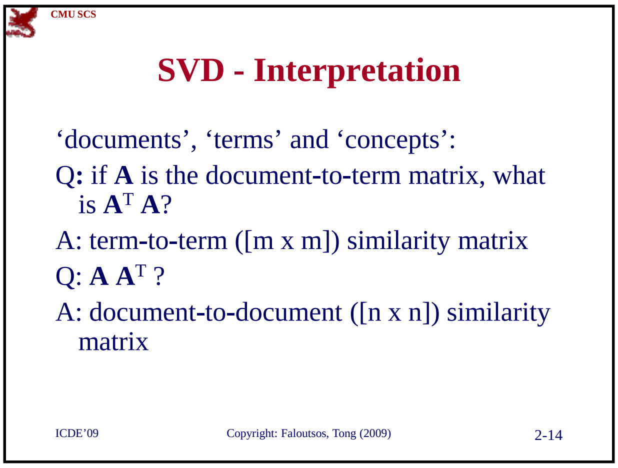
SVD - Example

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

term-to-concept similarity matrix

CS-concept

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



CMU SCS

SVD - Interpretation

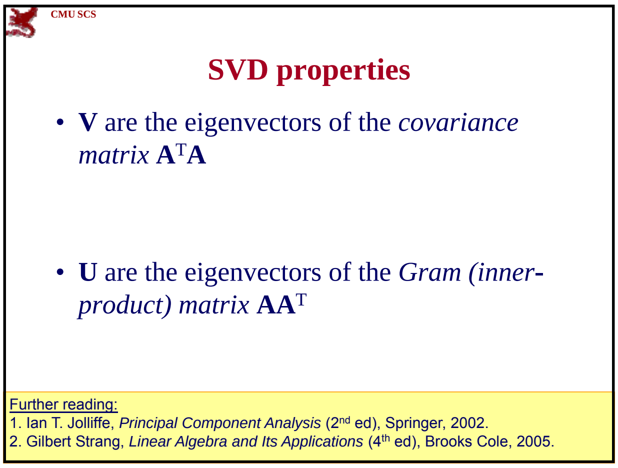
'documents', 'terms' and 'concepts':

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

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

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

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



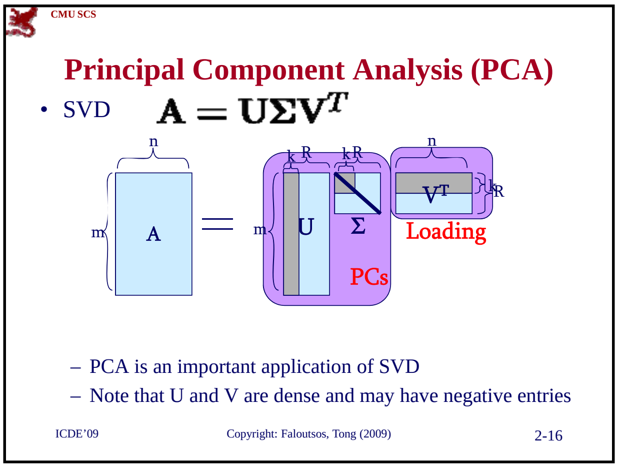
CMU SCS

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:

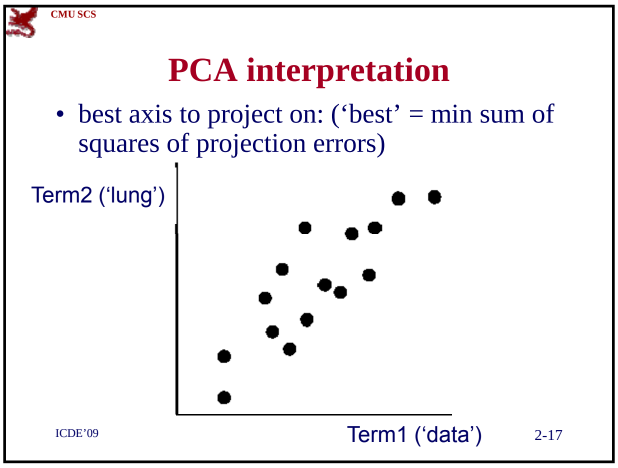
1. Ian T. Jolliffe, *Principal Component Analysis* (2nd ed), Springer, 2002.
2. Gilbert Strang, *Linear Algebra and Its Applications* (4th ed), Brooks Cole, 2005.



Principal Component Analysis (PCA)

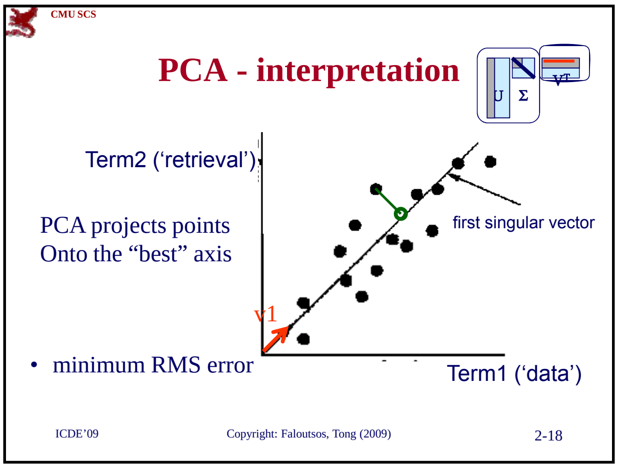
- SVD $\mathbf{A} = \mathbf{U \Sigma V}^T$

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



PCA interpretation

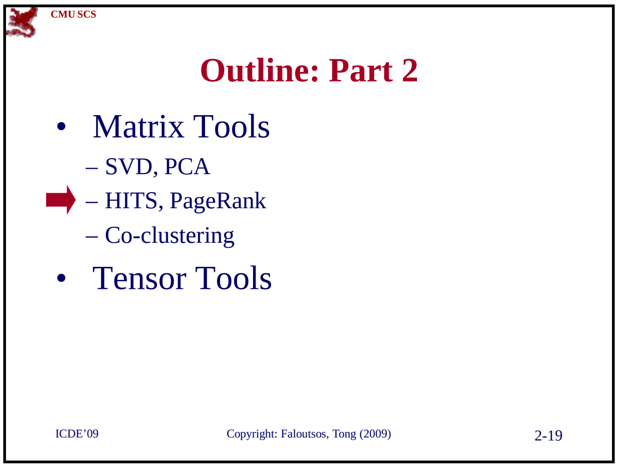
- best axis to project on: ('best' = min sum of squares of projection errors)



PCA - interpretation

PCA projects points Onto the "best" axis

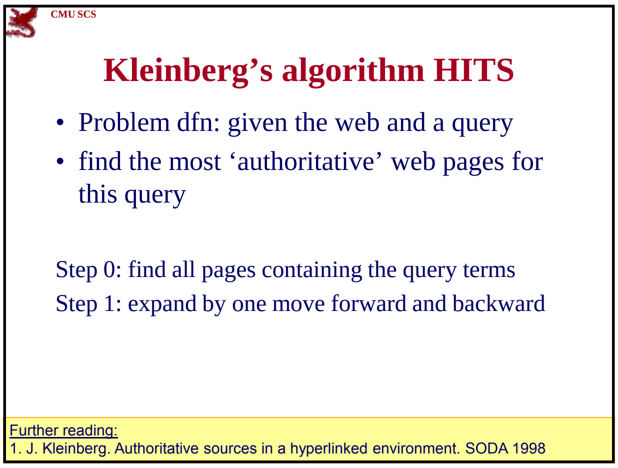
- minimum RMS error



CMU SCS

Outline: Part 2

- Matrix Tools
  - – SVD, PCA
  - ➡ – HITS, PageRank
  - – Co-clustering
- Tensor Tools



CMU SCS

Kleinberg's algorithm HITS

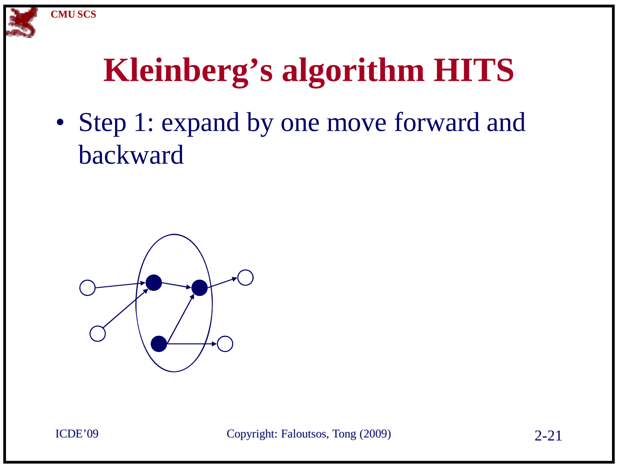
- Problem dfn: given the web and a query
- find the most 'authoritative' web pages for this query

Step 0: find all pages containing the query terms

Step 1: expand by one move forward and backward

Further reading:

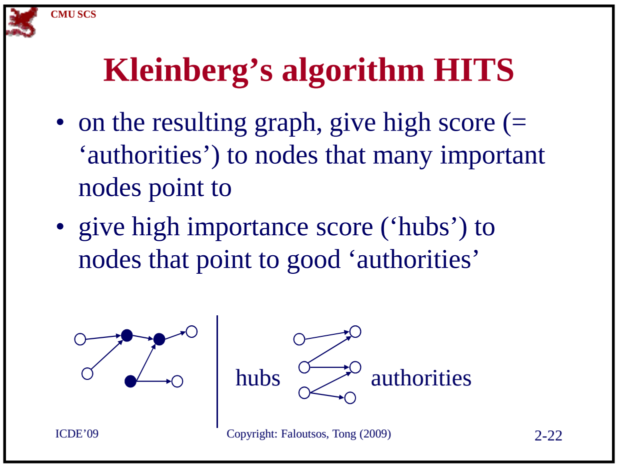
1. J. Kleinberg. Authoritative sources in a hyperlinked environment. SODA 1998



CMU SCS

Kleinberg's algorithm HITS

- Step 1: expand by one move forward and backward

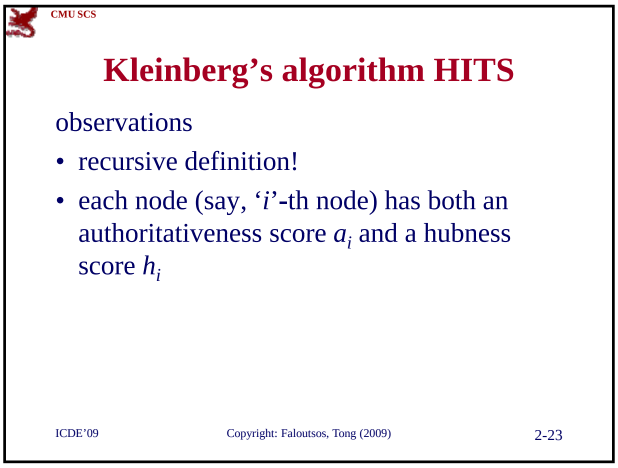


CMU SCS

Kleinberg’s algorithm HITS

- on the resulting graph, give high score (= ‘authorities’) to nodes that many important nodes point to
- give high importance score (‘hubs’) to nodes that point to good ‘authorities’

ICDE’09 Copyright: Faloutsos, Tong (2009) 2-22



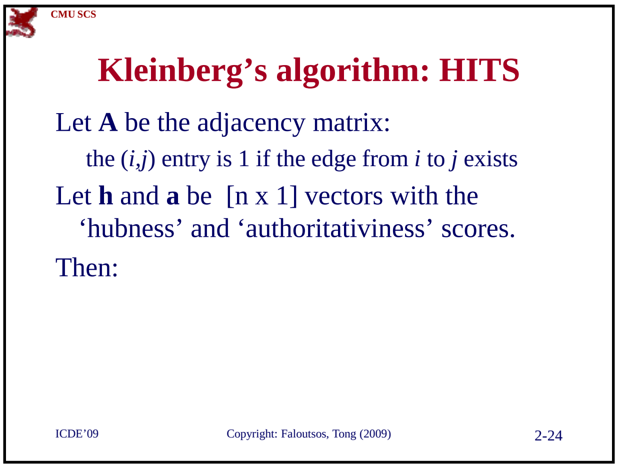
CMU SCS

Kleinberg’s algorithm HITS

observations

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

ICDE’09 Copyright: Faloutsos, Tong (2009) 2-23



CMU SCS

Kleinberg’s algorithm: HITS

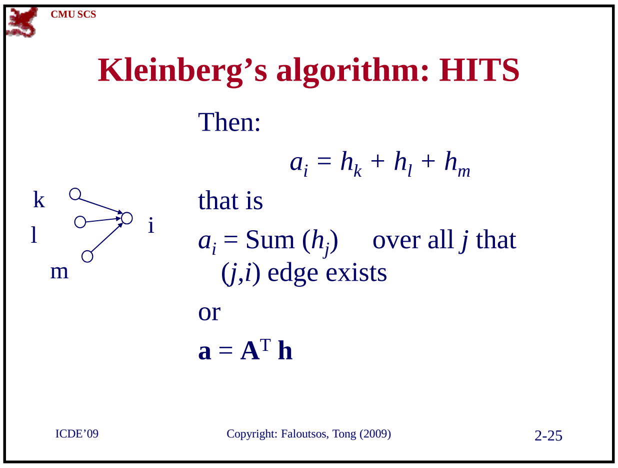
Let **A** be the adjacency matrix:

the (i,j) entry 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:

ICDE’09 Copyright: Faloutsos, Tong (2009) 2-24



Kleinberg's algorithm: HITS

Then:

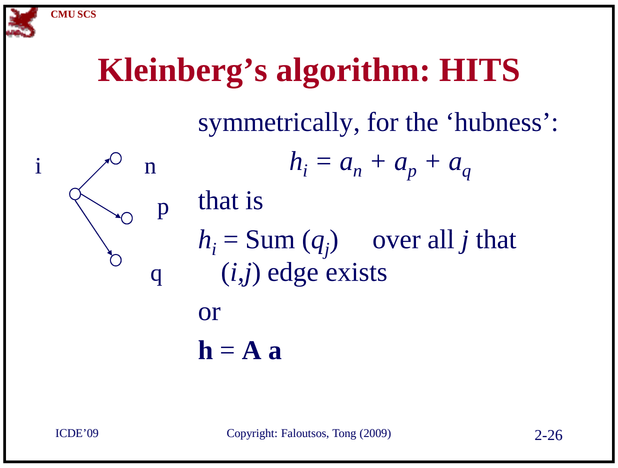
$$a_i = h_k + h_l + h_m$$

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: HITS

symmetrically, for the 'hubness':

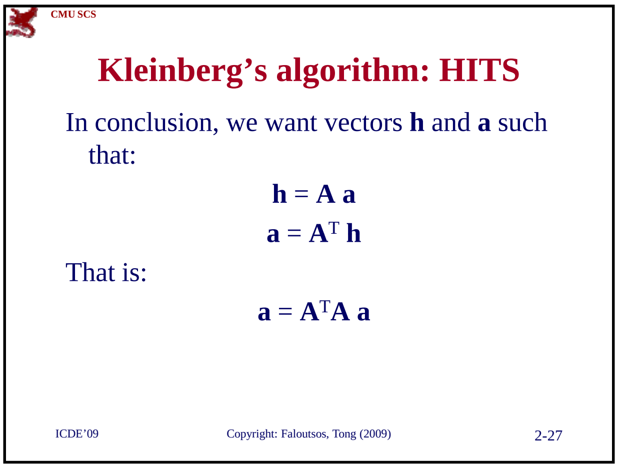
$$h_i = a_n + a_p + a_q$$

that is

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

or

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



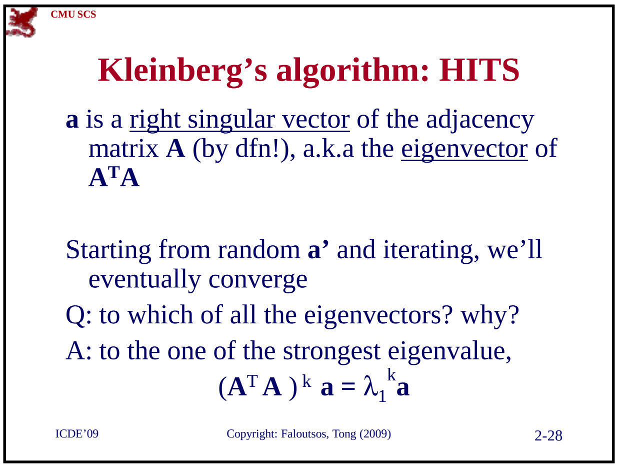
Kleinberg's algorithm: HITS

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

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

That is:

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



CMU SCS

Kleinberg's algorithm: HITS

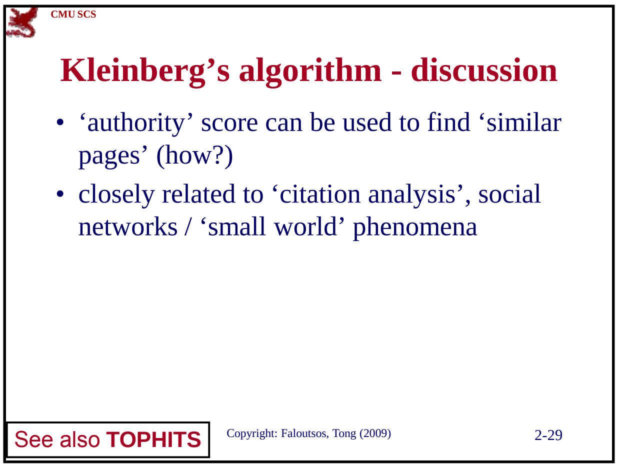
a is a right singular vector of the adjacency matrix **A** (by dfn!), a.k.a the eigenvector of $\mathbf{A}^T\mathbf{A}$

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

Q: to which of all the eigenvectors? why?

A: to the one of the strongest eigenvalue,

$$(\mathbf{A}^T\mathbf{A})^k\ \mathbf{a} = \lambda_1^k\mathbf{a}$$

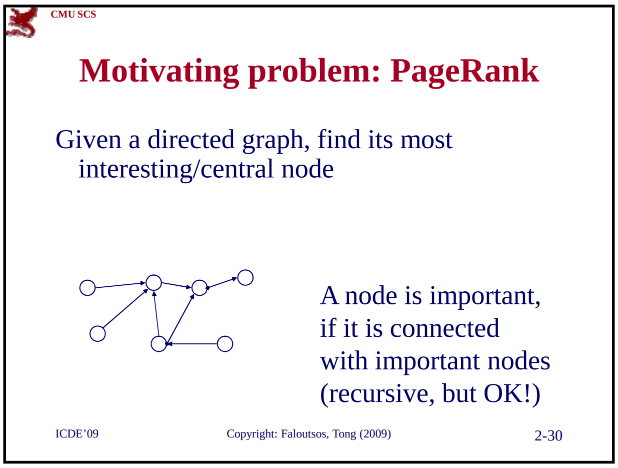


CMU SCS

Kleinberg's algorithm - discussion

- 'authority' score can be used to find 'similar pages' (how?)
- closely related to 'citation analysis', social networks / 'small world' phenomena

See also **TOPHITS**

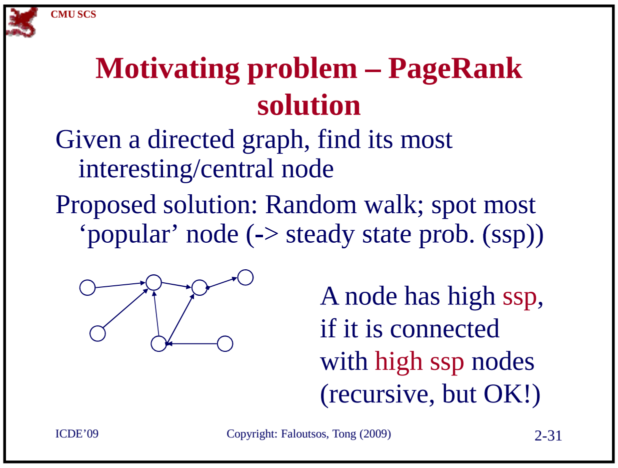


CMU SCS

Motivating 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!)



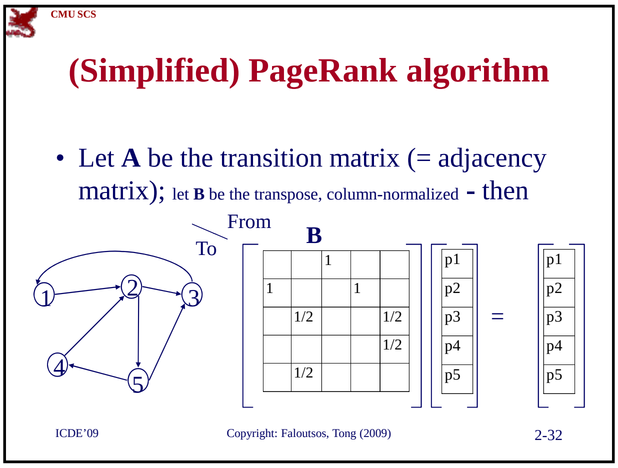
CMU SCS

Motivating problem – PageRank solution

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

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

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



CMU SCS

(Simplified) PageRank algorithm

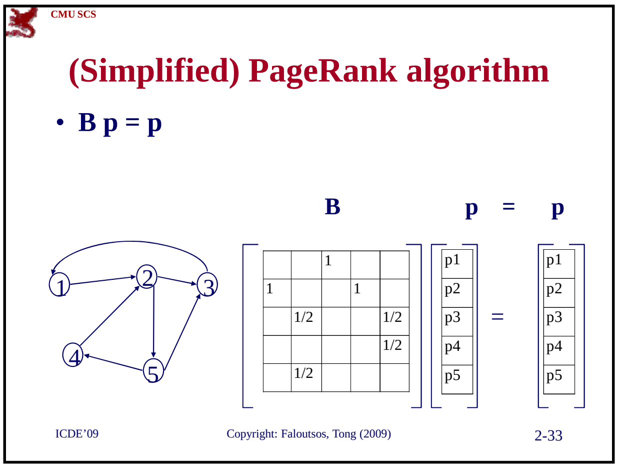
- Let **A** be the transition matrix (= adjacency matrix); let **B** be the transpose, column-normalized - then

From

To

B

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



CMU SCS

(Simplified) PageRank algorithm

- **B p = p**

B **p** = **p**

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

(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

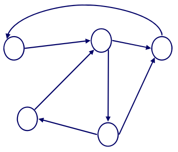
Why? To make the matrix irreducible

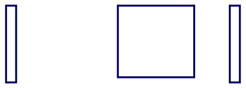
Full Algorithm

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

p = c **B p** + (1-c)/n **1** =>

p = (1-c)/n [**I** - c **B**]$^{-1}$ **1**



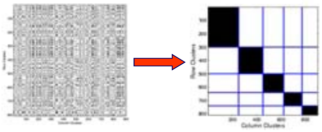


Outline: Part 2

- Matrix Tools
  - SVD, PCA
  - HITS, PageRank
  - ➡ Co-clustering
- Tensor Tools

Co-clustering

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

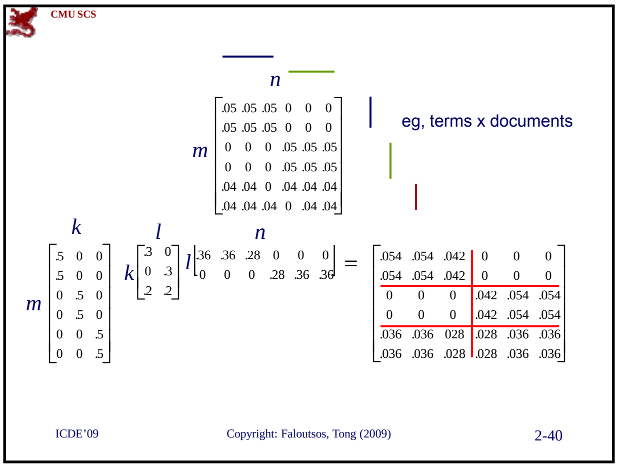


Co-clustering

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

Reference:

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

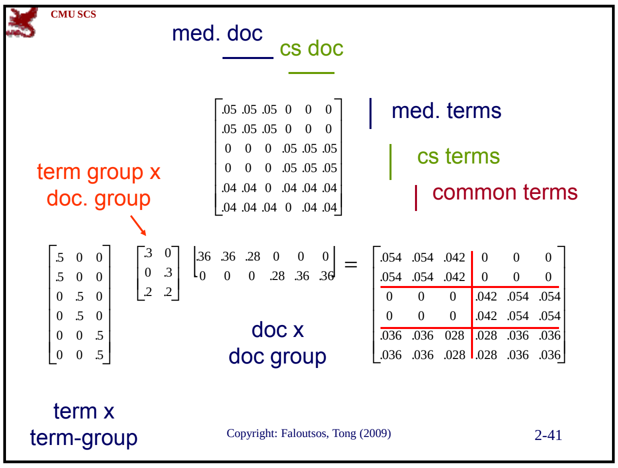


$$m\overset{n}{\begin{bmatrix} .05 & .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ .04 & .04 & 0 & .04 & .04 & .04 \\ .04 & .04 & .04 & 0 & .04 & .04 \end{bmatrix}}$$

eg, terms x documents

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

ICDE’09 Copyright: Faloutsos, Tong (2009)



med. doc

cs doc

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

med. terms

cs terms

common terms

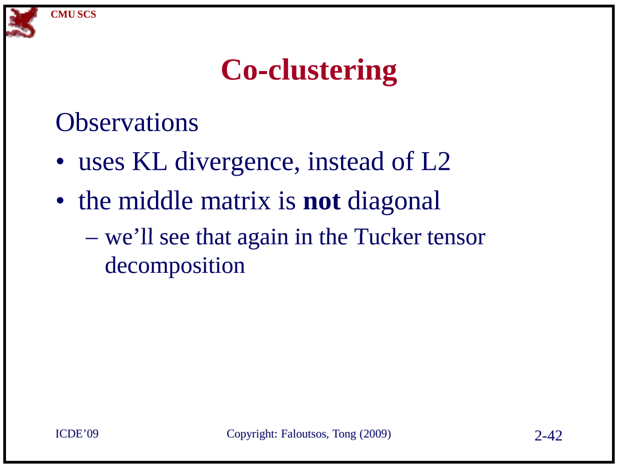
term group x doc. group

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

doc x doc group

term x term-group

Copyright: Faloutsos, Tong (2009)



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

ICDE’09 Copyright: Faloutsos, Tong (2009)

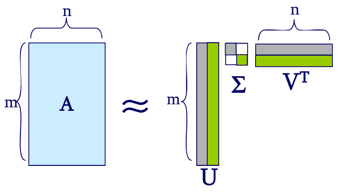
Outline: Part 2

- Matrix Tools
- Tensor Tools
  - Tensor Basics
  - Tucker
    - Tucker 1
    - Tucker 2
    - Tucker 3
  - PARAFAC
  - Incrementalization

Tensor Basics

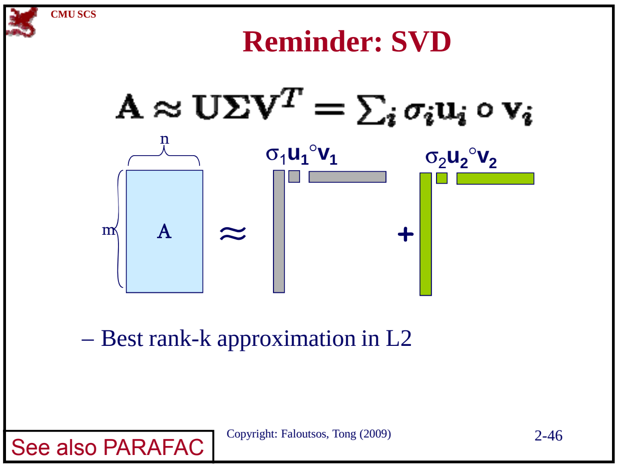
Reminder: SVD

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



- Best rank-k approximation in L2

See also PARAFAC



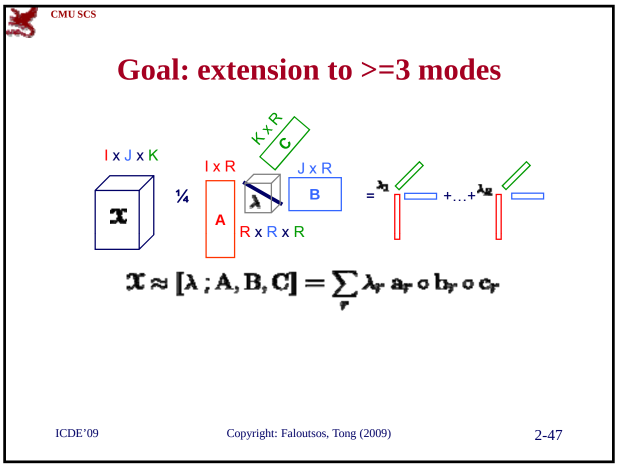
CMU SCS

Reminder: SVD

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

– Best rank-k approximation in L2

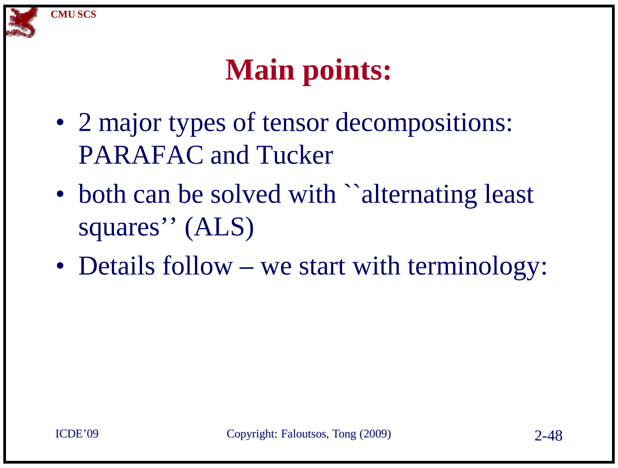
See also PARAFAC



CMU SCS

Goal: extension to >=3 modes

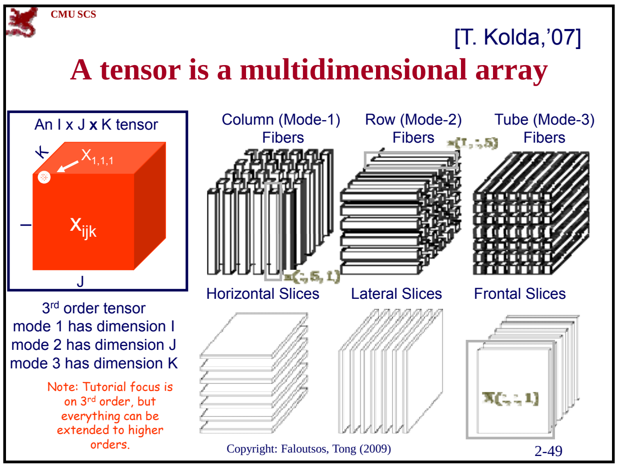
$$\mathcal{X} \approx [\![\lambda\, ; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_r \lambda_r\, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$



CMU SCS

Main points:

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



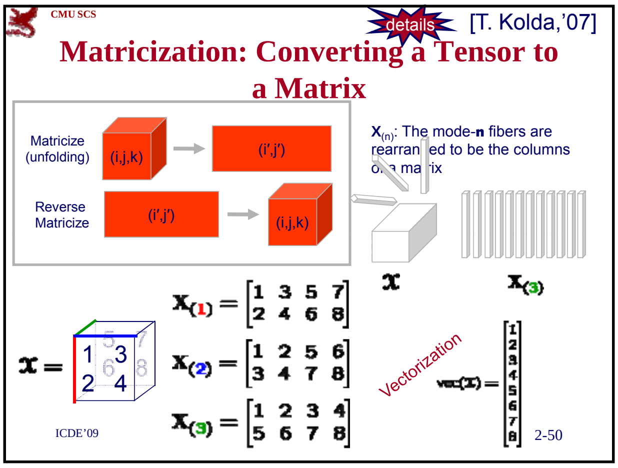
CMU SCS

[T. Kolda,’07]

A tensor is a multidimensional array

3rd order tensor
mode 1 has dimension I
mode 2 has dimension J
mode 3 has dimension K

Note: Tutorial focus is on 3rd order, but everything can be extended to higher orders.



CMU SCS

details

[T. Kolda,’07]

Matricization: Converting a Tensor to a Matrix

$X_{(n)}$: The mode-**n** fibers are rearranged to be the columns of a matrix

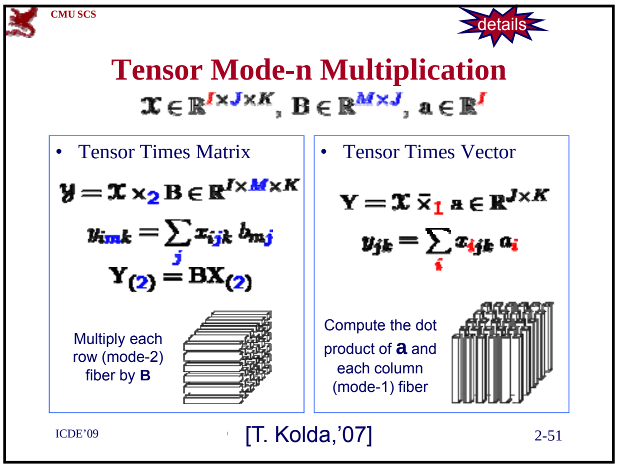
$$\mathcal{X} = \begin{matrix} 1 & 3 \\ 2 & 4 \end{matrix} \quad \begin{matrix} 5 & 7 \\ 6 & 8 \end{matrix}$$

$$X_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}$$

$$X_{(2)} = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}$$

$$X_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

$$\mathrm{vec}(\mathcal{X}) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$$



CMU SCS

details

Tensor Mode-n Multiplication

$$\mathcal{X} \in \mathbb{R}^{I \times J \times K},\ \mathbf{B} \in \mathbb{R}^{M \times J},\ \mathbf{a} \in \mathbb{R}^{I}$$

- Tensor Times Matrix

$$\mathcal{Y} = \mathcal{X} \times_2 \mathbf{B} \in \mathbb{R}^{I \times M \times K}$$

$$y_{imk} = \sum_j x_{ijk}\, b_{mj}$$

$$\mathbf{Y}_{(2)} = \mathbf{B}\mathbf{X}_{(2)}$$

Multiply each row (mode-2) fiber by **B**

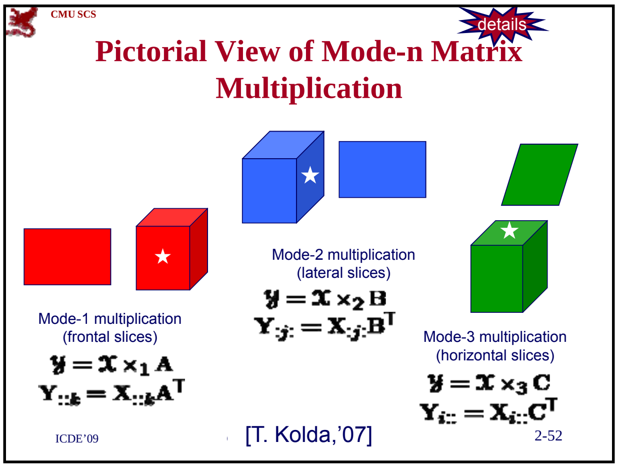
- Tensor Times Vector

$$\mathbf{Y} = \mathcal{X}\ \bar{\times}_1\ \mathbf{a} \in \mathbb{R}^{J \times K}$$

$$y_{jk} = \sum_i x_{ijk}\, a_i$$

Compute the dot product of **a** and each column (mode-1) fiber

[T. Kolda,’07]



Pictorial View of Mode-n Matrix Multiplication

Mode-1 multiplication (frontal slices)

$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}$

$\mathbf{Y}_{::k} = \mathbf{X}_{::k}\mathbf{A}^{\mathsf{T}}$

Mode-2 multiplication (lateral slices)

$\mathcal{Y} = \mathcal{X} \times_2 \mathbf{B}$

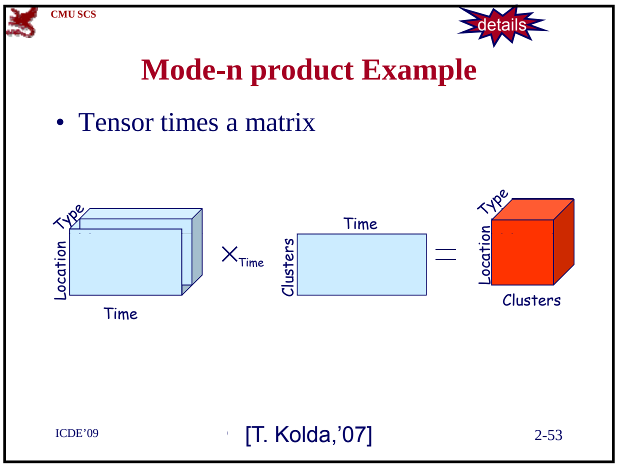
$\mathbf{Y}_{:j:} = \mathbf{X}_{:j:}\mathbf{B}^{\mathsf{T}}$

Mode-3 multiplication (horizontal slices)

$\mathcal{Y} = \mathcal{X} \times_3 \mathbf{C}$

$\mathbf{Y}_{i::} = \mathbf{X}_{i::}\mathbf{C}^{\mathsf{T}}$

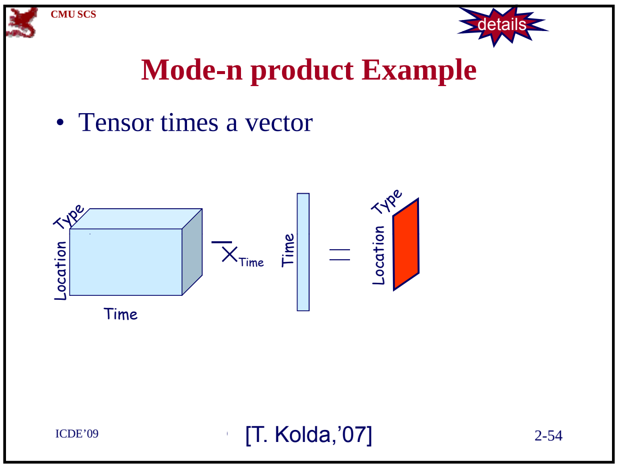
[T. Kolda,'07]



Mode-n product Example

- Tensor times a matrix

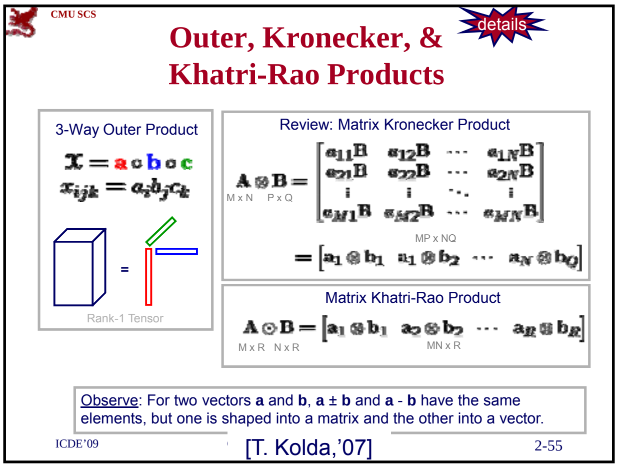
[T. Kolda,'07]



Mode-n product Example

- Tensor times a vector

[T. Kolda,'07]



CMU SCS

Outer, Kronecker, & Khatri-Rao Products

details

3-Way Outer Product

$$\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$$

$$x_{ijk} = a_i b_j c_k$$

Review: Matrix Kronecker Product

$$\underset{M \times N}{\mathbf{A}} \otimes \underset{P \times Q}{\mathbf{B}} = \underset{MP \times NQ}{\begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1N}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2N}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1}\mathbf{B} & a_{M2}\mathbf{B} & \cdots & a_{MN}\mathbf{B} \end{bmatrix}}$$

$$= \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_1 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_N \otimes \mathbf{b}_Q \end{bmatrix}$$

Matrix Khatri-Rao Product

$$\underset{M \times R}{\mathbf{A}} \odot \underset{N \times R}{\mathbf{B}} = \underset{MN \times R}{\begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_R \otimes \mathbf{b}_R \end{bmatrix}}$$

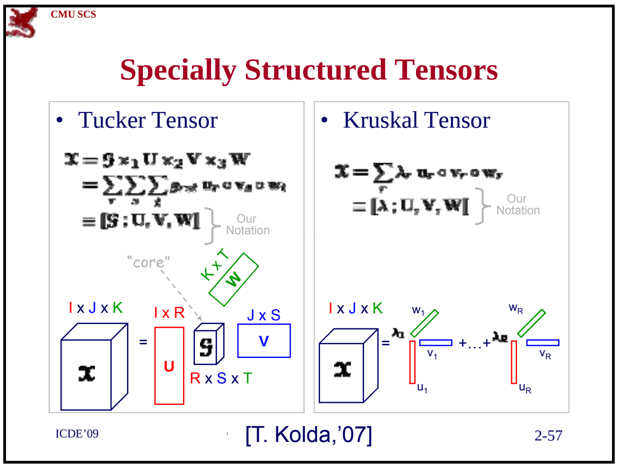
<u>Observe</u>: For two vectors **a** and **b**, **a** ± **b** and **a** - **b** have the same elements, but one is shaped into a matrix and the other into a vector.

ICDE'09 [T. Kolda,'07] 2-55

CMU SCS

Specially Structured Tensors

ICDE'09 Copyright: Faloutsos, Tong (2009) 2-56



CMU SCS

Specially Structured Tensors

- Tucker Tensor

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

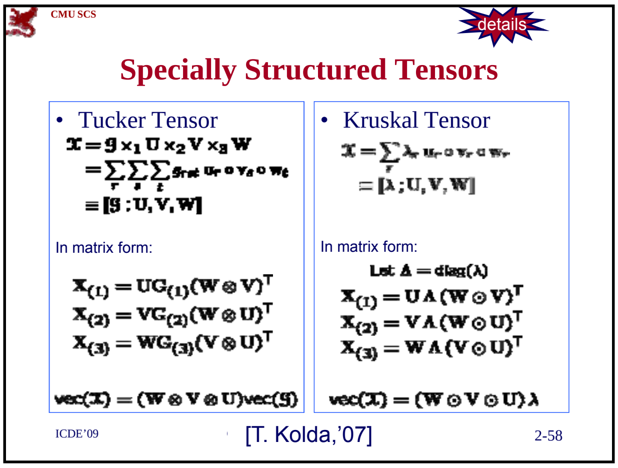
Our Notation

- Kruskal Tensor

$$\mathcal{X} = \sum_r \lambda_r\, \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r$$
$$= [\![\lambda ; \mathbf{U}, \mathbf{V}, \mathbf{W}]\!]$$

Our Notation

ICDE'09 [T. Kolda,'07] 2-57



CMU SCS

details

Specially Structured Tensors

- Tucker Tensor

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

In matrix form:

$$\mathbf{X}_{(1)} = \mathbf{U}\mathbf{G}_{(1)}(\mathbf{W} \otimes \mathbf{V})^{\mathsf{T}}$$
$$\mathbf{X}_{(2)} = \mathbf{V}\mathbf{G}_{(2)}(\mathbf{W} \otimes \mathbf{U})^{\mathsf{T}}$$
$$\mathbf{X}_{(3)} = \mathbf{W}\mathbf{G}_{(3)}(\mathbf{V} \otimes \mathbf{U})^{\mathsf{T}}$$

$$\mathrm{vec}(\mathcal{X}) = (\mathbf{W} \otimes \mathbf{V} \otimes \mathbf{U})\mathrm{vec}(\mathcal{G})$$

- Kruskal Tensor

$$\mathcal{X} = \sum_r \lambda_r\, \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r = [\![\lambda \,; \mathbf{U}, \mathbf{V}, \mathbf{W}]\!]$$

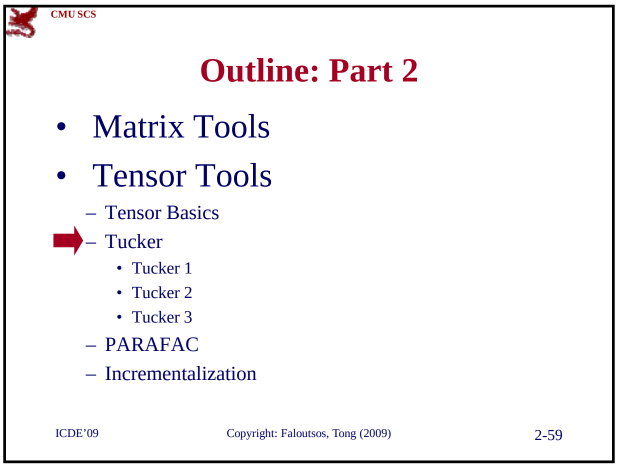
In matrix form:

Let $\mathbf{\Lambda} = \mathrm{diag}(\lambda)$

$$\mathbf{X}_{(1)} = \mathbf{U}\mathbf{\Lambda}(\mathbf{W} \odot \mathbf{V})^{\mathsf{T}}$$
$$\mathbf{X}_{(2)} = \mathbf{V}\mathbf{\Lambda}(\mathbf{W} \odot \mathbf{U})^{\mathsf{T}}$$
$$\mathbf{X}_{(3)} = \mathbf{W}\mathbf{\Lambda}(\mathbf{V} \odot \mathbf{U})^{\mathsf{T}}$$

$$\mathrm{vec}(\mathcal{X}) = (\mathbf{W} \odot \mathbf{V} \odot \mathbf{U})\lambda$$

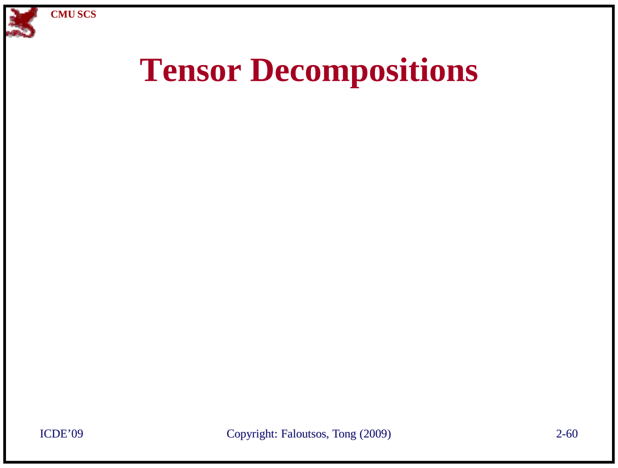
ICDE'09 [T. Kolda,'07]



CMU SCS

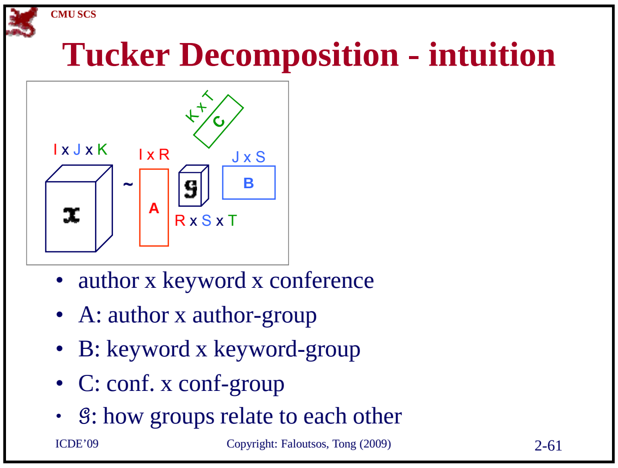
Outline: Part 2

- Matrix Tools
- Tensor Tools
  - Tensor Basics
  - ➡ Tucker
    - Tucker 1
    - Tucker 2
    - Tucker 3
  - PARAFAC
  - Incrementalization



CMU SCS

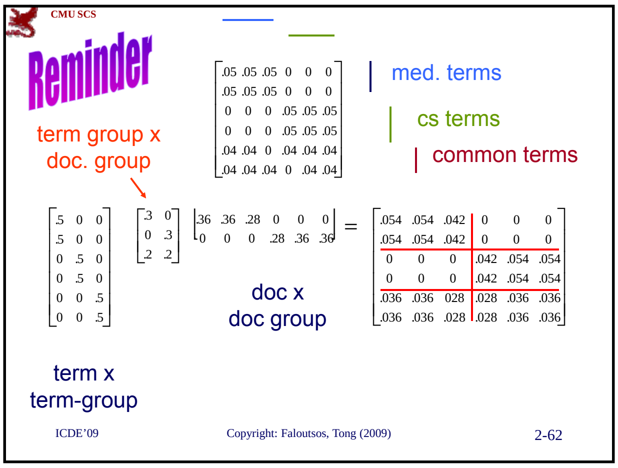
Tensor Decompositions



Tucker Decomposition - intuition

- author x keyword x conference
- A: author x author-group
- B: keyword x keyword-group
- C: conf. x conf-group
- $\mathcal{G}$: how groups relate to each other

ICDE'09 Copyright: Faloutsos, Tong (2009) 2-61



Reminder

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

med. terms

cs terms

common terms

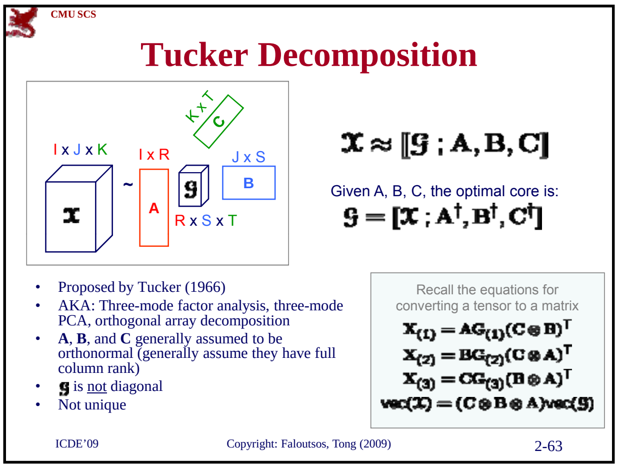
term group x doc. group

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

doc x doc group

term x term-group

ICDE'09 Copyright: Faloutsos, Tong (2009) 2-62



Tucker Decomposition

$$\mathcal{X} \approx [\![\mathcal{G} \,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

Given A, B, C, the optimal core is:

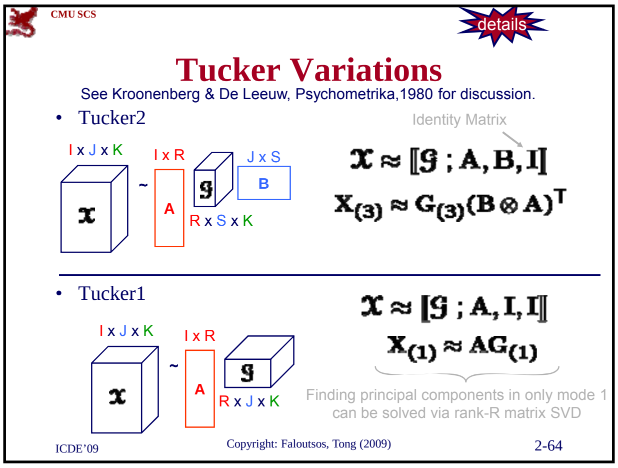
$$\mathcal{G} = [\![\mathcal{X} \,; \mathbf{A}^{\dagger}, \mathbf{B}^{\dagger}, \mathbf{C}^{\dagger}]\!]$$

- Proposed by Tucker (1966)
- AKA: Three-mode factor analysis, three-mode PCA, orthogonal array decomposition
- **A**, **B**, and **C** generally assumed to be orthonormal (generally assume they have full column rank)
- $\mathcal{G}$ is not diagonal
- Not unique

Recall the equations for converting a tensor to a matrix

$$\mathbf{X}_{(1)} = \mathbf{A}\mathbf{G}_{(1)}(\mathbf{C} \otimes \mathbf{B})^{\mathsf{T}}$$
$$\mathbf{X}_{(2)} = \mathbf{B}\mathbf{G}_{(2)}(\mathbf{C} \otimes \mathbf{A})^{\mathsf{T}}$$
$$\mathbf{X}_{(3)} = \mathbf{C}\mathbf{G}_{(3)}(\mathbf{B} \otimes \mathbf{A})^{\mathsf{T}}$$
$$\mathrm{vec}(\mathcal{X}) = (\mathbf{C} \otimes \mathbf{B} \otimes \mathbf{A})\mathrm{vec}(\mathcal{G})$$

ICDE'09 Copyright: Faloutsos, Tong (2009) 2-63



Tucker Variations

See Kroonenberg & De Leeuw, Psychometrika,1980 for discussion.

- Tucker2

Identity Matrix

$$\mathcal{X} \approx [\![\mathcal{G}\,;\mathbf{A},\mathbf{B},\mathbf{I}]\!]$$

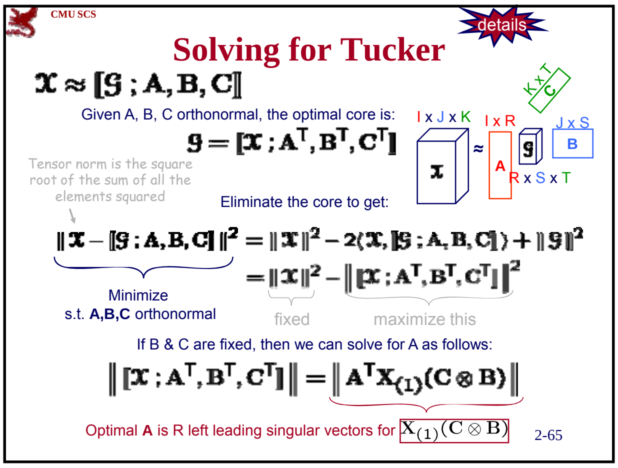
$$\mathbf{X}_{(3)} \approx \mathbf{G}_{(3)}(\mathbf{B}\otimes\mathbf{A})^{\mathsf{T}}$$

- Tucker1

$$\mathcal{X} \approx [\![\mathcal{G}\,;\mathbf{A},\mathbf{I},\mathbf{I}]\!]$$

$$\underbrace{\mathbf{X}_{(1)} \approx \mathbf{A}\mathbf{G}_{(1)}}_{\text{Finding principal components in only mode 1 can be solved via rank-R matrix SVD}}$$

ICDE'09 Copyright: Faloutsos, Tong (2009) 2-64



Solving for Tucker

$$\mathcal{X} \approx [\![\mathcal{G}\,;\mathbf{A},\mathbf{B},\mathbf{C}]\!]$$

Given A, B, C orthonormal, the optimal core is:

$$\mathcal{G} = [\![\mathcal{X}\,;\mathbf{A}^{\mathsf{T}},\mathbf{B}^{\mathsf{T}},\mathbf{C}^{\mathsf{T}}]\!]$$

Tensor norm is the square root of the sum of all the elements squared

Eliminate the core to get:

$$\underbrace{\|\mathcal{X} - [\![\mathcal{G}\,;\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|^2}_{\text{Minimize s.t. A,B,C orthonormal}} = \|\mathcal{X}\|^2 - 2\langle\mathcal{X},[\![\mathcal{G}\,;\mathbf{A},\mathbf{B},\mathbf{C}]\!]\rangle + \|\mathcal{G}\|^2$$

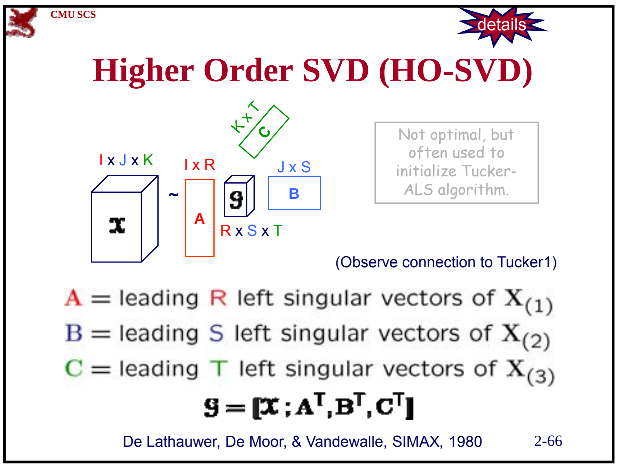
$$= \underbrace{\|\mathcal{X}\|^2}_{\text{fixed}} - \underbrace{\|[\![\mathcal{X}\,;\mathbf{A}^{\mathsf{T}},\mathbf{B}^{\mathsf{T}},\mathbf{C}^{\mathsf{T}}]\!]\|^2}_{\text{maximize this}}$$

If B & C are fixed, then we can solve for A as follows:

$$\|[\![\mathcal{X}\,;\mathbf{A}^{\mathsf{T}},\mathbf{B}^{\mathsf{T}},\mathbf{C}^{\mathsf{T}}]\!]\| = \underbrace{\|\mathbf{A}^{\mathsf{T}}\mathbf{X}_{(1)}(\mathbf{C}\otimes\mathbf{B})\|}$$

Optimal **A** is R left leading singular vectors for $\boxed{\mathrm{X}_{(1)}(\mathrm{C}\otimes\mathrm{B})}$

2-65



Higher Order SVD (HO-SVD)

Not optimal, but often used to initialize Tucker-ALS algorithm.

(Observe connection to Tucker1)

A = leading R left singular vectors of $\mathrm{X}_{(1)}$

B = leading S left singular vectors of $\mathrm{X}_{(2)}$

C = leading T left singular vectors of $\mathrm{X}_{(3)}$

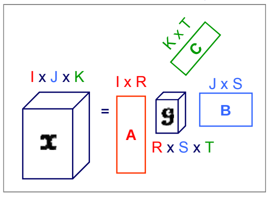
$$\mathcal{G} = [\![\mathcal{X}\,;\mathbf{A}^{\mathsf{T}},\mathbf{B}^{\mathsf{T}},\mathbf{C}^{\mathsf{T}}]\!]$$

De Lathauwer, De Moor, & Vandewalle, SIMAX, 1980 2-66

CMU SCS

Tucker-Alternating Least Squares (ALS)

*Successively solve for each component (**A**,**B**,**C**).*



- Initialize
  - Choose R, S, T
  - Calculate **A**, **B**, **C** via HO-SVD
- Until converged do…
  - **A** = R leading left singular vectors of $\mathbf{X}_{(1)}(\mathbf{C}\text{-}\mathbf{B})$
  - **B** = S leading left singular vectors of $\mathbf{X}_{(2)}(\mathbf{C}\text{-}\mathbf{A})$
  - **C** = T leading left singular vectors of $\mathbf{X}_{(3)}(\mathbf{B}\text{-}\mathbf{A})$
- Solve for core:

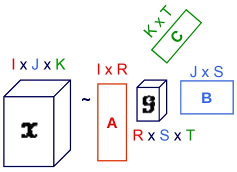
$$\mathcal{G} = [\![\mathcal{X}; \mathbf{A}^T, \mathbf{B}^T, \mathbf{C}^T]\!]$$

Kroonenberg & De Leeuw, Psychometrika, 1980

CMU SCS

details

Tucker in Not Unique



Tucker decomposition is *not* unique. Let Y be an RxR orthogonal matrix. Then…

$$\mathcal{X} \approx \mathcal{G}\times_1 \mathbf{A}\times_2 \mathbf{B}\times_3 \mathbf{C} = \left(\mathcal{G}\times_1 \mathbf{Y}^T\right)\times_1(\mathbf{A}\mathbf{Y})\times_2 \mathbf{B}\times_3 \mathbf{C}$$

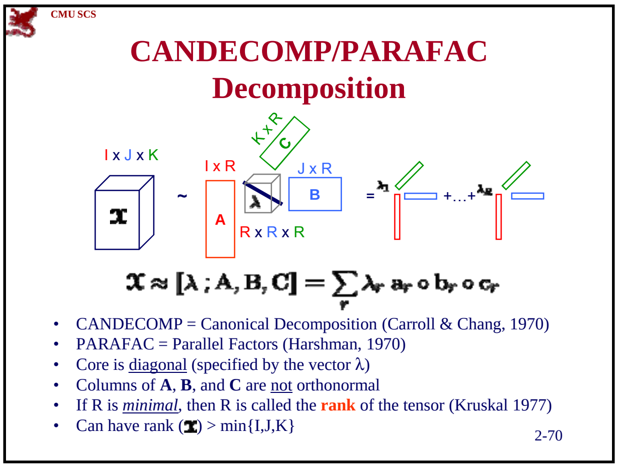
$$\mathbf{X}_{(1)} \approx \mathbf{A}\mathbf{G}_{(1)}(\mathbf{C}\otimes\mathbf{B})^T = \mathbf{A}\mathbf{Y}\mathbf{Y}^T\mathbf{G}_{(1)}(\mathbf{C}\otimes\mathbf{B})^T$$

[T. Kolda,'07]

CMU SCS

Outline: Part 2

- Matrix Tools
- Tensor Tools
  - Tensor Basics
  - Tucker
    - Tucker 1
    - Tucker 2
    - Tucker 3
  - ➡ PARAFAC

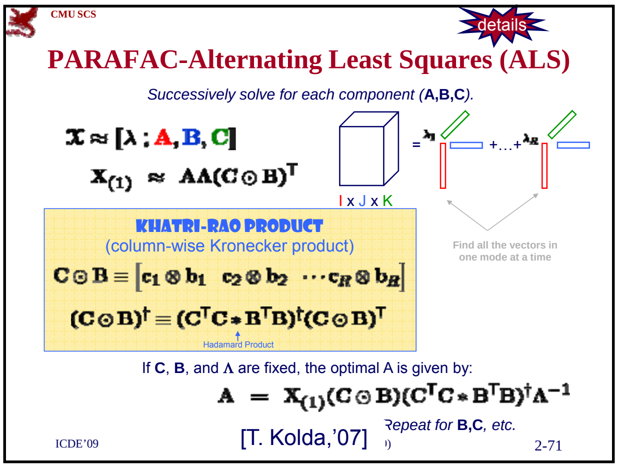


CMU SCS

CANDECOMP/PARAFAC Decomposition

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

- CANDECOMP = Canonical Decomposition (Carroll & Chang, 1970)
- PARAFAC = Parallel Factors (Harshman, 1970)
- Core is diagonal (specified by the vector λ)
- Columns of **A**, **B**, and **C** are not orthonormal
- If R is *minimal*, then R is called the **rank** of the tensor (Kruskal 1977)
- Can have rank ($\mathcal{X}$) > min{I,J,K}



CMU SCS

details

PARAFAC-Alternating Least Squares (ALS)

*Successively solve for each component (**A**,**B**,**C**).*

$$\mathcal{X} \approx [\![\lambda\, ; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

$$\mathbf{X}_{(1)} \approx \mathbf{A}\Lambda(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}$$

KHATRI-RAO PRODUCT
(column-wise Kronecker product)

$$\mathbf{C} \odot \mathbf{B} \equiv \left[\mathbf{c}_1 \otimes \mathbf{b}_1 \quad \mathbf{c}_2 \otimes \mathbf{b}_2 \quad \cdots \mathbf{c}_R \otimes \mathbf{b}_R\right]$$

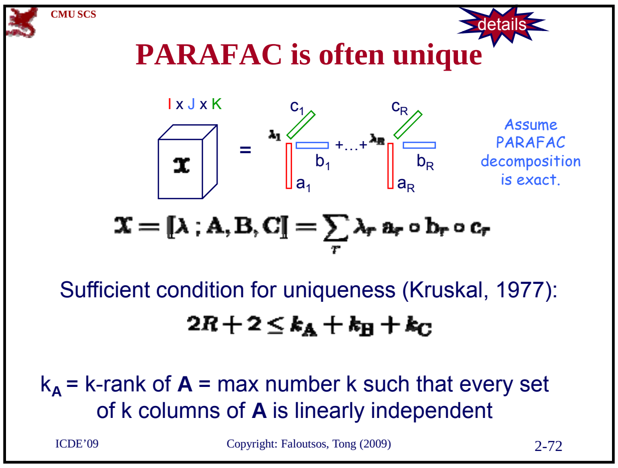
$$(\mathbf{C} \odot \mathbf{B})^{\dagger} \equiv (\mathbf{C}^{\mathsf{T}}\mathbf{C} \underset{\substack{\uparrow \\ \text{Hadamard Product}}}{*} \mathbf{B}^{\mathsf{T}}\mathbf{B})^{\dagger}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}$$

If **C**, **B**, and Λ are fixed, the optimal A is given by:

$$\mathbf{A} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^{\mathsf{T}}\mathbf{C} * \mathbf{B}^{\mathsf{T}}\mathbf{B})^{\dagger}\Lambda^{-1}$$

*Repeat for **B**,**C**, etc.*

[T. Kolda,'07]



CMU SCS

details

PARAFAC is often unique

Assume PARAFAC decomposition is exact.

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

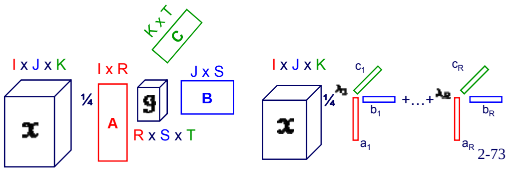
Sufficient condition for uniqueness (Kruskal, 1977):

$$2R + 2 \leq k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}}$$

k_A = k-rank of **A** = max number k such that every set of k columns of **A** is linearly independent

Tucker vs. PARAFAC Decompositions

- Tucker
  - Variable transformation in each mode
  - Core G may be dense
  - A, B, C generally orthonormal
  - Not unique
- PARAFAC
  - Sum of rank-1 components
  - No core, i.e., superdiagonal core
  - A, B, C may have linearly dependent columns
  - Generally unique



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

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, to appear (accepted June 2008)
- 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)