

CMU SCS Sandia National Laboratories

Mining Large Time-evolving Data Using Matrix and Tensor Tools

Christos Faloutsos Carnegie Mellon Univ.
Tamara G. Kolda Sandia National Labs
Jimeng Sun Carnegie Mellon Univ.

CMU SCS Sandia National Laboratories

About the tutorial


- Introduce **matrix and tensor tools** through real mining applications
- **Goal: find patterns, rules, clusters, outliers, ...**
 - in matrices and
 - in tensors

SDM'07 Faloutsos, Kolda, Sun 1-2

CMU SCS Sandia National Laboratories

Motivation 1: Why “matrix”?

- Why matrices are important?



SDM'07 Faloutsos, Kolda, Sun 1-3

CMU SCS Sandia National Laboratories

Examples of Matrices: Graph - social network

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

SDM'07 Faloutsos, Kolda, Sun 1-4

CMU SCS Sandia National Laboratories

Examples of Matrices: cloud of n-d points

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

SDM'07 Faloutsos, Kolda, Sun 1-5

CMU SCS Sandia National Laboratories

Examples of Matrices: Market basket

- **market basket** as in Association Rules

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

SDM'07 Faloutsos, Kolda, Sun 1-6

Examples of Matrices:
Documents and terms

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

SDM'07 Faloutsos, Kolda, Sun 1-7

Examples of Matrices:
Authors and terms

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

SDM'07 Faloutsos, Kolda, Sun 1-8


Examples of Matrices:
sensor-ids and time-ticks

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

SDM'07 Faloutsos, Kolda, Sun 1-9

Motivation 2: Why tensor?

- Q: what is a tensor?



SDM'07 Faloutsos, Kolda, Sun 1-10

Motivation 2: Why tensor?

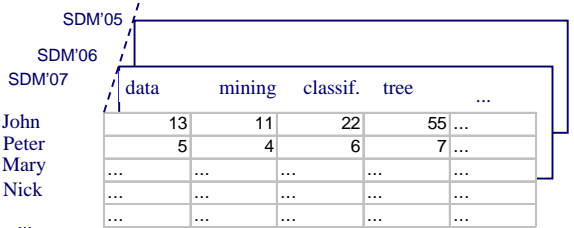
- A: N-D generalization of matrix:

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

SDM'07 Faloutsos, Kolda, Sun 1-11

Motivation 2: Why tensor?

- A: N-D generalization of matrix:



SDM'07 Faloutsos, Kolda, Sun 1-12

Tensors are useful for 3 or more modes

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

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

SDM'07 Mode (== aspect) #1 1-13

Motivating Applications

- Why matrices are important?
- Why tensors are useful?
 - P1: environmental sensors
 - P2: data center monitoring ('autonomic')
 - P3: social networks
 - P4: network forensics
 - P5: web mining

SDM'07 Faloutsos, Kolda, Sun 1-14

P1: Environmental sensor monitoring

Data in three aspects (time, location, type)

Faloutsos, Kolda, Sun 1-15

P2: Clusters/data center monitoring

- Monitor correlations of multiple measurements
- Automatically flag anomalous behavior
- Intemon: intelligent monitoring system
 - Prof. Greg Ganger and PDL
 - >100 machines in a data center
 - warsteiner.db.cs.cmu.edu/demo/intemon.jsp

SDM'07 1-16

P3: Social network analysis

- Traditionally, people focus on static networks and find community structures
- We plan to monitor the change of the community structure over time

SDM'07 1-17

P4: Network forensics

- Directional network flows
- A large ISP with 100 POPs, each POP 10Gbps link capacity [Hotnets2004]
 - 450 GB/hour with compression
- Task: Identify abnormal traffic pattern and find out the cause

SDM'07 Collaboration with Prof. Hui Zhang and Dr. Yinglian Xie 1-18

P5: Web graph mining

- How to order the importance of web pages?
 - Kleinberg’s algorithm HITS
 - PageRank
 - Tensor extension on HITS (**TOPHITS**)
 - context-sensitive hypergraph analysis

SDM'07 Faloutsos, Kolda, Sun 1-19

Static Data model

- Tensor
 - Formally, $\mathcal{X} \in \mathbb{R}^{N_1 \times \dots \times N_M}$
 - Generalization of matrices
 - Represented as multi-array, (~ data cube).

Order	1st	2nd	3rd
Correspondence	Vector	Matrix	3D array
Example			

SDM'07 1-20

Dynamic Data model

- Tensor Streams
 - A sequence of Mth order tensor

$\mathcal{X}_1 \dots \mathcal{X}_t$ where $\mathcal{X}_t \in \mathbb{R}^{N_1 \times \dots \times N_M}$
 t is increasing over time

Order	1st	2nd	3rd
Correspondence	Multiple streams	Time evolving graphs	3D arrays
Example			

SDM'07 1-21

Roadmap

- Motivation
- Matrix tools
- Tensor basics
- Tensor extensions
- Software demo
- Case studies


SDM'07 Faloutsos, Kolda, Sun 1-22

CMU SCS Carnegie Mellon University

Roadmap

- Motivation
- **Matrix tools**
- Tensor basics
- Tensor extensions
- Software demo
- Case studies

- SVD, PCA
- HITS, PageRank
- CUR
- Co-clustering
- Nonnegative Matrix factorization



SDM'07 Faloutsos, Kolda, Sun 2-1

CMU SCS Carnegie Mellon University

General goals

- Patterns
- Anomaly detection
- Compression

SDM'07 Faloutsos, Kolda, Sun 2-2

CMU SCS Carnegie Mellon University

Examples of Matrices

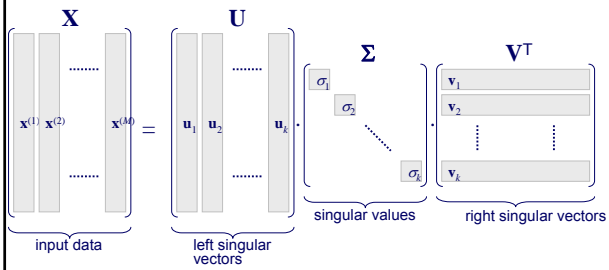
- Example/Intuition: Documents and terms
- Find patterns, groups, concepts

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

SDM'07 Faloutsos, Kolda, Sun 2-3

CMU SCS Carnegie Mellon University

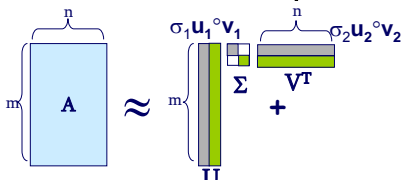
Singular Value Decomposition (SVD)

$$X = U \Sigma V^T$$


SDM'07 Faloutsos, Kolda, Sun 2-4

CMU SCS Carnegie Mellon University

SVD as spectral decomposition

$$A \approx U \Sigma V^T = \sum_i \sigma_i u_i \circ v_i$$


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

SDM'07 Faloutsos, Kolda, Sun 2-5

See also PARAFAC

CMU SCS Carnegie Mellon University


SVD - Example

- $A = U \Sigma V^T$ - example:

		retrieval		
		inf. ↓	brain	lung
data				
↑ CS	↓			
↑ MD	↓			

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


SDM'07 Faloutsos, Kolda, Sun 2-6

CMU SCS 

SVD - Interpretation

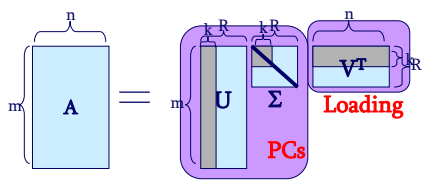
‘documents’, ‘terms’ and ‘concepts’:
 Q: if A is the document-to-term matrix, what is $A^T A$?
 A: term-to-term ($[m \times m]$) similarity matrix
 Q: $A A^T$?
 A: document-to-document ($[n \times n]$) similarity matrix

SDM07 Faloutsos, Kolda, Sun 2-13

CMU SCS 


Principal Component Analysis (PCA)

- SVD $A = U \Sigma V^T$



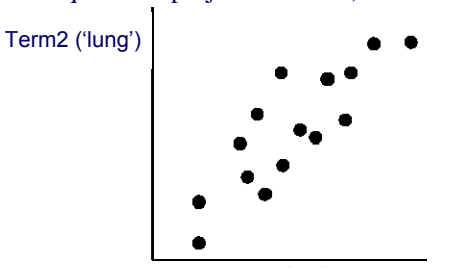
– PCA is an important application of SVD
 – Note that U and V are dense and may have negative entries

SDM07 Faloutsos, Kolda, Sun 2-14


CMU SCS 

PCA interpretation

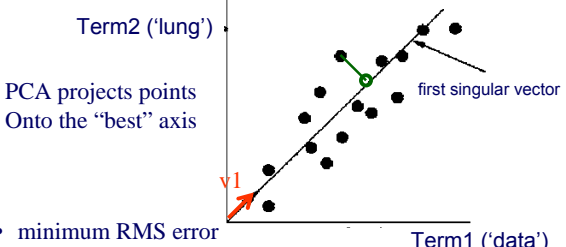
- best axis to project on: (‘best’ = min sum of squares of projection errors)



SDM07 2-15


CMU SCS 

PCA - interpretation



• minimum RMS error


SDM07 Faloutsos, Kolda, Sun 2-16

CMU SCS 


Roadmap

- Motivation
- Matrix tools
- Tensor basics
- Tensor extensions
- Software demo
- Case studies

- SVD, PCA
- **HITS, PageRank**
- CUR
- Co-clustering
- Nonnegative Matrix factorization



SDM07 Faloutsos, Kolda, Sun 2-17


CMU SCS 

Kleinberg’s algorithm HITS

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

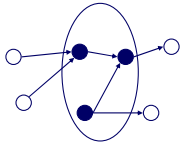
Step 0: find all pages containing the query terms
 Step 1: expand by one move forward and backward

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

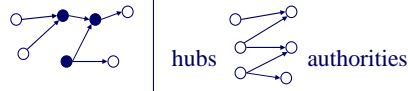


SDM'07 Faloutsos, Kolda, Sun 2-19


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'



SDM'07 Faloutsos, Kolda, Sun 2-20


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

SDM'07 Faloutsos, Kolda, Sun 2-21

CMU SCS 


Kleinberg's algorithm: HITS

Let \mathbf{A} be the adjacency matrix:
the (i,j) entry is 1 if the edge from i to j exists

Let \mathbf{h} and \mathbf{a} be $[n \times 1]$ vectors with the 'hubness' and 'authoritativeness' scores.

Then:

SDM'07 Faloutsos, Kolda, Sun 2-22

CMU SCS 

Kleinberg's algorithm: HITS

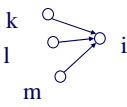
Then:

$$a_i = h_k + h_l + h_m$$


that is

$$a_i = \text{Sum}(h_j) \quad \text{over all } j \text{ that } (j,i) \text{ edge exists}$$

or

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


SDM'07 Faloutsos, Kolda, Sun 2-23

CMU SCS 

Kleinberg's algorithm: HITS

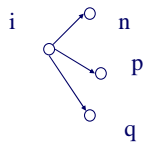
symmetrically, for the 'hubness':

$$h_i = a_n + a_p + a_q$$


that is

$$h_i = \text{Sum}(a_j) \quad \text{over all } j \text{ that } (i,j) \text{ edge exists}$$

or

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


SDM'07 Faloutsos, Kolda, Sun 2-24

CMU SCS 

Kleinberg's algorithm: HITS

In conclusion, we want vectors \mathbf{h} and \mathbf{a} such that:


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

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

That is:

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

SDM07 Faloutsos, Kolda, Sun 2-25

CMU SCS 

Kleinberg's algorithm: HITS

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


Starting from random \mathbf{a}' and iterating, we'll eventually converge

Q: to which of all the eigenvectors? why?

A: to the one of the strongest eigenvalue,

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

SDM07 Faloutsos, Kolda, Sun 2-26


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


Faloutsos, Kolda, Sun 2-27

CMU SCS 


Roadmap

- Motivation
- **Matrix tools**
- Tensor basics
- Tensor extensions
- Software demo
- Case studies

- SVD, PCA
- HITS, **PageRank**
- CUR
- Co-clustering
- Nonnegative Matrix factorization

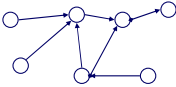


SDM07 Faloutsos, Kolda, Sun 2-28

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!)

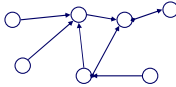
SDM07 Faloutsos, Kolda, Sun 2-29

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))



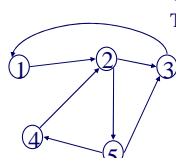
A node has high ssp, if it is connected with **high ssp** nodes (recursive, but OK!)

SDM07 Faloutsos, Kolda, Sun 2-30

CMU SCS Sanda National Laboratories

(Simplified) PageRank algorithm

- Let A be the transition matrix (= adjacency matrix); let A^T become column-normalized - then



From A^T
To

		1		
1			1	
	1/2			1/2
				1/2
	1/2			

=

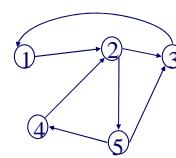
p1
p2
p3
p4
p5

SDM07 Faloutsos, Kolda, Sun 2-31

CMU SCS Sanda National Laboratories

(Simplified) PageRank algorithm

- $A^T p = p$



A^T

		1		
1			1	
	1/2			1/2
				1/2
	1/2			

$p = p$

p1
p2
p3
p4
p5

SDM07 Faloutsos, Kolda, Sun 2-32

CMU SCS Sanda National Laboratories

(Simplified) PageRank algorithm

- $A^T p = \mathbf{1} * p$
- thus, p is the **eigenvector** that corresponds to the highest eigenvalue (=1, since the matrix is column-normalized)
- Why does it exist such a p ?
 - p exists if A is $n \times n$, nonnegative, irreducible [Perron-Frobenius theorem]

SDM07 Faloutsos, Kolda, Sun 2-33

CMU SCS Sanda National Laboratories

(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

SDM07 Faloutsos, Kolda, Sun 2-34

CMU SCS Sanda National Laboratories

Full Algorithm

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

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

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

--

--

--


SDM07 Faloutsos, Kolda, Sun 2-35

CMU SCS Sanda National Laboratories


Roadmap

- Motivation
- Matrix tools**
- Tensor basics
- Tensor extensions
- Software demo
- Case studies

- SVD, PCA
- HITS, PageRank
- CUR**
- Co-clustering
- Nonnegative Matrix factorization




SDM07 Faloutsos, Kolda, Sun 2-36

CMU SCS 

Motivation of CUR or CMD

- SVD, PCA all transform data into some abstract space (specified by a set basis)
 - Interpretability problem
 - Loss of sparsity


SDM'07 Faloutsos, Kolda, Sun 2-37

CMU SCS 

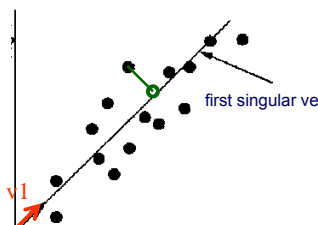
Interpretability problem

- Each column of projection matrix U_i is a linear combination of all dimensions along certain mode $U_i(:,1) = [0.5; -0.5; 0.5; 0.5]$
- All the data are projected onto the span of U_i
- It is hard to interpret the projections

SDM'07 Faloutsos, Kolda, Sun 2-38

CMU SCS 

PCA - interpretation



Term2 ('lung')


PCA projects points onto the "best" axis

first singular vector


- minimum RMS error

Term1 ('data')


SDM'07 Faloutsos, Kolda, Sun 2-39

CMU SCS 

The sparsity property


 **SVD: $A = U \Sigma V^T$**

Labels: sparse and small (pointing to Σ), Big but sparse (pointing to U), Big and dense (pointing to V)

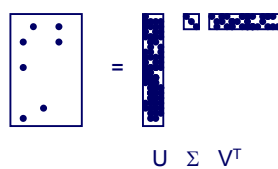
 **CUR: $A = C U R$**

Labels: dense but small (pointing to U), Big but sparse (pointing to C), Big but sparse (pointing to R)

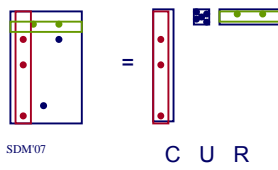
SDM'07 2-40

CMU SCS 

The sparsity property – pictorially:




SVD/PCA:
Destroys sparsity

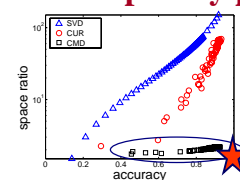


CUR: maintains sparsity

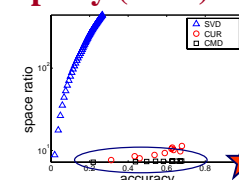
SDM'07 2-41

CMU SCS 

The sparsity property (cont.)




Network



DBLP

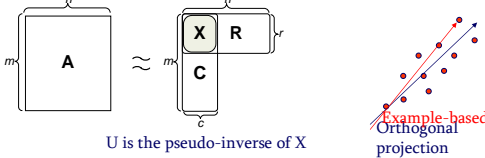
- CMD uses much smaller space to achieve the same accuracy
- CUR limitation: duplicate columns and rows
- SVD limitation: orthogonal projection densifies the data

Reference:
Sun et al. Less is More: Compact Matrix Decomposition for Large Sparse Graphs, SDM'07

CMU SCS 


CUR

- **Example-based projection:** use actual rows and columns to specify the subspace
- Given a matrix $A \in \mathbb{R}^{m \times n}$, find three matrices $C \in \mathbb{R}^{m \times c}$, $U \in \mathbb{R}^{c \times r}$, $R \in \mathbb{R}^{r \times n}$, such that $\|A - CUR\|$ is small



U is the pseudo-inverse of X

SDM'07 Faloutsos, Kolda, Sun 2-43

CMU SCS 


CUR (cont.)

- **Key question:**
 - How to select/sample the columns and rows?
- **Uniform sampling**
- **Biased sampling**
 - CUR w/ absolute error bound
 - CUR w/ relative error bound

Reference:

1. Tutorial: Randomized Algorithms for Matrices and Massive Datasets, SDM'06
2. Drineas et al. Subspace Sampling and Relative-error Matrix Approximation: Column-Row-Based Methods, ESA2006
3. Drineas et al., Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition, SIAM Journal on Computing, 2006.

SDM'07 Faloutsos, Kolda, Sun 2-43


CMU SCS 

Roadmap


- Motivation
- **Matrix tools**
- Tensor basics
- Tensor extensions
- Software demo
- Case studies

}

- SVD, PCA
- HITS, PageRank
- CUR
- **Co-clustering etc**
- Nonnegative Matrix factorization



SDM'07 Faloutsos, Kolda, Sun 2-45

CMU SCS 


Co-clustering

- Let X and Y be discrete random variables
 - X and Y take values in $\{1, 2, \dots, m\}$ and $\{1, 2, \dots, n\}$
 - $p(X, Y)$ 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



SDM'07 Faloutsos, Kolda, Sun 2-45

CMU SCS 

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
- **Key goal is to exploit the “duality” between row and column clustering to overcome sparsity and noise**

SDM'07 Faloutsos, Kolda, Sun 2-47

CMU SCS  

Information Theory Concepts

- Entropy of a random variable X with probability distribution p :

$$H(p) = -\sum_x p(x) \log p(x)$$
- The Kullback-Leibler (KL) Divergence or “Relative Entropy” between two probability distributions p and q :

$$KL(p, q) = \sum_x p(x) \log(p(x)/q(x))$$
- Mutual Information between random variables X and Y :

$$I(X, Y) = \sum_x \sum_y p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right)$$

SDM'07 Faloutsos, Kolda, Sun 2-48

CMU SCS details

Information-Theoretic Co-Clustering

- View (scaled) co-occurrence matrix as a **joint probability** distribution between row & column random variables

$$p(x, y) = \frac{\#co-occurrence(x, y)}{\sum_{x, y} \#co-occurrence(x, y)}$$

- We seek a **hard-clustering** of both dimensions such that loss in "Mutual Information"

$$I(X, Y) - I(\hat{X}, \hat{Y})$$
 is minimized **given a fixed no. of row & col. clusters**

SDM'07 Faloutsos, Kolda, Sun 2-49

CMU SCS details

Information Theoretic Co-clustering

- "Loss in mutual information" equals

$$I(X, Y) - I(\hat{X}, \hat{Y}) = KL(p(x, y) || q(x, y))$$

$$= H(\hat{X}, \hat{Y}) + H(X | \hat{X}) + H(Y | \hat{Y}) - H(X, Y)$$
- p is the input distribution
- q is an *approximation* to p

$$q(x, y) = p(\hat{x}, \hat{y})p(x | \hat{x})p(y | \hat{y}), x \in \hat{x}, y \in \hat{y}$$

– Can be shown that $q(x, y)$ is a **maximum entropy approximation** subject to cluster constraints.

SDM'07 Faloutsos, Kolda, Sun 2-50

CMU SCS

$$p(x, y) = \begin{bmatrix} .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & .05 & .05 & 0 \\ 0 & 0 & .05 & .05 & 0 \\ .04 & .04 & 0 & .04 & .04 \\ .04 & .04 & 0 & .04 & .04 \end{bmatrix}$$

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

$$p(x | \hat{x}) \qquad \qquad \qquad q(x, y)$$

#parameters that determine $q(x, y)$ are: $(m-k) + (kl-1) + (n-l)$

SDM'07 Faloutsos, Kolda, Sun 2-51

CMU SCS

Problem with Information Theoretic Co-clustering

- Number of row and column groups must be specified

Desiderata:

- ✓ Simultaneously discover row and column groups
- ✗ Fully Automatic: No "magic numbers"
- ✓ Scalable to large graphs

SDM'07 Faloutsos, Kolda, Sun 2-52

CMU SCS

Cross-association

Desiderata:

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


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

CMU SCS

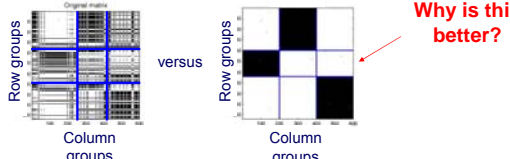
What makes a cross-association "good"?

Why is this better?

SDM'07 Faloutsos, Kolda, Sun 2-54

CMU SCS 

What makes a cross-association "good"?



Original matrix

Row groups

Column groups

versus


Row groups

Column groups


Why is this better?

simpler; easier to describe
easier to compress!

SDM'07 Faloutsos, Kolda, Sun 2-55

CMU SCS 


What makes a cross-association "good"?



Problem definition: given an encoding scheme

- decide on the # of col. and row groups k and l
- and reorder rows and columns,
- to achieve best compression

SDM'07 Faloutsos, Kolda, Sun 2-56

CMU SCS  details


Main Idea

Good Compression \rightarrow Better Clustering

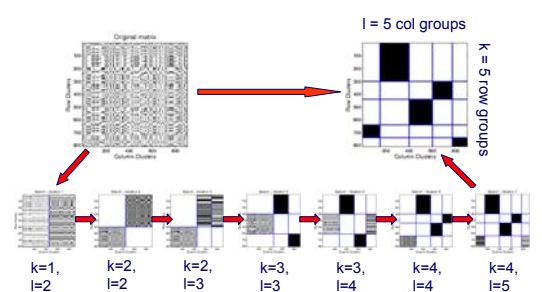
$$\text{Total Encoding Cost} = \underbrace{\sum_i \text{size}_i * H(x_i)}_{\text{Code Cost}} + \underbrace{\text{Cost of describing cross-associations}}_{\text{Description Cost}}$$

Minimize the total cost (# bits)
for lossless compression

SDM'07 Faloutsos, Kolda, Sun 2-57

CMU SCS 

Algorithm




Original matrix

$l = 5$ col groups

$k = 5$ row groups

$k=1, l=2$ $k=2, l=2$ $k=2, l=3$ $k=3, l=3$ $k=3, l=4$ $k=4, l=4$ $k=4, l=5$

SDM'07 Faloutsos, Kolda, Sun 2-58

CMU SCS 

Algorithm


Code for cross-associations (matlab):

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

Variations and extensions:

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


SDM'07 Faloutsos, Kolda, Sun 2-59

CMU SCS 

Cross-Associations vs. Co-clustering

Information-theoretic co-clustering	Cross-Associations
<ol style="list-style-type: none"> 1. For any nonnegative matrix 2. Lossy Compression. 3. Approximates the original matrix, while trying to minimize KL-divergence. 4. The number of row and column groups must be given by the user. 	<ol style="list-style-type: none"> 1. For binary matrix 2. Lossless Compression. 3. Always provides complete information about the matrix, for any number of row and column groups. 4. Chosen automatically using the MDL principle.


SDM'07 Faloutsos, Kolda, Sun 2-60

CMU SCS 

Roadmap

- Motivation
- **Matrix tools**
- Tensor basics
- Tensor extensions
- Software demo
- Case studies

- SVD, PCA
- HITS, PageRank
- CUR
- Co-clustering, etc
- **Nonnegative Matrix factorization**



SDM'07 Faloutsos, Kolda, Sun 2-61

CMU SCS 

Nonnegative Matrix Factorization

- Coming up soon with **nonnegative tensor factorization**


SDM'07 Faloutsos, Kolda, Sun 2-62

CMU SCS Gordon and Betty Moore Foundation

Roadmap

- Motivation
- Matrix tools
- **Tensor basics**
- Tensor extensions
- Software demo
- Case studies

- Tensor Basics
- Tucker
 - Tucker 1 (PCA)
 - Tucker 2
 - Tucker 3 (HOSVD)
- PARAFAC



3-1

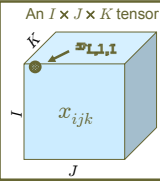
CMU SCS Gordon and Betty Moore Foundation

Tensor Basics

CMU SCS Gordon and Betty Moore Foundation


A tensor is a multidimensional array

An $I \times J \times K$ tensor

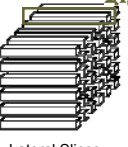


x_{ijk}

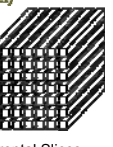
Column (Mode-1) Fibers



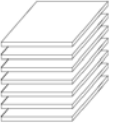
Row (Mode-2) Fibers



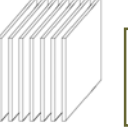
Tube (Mode-3) Fibers




Horizontal Slices



Lateral Slices



Frontal Slices




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

Note: Tutorial focus is on 3 dimensions, but everything can be extended to higher dimensionality.


CMU SCS Gordon and Betty Moore Foundation

Matricize: Converting a Tensor to a Matrix

Matricize (unfolding)




Reverse Matricize



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

\mathbf{x}



$\mathbf{x}_{(1)}$

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

$\mathbf{x}_{(2)}$

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

$\mathbf{x}_{(3)}$

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

3-4

CMU SCS Gordon and Betty Moore Foundation

Tensor Mode- n Multiplication

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


• Tensor Times Matrix

$$\mathbf{y} = \mathbf{X} \times_2 \mathbf{B}$$

$$y_{mk} = \sum_j x_{ijk} b_{mj}$$

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

Multiply each row (mode-2) fiber by \mathbf{B}




• Tensor Times Vector

$$\mathbf{Y} = \mathbf{X} \times_1 \mathbf{a}$$

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

$$\text{vec}(\mathbf{Y}) = \mathbf{X}_{(1)}^T \mathbf{a}$$

Compute the dot product of \mathbf{a} and each column (mode-1) fiber



3-5

CMU SCS Gordon and Betty Moore Foundation

Pictorial View of Mode- n Matrix Multiplication

Mode-1 multiplication (frontal slices)

$$\mathbf{y} = \mathbf{X} \times_1 \mathbf{A}$$

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

Mode-2 multiplication (lateral slices)

$$\mathbf{y} = \mathbf{X} \times_2 \mathbf{B}$$

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

Mode-3 multiplication (horizontal slices)

$$\mathbf{y} = \mathbf{X} \times_3 \mathbf{C}$$

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

3-6

CMU SCS Carnegie Mellon University

Mode-n product Example

- Tensor times a matrix

3-7

CMU SCS Carnegie Mellon University

Mode-n product Example

- Tensor times a vector

3-8

CMU SCS Carnegie Mellon University

Outer, Kronecker, & Khatri-Rao Products

Outer Product

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

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

Kronecker Product

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1N}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2N}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1}\mathbf{B} & a_{M2}\mathbf{B} & \dots & a_{MN}\mathbf{B} \end{bmatrix}$$

$M \times N \quad P \times Q \quad \quad \quad MP \times NQ$

Khatri-Rao Product

$$\mathbf{A} \circledast \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \dots \quad \mathbf{a}_R \otimes \mathbf{b}_R]$$

$M \times R \quad N \times R \quad \quad \quad MN \times R$

Observe: $\mathbf{a} \circ \mathbf{b}$ and $\mathbf{a} \otimes \mathbf{b}$ have the same elements, but one is shaped into a matrix and the other into a vector.

3-9

CMU SCS Carnegie Mellon University

Specially Structured Tensors

CMU SCS Carnegie Mellon University

Specially Structured Tensors

- Tucker Tensor

$$\mathbf{X} = \mathbf{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$

$$= \sum_i \sum_j \sum_k g_{ijk} u_i \circ v_j \circ w_k$$

$$\equiv [\mathbf{G}; \mathbf{U}, \mathbf{V}, \mathbf{W}]$$

- Kruskal Tensor

$$\mathbf{X} = \sum_r \lambda_r \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r$$

$$\equiv [\boldsymbol{\lambda}; \mathbf{U}, \mathbf{V}, \mathbf{W}]$$

3-11

CMU SCS Carnegie Mellon University

Specially Structured Tensors

- Tucker Tensor

$$\mathbf{X} = \mathbf{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$

$$= \sum_i \sum_j \sum_k g_{ijk} u_i \circ v_j \circ w_k$$

$$\equiv [\mathbf{G}; \mathbf{U}, \mathbf{V}, \mathbf{W}]$$

In matrix form:

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

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

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

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

- Kruskal Tensor

$$\mathbf{X} = \sum_r \lambda_r \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r$$

$$\equiv [\boldsymbol{\lambda}; \mathbf{U}, \mathbf{V}, \mathbf{W}]$$

In matrix form:

Let $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$

$$\mathbf{X}_{(1)} = \mathbf{U} \boldsymbol{\Lambda} (\mathbf{W} \otimes \mathbf{V})^T$$

$$\mathbf{X}_{(2)} = \mathbf{V} \boldsymbol{\Lambda} (\mathbf{W} \otimes \mathbf{U})^T$$

$$\mathbf{X}_{(3)} = \mathbf{W} \boldsymbol{\Lambda} (\mathbf{V} \otimes \mathbf{U})^T$$


$$\text{vec}(\mathbf{X}) = (\mathbf{W} \otimes \mathbf{V} \otimes \mathbf{U}) \boldsymbol{\lambda}$$

3-12

CMU SCS Carnegie Mellon University

Matrix SVD is a Tucker or Kruskal Tensor

Matrix SVD:

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


Tucker Tensor:

$$\mathbf{X} = \mathbf{\Sigma} \times_1 \mathbf{U} \times_2 \mathbf{V} = [\mathbf{\Sigma}; \mathbf{U}, \mathbf{V}]$$

Kruskal Tensor:

$$\mathbf{X} = \sum_{r=1}^R \sigma_r \mathbf{u}_r \circ \mathbf{v}_r = [\sigma; \mathbf{U}, \mathbf{V}]$$

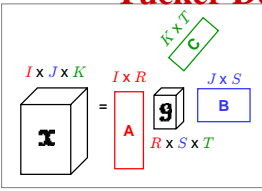
3-13

CMU SCS Carnegie Mellon University

Tensor Decompositions

CMU SCS Carnegie Mellon University

Tucker Decomposition



$$\mathbf{X} = [\mathbf{g}; \mathbf{A}, \mathbf{B}, \mathbf{C}]$$

$$\mathbf{g} = [\mathbf{x}; \mathbf{A}^T, \mathbf{B}^T, \mathbf{C}^T]$$

- 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)
- g is not diagonal
- Not unique

Recall the equations for converting a tensor to a matrix

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

$$\mathbf{X}_{(2)} = \mathbf{B}\mathbf{G}_{(2)}(\mathbf{C} \otimes \mathbf{A})^T$$

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

$$\text{vec}(\mathbf{X}) = (\mathbf{C} \otimes \mathbf{B} \otimes \mathbf{A})\text{vec}(\mathbf{g})$$

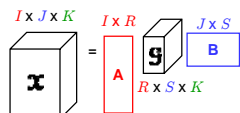
3-15

CMU SCS Carnegie Mellon University

Tucker Variations

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

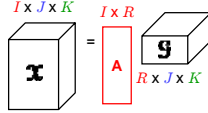
- Tucker2



$$\mathbf{X} = [\mathbf{g}; \mathbf{A}, \mathbf{B}, \mathbf{I}]$$

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

- Tucker1



$$\mathbf{X} = [\mathbf{g}; \mathbf{A}, \mathbf{I}, \mathbf{I}]$$

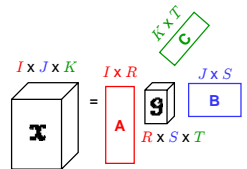
$$\mathbf{X}_{(1)} = \mathbf{A}\mathbf{G}_{(1)}$$

Finding principal components in only mode 1. Can be solved via rank-R matrix SVD

3-16

CMU SCS Carnegie Mellon University

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 $\mathbf{X}_{(1)}$
 B = leading S left singular vectors of $\mathbf{X}_{(2)}$
 C = leading T left singular vectors of $\mathbf{X}_{(3)}$

$$\mathbf{g} = [\mathbf{x}; \mathbf{A}^T, \mathbf{B}^T, \mathbf{C}^T]$$

De Lathauwer, De Moor, & Vandewalle, SIMAX, 1980 3-17

CMU SCS Carnegie Mellon University

Solving for Tucker

$$\|\mathbf{X} - [\mathbf{g}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\|^2 = \|\mathbf{X}\|^2 - 2\langle \mathbf{X}, [\mathbf{g}; \mathbf{A}, \mathbf{B}, \mathbf{C}] \rangle + \|\mathbf{g}\|^2$$

$$= \|\mathbf{X}\|^2 - \|\mathbf{g}\|^2$$

Minimize s.t. A, B, C orthonormal

$$\|\mathbf{g}\| \equiv \|\mathbf{[X; A^T, B^T, C^T]}\| = \|\mathbf{A}^T \mathbf{X}_{(1)} (\mathbf{C} \otimes \mathbf{B})\|$$

Maximize s.t. A, B, C orthonormal

To solve for A (assuming B and C are fixed):
 Calculate R leading left singular vectors of $\mathbf{X}_{(1)} (\mathbf{C} \otimes \mathbf{B})$

3-18

CMU SCS Gerdle National Laboratories

Alternating Least Squares (ALS) for Tucker

- Initialize
 - Choose R, S, T
 - Calculate A, B, C via HO-SVD
- Until converged do...
 - A = R leading left singular vectors of $X_{(1)}(C \otimes B)$
 - B = S leading left singular vectors of $X_{(2)}(C \otimes A)$
 - C = T leading left singular vectors of $X_{(3)}(B \otimes A)$

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

Kroonenberg & De Leeuw, Psychometrika, 1980 3-19

CMU SCS Gerdle National Laboratories

CANDECOMP/PARAFAC Decomposition

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

- 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 $\text{rank}(\mathcal{X}) > \min\{I, J, K\}$

3-20

CMU SCS Gerdle National Laboratories

Alternating Least Squares (ALS) for PARAFAC

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

To solve for A, consider:

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


By the properties of the Khatri-Rao product, we have:

$$(\mathbf{C} \otimes \mathbf{B})^\dagger \equiv (\mathbf{C}^T \mathbf{C} * \mathbf{B}^T \mathbf{B})^\dagger (\mathbf{C} \otimes \mathbf{B})^T$$

Thus:

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


Do the same for B and C, then repeat... 3-21

CMU SCS 


Roadmap

- Motivation
- Matrix tools
- Tensor basics
- **Tensor extensions**
- Software demo
- Case studies


- Other decompositions
- Nonnegativity
- Missing values
- Incrementalization



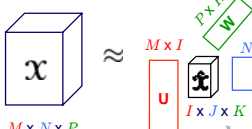
4-1

CMU SCS 

Other Tensor Decompositions

CMU SCS 

Combining Tucker & PARAFAC

$$\mathcal{X} \approx [\mathcal{X}; \mathbf{U}, \mathbf{V}, \mathbf{W}]$$



- Step 1: Choose orthonormal matrices U, V, W to compress tensor (Tucker tensor!)
 - Typically HO-SVD can be used
- Step 2: Run PARAFAC on smaller tensor
- Step 3: Reassemble result

$$\mathcal{X} = [\mathcal{X}; \mathbf{U}^T, \mathbf{V}^T, \mathbf{W}^T]$$

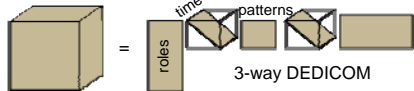
$$\mathcal{X} \approx [\mathcal{X}; \mathbf{A}, \mathbf{B}, \mathbf{C}]$$

$$\mathcal{X} \approx [\mathbf{A}; \mathbf{U}\mathbf{A}, \mathbf{V}\mathbf{B}, \mathbf{W}\mathbf{C}] \equiv [\mathbf{A}; \mathbf{A}, \mathbf{B}, \mathbf{C}]$$

Bro and Andersson, 1998 4-3

CMU SCS 

3-Way DEDICOM




3-way DEDICOM


$$\mathbf{X}_{::k} = \mathbf{A} \mathbf{D}_{::k} \mathbf{R} \mathbf{D}_{::k} \mathbf{A}^T$$

- 2-way DEDICOM introduced by Harshman, 1978
- 3-way DEDICOM due to Kiers, 1993
- Idea is to capture asymmetric relationships among different "roles"
- If third dimension is time, than diagonal slices capture participation of each role at each time

See, e.g., Bader, Harshman, Kolda, SAND2006-2161 4-4

CMU SCS 

Computations with Tensors

CMU SCS 

Dense Tensors

- Largest tensor that can be stored on a laptop is 200 x 200 x 200
- Typically, tensor operations are reduced to matrix operations
 - Requires permuting and reshaping the tensor
- Example: Mode-n tensor-matrix multiply


Example: Mode-1 Matrix Multiply

$$\mathbf{y} = \mathcal{X} \times_1 \mathbf{U}$$

$$\mathbf{Y}_{(n)} = \mathbf{U} \mathbf{X}_{(n)}$$

CMU SCS Gandhi National Laboratories

Sparse Tensors: Only Store Nonzeros



Store just the nonzeros of a tensor (assume coordinate format)

Example: Tensor-Vector Multiply (in all modes)

$$\alpha = \mathbf{X} \bar{x}_1 \mathbf{a} \bar{x}_2 \mathbf{b} \bar{x}_3 \mathbf{c}$$

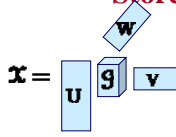
$$= \sum_i \sum_j \sum_k x_{ijk} a_i b_j c_k$$

$$= \sum_p v_p a_{p(1)} b_{p(2)} c_{p(3)}$$

p th nonzero 1st subscript of p th nonzero 2nd subscript of p th nonzero 3rd subscript of p th nonzero
4-7

CMU SCS Gandhi National Laboratories

Tucker Tensors: Store Core & Factors



Tucker tensor stores the core (which can be dense, sparse, or structured) and the factors.

Example: Mode-3 Tensor-Vector Multiply

$$\mathbf{Y} = \mathbf{X} \bar{x}_3 \mathbf{z}$$

$$= (\mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}) \bar{x}_3 \mathbf{z}$$

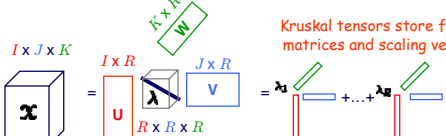
$$= \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \bar{x}_3 \mathbf{W}^T \mathbf{z}$$

$$= \underbrace{\mathcal{G} \bar{x}_3 \mathbf{W}^T \mathbf{z}}_{\mathbf{3}\mathbf{c}} \times_1 \mathbf{U} \times_2 \mathbf{V} = [\mathbf{3}\mathbf{c}; \mathbf{U}, \mathbf{V}]$$

Result is a Tucker Tensor 4-8

CMU SCS Gandhi National Laboratories

Kruskal Example: Store Factors



Kruskal tensors store factor matrices and scaling vector.

Example: Norm

$$\|\mathbf{X}\|^2 = \|\lambda; \mathbf{U}, \mathbf{V}, \mathbf{W}\|^2$$

$$= \|(\mathbf{W} \odot \mathbf{V} \odot \mathbf{U}) \lambda\|^2$$

$$= \lambda^T (\mathbf{W} \odot \mathbf{V} \odot \mathbf{U})^T (\mathbf{W} \odot \mathbf{V} \odot \mathbf{U}) \lambda$$

$$= \lambda^T (\mathbf{W}^T \mathbf{W} * \mathbf{V}^T \mathbf{V} * \mathbf{U}^T \mathbf{U}) \lambda$$

4-9

CMU SCS Gandhi National Laboratories

Nonnegativity

CMU SCS Gandhi National Laboratories

Non-negative Matrix Factorization

$$\|\mathbf{X} - \mathbf{A}\mathbf{B}^T\| \leftarrow \text{Minimize subject to elements of A and B being positive.}$$

Update formulas (do not increase objective function):

$$\mathbf{A} = \mathbf{A} * (\mathbf{X}\mathbf{B}) \oslash (\mathbf{A}\mathbf{B}^T\mathbf{B})$$

$$\mathbf{B} = \mathbf{B} * (\mathbf{X}^T\mathbf{A}) \oslash (\mathbf{B}\mathbf{A}^T\mathbf{A})$$

↑ Elementwise multiply (Hadamard product) ↑ Elementwise divide

Lee & Seung, Nature, 1999 4-11

CMU SCS Gandhi National Laboratories

Non-negative 3-Way PARAFAC Factorization

$$\|\mathbf{X} - [\mathbf{A}, \mathbf{B}, \mathbf{C}]\| \leftarrow \text{Minimize subject to elements of A, B and C being positive.}$$

Lee-Seung-like update formulas can be derived for 3D and higher:

$$\mathbf{A} = \mathbf{A} * (\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})) \oslash (\mathbf{A}(\mathbf{C}^T\mathbf{C} * \mathbf{B}^T\mathbf{B}))$$

$$\mathbf{B} = \mathbf{B} * (\mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})) \oslash (\mathbf{B}(\mathbf{C}^T\mathbf{C} * \mathbf{A}^T\mathbf{A}))$$

$$\mathbf{C} = \mathbf{C} * (\mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})) \oslash (\mathbf{C}(\mathbf{B}^T\mathbf{B} * \mathbf{A}^T\mathbf{A}))$$

↑ Elementwise multiply (Hadamard product) ↑ Elementwise divide

M. Mørup, L. K. Hansen, J. Parnas, S. M. Arnfred, *Decomposing the time-frequency representation of EEG using non-negative matrix and multi-way factorization*, 2006

CMU SCS

Handling Missing Data

CMU SCS

A Quick Overview on Handling Missing Data

- Consider sparse PARAFAC where all the zero entries represent missing data

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

- Typically, missing values are just set to zero
- There are more sophisticated approaches for handling missing values:
 - Weighted approximation
 - Data imputation to estimate missing values

4-14
See, e.g., Kiers, Psychometrika, 1997 and Srebro & Jaakkola, ICML 2003

CMU SCS

Weighted Least Squares

$$w_{ijk} = \begin{cases} 1 & x_{ijk} \text{ is known} \\ 0 & \text{otherwise} \end{cases} \quad \text{Weight Tensor}$$

- Weight the least squares problem so that the missing elements are ignored:

Weighted Least Squares

$$\sum_i \sum_j \sum_k w_{ijk} (x_{ijk} - a_i b_j c_k)^2$$

- But this problem is often too hard to solve directly!

4-15

CMU SCS

Missing Value Imputation

- Use the current estimate to fill in the missing values

$$\mathcal{P} = [\mathbf{A}, \mathbf{B}, \mathbf{C}] \quad \text{Current Estimate}$$

- The tensor for the next iteration of the algorithm is:

$$\hat{\mathcal{X}} = \underbrace{\mathcal{W} * \mathcal{X}}_{\text{Known Values}} + \underbrace{(\mathbf{1} - \mathcal{W}) * \mathcal{P}}_{\text{Estimates of Unknowns}}$$

$$= \underbrace{\mathcal{X}}_{\text{Sparse!}} - \underbrace{\mathcal{W} * \mathcal{P}}_{\text{Kruskal Tensor}} + \mathcal{P}$$

- Challenge is finding a good initial estimate

4-16

CMU SCS

Roadmap

- Motivation
- Matrix tools
- Tensor basics
- Tensor extensions
- Software demo
- Case studies



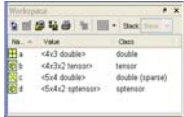
4-17

CMU SCS

Tensor Toolbox for MATLAB

<http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox>

- Six object-oriented tensor classes
 - Working with tensors is easy
- Most comprehensive set of kernel operations in any language
 - E.g., arithmetic, logical, multiplication operations
- Sparse tensors are unique
 - Speed-ups of two orders of magnitude for smaller problems
 - Larger problems than ever before



- Free for research or evaluations purposes
- 297 unique registered users from all over the world (as of January 17, 2006)

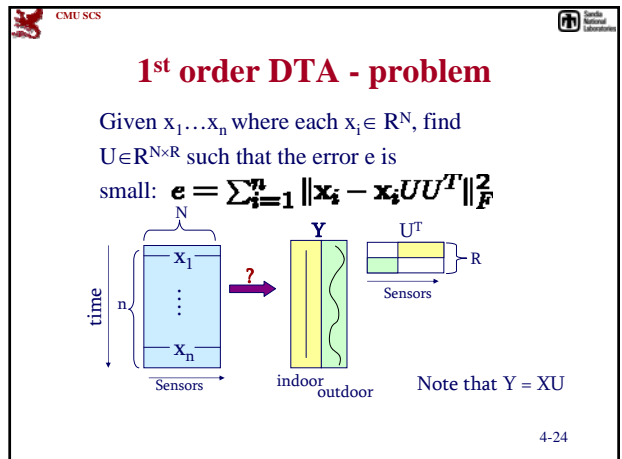
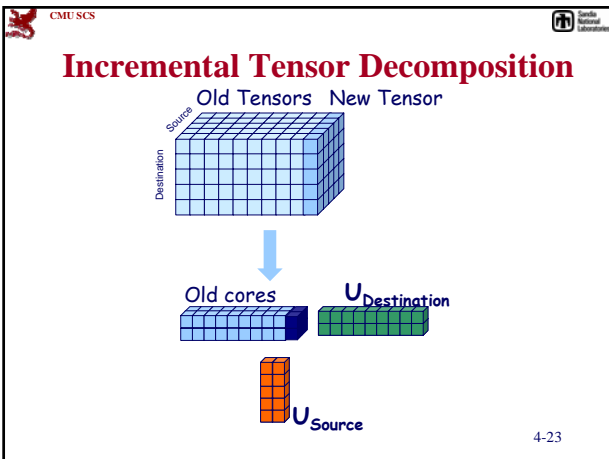
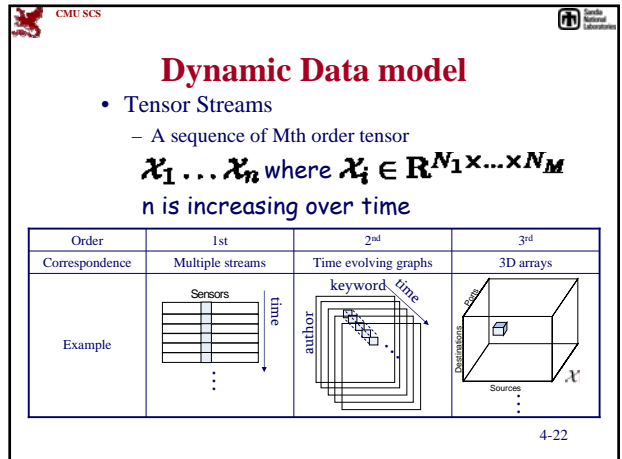
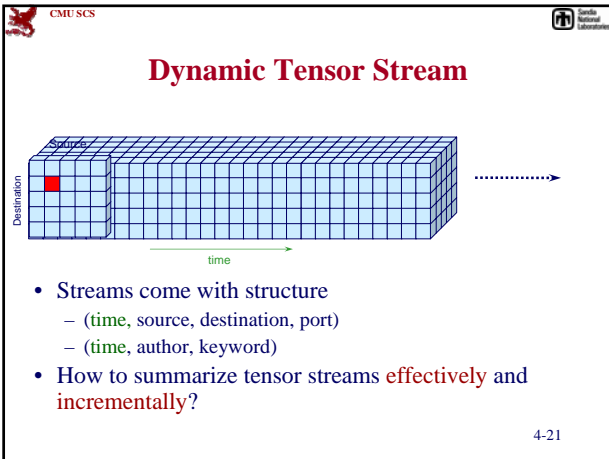
Bader & Kolda, ACM TOMS 2006 & SAND2006-7592


4-18

CMU SCS

Incrementalization

- CMU SCS
- ## Incremental Tensor Decomposition
- Dynamic data model
 - Tensor Streams
 - Dynamic Tensor Decomposition (DTA)
 - Streaming Tensor Decomposition (STA)
 - Window-based Tensor Decomposition (WTA)
- 4-20



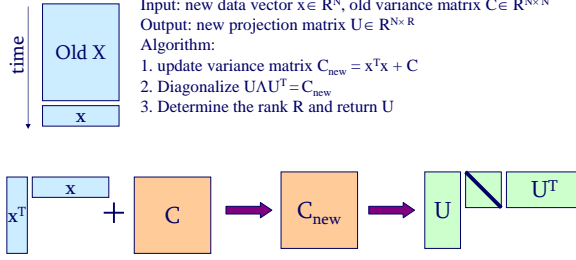
CMU SCS 

1st order Dynamic Tensor Analysis

Input: new data vector $x \in \mathbb{R}^N$, old variance matrix $C \in \mathbb{R}^{N \times N}$
 Output: new projection matrix $U \in \mathbb{R}^{N \times R}$


Algorithm:

1. update variance matrix $C_{\text{new}} = x^T x + C$
2. Diagonalize $U \Lambda U^T = C_{\text{new}}$
3. Determine the rank R and return U

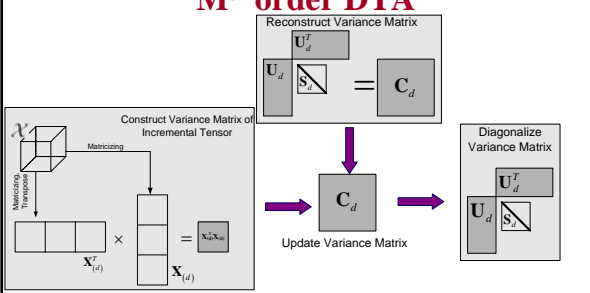


Diagonalization has to be done for **every** new $x!$

4-25

CMU SCS 

Mth order DTA




Reconstruct Variance Matrix $U_d^T S_d U_d = C_d$

Update Variance Matrix C_d

Diagonalize Variance Matrix $U_d^T S_d U_d$

4-26

CMU SCS 


Mth order DTA – complexity

Storage:
 $O(\prod N_i)$, i.e., size of an input tensor at a single timestamp

Computation:
 $\sum N_i^3$ (or $\sum N_i^2$) diagonalization of C
 $+\sum N_i \prod N_i$ matrix multiplication $X_{(d)}^T X_{(d)}$

For low order tensor (<3), diagonalization is the main cost
 For high order tensor, matrix multiplication is the main cost

4-27

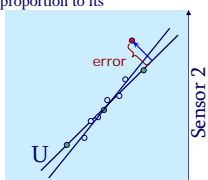
CMU SCS 

1st order Streaming Tensor Analysis (STA)


- Adjust U smoothly when new data arrive **without diagonalization** [VLDB05]
- For each new point x
 - Project onto current line
 - Estimate error
 - Rotate line in the direction of the error and in proportion to its magnitude

For each new point x and for $i = 1, \dots, k$:

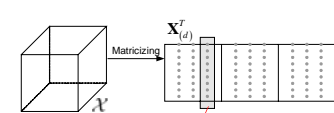
- $y_i := U_i^T x$ (proj. onto U_i)
- $d_i \leftarrow \lambda d_i + y_i^2$ (energy $\propto i$ -th eigenval.)
- $e_i := x - y_i U_i$ (error)
- $U_i \leftarrow U_i + (1/d_i) y_i e_i$ (update estimate)
- $x \leftarrow x - y_i U_i$ (repeat with remainder)



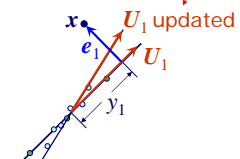
Sensor 1 4-28

CMU SCS 


Mth order STA



- Run 1st order STA along each mode
- Complexity:
 - Storage: $O(\prod N_i)$
 - Computation: $\sum R_i \prod N_i$ which is smaller than DTA



4-29

CMU SCS 

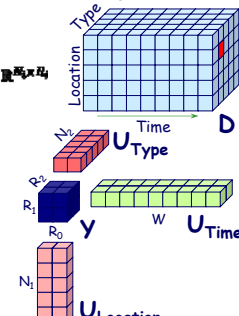
Meta-algorithm for window-based tensor analysis

Input: The tensor window $\mathcal{D} \in \mathbb{R}^{W \times N_L \times N_T \times \dots \times N_M}$

Output: The projection matrix $U_0 \in \mathbb{R}^{W \times R_0}$, $U_i \in \mathbb{R}^{N_i \times R_i}$ and the core tensor \mathcal{Y} .

Algorithm:

1. Initialize U_i for $i=0, \dots, M$
2. Conduct 3 - 5 iteratively
3. For $k=0$ to M
4. Fix U_i for $i \neq k$ and find the U_k that minimizes $d(\mathcal{D}, \mathcal{D} \prod_{i=0}^M (U_i U_i^T))$
5. Check convergence
6. Calculate the core tensor $\mathcal{Y} = \mathcal{D} \prod_{i=0}^M U_i$



4-30

CMU SCS Google
National
Laboratories

Moving Window scheme (MW)

- Update the variance matrix $C_{(i)}$ **incrementally**
- Diagonalize $C(i)$ to find $U(i)$

Tensor Streams

Time

$D_{(n-1, W)}$ $D_{(n, W)}$

Update variance matrix

$U_{(d)}$ Diagonalize

A good and efficient initialization

4-31


CMU SCS 

Roadmap

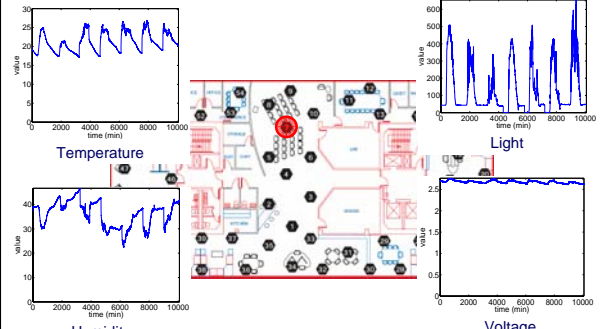
- Motivation
- Matrix tools
- Tensor basics
- Tensor extensions
- Software demo
- **Case studies**




SDM'07 Faloutsos, Kolda, Sun 5-1

CMU SCS 

P1: Environmental sensor monitoring

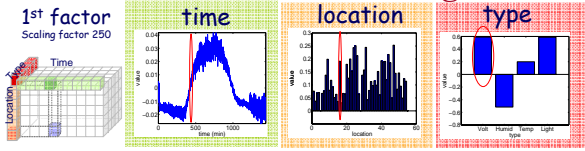


SDM'07 Faloutsos, Kolda, Sun 5-2

CMU SCS 


P2: sensor monitoring

1st factor
Scaling factor 250



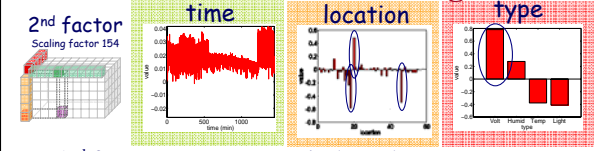
- 1st factor consists of the main trends:
 - Daily periodicity on time
 - Uniform on all locations
 - Temp, Light and Volt are positively correlated while negatively correlated with Humid

SDM'07 Faloutsos, Kolda, Sun 5-3

CMU SCS 


P2: sensor monitoring

2nd factor
Scaling factor 154



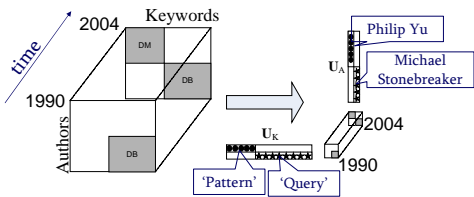
- 2nd factor captures an atypical trend:
 - Uniformly across all time
 - Concentrating on 3 locations
 - Mainly due to voltage
- Interpretation: two sensors have low battery, and the other one has high battery.

SDM'07 Faloutsos, Kolda, Sun 5-4


CMU SCS 

P3: Social network analysis

- Multiway latent semantic indexing (LSI)
 - Monitor the change of the community structure over time



SDM'07 Faloutsos, Kolda, Sun 5-5

CMU SCS 

P3: Social network analysis (cont.)

Authors	Keywords	Year
michael carey, michael stonebreaker, h. jagadish, hector garcia-molina	hier, parallel, optimization, concurr, ent	1995
surajit chaudhuri, mitch chemack, michael stonebreaker, agus etintemel	distribut, systems, view, storage, servic, process, cache	2004
jianwei han, jian pei, philip s. yu, jianyong wang, charu c. aggarwal	data, min, support, cluster, quer, queri	2004

- Two groups are correctly identified: Databases and Data mining
- People and concepts are drifting over time

SDM'07 Faloutsos, Kolda, Sun 5-6

P4: Network anomaly detection

Abnormal traffic Reconstruction error over time Normal traffic

- Reconstruction error gives indication of anomalies.
- Prominent difference between normal and abnormal ones is mainly due to the unusual scanning activity (confirmed by the campus admin).

SDM07 Faloutsos, Kolda, Sun 5-7

P5: Web graph mining

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

SDM07 Faloutsos, Kolda, Sun 5-8

Kleinberg's Hubs and Authorities (the HITS method)

Endangered Species: Federal policy was being threatened by a variety of environmental pressures. For example, the impact of logging permits in the world's great forests is being questioned by some environmentalists.

Jaguar FAQ: Jaguars are an endangered species that live in the tropical rain forests of Central and South America. They live about 15 years in the wild and up to 22 years in a zoo.

Orinco Atlas: View maps of animal habitats from around the world, including those of endangered animals in North, South, and Central America.

Rain Forest Zoo: We have a new online opening night month highlighting the endangered species of the Americas, including the jaguar.

Chitra Atlas: View maps of animal habitats from around the world, including those of endangered animals in North, South, and Central America.

WebSite 1, WebSite 2, WebSite 3, WebSite 4

Sparse adjacency matrix and its SVD:

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

$$X \approx \sum_r \lambda_r h_r o_r$$

authority scores for 1st topic authority scores for 2nd topic

hub scores for 1st topic hub scores for 2nd topic

SDM07 Faloutsos, Kolda, Sun 5-9

Kleinberg, JACM, 1999

HITS Authorities on Sample Data

1st Principal Factor	
.97	www.ibm.com
.24	www.alpha
.08	www.develop
.02	www.resour
.01	www.redbo
.01	news.com.c

2nd Principal Factor	
.99	www.lehigh.edu
.11	www2.lehigh.edu
.06	www.lehigh
.06	www.lehighs
.02	www.bethle
.02	www.adobe
.02	lewisweb.c
.02	www.leo.leh
.02	www.distanc
.12	blogs.sun.c
.08	sunsolve.sun.c
.09	www.sun-catal
.08	news.com.c

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

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

6th Principal Factor	
.97	mathpost.asu.edu
.18	math.la.asu.edu
.20	www.ssa.g
.17	www.asu.edu
.04	www.act.org
.03	www.sas.asu.edu
.02	archives.math.uk.edu
.02	www.geom.uiuc.edu
.02	www.fulton.asu.edu
.02	www.amstat.org
.02	www.maa.org

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

SDM07 Faloutsos, Kolda, Sun 5-10

Three-Dimensional View of the Web

Endangered Species: Federal policy was being threatened by a variety of environmental pressures. For example, the impact of logging permits in the world's great forests is being questioned by some environmentalists.

Jaguar FAQ: Jaguars are an endangered species that live in the tropical rain forests of Central and South America. They live about 15 years in the wild and up to 22 years in a zoo.

Orinco Atlas: View maps of animal habitats from around the world, including those of endangered animals in North, South, and Central America.

Rain Forest Zoo: We have a new online opening night month highlighting the endangered species of the Americas, including the jaguar.

Chitra Atlas: View maps of animal habitats from around the world, including those of endangered animals in North, South, and Central America.

WebSite 1, WebSite 2, WebSite 3, WebSite 4

Observe that this tensor is very sparse!

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

SDM07 Faloutsos, Kolda, Sun 5-11

Kolda, Bader, Kenny, ICDM05

Topical HITS (TOPHITS)

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

$$X \approx \sum_{r=1}^R \lambda_r h_r o_r$$

SDM07 Faloutsos, Kolda, Sun 5-12

