Tensor Decompositions

Multilinear operators for higher-order decompositions
A tensor is a multidimensional array

An $I \times J \times K$ tensor

$X = [x_{ijk}]$

$3^{rd}$ order tensor

mode 1 has dimension $I$
mode 2 has dimension $J$
mode 3 has dimension $K$
Matrization: Converting a Tensor to a Matrix

Matricize (unfolding) $(i,j,k) \rightarrow (i',j')$

Reverse Matricize $(i',j') \rightarrow (i,j,k)$

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

$x = \begin{pmatrix} 1 & 5 & 3 & 7 \\ 2 & 6 & 4 & 8 \end{pmatrix}$

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

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

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

$\text{vec}(x) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{pmatrix}$
Tensor Mode-n Multiplication

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

- **Tensor Times Matrix**

\[ y_{imk} = \sum_{j} x_{ijk} \ b_{mj} \]

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

- **Tensor Times Vector**

\[ y_{jk} = \sum_{i} x_{ijk} \ a_{i} \]

Compute the dot product of \( \mathbf{a} \) and each column (mode-1) fiber.
Pictorial View of Mode-n Matrix Multiplication

Mode-1 multiplication (frontal slices)
\[ y = X \times_1 A \]
\[ Y::k = X::k A^T \]

Mode-2 multiplication (lateral slices)
\[ y = X \times_2 B \]
\[ Y::j: = X::j: B^T \]

Mode-3 multiplication (horizontal slices)
\[ y = X \times_3 C \]
\[ Y_{i::} = X_{i::} C^T \]
Outer, Kronecker, & Khatri-Rao Products

3-Way Outer Product
\[ \mathbf{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \]
\[ x_{ijk} = a_i b_j c_k \]

Review: Matrix Kronecker Product
\[
\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix}
ad_{11} \mathbf{B} & ad_{12} \mathbf{B} & \cdots & ad_{1N} \mathbf{B} \\
ad_{21} \mathbf{B} & ad_{22} \mathbf{B} & \cdots & ad_{2N} \mathbf{B} \\
\vdots & \vdots & \ddots & \vdots \\
ad_{M1} \mathbf{B} & ad_{M2} \mathbf{B} & \cdots & ad_{MN} \mathbf{B}
\end{bmatrix}_{MP \times NQ}
= \begin{bmatrix}
a_1 \otimes b_1 & a_1 \otimes b_2 & \cdots & a_N \otimes b_Q
\end{bmatrix}
\]

Matrix Khatri-Rao Product
\[
\mathbf{A} \odot \mathbf{B} = \begin{bmatrix}
a_1 \otimes b_1 & a_2 \otimes b_2 & \cdots & a_R \otimes b_R
\end{bmatrix}_{M \times R \times N \times R}
\]

Observe: For two vectors \( \mathbf{a} \) and \( \mathbf{b} \), \( \mathbf{a} \circ \mathbf{b} \) and \( \mathbf{a} \otimes \mathbf{b} \) have the same elements, but one is shaped into a matrix and the other into a vector.
Specially Structured Tensors

**Tucker Tensor**

\[ \mathbf{x} = \mathbf{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \]

= \sum_{r} \sum_{s} \sum_{t} g_{rst} \mathbf{u}_r \circ \mathbf{v}_s \circ \mathbf{w}_t

\equiv [\mathbf{G}; \mathbf{U}, \mathbf{V}, \mathbf{W}] \quad \text{Our Notation}

**Kruskal Tensor**

\[ \mathbf{x} = \sum_{r} \lambda_r \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r \]

\equiv [\lambda; \mathbf{U}, \mathbf{V}, \mathbf{W}] \quad \text{Our Notation}
Specially Structured Tensors

- **Tucker Tensor**

\[
X = \mathcal{G} \times_1 U \times_2 V \times_3 W = \sum_r \sum_s \sum_t g_{rst} u_r \circ v_s \circ w_t \equiv \mathcal{G} ; U, V, W
\]

In matrix form:

\[
X_{(1)} = U G_{(1)} (W \otimes V)^T \\
X_{(2)} = V G_{(2)} (W \otimes U)^T \\
X_{(3)} = W G_{(3)} (V \otimes U)^T
\]

\[
\text{vec}(X) = (W \otimes V \otimes U) \text{vec}({\mathcal{G}})
\]

- **Kruskal Tensor**

\[
X = \sum_r \lambda_r u_r \circ v_r \circ w_r \equiv [\lambda ; U, V, W]
\]

In matrix form:

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

\[
X_{(1)} = U \Lambda (W \otimes V)^T \\
X_{(2)} = V \Lambda (W \otimes U)^T \\
X_{(3)} = W \Lambda (V \otimes U)^T
\]

\[
\text{vec}(X) = (W \circ V \circ U) \lambda
\]
What is the HO Analogue of the Matrix SVD?

Matrix SVD:

\[ X = U \Sigma V^T = \begin{bmatrix} \sigma_1 & \sigma_2 & \ldots & \sigma_R \end{bmatrix} + \ldots + \]

Tucker Tensor (finding bases for each subspace):

\[ X = \Sigma \times_1 U \times_2 V = [\Sigma ; U, V] \]

Kruskal Tensor (sum of rank-1 components):

\[ X = \sum_{r=1}^{R} \sigma_r u_r \circ v_r = [\sigma ; U, V] \]
Tucker Decomposition
Identifies Subspaces

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

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

$$G = [X ; A^\dagger, B^\dagger, C^\dagger]$$

Recall the equations for converting a tensor to a matrix

$$X_{(1)} \approx AG_{(1)}(C \otimes B)^T$$
$$X_{(2)} \approx BG_{(2)}(C \otimes A)^T$$
$$X_{(3)} \approx CG_{(3)}(B \otimes A)^T$$

$$\text{vec}(X) \approx (C \otimes B \otimes A)\text{vec}(G)$$
TensorFaces: An Application of the Tucker Decomposition

M.A.O. Vasilescu & D. Terzopoulos, CVPR’03

- Example: 7942 pixels x 16 illuminations x 11 subjects
- PCA (eigenfaces): SVD of 7942 x 176 matrix
- Tensorfaces: Tucker-2 decomposition of 7942 x 16 x 11 tensor

\[
\mathbf{X} \approx \mathbf{E} \times_2 \mathbf{V}
\]

eigenfaces    loadings

An image is represented by a linear combination of 33 eigenfaces.

\[
\mathbf{X} \approx \mathcal{T} \times_2 \mathbf{U}_{\text{illum}} \times_3 \mathbf{U}_{\text{person}}
\]
tensorfaces    illumination    subjects

An image is represented by a multilinear combination of 33 tensorfaces using the outer product (or Kronecker product) of a length-3 illumination vector and a length-11 person vector.
CANDECOMP/PARAFAC is Sum of Rank-1 Tensors

\[ \mathbf{X} \approx [\lambda ; \mathbf{A}, \mathbf{B}, \mathbf{C}] = \sum_{r} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \]

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

$$X(1) \approx A\Lambda(C \odot B)^\top$$

**Khatri-Rao Product**
(column-wise Kronecker product)

$$C \odot B \equiv \begin{bmatrix} c_1 \otimes b_1 & c_2 \otimes b_2 & \cdots & c_R \otimes b_R \end{bmatrix}$$

$$(C \odot B)^\dagger \equiv (C^\top C \ast B^\top B)^\dagger(C \odot B)^\top$$

If $C$, $B$, and $\Lambda$ are fixed, the optimal $A$ is given by:

$$A = X(1)(C \odot B)(C^\top C \ast B^\top B)^\dagger \Lambda^{-1}$$

*Repeat for $B, C$, etc.*
Application: PARAFAC & Web Data

Higher-order web link analysis using multilinear algebra
(with B. W. Bader, and J. P. Kenny)
In ICDM 2005, pp. 242–249, 2005

The TOPHITS model for higher-order web link analysis (with B. Bader)
In Workshop on Link Analysis, Counterterrorism and Security, 2006

Thanks also to Travis Bauer, and Ken Kolda for data
Hubs and Authorities (the HITS method)

Sparse adjacency matrix and its SVD:

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

\[
X \approx [\sigma; H, A] = \sum_r \sigma_r \cdot h_r \circ a_r
\]

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

<table>
<thead>
<tr>
<th>1st Principal Factor</th>
<th>2nd Principal Factor</th>
<th>3rd Principal Factor</th>
<th>4th Principal Factor</th>
<th>5th Principal Factor</th>
<th>6th Principal Factor</th>
</tr>
</thead>
</table>

Authority scores for 1st topic: from to

Hub scores for 1st topic: from to

Authority scores for 2nd topic: from to

Hub scores for 2nd topic: from to

Sandia National Laboratories
Three-Dimensional View of the Web

Observe that this tensor is very sparse!

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

with term $k$.
Topical HITS (TOPHITS)

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

\[
\mathbf{X} \approx [\lambda ; \mathbf{H}, \mathbf{A}, \mathbf{T}] = \sum_{r=1}^{R} \lambda_r \mathbf{h}_r \odot \mathbf{a}_r \odot \mathbf{t}_r
\]

TOPHITS Terms & Authorities on Sample Data

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

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

\( w_k = \# \text{ unique links using term } k \)

Tensor PARAFAC

\( \approx \) term scores for 1st topic + term scores for 2nd topic + term scores for 1st topic + ...
Google for “Tensor Toolbox” to find on the web

Efficient MATLAB computations with sparse and factored tensors (with B. W. Bader)

Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping (with B.W. Bader)
A Brief History of Tensors in MATLAB

• MATLAB (~1997)
  ▪ Version 5.0 adds support for multidimensional arrays (MDAs)

• N-way Toolbox (<1997)
  ▪ Extensive collection of functions and algorithms for analyzing multiway data
    • Handles constraints
    • Handles missing data
    • Etc.

• Tensor Toolbox V1.0 (7/2005)
  ▪ MATLAB classes for dense tensors, etc.
  ▪ Extends MDA capabilities to include multiplication, matrization, etc.
  ▪ Published in ACM TOMS

• Tensor Toolbox V2.0 (9/2006)
  ▪ Adds support for sparse tensors, etc.
  ▪ 489 registered users as of April 4th
  ▪ In review for SIAM Journal on Scientific Computing
Tensor Toolbox V2.0 supports 4 types of tensors

**Dense Tensors**  
- tensor
- Extends MATLAB’s native MDA capabilities
- Can be converted to a matrix and vice versa

**Sparse Tensors**  
- sptensor
- Unique to Tensor Toolbox
- Can be converted to a (sparse) matrix and vice versa
- Effort to choose suitable representation
- Efficient functions for computation

**Tucker Tensors**  
- ttensor
- Stores a tensor in decomposed form
- A different way to store a large-scale dense tensor
- Can do many operations in factored form

**Kruskal Tensors**  
- ktensor
- Stores a tensor as sum of rank-1 tensors
- A different way to store a large-scale dense tensor
- Can do many operations in factored form
Dense Tensors

• Largest tensor that can be stored on my 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

\[
\mathbf{y} = \mathbf{X} \times_1 \mathbf{U}
\]

\[
\mathbf{Y}(n) = \mathbf{U} \mathbf{X}(n)
\]
Sparse Tensors: Only Store Nonzeros

Example: Tensor-Vector Multiply (in all modes)

\[ \alpha = X \times_1 a \times_2 b \times_3 c \]

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

\[ = \sum_p \upsilon_p a_{s(p,1)} b_{s(p,2)} c_{s(p,3)} \]

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

\[ Y = X \times_3 z \]
\[ = (\mathcal{G} \times_1 U \times_2 V \times_3 W) \times_3 z \]
\[ = \mathcal{G} \times_1 U \times_2 V \times_3 W^T z \]
\[ = \mathcal{G} \times_3 W^T z \times_1 U \times_2 V = [\mathcal{H} ; U, V] \]

Result is a Tucker Tensor
Kruskal Example: Store Factors

\[ I \times J \times K \]

\[ I \times R \]

\[ J \times R \]

\[ R \times R \times R \]

\[ K \times R \]

\[ x \]

Kruskal tensors store factor matrices and scaling vector.

\[ \| x \|_2^2 = \| \left[ \lambda ; U, V, W \right] \|_2^2 \]

\[ = \| (W \odot V \odot U) \lambda \|_2^2 \]

\[ = \lambda^T (W \odot V \odot U)^T (W \odot V \odot U) \lambda \]

\[ = \lambda^T (W^T W \ast V^T V \ast U^T U) \lambda \]
Application: Cross-Language Information Retrieval

with Peter Chew, Brett Bader
Goal is to Cluster Independent of Language

- WWW includes many languages
- Our goal: Cluster documents regardless of language
- Tools
  - Latent Semantic Indexing (LSI)
  - Parallel corpora from the bible
- Challenge – extend LSI to cross-language context
Latent Semantic Indexing (LSI) & Clustering

Any *new* document is mapped into a $k$-dimensional vector in concept space by multiplying it by $U^T$. 

Term-Doc Matrix $\approx$ Term-Concept Matrix $\approx$ Concept-Doc Matrix

Rank-$k$ SVD Approximation

$X \approx UV^T \Sigma$
Standard Approach: One Matrix with All Languages

Use the Bible for parallel texts.

Run LSI on concatenated matrix. A single $U$ matrix is produced for all languages.
PARAFAC2 Model

PARAFAC2 produces a different U matrix for each language.

Trained on Bible. Tested on Quran.

**SVD**

Rank-300

For each document in each language on the vertical axis, we ranked documents in each of the other languages. The bar represents the average rank of the correct document. Rank 1 is ideal.

**PARAFAC2**

Rank-240

Closer to 1.0 is better
Application: DEDICOM and Enron Data

Temporal analysis of social networks using three-way DEDICOM
(with B. W. Bader and R. Harshman)
3-Way Dedicom Uncovers Latent Patterns in Data

We use DEDICOM to analyze a time-series graph of email communications.

We construct a tensor $\mathbf{X}$ such that $x_{ijk}$ is nonzero if person $i$ sent an email to person $j$ in time period $k$. 

$$
\mathbf{X} \approx \mathbf{A} \mathbf{D} \mathbf{R} \mathbf{D} \mathbf{A}^T
$$
Three-way DEDCOM with 4 factors identified the four roles roughly labeled as Executive, Govt. Affairs, Legal, and Pipeline.

Graph for \( k = \text{Oct 2000} \) (pre-crisis)

Graph for \( k = \text{Oct 2001} \) (during crisis)
Other Applications

- Bibliometric Data
  - Author x Term x Year
  - Doc x Doc x Connection Type
- Social Network Analysis
  - Person x Person x Connection Type x Date
- Network Traffic Analysis
  - HTTP logs, etc.
- Also working on tools for creating *semantic* graphs from raw data
- Joint with Ann Yoshimura, SNL
Tensor Toolbox is Software for Working with Multidimensional Arrays

- New software makes working with tensors easy
- Lots of applications of tensors within data mining
  - Web analysis
  - Cross-language IR
  - Email analysis
  - Etc.
- “New” area for math & CS for research

More information:
- tgkolda@sandia.gov
- http://csmr.ca.sandia.gov/~tgkolda/
Questions?

Tammy Kolda
tgkolda@sandia.gov
http://csmr.ca.sandia.gov/~tgkolda/