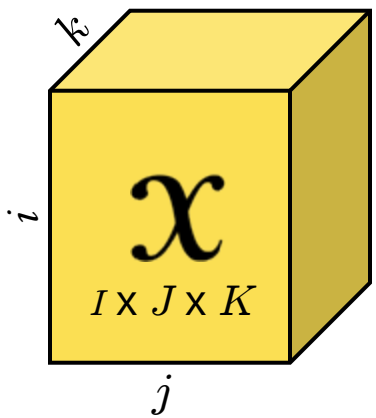
Efficient Computations with Tensors and Examples from Data Mining

Tamara G. Kolda
Sandia National Laboratories

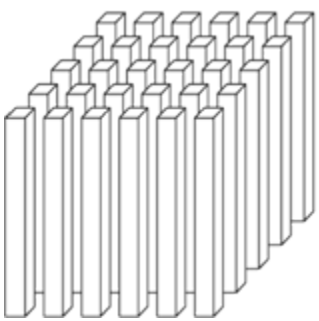
Collaborators
Brett Bader and Peter Chew
Sandia National Laboratories

Tensor Background

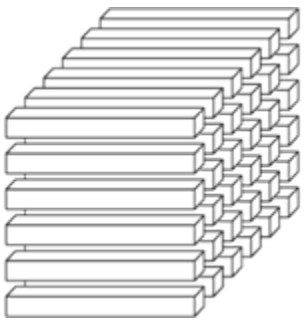
Tensor Basics: Fibers and Matricizing



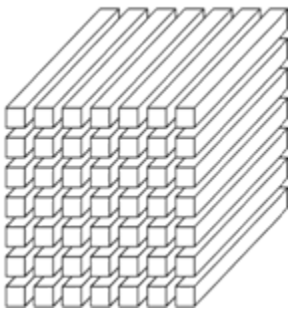
Column (Mode-1) Fibers



Row (Mode-2) Fibers



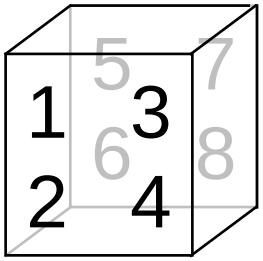
Tube (Mode-3) Fibers



Matricizing/Unfolding

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

$$\mathcal{X} =$$



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

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

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

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

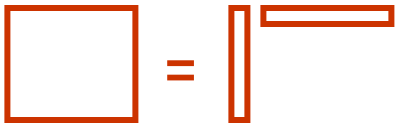
Sandia National Laboratories

Vector Outer and Kronecker Products

2-Way Outer Product
($I \times J$ Rank-1 Matrix)

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

$$x_{ij} = a_i b_j \ \forall\ i, j$$



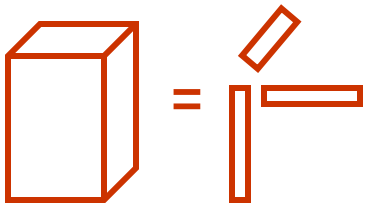
2-Way Kronecker Product
(IJ-Vector)

$$\text{vec}(\mathbf{X}) = \mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1\mathbf{b} \\ a_2\mathbf{b} \\ \vdots \\ a_N\mathbf{b} \end{bmatrix}$$

3-Way Outer Product
($I \times J \times K$ Rank-1 Tensor)

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

$$x_{ijk} = a_i b_j c_k \ \forall\ i, j, k$$



3-Way Kronecker Product
(IJK-Vector)

$$\text{vec}(\mathcal{X}) = \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} = \begin{bmatrix} a_1(\mathbf{b} \otimes \mathbf{c}) \\ a_2(\mathbf{b} \otimes \mathbf{c}) \\ \vdots \\ a_N(\mathbf{b} \otimes \mathbf{c}) \end{bmatrix}$$

Sandia National Laboratories

Matrix Kronecker & Khatri-Rao Products

Kronecker Product
(MP x NQ Matrix)

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

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

Khatri-Rao Product
(MN x R Matrix)

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

$$(\mathbf{A} \odot \mathbf{B})^\mathsf{T}(\mathbf{A} \odot \mathbf{B}) = (\mathbf{A}^\mathsf{T}\mathbf{A} * \mathbf{B}^\mathsf{T}\mathbf{B})$$

$$(\mathbf{A} \odot \mathbf{B})^\dagger \equiv \underbrace{(\mathbf{A}^\mathsf{T}\mathbf{A} * \mathbf{B}^\mathsf{T}\mathbf{B})^\dagger}_{R \times R}(\mathbf{A} \odot \mathbf{B})^\mathsf{T}$$

Hadamard (Elementwise) Product

Sandia National Laboratories

Tensor Times Matrix

Tensor Times Matrix in Mode-1

$$\overset{M \times J \times K}{\mathcal{Y}} = \overset{I \times J \times K}{\mathcal{X}} \times_1 \overset{M \times I}{\mathbf{A}}$$

$$y_{mjk} = \sum_i x_{ijk} a_{mi}$$

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

Tensor Times Matrix in All Modes

$$\overset{M \times N \times P}{\mathcal{Y}} = \overset{I \times J \times K}{\mathcal{X}} \times_1 \overset{M \times I}{\mathbf{A}} \times_2 \overset{N \times J}{\mathbf{B}} \times_3 \overset{P \times K}{\mathbf{C}}$$

$$y_{mnp} = \sum_i \sum_j \sum_k x_{ijk} a_{mi} b_{nj} c_{pk}$$

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

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

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

$$\mathsf{vec}(\mathcal{Y}) = (\mathbf{C} \otimes \mathbf{B} \otimes \mathbf{A})\mathsf{vec}(\mathcal{X})$$

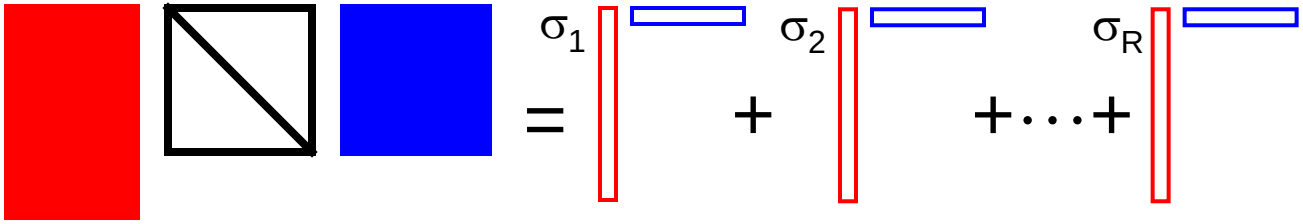
Sandia National Laboratories

Primary Tensor Decompositions

What is the higher-order analogue of the Matrix SVD?

Two views of the matrix SVD:

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



Finding bases for row and column subspaces:

$$\mathbf{X} = \Sigma \times_1 \mathbf{U} \times_2 \mathbf{V}$$

→ Tucker Decomposition

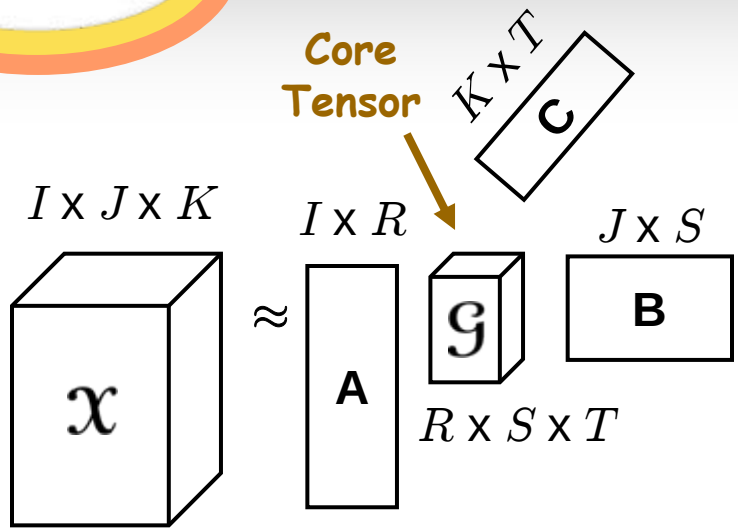
Sum of R rank-1 matrix factors (where R is the rank):

$$\mathbf{X} = \sum_{r=1}^{R} \sigma_r \, \mathbf{u}_r \circ \mathbf{v}_r$$

→ CANDECOMP/ PARAFAC

Sandia National Laboratories

Tucker Decomposition



$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

$$x_{ijk} \approx \sum_r \sum_s \sum_t g_{rst}\, a_{ir}\, b_{js}\, c_{kt}$$

$$\mathcal{X} \approx \sum_r \sum_s \sum_t g_{rst}\, \mathbf{a}_r \circ \mathbf{b}_s \circ \mathbf{c}_t$$

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

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

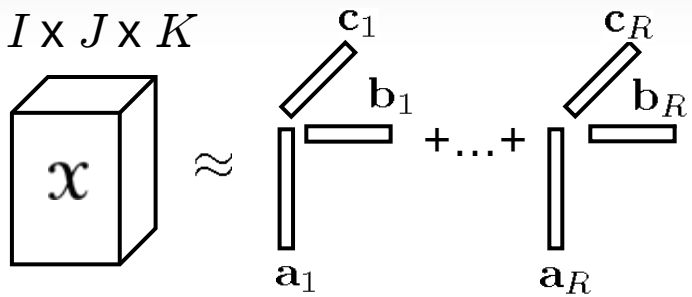
$$\mathbf{X}_{(3)} \approx \mathbf{C}\mathbf{G}_{(3)}(\mathbf{B} \otimes \mathbf{A})^{\mathsf{T}}$$

$$\mathrm{vec}(\mathcal{X}) = (\mathbf{C} \otimes \mathbf{B} \otimes \mathbf{A})\mathrm{vec}(\mathcal{G})$$

- Also known as: three-mode factor analysis, three-mode PCA, orthogonal array decomposition
- Sizes R, S, T chosen by the user.
- **A**, **B**, and **C** may be orthonormal (generally assume full column rank)
- Core is not diagonal
- Not unique

See Tucker, *Psychometrika*, 1966; see also Hitchcock, 1927.

CANDECOMP/PARAFAC (CP)



$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_R \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_R \end{bmatrix}$$
$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_R \end{bmatrix}$$

$$\mathcal{X} \approx \sum_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

$$x_{ijk} \approx \sum_r a_{ir}\, b_{jr}\, c_{kr}$$

$$\mathbf{X}_{(1)} \approx \mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}$$
$$\mathbf{X}_{(2)} \approx \mathbf{B}(\mathbf{C} \odot \mathbf{A})^{\mathsf{T}}$$
$$\mathbf{X}_{(3)} \approx \mathbf{C}(\mathbf{B} \odot \mathbf{A})^{\mathsf{T}}$$

$$\mathrm{vec}(\mathcal{X}) = (\mathbf{C} \odot \mathbf{B} \odot \mathbf{A})\mathbf{1}$$

- CANDECOMP = Canonical Decomposition
- PARAFAC = Parallel Factors
- Columns of **A**, **B**, and **C** are not orthonormal
- Exact decomposition is often unique

Carroll & Chang, *Psychometrika*, 1970, Harshman, 1970 – plus Hitchcock, 1927.

MATLAB Tensor Toolbox

MATLAB has MDAs

- Standard Operations
  - Subscripted reference and assignment
  - Size queries (size, ndims, nnz)
  - Permute/squeeze
  - Elementwise and scalar operations (+,-,*,/,^,etc)
  - Logical operations (and,or,xor,not)
  - Comparisons (==,>,<,>=, <=,~=)

Multidimensional Arrays (MDAs)

Dense Only!

No support for multiplication, etc.

Tensor Toolbox adds functionality & sparse support

- Standard Operations
  - Subscripted reference and assignment
  - Size queries (size, ndims, nnz)
  - Permute/squeeze
  - Elementwise and scalar operations (+,-,*,/,^,etc)
  - Logical operations (and,or,xor,not)
  - Comparisons (==,>,<,>=, <=,~=)
- Tensor-Specific Operations
  - Matricize
  - Tensor multiplication
    - outer product, etc.
  - Contraction
  - Norm
- Other Tensor Operations
  - Collapse/scale
  - Matricized-tensor-times-Khatrio-Rao-product
  - Mode-n singular vectors
- Khatri-Rao product, etc.

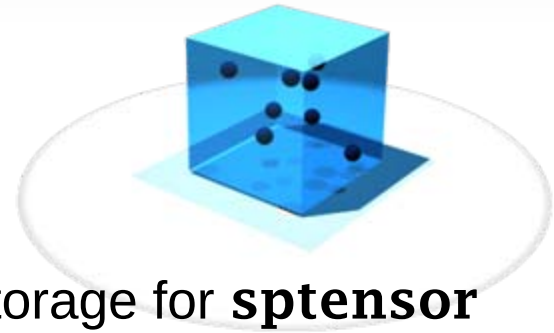
For dense, sparse, and structured tensors.

Fully object-oriented.

Sparse Tensors

Exploiting Sparsity: Sparse Tensors (sptensor)

- *Sparse* if majority of entries (x_{ijk}) are zero
- Some storage options
  - Each two-dimensional slice stored as sparse matrix
  - Unfold and store as sparse matrix
    - Lin, Liu, Chung, IEEE Trans. Computers, 2002 & 2003
  - Coordinate format



- Storage for **sptensor**
  - P = # nonzeros
  - **vals** = P x 1 vector of nonzero values
  - **subs** = P x 3 matrix of subscripts

Norm of sparse tensor

$$\|\mathcal{X}\| = \sqrt{\sum_{ijk} x_{ijk}^2} = \sqrt{\sum_p [\mathtt{vals}(p)]^2}$$

Sparse Storage Example

2 x 2 x 2 Tensor with P = 4 Nonzeros

$$x_{111} = 1.5$$
$$x_{121} = 2.7$$
$$x_{212} = 3.3$$
$$x_{222} = 8.5$$

$$\texttt{subs} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \\ 2 & 2 & 2 \end{bmatrix} \qquad \texttt{vals} = \begin{bmatrix} 1.5 \\ 2.7 \\ 3.3 \\ 8.5 \end{bmatrix}$$

Tucker for Sparse

Fitting Tucker

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

Fact 1: Optimal core exists

Assume A, B, C orthonormal

$$\text{vec}(\mathcal{X}) = (\mathbf{C} \otimes \mathbf{B} \otimes \mathbf{A})\text{vec}(\mathcal{G})$$
$$\Rightarrow \mathcal{G} = \mathcal{X} \times_1 \mathbf{A}^\top \times_2 \mathbf{B}^\top \times_3 \mathbf{C}^\top$$

Fact 2: Core can be eliminated to form objective in A,B,C

$$\min \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}\|^2$$
$$\min \|\mathcal{X}\|^2 - 2\langle \mathcal{X}, \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \rangle + \|\mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}\|^2$$
$$\min \|\mathcal{X}\|^2 - 2\langle \mathcal{X} \times_1 \mathbf{A}^\top \times_2 \mathbf{B}^\top \times_3 \mathbf{C}^\top, \mathcal{G} \rangle + \|\mathcal{G}\|^2$$
$$\min \|\mathcal{X}\|^2 - 2\langle \mathcal{G}, \mathcal{G} \rangle + \|\mathcal{G}\|^2$$
$$\max \|\mathcal{G}\|^2$$
$$\max \|\mathcal{X} \times_1 \mathbf{A}^\top \times_2 \mathbf{B}^\top \times_3 \mathbf{C}^\top\|^2$$
$$\max \|\mathbf{A}^\top \mathbf{X}_{(1)} (\mathbf{C} \otimes \mathbf{B})\|$$

If B & C are completely unknown, solve:

$$\max \|\mathbf{A}\mathbf{X}_{(1)}\|$$

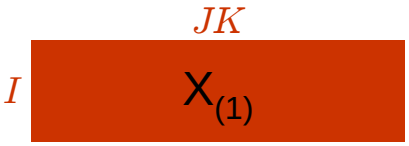
Fixing B & C, can solve this equation for A

HO-SVD (Tucker1)

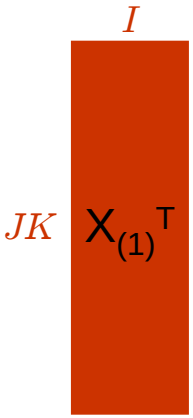
Simplest approach; much more sophisticated methods exist.

Find optimal component w/o knowledge of other components:

$$\left.\begin{array}{l} \max \|\mathbf{A}\mathbf{X}_{(1)}\| \\ \max \|\mathbf{B}\mathbf{X}_{(2)}\| \\ \max \|\mathbf{C}\mathbf{X}_{(3)}\| \end{array}\right\} \text{Need to find leading left singular vectors of } X_{(n)}$$

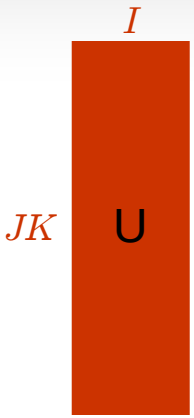


- Convert tensor to MATLAB sparse matrix
- Bad: $X_{(1)}$ is a wide, short matrix
  - Size I x JK
  - Worst possible aspect ratio for MATLAB's CSC format
- Good: $U = X_{(1)}^T$, which is tall and skinny
  - Size JK x I
- To compute left singular vectors of $X_{(1)}$, calculate eigenvalues of $V = UU^T$
  - Size I x I



De Lathauwer, De Moor, & Vandewalle, SIMAX, 2000.
Also known as "Method 1" in Tucker, 1966.

eigs vs. svds



Create 100x100x100 (matricized) tensor with 5000 nonzeros.

```
>> I = 100; J = 100; K = 100; P = 5000;
>> U = sprand(J*K, I, P/(I*J*K));
>> R = 10;
```

Calculate eigenvectors of $U^T U$

```
>> tic; V = U'*U; [U1,D1] = eigs(V,R,'LM'); toc
Elapsed time is 0.023073 seconds.
```

Calculate singular vectors of U

```
>> tic; [U2,S2,V2] = svds(U,R); toc
Elapsed time is 0.237503 seconds.
```

In MATLAB, eigenvalue calculation is 10x faster than SVD.

Computing Core Tensor is Difficult

$R \times S \times T$ dense $I \times J \times K$ sparse $R \times I$ $S \times J$ $T \times K$

$$\mathcal{G} = \underbrace{\mathcal{X} \times_1 \mathbf{A}^\top}_{R \times J \times K \text{ dense}} \times_2 \mathbf{B}^\top \times_3 \mathbf{C}^\top$$

- Final core is small
- But intermediate results are large
- Requires too much time and memory for even moderate sizes (1000 x 1000 x 1000)
- Currently researching ways to compute this efficiently…

Sandia National Laboratories

CP for Sparse

Fitting CP

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

shorthand notation for sum – corresponds to **ktensor** in the Tensor Toolbox

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \| \mathcal{X} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \|$$

$$\left.\begin{aligned} &\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}\| \\ &\min_{\mathbf{B}} \|\mathbf{X}_{(2)} - \mathbf{B}(\mathbf{C} \odot \mathbf{A})^{\mathsf{T}}\| \\ &\min_{\mathbf{C}} \|\mathbf{X}_{(3)} - \mathbf{C}(\mathbf{B} \odot \mathbf{A})^{\mathsf{T}}\| \end{aligned}\right\}$$

Successively solve least squares problems.

Exploit structure of Khatri-Rao inverse.

Continue until fit ceases to improve:

$$\begin{aligned} \text{fit} &= \|\mathcal{X} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|^2 \\ &= \|\mathcal{X}\|^2 - 2\langle \mathcal{X}, [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \rangle + \|[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|^2 \end{aligned}$$

Sandia National Laboratories

CP-ALS Algorithm

Specify R. $\boxed{\text{Initialize } \mathbf{B}, \mathbf{C}.}$ Tucker1

Repeat

Matricized tensor times Khatri-Rao product (mttkrp)

$$\mathbf{A} = \hat{\mathbf{A}}(\mathbf{C}^\top\mathbf{C} * \mathbf{B}^\top\mathbf{B})^\dagger \text{ with } \hat{\mathbf{A}} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$$
$$\mathbf{B} = \hat{\mathbf{B}}(\mathbf{C}^\top\mathbf{C} * \mathbf{A}^\top\mathbf{A})^\dagger \text{ with } \hat{\mathbf{B}} = \mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})$$
$$\mathbf{C} = \hat{\mathbf{C}}(\mathbf{B}^\top\mathbf{B} * \mathbf{A}^\top\mathbf{A})^\dagger \text{ with } \hat{\mathbf{C}} = \mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})$$

$$\text{fit} = \underbrace{\boxed{\|\mathcal{X}\|^2}}_{\text{Norm}} - 2\underbrace{\boxed{\langle \mathcal{X}, [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \rangle}}_{\text{Inner Product}} + \underbrace{\boxed{\| [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \|^2}}_{\text{Norm}}$$

Until $\Delta_{\text{fit}} < \text{tol}$

Matricized Tensor Times Khatri-Rao Product (`mttkrp`)

$$\hat{\mathbf{A}} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$$

I x JK sparse

JK x R very big!

Don't want to compute explicitly

Trick 1: compute solution column-wise (for r=1,…,R):

$$\hat{\mathbf{a}}_r = \mathbf{X}_{(1)}(\mathbf{c}_r \otimes \mathbf{b}_r)$$

Trick 2: Do not form unfolded tensor or Kronecker product

$$(\mathbf{a}_r)_i = \sum_j \sum_k x_{ijk} (\mathbf{b}_r)_j (\mathbf{c}_r)_k$$

Trick 3: Optimize in MATLAB by avoiding loops

Avoiding loops in mttkrp

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

Storage for **`sptensor`**

- P = # nonzeros
- **`vals`** = P x 1 vector of nonzero values
- **`subs`** = P x 3 matrix of subscripts

2 x 2 x 2 tensor with 4 nonzeros

$$\texttt{subs} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \\ 2 & 2 & 2 \end{bmatrix} \quad \texttt{vals} = \begin{bmatrix} 1.5 \\ 2.7 \\ 3.3 \\ 8.5 \end{bmatrix}$$

Vectors

$$\mathbf{b} = \begin{bmatrix} .1 \\ .2 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 30 \\ 40 \end{bmatrix}$$

```
z = vals .* b(subs(:,2)) .* c(subs(:.3))
a = accumarray(z, subs(:,1))
```

$$\begin{bmatrix} 1.5 \\ 2.7 \\ 3.3 \\ 8.5 \end{bmatrix} .* \begin{bmatrix} .1 \\ .2 \\ .1 \\ .2 \end{bmatrix} .* \begin{bmatrix} 30 \\ 30 \\ 40 \\ 40 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 16.2 \\ 13.2 \\ 68.0 \end{bmatrix} \rightarrow \begin{bmatrix} 20.7 \\ 81.2 \end{bmatrix} = \mathbf{a}$$

Sandia National Laboratories

CP-ALS Algorithm

Specify R. $\boxed{\text{Initialize } \mathbf{B}, \mathbf{C}.}$ Tucker1

Repeat

Matricized tensor times Khatri-Rao product (mttkrp)

$$\mathbf{A} = \hat{\mathbf{A}}(\mathbf{C}^\top\mathbf{C} * \mathbf{B}^\top\mathbf{B})^\dagger \text{ with } \hat{\mathbf{A}} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$$
$$\mathbf{B} = \hat{\mathbf{B}}(\mathbf{C}^\top\mathbf{C} * \mathbf{A}^\top\mathbf{A})^\dagger \text{ with } \hat{\mathbf{B}} = \mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})$$
$$\mathbf{C} = \hat{\mathbf{C}}(\mathbf{B}^\top\mathbf{B} * \mathbf{A}^\top\mathbf{A})^\dagger \text{ with } \hat{\mathbf{C}} = \mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})$$

$$\text{fit} = \underbrace{\boxed{\|\mathcal{X}\|^2}}_{\text{Norm}} - 2\underbrace{\boxed{\langle \mathcal{X}, [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \rangle}}_{\text{Inner Product}} + \underbrace{\boxed{\| [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \|^2}}_{\text{Norm}}$$

Until $\Delta_{\text{fit}} < \text{tol}$

Norm of ktensor

Cannot form tensor explicitly because it would be too large.

$$\alpha = \| \underbrace{[\![\overset{I \times R}{\mathbf{A}}, \overset{J \times R}{\mathbf{B}}, \overset{K \times R}{\mathbf{C}}]\!]}_{\substack{I \times J \times K \\ \text{dense}}} \|^2$$

$$\begin{aligned}
\alpha &= \| [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \|^2 \\
&= \langle \mathrm{vec}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]), \mathrm{vec}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) \rangle \\
&= \mathbf{1}^\mathsf{T} (\mathbf{C} \odot \mathbf{B} \odot \mathbf{A})^\mathsf{T} (\mathbf{C} \odot \mathbf{B} \odot \mathbf{A}) \mathbf{1} \\
&= \mathbf{1}^\mathsf{T} (\underbrace{\mathbf{C}^\mathsf{T}\mathbf{C} * \mathbf{B}^\mathsf{T}\mathbf{B} * \mathbf{A}^\mathsf{T}\mathbf{A}}_{R \times R}) \mathbf{1}
\end{aligned}$$

Inner Product: `sptensor` & `ktensor`

$$\langle \mathcal{X}, [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \rangle$$

$$= \sum_i \sum_j \sum_k x_{ijk} \left(\sum_r a_{ir} b_{jr} c_{kr} \right)$$

write out product

$$= \sum_r \underbrace{\left(\sum_i \sum_j \sum_k x_{ijk} a_{ir} b_{jr} c_{kr} \right)}$$

rearrange terms

```
sum( vals .* ar(subs(:,1)) .* br(subs(:,2)) .* cr(subs(:.3)) )
```

Toolbox Numerical Results

Numerical Results: Dense vs. Sparse

	Sparse	Dense	Speed-Up	MATLAB	Speed-Up
Permute 1st and 2nd dims	0.0019s	0.3307s	174x	0.2845s	150x
Norm of X	0.0007s	0.0609s	87x	0.0874s	125x
Logical operation: X > 5	0.0588s	0.2570s	4x	0.0796s	None
Scalar Multiply: X*5	0.0972s	0.2402s	2x	0.1641s	2x
Plus: X+X	0.2455s	0.4223s	2x	0.2540s	None
Find all nonzeros	0.0006s	0.2079s	347x	0.1845s	308x
Sum all nonzeros	0.0013s	0.3786s	291x	0.1078s	83x
Tensor-Vector Multiply	0.0898s	0.8174s	9x	N/A	
Contraction	0.0035s	0.4479s	128x	N/A	
5 Dominant Mode-1 Vectors	0.1036s	5.2188s	50x	N/A	

Tests on 256 x 256 x 256 tensor with 32,000 nonzeros.
1.66GHz Intel CoreDuo laptop with 2GB of RAM

Numerical Results: Sparse – 500,000 nonzeros

	Sparse
Permute 1st and 2nd dims	0.0312s
Norm of X	0.0040s
Logical operation: X > 5	0.9934s
Scalar Multiply: X*5	1.6355s
Plus: X+X	4.2492s
Find all nonzeros	0.0022s
Sum all nonzeros	0.0043s
Tensor-Vector Multiply	0.0043s
Contraction	0.0312s
Dominant Mode-1 Vectors	5.7626s

Tests on a 10,000 x 10,000 x 10,000 tensor with ½ million nonzeros.
1.66GHz Intel CoreDuo laptop with 2GB of RAM

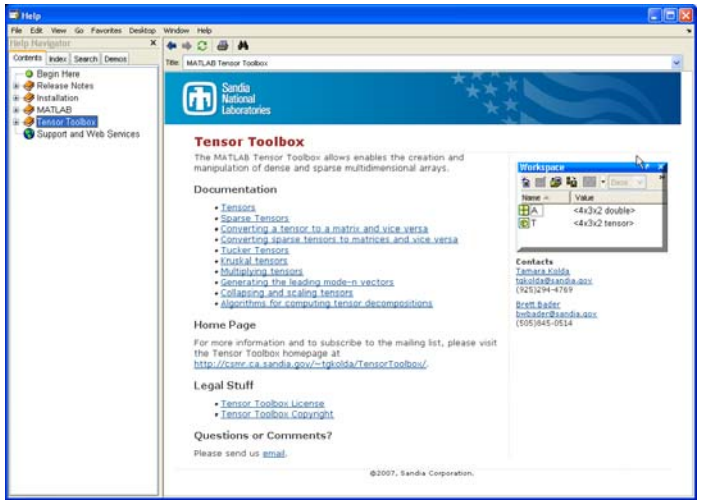
Sandia National Laboratories

Toolbox Summary

Tensor Toolbox

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

- Seamless integration into MATLAB
- Object-oriented classes
- Enables storage of large-scale sparse tensors
- Most extensive library of tensor operations available
- Documentation available within MATLAB
- Over 1000 unique registered users since release in 9/06



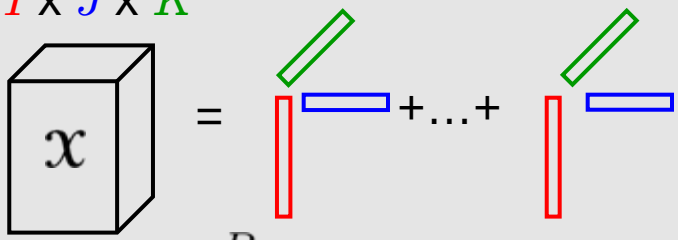
- Areas for toolbox improvement
  - Smarter memory manipulation in dense operations; avoid memory copies
  - Extend to other languages (C++)
  - More and better decomposition methods
  - Suggestions welcome!

Sandia National Laboratories

PARAFAC2 and an Application in Data Mining

Yet another view of PARAFAC

$I \times J \times K$



$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

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

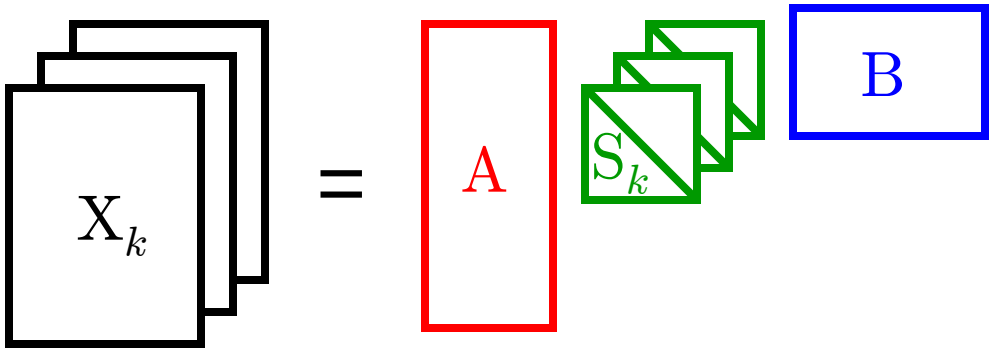
$$\mathbf{X}_{(2)} \approx \mathbf{B}(\mathbf{C} \odot \mathbf{A})^\mathsf{T}$$

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

$$\mathrm{vec}(\mathcal{X}) \approx (\mathbf{C} \odot \mathbf{B} \odot \mathbf{A})\mathbf{1}$$

$$\mathbf{X}_k = \mathbf{A}\mathbf{S}_k\mathbf{B}^\mathsf{T}$$

$\mathrm{S}_k = \mathrm{diag}(k^{\text{th}}$ <u>*row*</u> of C)



This representation only works for 3rd-order tensors.

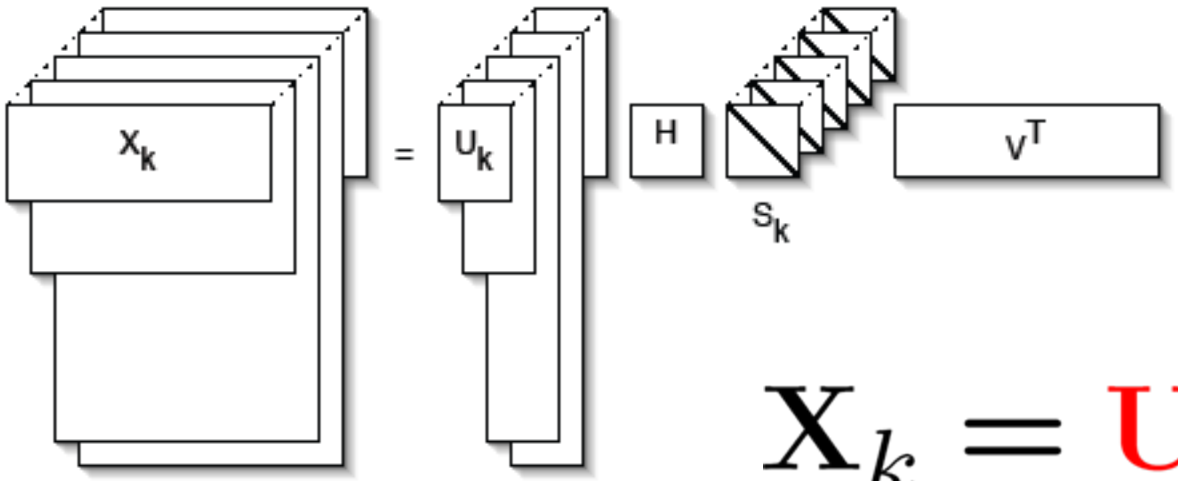
Looks like SVD.

PARAFAC2

(not, strictly speaking, a tensor decomposition)

PARAFAC

$$\mathbf{X}_k = \mathbf{A}\mathbf{S}_k\mathbf{B}^\mathsf{T}$$



$$\mathbf{X}_k = \mathbf{U}_k\mathbf{H}\mathbf{S}_k\mathbf{V}^\mathsf{T}$$

Not a tensor, but similar

orthonormal columns

used to enforce uniqueness

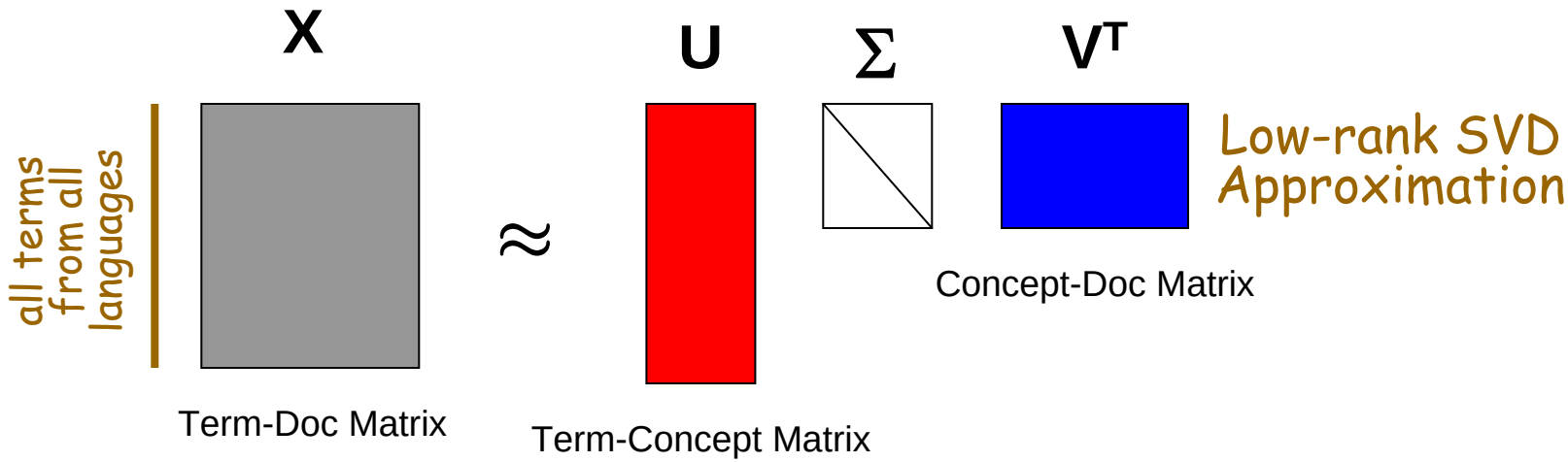
diagonal

R. A. Harshman, *UCLA Working Papers in Phonetics*, 1972.

Application: Cross-Language Information Retrieval

Latent Semantic Indexing (LSI) in Multilingual Environment

Step 1: Compute SVD on Parallel Corpus for training. Each "document" consists of <u>all</u> its translations.



Step 2: Map test documents to concept space. Each document is only a <u>single</u> translation.

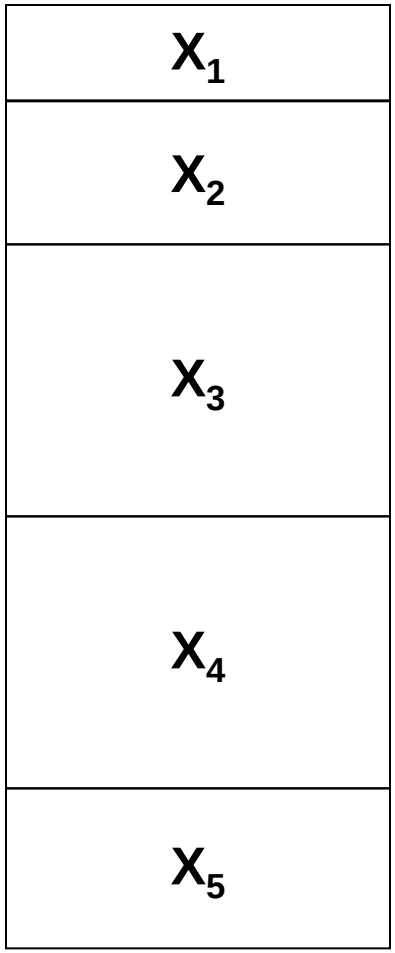
$$\hat{\mathbf{Y}} = \mathbf{U}\mathbf{Y}$$

Same U for all languages.

Goal

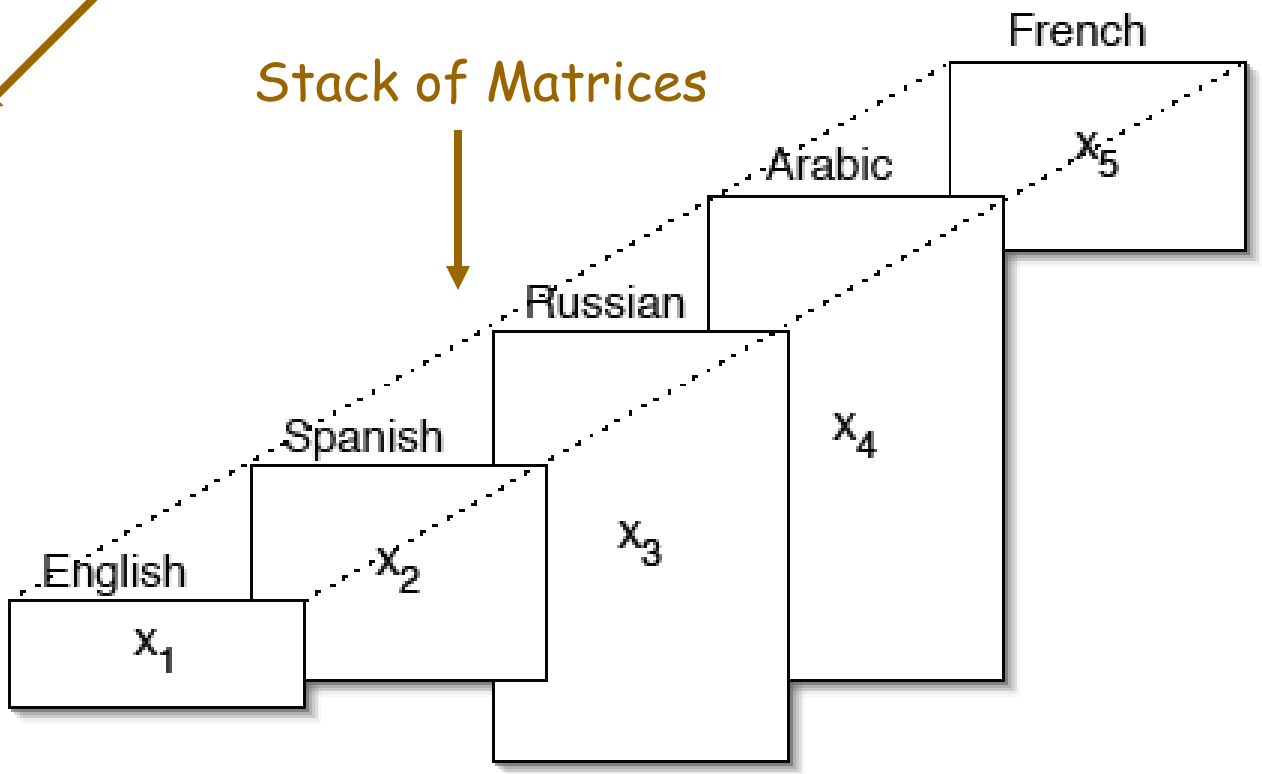
Different translations of the same document should be nearby in concept space.

A Different View



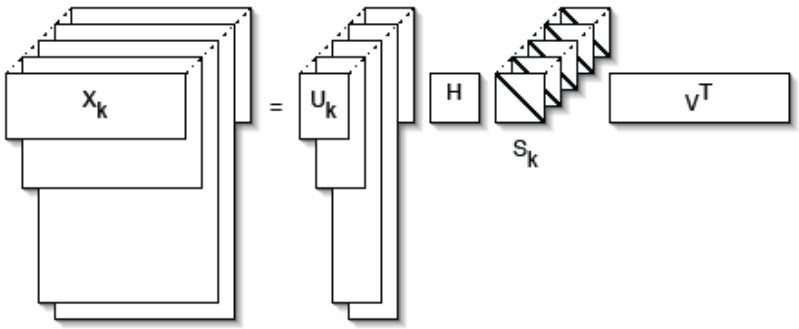
LSI Matrix (though terms are mixed)

Stack of Matrices



PARAFAC2 Model

Step 1: Compute PARAFAC2 on Parallel Corpus for training. Each "document" consists of <u>all</u> its translations.



Step 2: Map test documents to concept space. Each document is only a <u>single</u> translation.

$$\hat{\mathbf{Y}}_k = \mathbf{U}_k \mathbf{Y}_k$$

Minor Drawback

Need to know language of test document.

Results Comparison

Trained on Bible.
Tested on Quran.

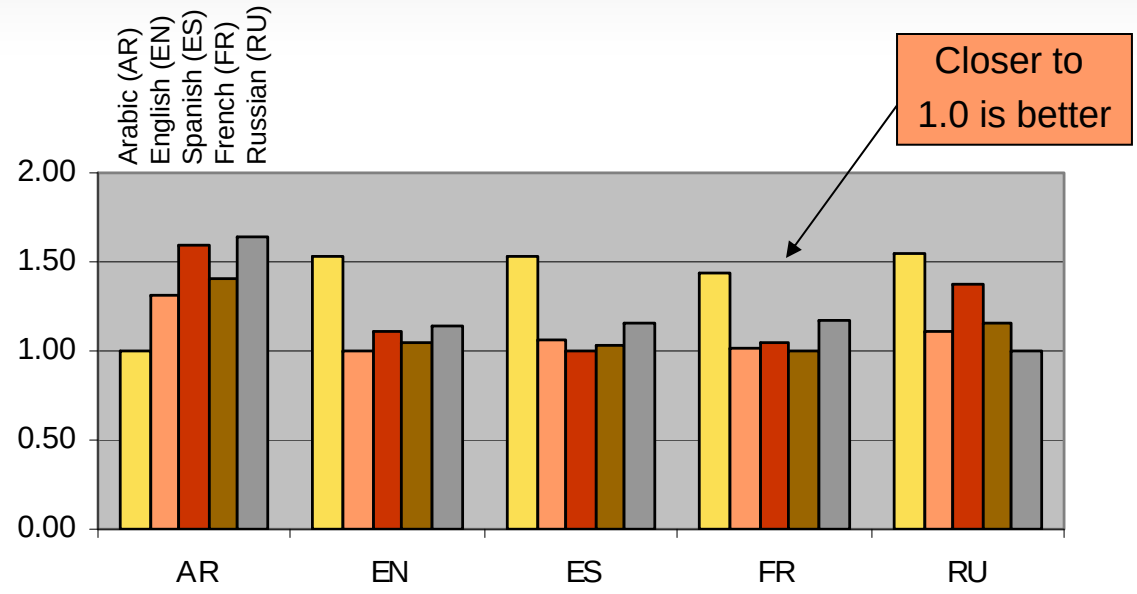
SVD

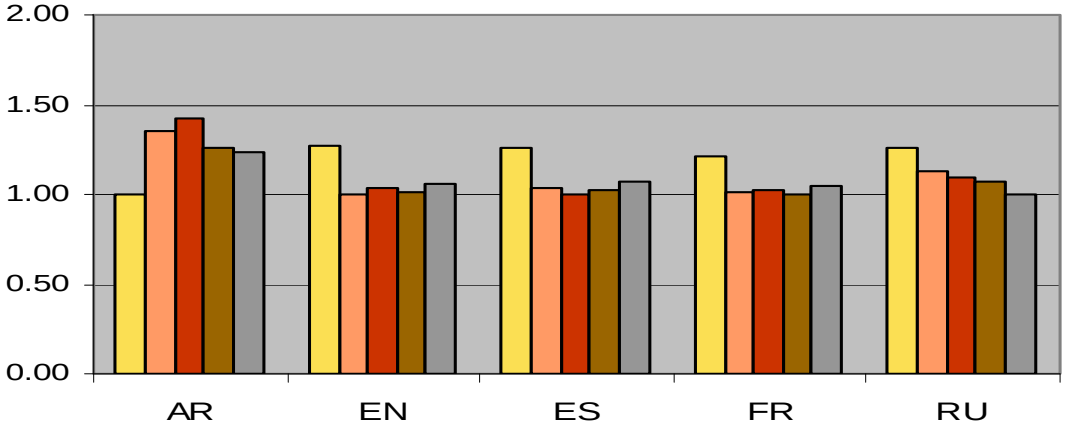
Rank-300

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



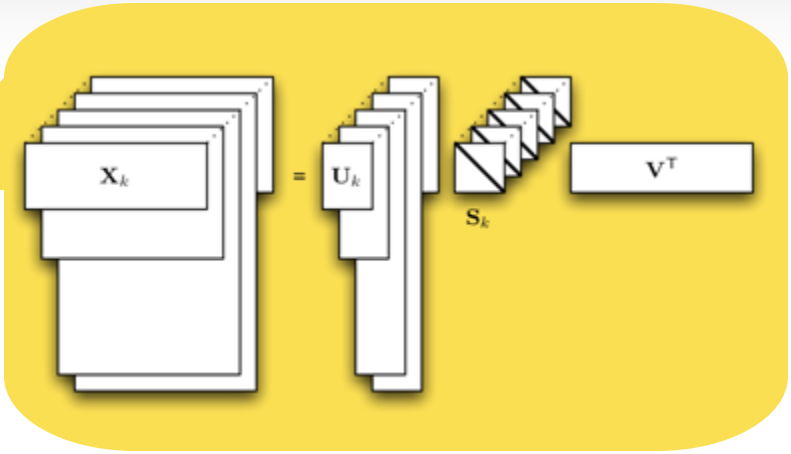


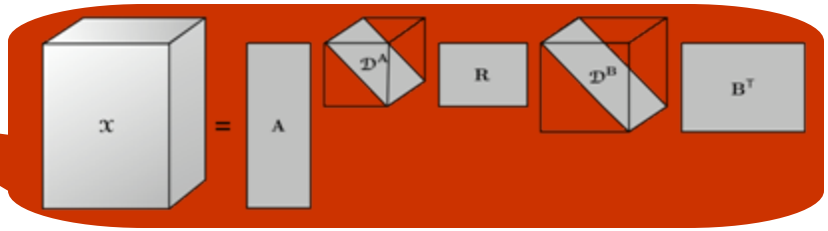
Sandia National Laboratories

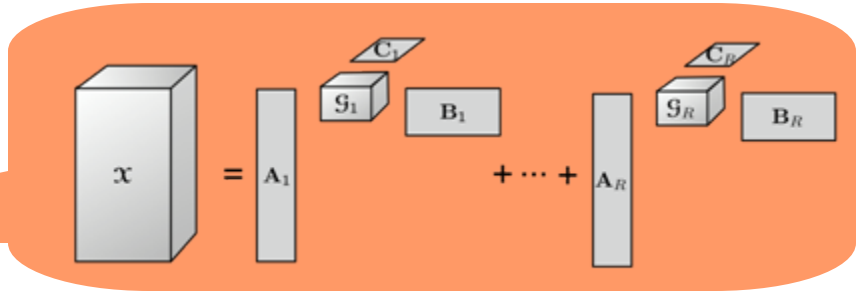
Other Decompositions

Other Decompositions

- **INDSCAL:** Individual Differences in Scaling (Carroll & Chang, 1972)
- **PARAFAC2:** (Harshman, 1978)
- **CANDELINC:** Linearly constrained CP (Carroll, Pruzansky, Kruskal, 1980)
- **DEDICOM:** Decomposition into directional components (Harshman, 1972)
- **PARATUCK2:** Generalization of DEDICOM (Harshman & Lundy, 1996)
- **Nonnegative tensor factorizations:** (Bro and De Jung, 1997; Paatero, 1997; Welling and Weber, 2001; etc.)
- **Block factorizations:** (De Lathauwer, 2007; etc.)

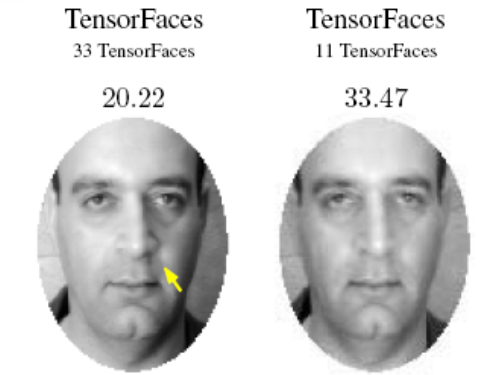


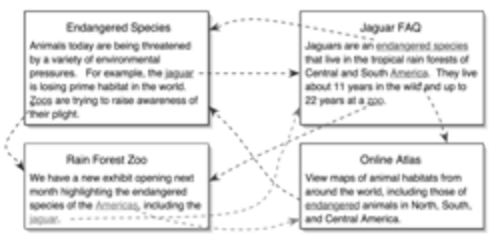


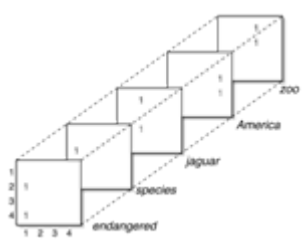


Sandia National Laboratories

Other Data Mining Applications

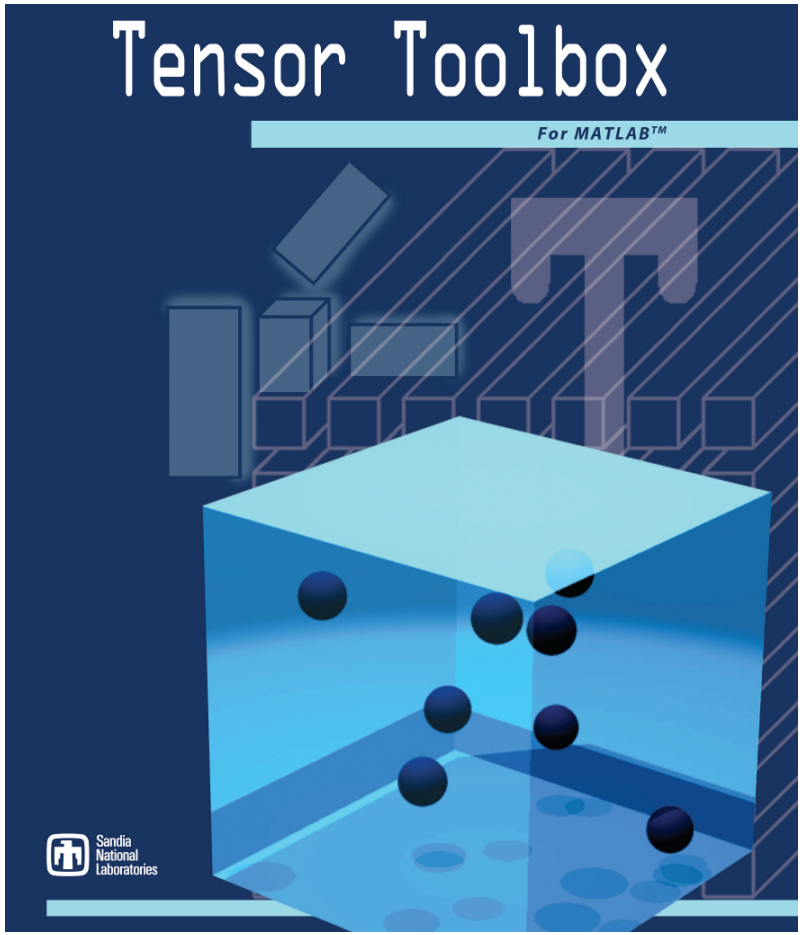






- **Higher-Order PCA** – Tucker or CP to decompose a data stream. Useful in a variety of contexts such as chemometrics. (R. Bro, Critical Reviews in Analytical Chemistry, 2007)
- **TuckerFaces and Image Analysis** – HO-SVD of image tensor. (M.A.O. Vasilescu & D. Terzopoulos, CVPR, 2003)
- **Hand-Written Digit Analysis** – Classification problem. (Eldén and Savas, Pattern Recognition, 2007)
- **Chatroom Analysis(*)** – Comparison of Tucker and CP to distinguish conversations in chatrooms. (Acar et al, ISI 2005 and ISI 2006)
- **TOPHITS(*)** – CP of page x page x anchor text link tensor from the web graph to compute hubs, authorities, and topics. (Kolda, Bader, and Kenney, ICDM, 2005)
- **Window-Based Tensor Analysis and Dynamic Tensor Analysis(*)** – Network intrusion detection. (Sun et al., ICDM 2006 and KDD 2006)
- **Multi-way Clustering on Relational Graphs(*)** – Using a variety of metrics for clustering. (Banerjee, Basu, and Merugu, SDM 2007)
- **Enron email analysis(*)** – Using CP and nonnegative CP. (Bader, Berry, Browne, Text Analysis Workshop at SDM 2007)
- **EEG Analysis** – Detecting onset of epileptic seizure using CP and multiway PLS. (Acar et al., 2007)

Conclusions & Future Work



- Conclusions
  - Special data structures enable computations with large-scale tensors
  - Applications to data mining
- Future work
  - Tucker for sparse tensors (with J. Sun)
  - Tensor methods for clustering (with T. Selee)
  - Development of C++ tensor libraries (serial and parallel) with colleagues at Sandia
  - FAQ for Einstein notation and the tensor toolbox

References & Contact Info

- B. W. Bader and T. G. Kolda. ***Efficient MATLAB computations with sparse and factored tensors***. SIAM Journal on Scientific Computing, Volume 30, Number 1, Pages 205-231, December 2007. [DOI:10.1137/060676489]
- B. W. Bader and T. G. Kolda. ***Algorithm 862: MATLAB tensor classes for fast algorithm prototyping***. ACM Transactions on Mathematical Software, Volume 32, Number 4, Pages 635-653, December 2006. [DOI:10.1145/1186785.1186794]
- T. G. Kolda and B. W. Bader. ***Tensor Decompositions and Applications***. Technical Report Number SAND2007-6702, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, November 2007.
- P. A. Chew, B. W. Bader, T. G. Kolda and A. Abdelali. ***Cross-language information retrieval using PARAFAC2***. KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 143-152, ACM Press, 2007. [DOI:10.1145/1281192.1281211]

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