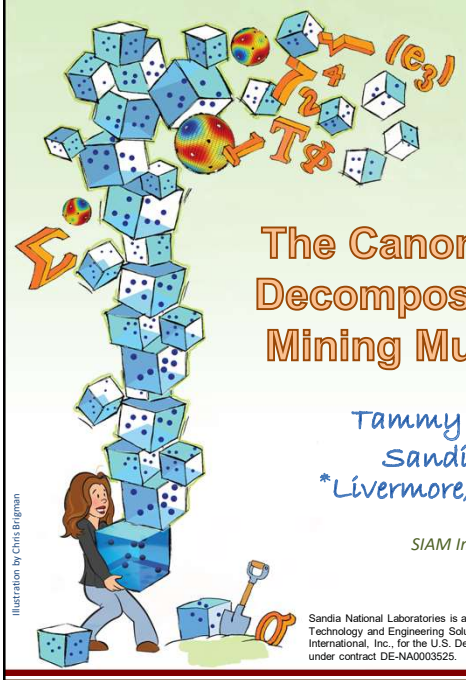


The Canonical Polyadic Tensor Decomposition and Variants for Mining Multi-Dimensional Data

Tammy Kolda*, Danny Dunlavy#
Sandia National Laboratories
*Livermore, CA and #Albuquerque, NM

SIAM International Conference on Data Mining
May 4, 2018

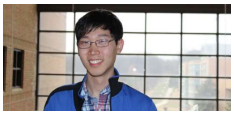
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Special Thanks to…

Kina Winoto (Sandia-CA) – For massive help in preparing the tutorial and helping in its presentation in other venues. All the labs that work are thanks to Kina!

Jed Duersch (Sandia-CA) – For preparing the examples for GCP, for creating additional versions of the gas data set, for preparing the social network data set.

David Hong (Univ. Michigan) – For work in preparing some of the labs for the first version of this tutorial in summer 2017.



Plus many other co-authors to be cited throughout the talk.

Sandia National Laboratories

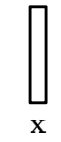
Tensors and Their Decompositions

Kolda & Bader 2009

Sandia National Laboratories

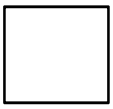
A Tensor is an d-Way Array

Vector
d = 1



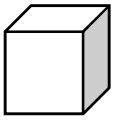
x

Matrix
d = 2



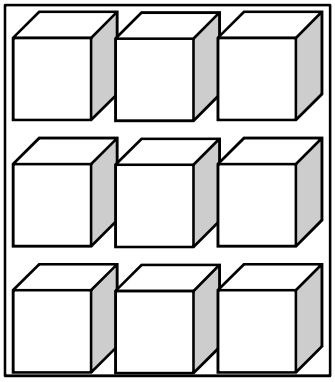
X

3rd-Order Tensor
d = 3



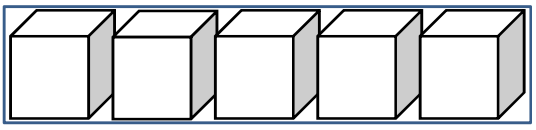
$\mathcal{X}$

5th-Order Tensor
d = 5



$\mathcal{X}$

4th-Order Tensor
d = 4



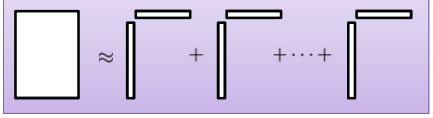
$\mathcal{X}$

From Matrices to Tensors: Two Points of View

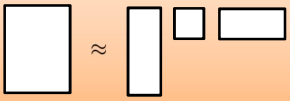
Sandia National Laboratories

Singular value decomposition (SVD), eigendecomposition (EVD), nonnegative matrix factorization (NMF), sparse SVD, CUR, etc.

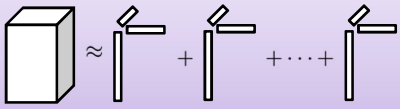
Viewpoint 1: Sum of outer products, useful for interpretation



Viewpoint 2: High-variance subspaces, useful for compression

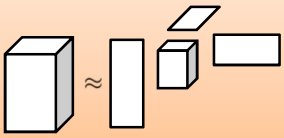


CP Model: Sum of d-way outer products, useful for interpretation



CANDECOMP, PARAFAC, Canonical Polyadic, CP

Tucker Model: Project onto high-variance subspaces to reduce dimensionality



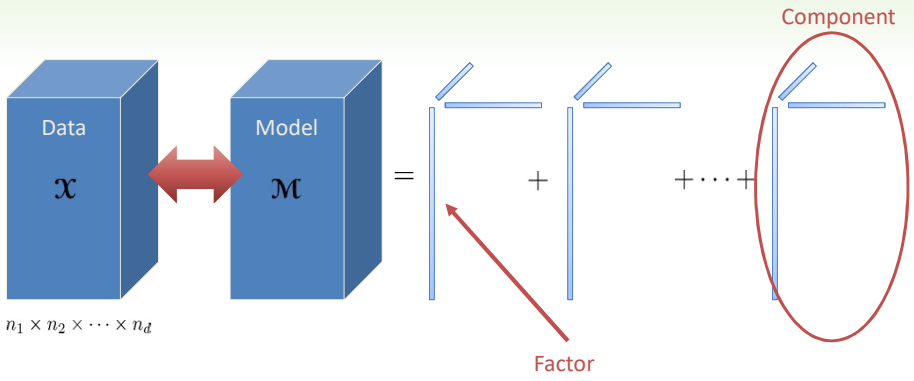
HO-SVD, Best Rank-$(R_1, R_2, \ldots, R_d)$ decomposition

Other models for compression include hierarchical Tucker and tensor train.

Sandia National Laboratories

Introducing the CP Decomposition

CP Tensor Decomposition: Sum of Outer Products



Goal: $\min \sum_i (x_i - m_i)^2$ subject to $\mathcal{M}$ having "CP structure"

What does CP mean?

Frank L. Hitchcock
MIT Professor
(1875–1957)

J. Douglas Carroll
Bell Labs
(1939-2011)

Jih-Jie Chang
Bell Labs
(1927-2007)

Richard A. Harshman
Univ. Ontario
(1943-2008)

- **Polyadic form of a tensor**, Hitchcock, 1927
- **CANDECOMP** or **CAND** (Canonical Decomposition), Carroll and Chang, 1970
- **PARAFAC** (Parallel Factors), Harshman, 1970
- **CP:** CANDECOMP/PARAFAC
  - Proposed by Kiers 2000
- **CP:** Canonical Polyadic
  - Reverse-engineered circa 2010 by Comon et al.

Sandia National Laboratories

Motivation: CP for Mouse Neural Activity

Alex Williams

A. H. Williams, T. H. Kim, F. Wang, S. Vyas, S. I. Ryu, K. V. Shenoy, M. Schnitzer, T. G. Kolda, S. Ganguli. Unsupervised Discovery of Demixed, Low-dimensional Neural Dynamics across Multiple Timescales through Tensor Components Analysis. bioRxiv, 2017. https://doi.org/10.1101/211128 (accepted to *Neuron*)

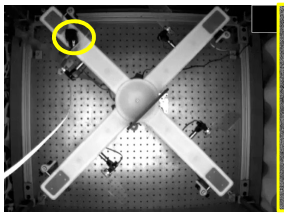
Sandia National Laboratories

Motivating Example: Neuron Activity in Learning

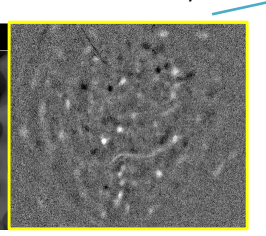
Thanks to Schnitzer Group @ Stanford
Mark Schnitzer, Fori Wang, Tony Kim

Microscope by Inscopix

mouse in "maze"

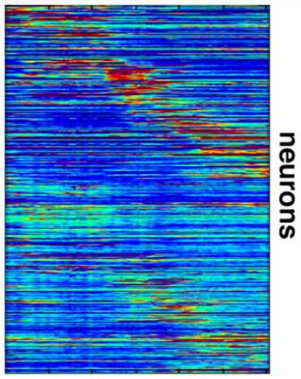


neural activity



One Column of Neuron x Time Matrix

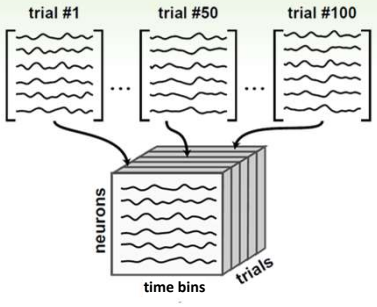
One Trial
300 neurons × 120 time bins

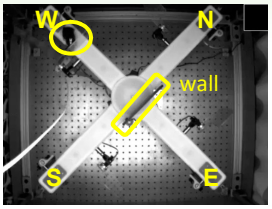


× 600 trials (over 5 days)

Williams, Ganguli, Kolda, et al. 2017

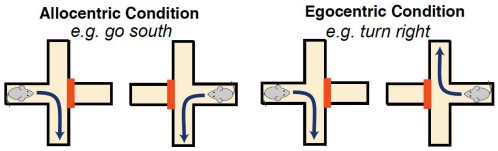
Trials Vary Start Position and Strategies





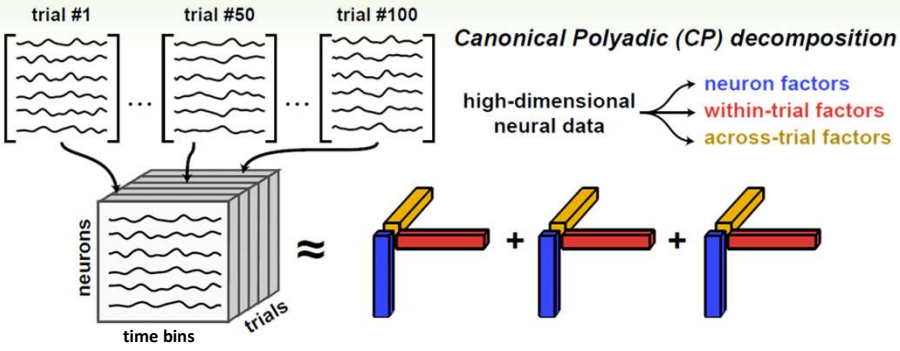
note different patterns on curtains

- 600 Trials over 5 Days
- Start West or East
- Conditions Swap Twice
  - ❖ Always Turn South
  - ❖ Always Turn Right
  - ❖ Always Turn South



Williams, Ganguli, Kolda, et al. 2017

CP for Simultaneous Analysis of Neurons, Time, and Trial



Canonical Polyadic (CP) decomposition

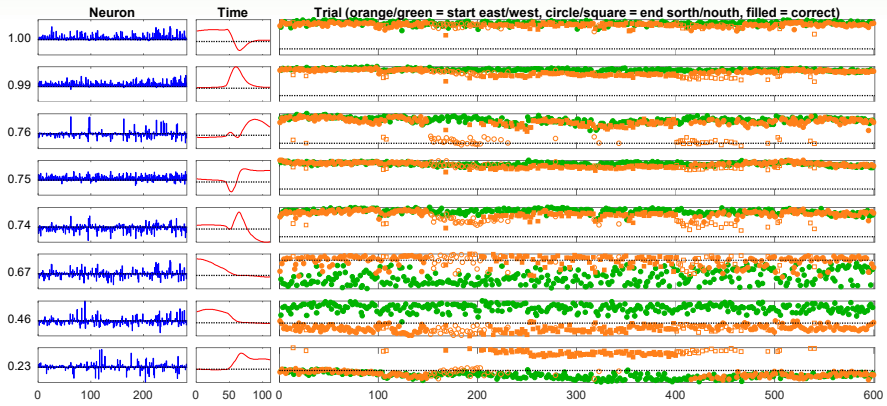
high-dimensional neural data → neuron factors, within-trial factors, across-trial factors

Past neuron-level work could only look at 2 factors at once: Time x Neuron or Trial x Neuron

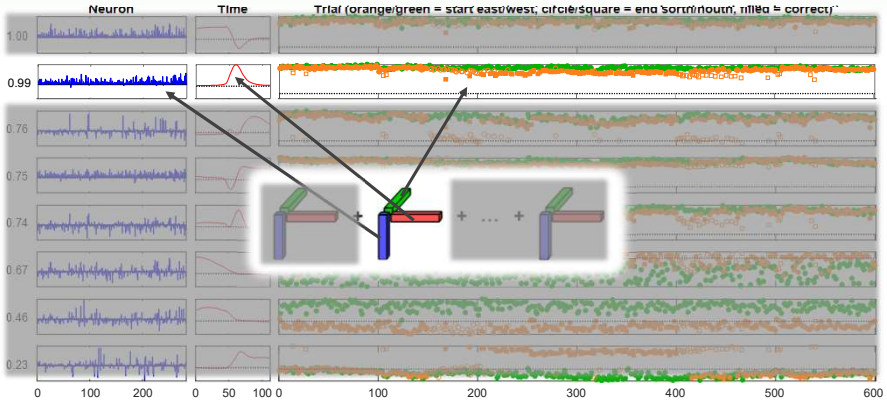
Prior tensor work in neuroscience for fMRI and EEG: Andersen and Rayens (2004), Mørup et al. (2004), Acar et al. (2007), De Vos et al. (2007), and more

Williams, Ganguli, Kolda, et al. 2017

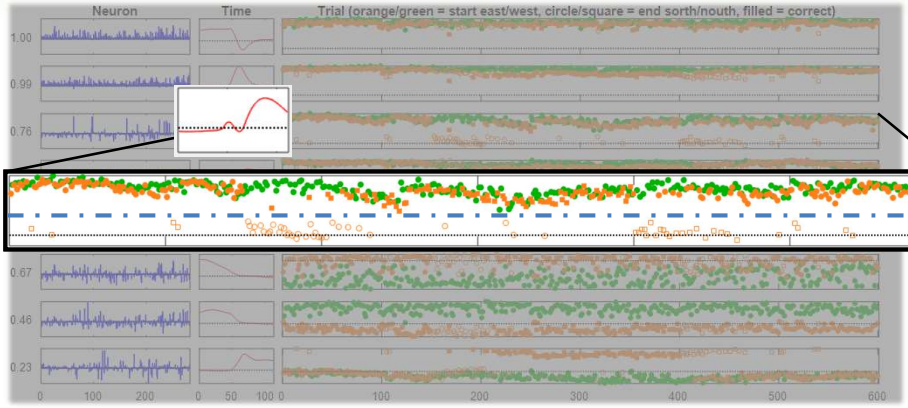
8-Component CP Decomposition of Mouse Neuron Data



8-Component CP Decomposition of Mouse Neuron Data

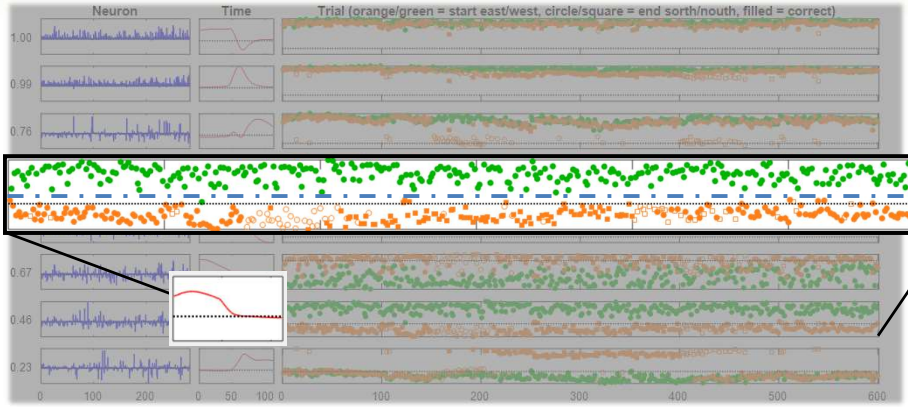


8-Component CP Decomposition of Mouse Neuron Data



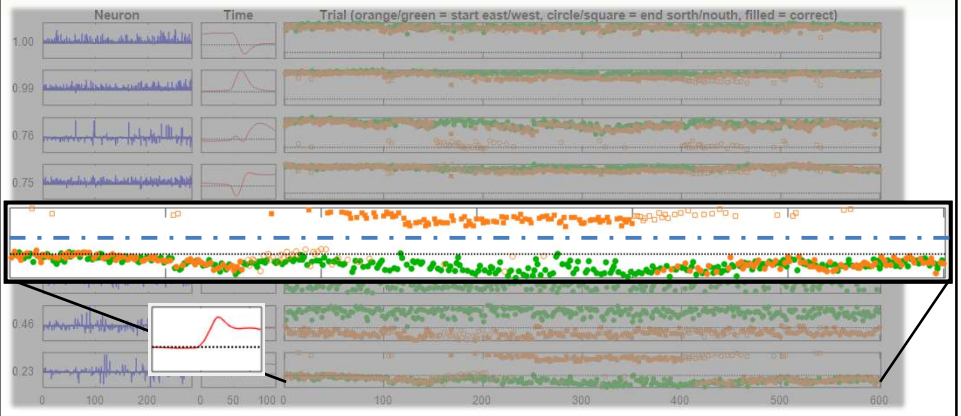
5/4/2018 Tensor Tutorial @ SDM18 15

8-Component CP Decomposition of Mouse Neuron Data



5/4/2018 Tensor Tutorial @ SDM18 16

8-Component CP Decomposition of Mouse Neuron Data



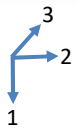
5/4/2018 Tensor Tutorial @ SDM18 17

Tensor Background

5/4/2018 Tensor Tutorial @ SDM18 24

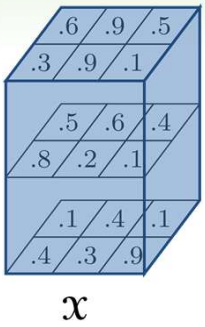
Working with Tensors

"Modes"



"Order" = Number of Modes or Ways

$d = 3$



$\mathcal{X}$

"Size"

$n_1 \times n_2 \times n_3$

$3 \times 3 \times 2$

"Frontal Slices"

$$\mathbf{X}(:,:,1) = \begin{bmatrix} 0.3 & 0.9 & 0.1 \\ 0.8 & 0.2 & 0.1 \\ 0.4 & 0.3 & 0.9 \end{bmatrix}$$

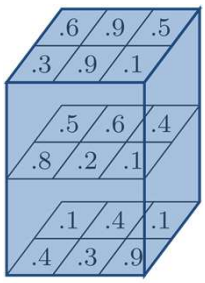
$$\mathbf{X}(:,:,2) = \begin{bmatrix} 0.6 & 0.9 & 0.5 \\ 0.5 & 0.6 & 0.4 \\ 0.1 & 0.4 & 0.1 \end{bmatrix}$$

This is how it's displayed in MATLAB.

The term "dimension" is overloaded and may be confusing.

Indexing a 3D Tensor

We use 1-based indexing throughout this talk to be consistent with MATLAB.



$\mathcal{X}$

Tensor Elements

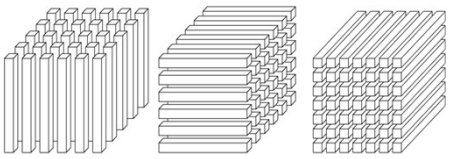
$x(1,1,1) = 0.3$

$x(2,1,1) = 0.8$

$x(1,1,2) = 0.6$

$x(3,2,1) = 0.3$

Tensor Fibers



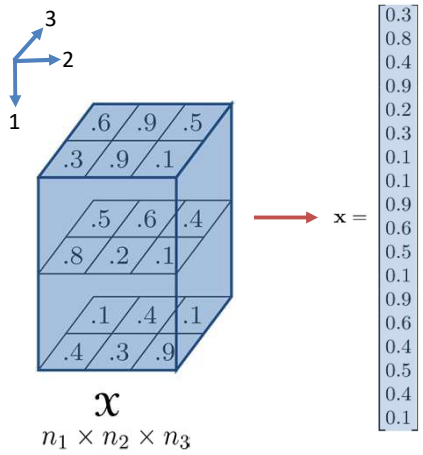
Mode-1 Fibers Mode-2 Fibers Mode-3 Fibers

$$\mathbf{x}(:,1,1) = \begin{bmatrix} .3 \\ .8 \\ .4 \end{bmatrix} \qquad \mathbf{x}(1,1,:) = \begin{bmatrix} .3 \\ .6 \end{bmatrix}$$

$$\mathbf{x}(1,:,1) = \begin{bmatrix} .3 \\ .9 \\ .1 \end{bmatrix}$$

Vectorizing a 3D Tensor (or How It's Stored in Memory!)

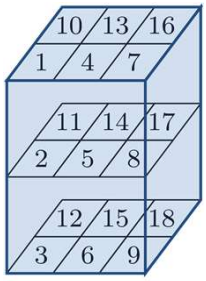
$$(i_1, i_2, i_3) \to i' = (i_3 - 1)n_2 n_1 + (i_2 - 1)n_1 + i_1$$



$\mathcal{X}$

$n_1 \times n_2 \times n_3$

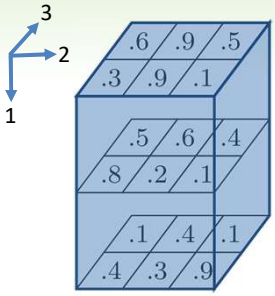
Ordering



$\mathcal{X}$

This is MATLAB's default order.
It may not be everyone's.

Unfolding a 3D Tensor



$\mathcal{X}$

$n_1 \times n_2 \times n_3$

$$(i_1, i_2, i_3) \to (i_i, i_1'), \quad i_1' = (i_3 - 1)n_2 + i_2$$

$$\mathbf{X}_{(1)} = \begin{bmatrix} 0.3 & 0.9 & 0.1 & 0.6 & 0.9 & 0.5 \\ 0.8 & 0.2 & 0.1 & 0.5 & 0.6 & 0.4 \\ 0.4 & 0.3 & 0.9 & 0.1 & 0.4 & 0.1 \end{bmatrix} \quad n_1 \times n_2 n_3$$

$$(i_1, i_2, i_3) \to (i_2, i_2'), \quad i_2' = (i_3 - 1)n_1 + i_1$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 0.3 & 0.8 & 0.4 & 0.6 & 0.5 & 0.1 \\ 0.9 & 0.2 & 0.3 & 0.9 & 0.6 & 0.4 \\ 0.1 & 0.1 & 0.9 & 0.5 & 0.4 & 0.1 \end{bmatrix} \quad n_2 \times n_1 n_3$$

$$(i_1, i_2, i_3) \to (i_3, i_3'), \quad i_3' = (i_2 - 1)n_1 + i_1$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 0.3 & 0.8 & 0.4 & 0.9 & 0.2 & 0.3 & 0.1 & 0.1 & 0.9 \\ 0.6 & 0.5 & 0.1 & 0.9 & 0.6 & 0.4 & 0.5 & 0.4 & 0.1 \end{bmatrix}$$

$n_3 \times n_1 n_2$

Unfolding also known as "matricization".

Working with d-way Tensors

Size: $n_1 \times n_2 \times \cdots \times n_d$

Single Element: $x(i_1, i_2, \ldots, i_d)$ or x_i "multiindex"

Set of All Indices: $\mathcal{I} = \{\, i \equiv (i_1, i_2, \ldots, i_d) \mid i_k \in \{1, \ldots, n_k\} \text{ for } k = 1, \ldots, d \,\}$

Geometric mean of the sizes: $n = \sqrt[d]{\prod_{k=1}^{d} n_k}$

Arithmetic mean of the sizes: $\bar{n} = \frac{1}{d} \sum_{k=1}^{d} n_k$

Total number of elements in the tensor:

$$|\mathcal{I}| = \prod_{k=1}^{d} n_k = n^d$$

5/4/2018 Tensor Tutorial @ SDM18 29

Vectorizing a d-way Tensor

Order: d

Size: $n_1 \times n_2 \times \cdots \times n_d$

"subscripts" "linear index"

$$(i_1, i_2, \ldots, i_d) \in \mathcal{I} \rightarrow i' \in \{\, 1, \ldots, n^d \,\}$$

$$i' = 1 + \sum_{k=1}^{d} (i_k - 1) n'_k$$

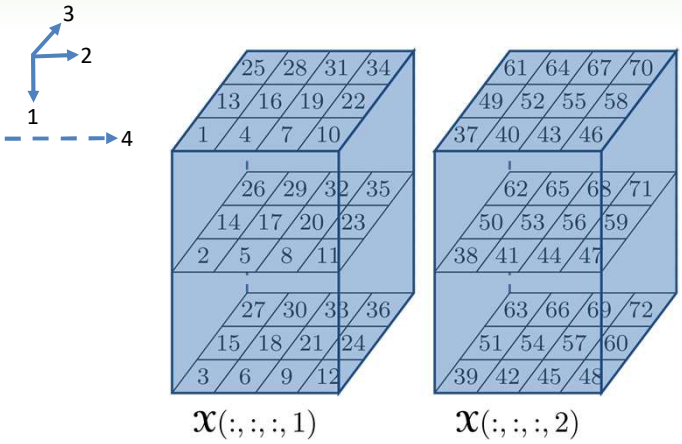
$$n'_k = \prod_{\ell=1}^{k-1} n_k$$

See MATLAB commands: tt_sub2ind, tt_ind2sub

5/4/2018 Tensor Tutorial @ SDM18 30

Example: Ordering of Elements in a 4D Tensor

4th-order tensor of size $3 \times 4 \times 3 \times 2$



$\mathcal{X}(:,:,:,1)$ $\mathcal{X}(:,:,:,2)$

Unfolding a d-way Tensor

Matricization or Unfolding: Rearrange d-way array into 2-way array.

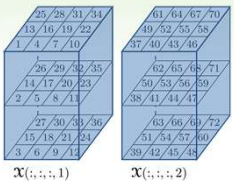
Size: $n_1 \times n_2 \times \cdots \times n_d$ Size: $n_k \times (n^d/n_k)$

Mode-k unfolding: $(i_1, i_2, \ldots, i_d) \rightarrow (i_k, i'_k)$

$$i'_k = 1 + \sum_{\ell=1}^{k-1} (i_\ell - 1) n'_\ell + \sum_{\ell=k+1}^{d} (i_\ell - 1)(n'_\ell / n_k) \qquad n'_k = \prod_{\ell=1}^{k-1} n_k$$

$$i'_k \in \{1, \ldots, n^d/n_k\}$$

Unfolding a 4D Tensor



Note the patterning!

block size = $n_k \times n'_k$

blocks = $n^d / n_k n'_k$

$$n'_k = \prod_{\ell=1}^{k-1} n_k$$

$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & \cdots & 58 & 61 & 64 & 67 & 70 \\ 2 & 5 & 8 & 11 & 14 & 17 & \cdots & 59 & 62 & 65 & 68 & 71 \\ 3 & 6 & 9 & 12 & 15 & 18 & \cdots & 60 & 63 & 66 & 69 & 72 \end{bmatrix}$$

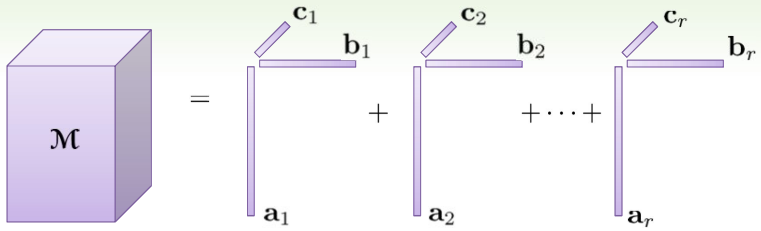
$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 & 25 & 26 & 27 & 37 & 38 & 39 & 49 & 50 & 51 & 61 & 62 & 63 \\ 4 & 5 & 6 & 16 & 17 & 18 & 28 & 29 & 30 & 40 & 41 & 42 & 52 & 53 & 54 & 64 & 65 & 66 \\ 7 & 8 & 9 & 19 & 20 & 21 & 31 & 32 & 33 & 43 & 44 & 45 & 55 & 56 & 57 & 67 & 68 & 69 \\ 10 & 11 & 12 & 22 & 23 & 24 & 34 & 35 & 36 & 46 & 47 & 48 & 58 & 59 & 60 & 70 & 71 & 72 \end{bmatrix}$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 37 & \cdots & 48 \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 49 & \cdots & 60 \\ 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 61 & \cdots & 72 \end{bmatrix}$$

$$\mathbf{X}_{(4)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & \cdots & 31 & 32 & 33 & 34 & 35 & 36 \\ 37 & 38 & 39 & 40 & 41 & 42 & \cdots & 67 & 68 & 69 & 70 & 71 & 72 \end{bmatrix}$$

Patterning: Battaglino, Perros, Sub, & Vuduc 2015; Austin, Ballard, & Kolda 2016

Ktensor: Sum of Outer Products



$$\mathcal{M} = \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \cdots + \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

$$m(i_1, i_2, i_3) = \sum_{j=1}^{r} a(i_1, j)\, b(i_2, j)\, c(i_3, j)$$

Bader & Kolda 2007

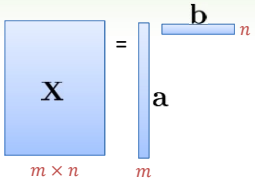
Vector Outer Product

Outer product of 2 vectors
⇒ rank-1 matrix

$$\mathbf{X} = \mathbf{a}\mathbf{b}^{\mathsf{T}}$$

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

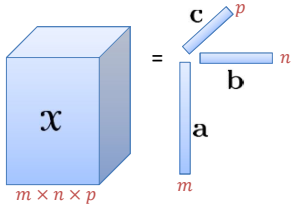
$$x(i_1, i_2) = a(i_1)\, b(i_2)$$



Outer product of 3 vectors
⇒ rank-1 tensor of order 3

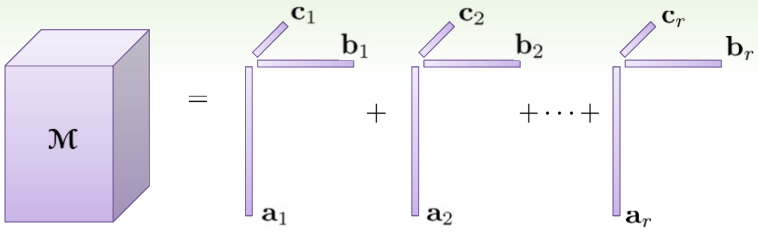
$$\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$$

$$x(i_1, i_2, i_3) = a(i_1)\, b(i_2)\, c(i_3)$$



5/4/2018 Tensor Tutorial @ SDM18 35

Ktensor: Sum of Outer Products



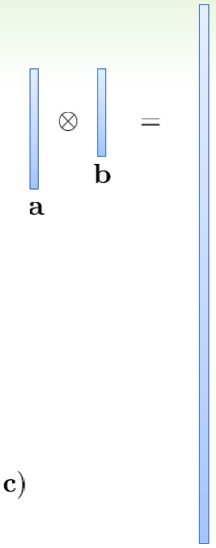
$$\mathcal{M} = \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \cdots + \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

Bader & Kolda 2007

5/4/2018 Tensor Tutorial @ SDM18 36

Vector Kronecker Product

$$\underset{m\ \ n}{\mathbf{a}\otimes\mathbf{b}} = \begin{bmatrix} a(1)\,\mathbf{b} \\ a(2)\,\mathbf{b} \\ \vdots \\ a(m)\,\mathbf{b} \end{bmatrix}_{mn} = \begin{bmatrix} a(1)\,b(1) \\ a(1)\,b(2) \\ \vdots \\ a(1)\,b(n) \\ \vdots \\ a(m)\,b(1) \\ a(m)\,b(2) \\ \vdots \\ a(m)\,b(n) \end{bmatrix}$$



Associative: $(\mathbf{a}\otimes\mathbf{b})\otimes\mathbf{c} = \mathbf{a}\otimes(\mathbf{b}\otimes\mathbf{c})$

Connection between Kronecker Product and Outer Products

$$\mathbf{X} = \mathbf{a}\circ\mathbf{b} \iff \text{vec}(\mathbf{X}) = \mathbf{b}\otimes\mathbf{a}$$

$$\mathbf{X} = \mathbf{a}\circ\mathbf{b}\circ\mathbf{c} \iff \begin{aligned} \text{vec}(\mathbf{X}) &= \mathbf{c}\otimes\mathbf{b}\otimes\mathbf{a} \\ \mathbf{X}_{(1)} &= \mathbf{a}(\mathbf{c}\otimes\mathbf{b})' \\ \mathbf{X}_{(2)} &= \mathbf{b}(\mathbf{c}\otimes\mathbf{a})' \\ \mathbf{X}_{(3)} &= \mathbf{c}(\mathbf{b}\otimes\mathbf{a})' \end{aligned}$$

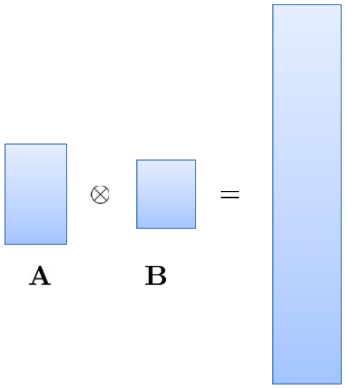
$$\mathcal{X} = \mathbf{a}_1\circ\mathbf{a}_2\circ\cdots\circ\mathbf{a}_d \iff \begin{aligned} \text{vec}(\mathbf{X}) &= \mathbf{a}_d\otimes\cdots\otimes\mathbf{a}_1 \\ \mathbf{X}_{(k)} &= \mathbf{a}_k(\mathbf{a}_d\otimes\cdots\otimes\mathbf{a}_{k+1}\otimes\mathbf{a}_{k-1}\otimes\cdots\otimes\mathbf{a}_1)' \end{aligned}$$

Khatri-Rao Product

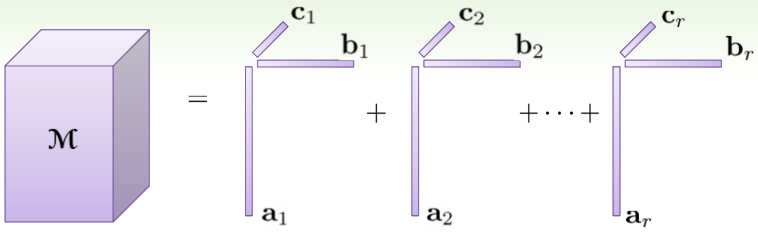
Input two matrices with same number of columns

Columnwise-Kronecker Product

$$\underset{m\times r}{\mathbf{A}} \odot \underset{n\times r}{\mathbf{B}} = \underset{mn\times r}{\left[\mathbf{a}(:,1)\otimes\mathbf{b}(:,1) \quad \mathbf{a}(:,2)\otimes\mathbf{b}(:,2) \quad \cdots \quad \mathbf{a}(:,r)\otimes\mathbf{b}(:,r)\right]}$$



Ktensors and Khatri-Rao Products



$$\mathcal{M} = \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \cdots + \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

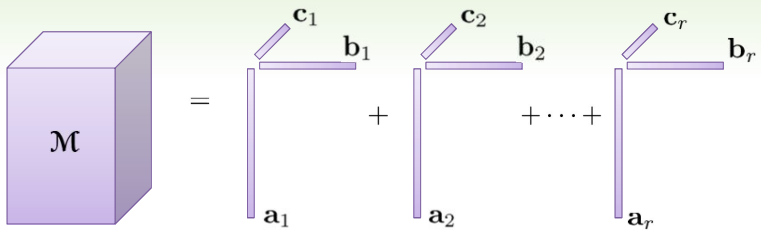
$$\mathbf{M}_{(1)} = \mathbf{a}_1(\mathbf{c}_1 \otimes \mathbf{b}_1)' + \mathbf{a}_2(\mathbf{c}_2 \otimes \mathbf{b}_2)' + \cdots + \mathbf{a}_r(\mathbf{c}_r \otimes \mathbf{b}_r)'$$

$$\mathbf{M}_{(1)} = \underbrace{\left[\mathbf{a}_1 \quad \cdots \quad \mathbf{a}_r\right]}_{\mathbf{A}} \underbrace{\left[\mathbf{c}_1 \otimes \mathbf{b}_1 \quad \cdots \quad \mathbf{c}_r \otimes \mathbf{b}_r\right]'}_{(\mathbf{C}\odot\mathbf{B})'}$$

$$\mathbf{M}_{(1)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})'$$

Bader & Kolda 2007

Ktensor Unfoldings



$$\mathbf{M}_{(1)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})'$$
$$\mathbf{M}_{(2)} = \mathbf{B}(\mathbf{C} \odot \mathbf{A})'$$
$$\mathbf{M}_{(3)} = \mathbf{C}(\mathbf{B} \odot \mathbf{A})'$$

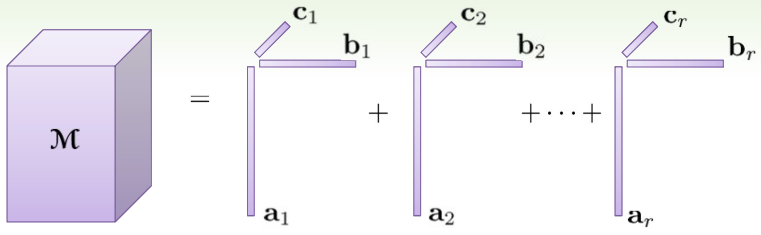
Factor Matrices

$\mathbf{A} = [\mathbf{a}_1\, \mathbf{a}_2 \cdots \mathbf{a}_r]$
$\mathbf{B} = [\mathbf{b}_1\, \mathbf{b}_2 \cdots \mathbf{b}_r]$
$\mathbf{C} = [\mathbf{c}_1\, \mathbf{c}_2 \cdots \mathbf{c}_r]$

d-way case $\mathbf{M}_{(k)} = \mathbf{A}_k \left(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1\right)'$

Bader & Kolda 2007

Ktensor: Shorthand

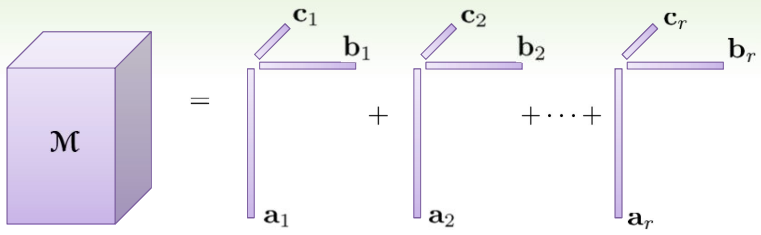


$$\mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

d-way case $\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$

Bader & Kolda 2007

Scaling Ambiguity



$$\mathcal{M} = \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \cdots + \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

$$\mathcal{M} = \mathbf{a}_1 \circ 2\,\mathbf{b}_1 \circ {}^1\!/_2\,\mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \cdots + \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

$$\mathcal{M} = \lambda_1\,\mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \lambda_2\,\mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \cdots + \lambda_r\,\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

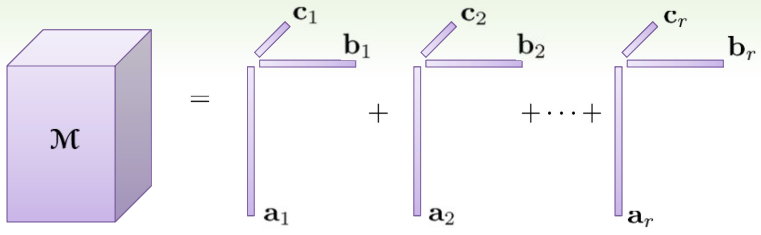
Require all factors (i.e., columns of the factor matrices) to have unit norm, e.g., $\|\mathbf{a}_1\| = 1$

Fixed by ktensor/normalize function.

$$\mathcal{M} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

Bader & Kolda 2007

Permutation Ambiguity in Ktensor



$$\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!] = [\![\mathbf{A}_1\boldsymbol{\Pi}, \mathbf{A}_2\boldsymbol{\Pi}, \ldots, \mathbf{A}_d\boldsymbol{\Pi}]\!]$$

where $\boldsymbol{\Pi}$ is a $p \times p$ permutation matrix.

Tensor Inner Product & Norm

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} x(i_1, i_2, \ldots, i_d) y(i_1, y_2, \ldots, y_d)$$

Shorthand: $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i \in \mathcal{I}} x_i y_i$

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \text{vec}(\mathcal{X})^\intercal \text{vec}(\mathcal{Y})$$

$$\|\mathcal{X}\|^2 = \langle \mathcal{X}, \mathcal{X} \rangle = \sum_{i \in \mathcal{I}} x_i^2$$

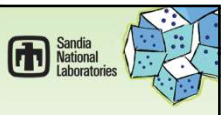
Matrix Hadamard Product

Input two same-sized matrices

$$\mathbf{A} * \mathbf{B} = \begin{bmatrix} a(1,1)b(1,1) & a(1,2)b(1,2) & \cdots & a(1,n)b(1,n) \\ a(2,1)b(2,1) & a(2,2)b(2,2) & \cdots & a(2,n)b(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ a(m,1)b(m,1) & a(m,2)b(m,2) & \cdots & a(m,n)b(m,n) \end{bmatrix}$$

In MATLAB: A .* B

Properties of Khatri-Rao Product



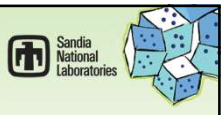
$$(\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C} = \mathbf{A} \odot (\mathbf{B} \odot \mathbf{C})$$

$$\underset{m \times r}{(\mathbf{A} \odot \mathbf{B})^\intercal} \underset{n \times r}{(\mathbf{A} \odot \mathbf{B})} = \underset{r \times r}{(\mathbf{A}^\intercal \mathbf{A}} * \underset{r \times r}{\mathbf{B}^\intercal \mathbf{B})}$$

$$\underset{mn \times r}{(\mathbf{A} \odot \mathbf{B})^\dagger} = (\mathbf{A}^\intercal \mathbf{A} * \underset{r \times r}{\mathbf{B}^\intercal \mathbf{B}})^\dagger \underset{mn \times r}{(\mathbf{A} \odot \mathbf{B})^\intercal}$$

Moore-Penrose Pseudo-Inverse

MTTKRP: Matricized-Tensor Times Khatri-Rao Product



Tensor of size $n_1 \times n_2 \times \cdots \times n_d$: $\mathcal{X}$

Matrices of size $n_k \times r$ for $k = 1, \ldots, d$: $\mathbf{A}_1, \ldots, \mathbf{A}_d$

Matricized Tensor

Khatri-Rao Product

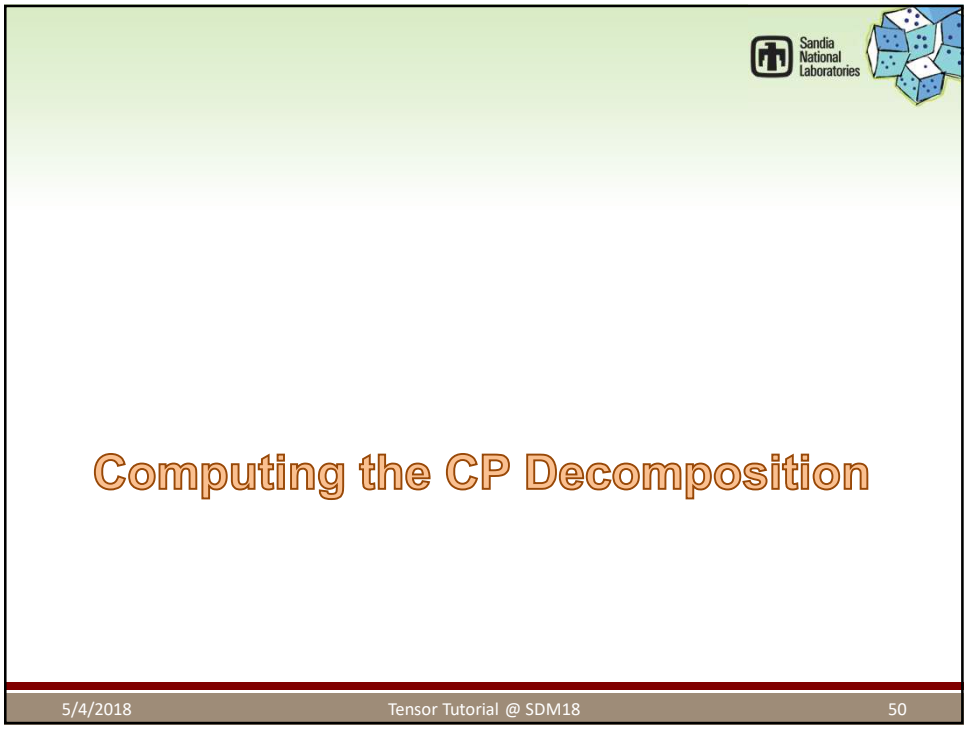
$n_k \times r$

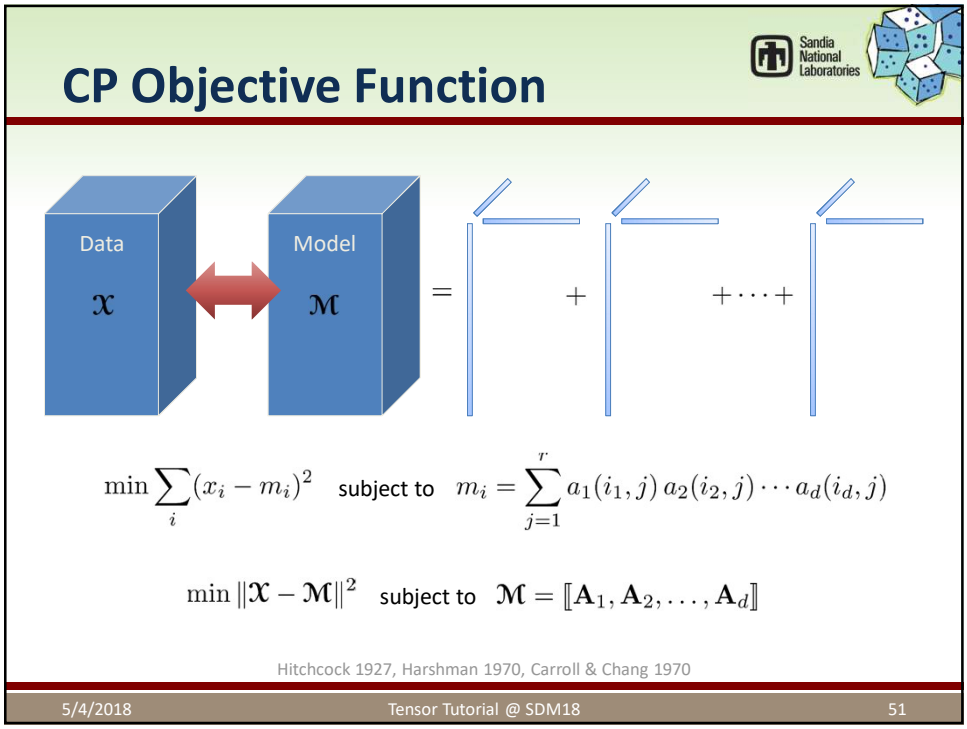
$$\mathbf{B} = \mathbf{X}_{(k)} (\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$$

$n_k \times (n^d/n_k)$ $\quad$ $(n^d/n_k) \times r$

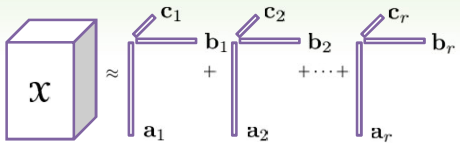
Same size as $\mathbf{A}_k$

Bader & Kolda 2007


Sandia National Laboratories
Computing the CP Decomposition
5/4/2018
Tensor Tutorial @ SDM18
50


CP Objective Function
Sandia National Laboratories
Data
$\mathcal{X}$
Model
$\mathcal{M}$
=
+
+ ⋯ +
$\min \sum_i (x_i - m_i)^2$ subject to $m_i = \sum_{j=1}^{r} a_1(i_1, j)\, a_2(i_2, j) \cdots a_d(i_d, j)$
$\min \|\mathcal{X} - \mathcal{M}\|^2$ subject to $\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$
Hitchcock 1927, Harshman 1970, Carroll & Chang 1970
5/4/2018
Tensor Tutorial @ SDM18
51

CP Optimization Problem



Number of observations = n^d

Number of free parameters = $dr\bar{n}$

Typically, $dr\bar{n} \ll n^d$

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathcal{X} - [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|^2$$

Nonconvex

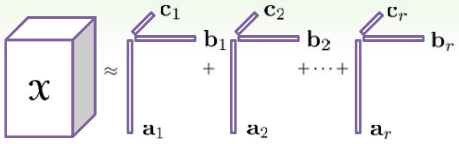
Permutation & Scaling Ambiguities

How to pick r?

General Form

$$\min_{\{\mathbf{A}_k\}} \|\mathcal{X} - [\![\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_d]\!]\|^2$$

CP-ALS: Fitting CP Model via Alternating Least Squares



Convex (linear least squares) subproblems can be solved exactly

$$f(\mathbf{A},\mathbf{B},\mathbf{C}) = \|\mathcal{X} - [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|^2$$

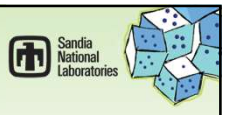
Repeat until convergence…

$$\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^\intercal\|^2$$

$$\min_{\mathbf{B}} \|\mathbf{X}_{(2)} - \mathbf{B}(\mathbf{C}\odot\mathbf{A})^\intercal\|^2$$

$$\min_{\mathbf{C}} \|\mathbf{X}_{(3)} - \mathbf{C}(\mathbf{B}\odot\mathbf{A})^\intercal\|^2$$

Special Structure of the Least Squares Problem



$$\min_{\mathbf{A}} \|\underbrace{\mathbf{X}_{(1)}}_{n_1 \times (n_2 n_3)} - \underbrace{\mathbf{A}}_{n_1 \times r}\underbrace{(\mathbf{C} \odot \mathbf{B})^\intercal}_{r \times (n_2 n_3)}\|^2$$

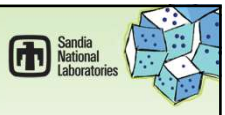
$$(\mathbf{C} \odot \mathbf{B})\mathbf{A}^\intercal = \mathbf{X}_{(1)}^\intercal$$

$$\mathbf{A}^\intercal = (\mathbf{C} \odot \mathbf{B})^\dagger \mathbf{X}_{(1)}^\intercal$$

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

$$\mathbf{A} = \underbrace{\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})}_{\text{MTTKRP}}(\mathbf{C}'\mathbf{C} * \mathbf{B}'\mathbf{B})^\dagger$$

Special Structure of the Least Squares Problem: d-way



$$\min_{\mathbf{A}_k} \|\mathbf{X}_{(k)} - \mathbf{A}_k(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)^\intercal\|^2$$

$$\mathbf{A}_k \leftarrow \mathbf{X}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)\mathbf{V}^\dagger$$

$$\mathbf{V} \leftarrow (\mathbf{A}_1^\intercal\mathbf{A}_1) * \cdots * (\mathbf{A}_{k-1}^\intercal\mathbf{A}_{k-1}) * (\mathbf{A}_{k+1}^\intercal\mathbf{A}_{k+1}) * \cdots * (\mathbf{A}_d^\intercal\mathbf{A}_d)$$

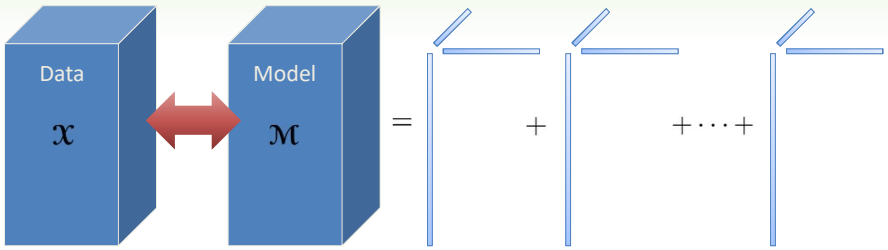
CP-ALS Algorithm + Nuances

1: $\text{fit} \leftarrow 1 - \left(\|\mathcal{X} - \mathcal{M}\|/\|\mathcal{X}\|\right)$ ← *Roughly, the proportion of the data explained by the model.*
2: **for** $\ell = 1, 2, \ldots$ **do**
3: $\quad \text{oldfit} \leftarrow \text{fit}$
4: $\quad$ **for** $k = 1, 2, \ldots, d$ **do**
5: $\quad\quad \mathbf{Z} \leftarrow \mathbf{X}_{(k)}\left(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1\right)$ {MTTKRP}
6: $\quad\quad \mathbf{V} \leftarrow (\mathbf{A}_1^\intercal \mathbf{A}_1) * \cdots * (\mathbf{A}_{k-1}^\intercal \mathbf{A}_{k-1}) * (\mathbf{A}_{k+1}^\intercal \mathbf{A}_{k+1}) * \cdots * (\mathbf{A}_d^\intercal \mathbf{A}_d)$
7: $\quad\quad \mathbf{A}_k \leftarrow \mathbf{Z}/\mathbf{V}$ {transposed backsolve}
8: $\quad\quad \boldsymbol{\lambda} \leftarrow$ column norms of $\mathbf{A}_k$
9: $\quad\quad \mathbf{A}_k \leftarrow \mathbf{A}_k/\text{diag}(\boldsymbol{\lambda})$ *Important to normalize!*
10: $\quad$ **end for**
11: $\quad \text{fit} \leftarrow 1 - \left(\|\mathcal{X} - \mathcal{M}\|/\|\mathcal{X}\|\right)$
12: $\quad$ **if** $|\text{oldfit} - \text{fit}| < \tau$ **then** *Stop when the fit stagnates.*
13: $\quad\quad$ **break**
14: $\quad$ **end if**
15: **end for**

See MATLAB command: cp_als

Harshman 1970, Carroll & Chang 1970

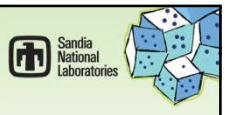
CP Objective Function



$$\min \tfrac{1}{2}\|\mathcal{X} - \mathcal{M}\|^2 \text{ subject to } \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

Hitchcock 1927, Harshman 1970, Carroll & Chang 1970

CP-OPT: Fitting CP Model via Direct Optimization



$$f = \tfrac{1}{2}\|\mathcal{X} - \mathcal{M}\|^2 = \tfrac{1}{2}\|\mathbf{X}_{(k)} - \mathbf{M}_{(k)}\|_F^2 \qquad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

$$\frac{\partial f}{\partial \mathbf{A}_k} = -\big(\mathbf{X}_{(k)} - \mathbf{M}_{(k)}\big)\frac{\partial \mathbf{M}_{(k)}}{\partial \mathbf{A}_k}$$

Note connect to ALS!

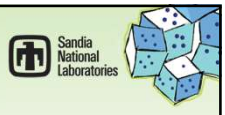
Recall: $\mathbf{M}_{(k)} = \mathbf{A}_k\big(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1\big)^\intercal$

$$\begin{aligned}\frac{\partial f}{\partial \mathbf{A}_k} &= -\big(\mathbf{X}_{(k)} - \mathbf{M}_{(k)}\big)\big(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1\big) \\ &= \underbrace{-\mathbf{X}_{(k)}\big(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1\big)}_{\text{MTTKRP}} + \mathbf{A}_k\mathbf{V}_k\end{aligned}$$

where $\mathbf{V}_k = (\mathbf{A}_1^\intercal\mathbf{A}_1) * \cdots * (\mathbf{A}_{k-1}^\intercal\mathbf{A}_{k-1}) * (\mathbf{A}_{k+1}^\intercal\mathbf{A}_{k+1}) * \cdots * (\mathbf{A}_d^\intercal\mathbf{A}_d)$

Acar, Dunlavy, Kolda 2011

CP-OPT



Calculating the function and gradient

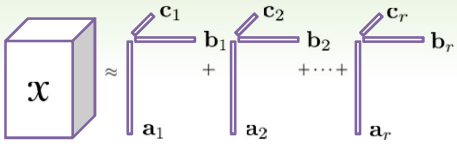
1: **for** $k = 1, 2, \ldots, d$ **do**
2: $\quad \mathbf{Z}_k \leftarrow \mathbf{X}_{(k)}\big(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1\big)$ {MTTKRP}
3: $\quad \mathbf{V}_k \leftarrow (\mathbf{A}_1^\intercal\mathbf{A}_1) * \cdots * (\mathbf{A}_{k-1}^\intercal\mathbf{A}_{k-1}) * (\mathbf{A}_{k+1}^\intercal\mathbf{A}_{k+1}) * \cdots * (\mathbf{A}_d^\intercal\mathbf{A}_d)$
4: $\quad \mathbf{\Delta}_k \leftarrow -\mathbf{Z}_k + \mathbf{A}_k\mathbf{V}_k$
5: **end for**
6: $f \leftarrow \|\mathcal{X}\|^2 + \mathbf{1}^\intercal\big[\mathbf{A}_d\mathbf{Z}_d\big]\mathbf{1} + \mathbf{1}^\intercal\big[\mathbf{A}_d^\intercal\mathbf{A}_d * \mathbf{V}_d\big]\mathbf{1}$

- Gradients not naturally in vector form.
- Need to convert before calling optimization method.

Scaling ambiguities can cause problems – may want to regularize to ameliorate

Acar, Dunlavy, Kolda 2011

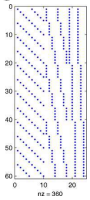
CP-OPT: Fitting CP via "All-at-once" Optimization



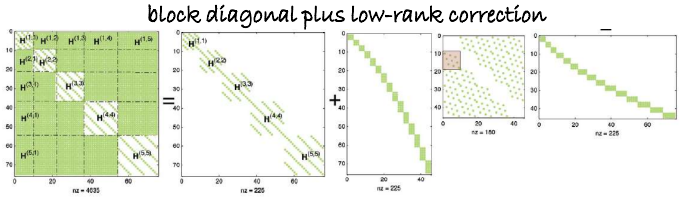
$$f(\mathbf{A},\mathbf{B},\mathbf{C}) = \|\mathcal{X} - [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|^2$$

- CP-OPT (Acar et al.): 1st-order method, better accuracy than ALS when r is "too big"
- CP-NLS (Paatero, Tomasi & Bro): Damped Gauss-Newton, accurate but slow
- CP-Newton (Phan et al.): Newton method, superior to CP-OPT for high order

Structured Jacobian



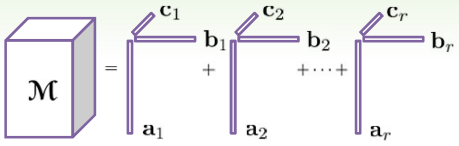
Structured Hessian can be written as block diagonal plus low-rank correction



Paatero 1997; Tomasi & Bro 2005, 2006; Acar, Dunlavy, & Kolda 2011; Phan, Tichavský, & Cichocki 2013

More on the CP Decomposition

Uniqueness ⇒ Interpretability



k-rank($\mathbf{A}$) = maximum value k such that *any* k columns of $\mathbf{A}$ are linearly independent

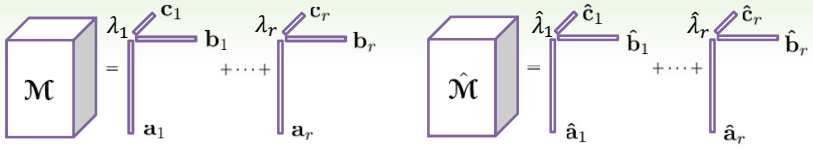
Model is **essentially unique** (i.e., up to permutation and scaling) under the condition that the sum of the factor matrix k-rank values is $\geq 2r + d - 1$

$$\text{k-rank}(\mathbf{A}) + \text{k-rank}(\mathbf{B}) + \text{k-rank}(\mathbf{C}) \geq 2r + 2$$

Matrix factorization does not share this property!

Kruskal 1977, Sidiropoulous & Bro 2000

Comparing two CP tensors



- Case 1: Single component
  - Normalize factors, absorbing weights into λ
  - Compute: $\text{score} = \rho(\mathbf{a}^\intercal\hat{\mathbf{a}})(\mathbf{b}^\intercal\hat{\mathbf{b}})(\mathbf{c}^\intercal\hat{\mathbf{c}}), \quad \rho = 1 - |\lambda - \hat{\lambda}|/\max\{\lambda, \hat{\lambda}\}$

    Cosine of angle between vectors
  - score $= 1$ ⇒ perfect match
- Case 2: Multiple components
  - Compute: score = *average* of single component scores
  - But need to match the components to maximize the score!
    - Exact solution too expensive, so use greedy heuristic

CP Rank

The (CP) **rank** of a tensor is the minimal value r such that the tensor can be expressed as the sum of exactly r rank-1 tensors and no fewer.

- The rank can be larger than the maximum dimension!

Tensor of size 2 x 2 x 2:

$$\mathbf{X}(:,:,1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{X}(:,:,2) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Rank is 3:

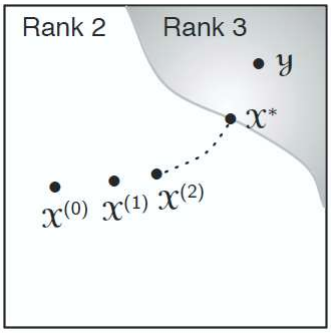
$$\mathcal{X} = [\![\underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}}_{\mathbf{A}}, \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}}_{\mathbf{B}}, \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}}_{\mathbf{C}}]\!]$$

Kruskal 1977, Krusal 1983, Kruskal 1989, Bini et al. 1979; see also Kolda & Bader 2009

CP Rank is NP-Hard

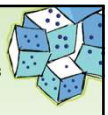
The (CP) **rank** of a tensor is the minimal value r such that the tensor can be expressed as the sum of exactly r rank-1 tensors and no fewer.

- Factorizations are **not nested** (Kolda 2001)
- The space of rank-r tensors is **not closed** except for $r = 1$ (Kruskal, Harshman, & Lundy 1989)
- The best rank-k approximation may **not exist** (de Silva & Lim 2006)
- Determining the rank of a tensor is **NP hard** (Håstad 1990, Hillar & Lim 2009)



Kolda & Bader 2009

Tensor of Unknown Rank



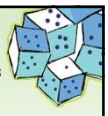
- Specific 9 x 9 x 9 tensor factorization problem
- Corresponds to being able to do fast matrix multiplication of two 3x3 matrices
- Rank is between 19 and 23 ⟶ at most 621 variables

$$
\begin{aligned}
x_{1,1,1} &= 1 & x_{4,2,1} &= 1 & x_{7,3,1} &= 1 \\
x_{1,4,2} &= 1 & x_{4,5,2} &= 1 & x_{7,6,2} &= 1 \\
x_{1,7,3} &= 1 & x_{4,8,3} &= 1 & x_{7,9,3} &= 1 \\
x_{2,1,4} &= 1 & x_{5,2,4} &= 1 & x_{8,3,4} &= 1 \\
x_{2,4,5} &= 1 & x_{5,5,5} &= 1 & x_{8,6,5} &= 1 \\
x_{2,7,6} &= 1 & x_{5,8,6} &= 1 & x_{8,9,6} &= 1 \\
x_{3,1,7} &= 1 & x_{6,2,7} &= 1 & x_{9,3,7} &= 1 \\
x_{3,4,8} &= 1 & x_{6,5,8} &= 1 & x_{9,6,8} &= 1 \\
x_{3,7,9} &= 1 & x_{6,8,9} &= 1 & x_{9,9,9} &= 1
\end{aligned}
$$

Laderman 1976; Bini et al. 1979; Bläser 2003; Benson & Ballard, PPoPP'15

5/4/2018 Tensor Tutorial @ SDM18 67

Lab 1 (40 minutes)



1. Download labs and toolboxes from:
   https://tinyurl.com/tensor-tutorial-sdm18
2. In MATLAB, change directory: `tensor_tutorial/labs`
3. Open and work through the following labs (in order):
   a) `lab_setup.mlx`
   b) `lab_intro_to_tensors.mlx`
   c) `lab_cpals_cpopt.mlx`
   d) `lab_cpals_cpopt_large_data.mlx`
4. You are free to continue working on the labs through the break.

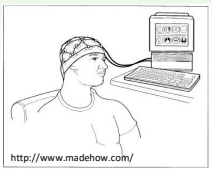
When you see this in the labs, **STOP** and try out the exercises on your own.

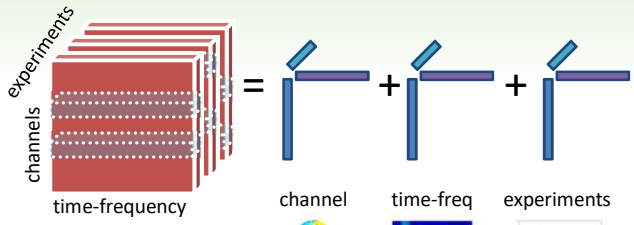
5/4/2018 Tensor Tutorial @ SDM18 70

Missing Data

Tammy Kolda, Evrim Acar, Morten Mørup, and Danny Dunlavy

Tensor Factorizations with Missing Data?





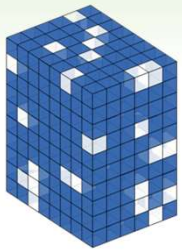
Biomedical signal processing

- EEG (electroencephalogram) signals can be recorded using electrodes placed on the scalp
- **Missing data problem** occurs when…
  - Electrodes get loose or disconnected, causing the signal to be unusable
  - Different experiments have over-lapping but not identical channels

Can we still do this calculation if data are missing?

Acar, Dunlavy, Kolda & Mørup 2010 & 2011

Approaches to Missing Data



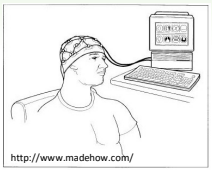
Ω = *set of known entries (blue)*

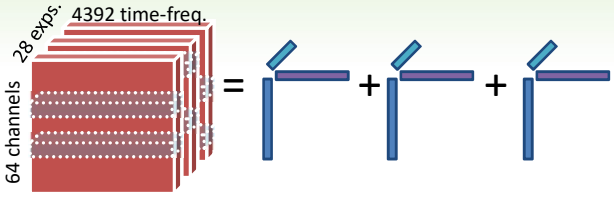
$$\min \sum_{i \in \Omega} (x_i - m_i)^2$$
$$\text{s.t. } \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

- Guess Missing Entries
- Expectation Maximization
  1. Generate initial model
  2. Estimate missing entries (using current model)
  3. Update model (data changed)
  4. Go back to step #2
- Ignore Missing Data
  - See equation at left
  - Closely related to ideas in matrix completion…except we haven't discussed about picking r

Acar, Dunlavy, Kolda & Mørup 2010 & 2011

Brain dynamics can be captured even extensive missing channels

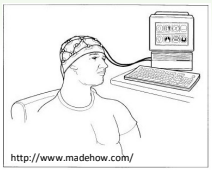


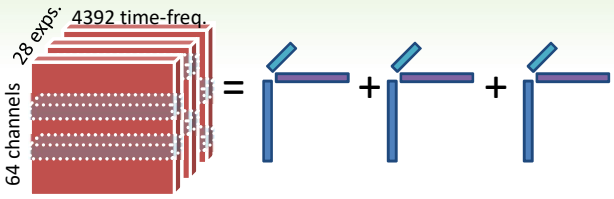


Number of Missing Channels	Replace Missing Entries with Mean
1	0.98
10	0.82
20	0.67
30	0.45
40	0.24

Acar, Dunlavy, Kolda & Mørup 2010 & 2011

Brain dynamics can be captured even extensive missing channels

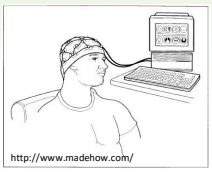


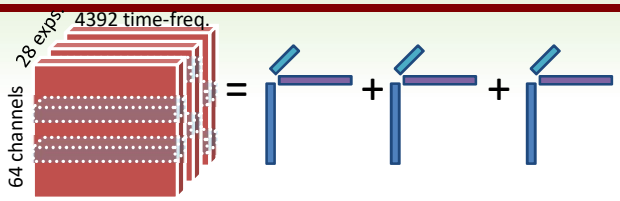


Number of Missing Channels	Replace Missing Entries with Mean	Ignore Missing Entries
1	0.98	1.00
10	0.82	0.98
20	0.67	0.95
30	0.45	0.89
40	0.24	0.65

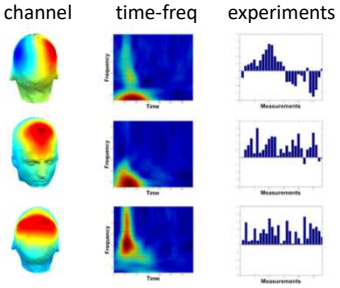
Acar, Dunlavy, Kolda & Mørup 2010 & 2011

Brain dynamics can be captured even extensive missing channels

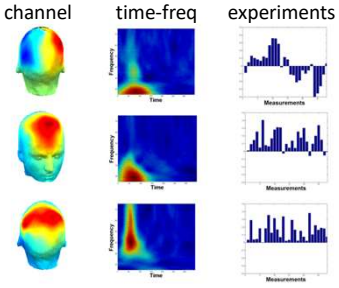




No Missing Data

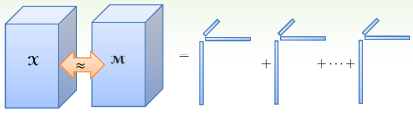


30 Chan./Exp. Missing



Acar, Dunlavy, Kolda & Mørup 2010 & 2011

Review: Handling Missing Data



$$\mathcal{X} \approx \mathcal{M} = \sum_{j=1}^{r} \mathbf{a}_j \circ \mathbf{b}_j \circ \mathbf{c}_j = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

Original problem:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \sum_i (x_i - m_i)^2 \text{ s.t. } \mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

$$F(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_i (x_i - m_i)^2$$

$$F(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{i \in \Omega} (x_i - m_i)^2$$

Just compute sum at known values

In practice, the user specifies a **mask** binary tensor as follows:

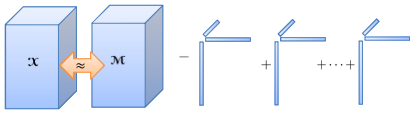
$$w_i = \begin{cases} 1 & \text{if } x_i \text{ known} \\ 0 & \text{if } x_i \text{ missing} \end{cases}$$

Anderson-Bergman, Duersch, Hong, Kolda 2017

Generalized CP Decomposition

Tammy Kolda, Jed Duersch (Sandia), David Hong (Michigan), Cliff Anderson-Bergman (Sandia)

Generalizing the Goodness-of-Fit Criteria



$$\mathcal{X} \approx \mathcal{M} = \sum_{j=1}^{r} \mathbf{a}_j \circ \mathbf{b}_j \circ \mathbf{c}_j = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

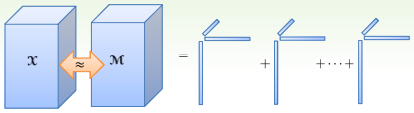
$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \sum_i (x_i - m_i)^2 \text{ s.t. } \mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

$$F(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_i (x_i - m_i)^2$$

$$F(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_i f(x_i, m_i)$$

Anderson-Bergman, Duersch, Hong, Kolda 2017

"Standard" CP via Maximum Likelihood



Typically: Consider data to be low-rank plus "white noise"

$$x_i = m_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma)$$

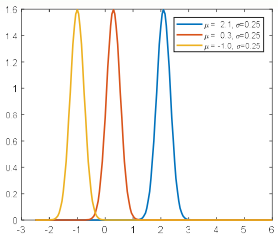
Equivalently, Gaussian with mean m_{ijk}

$$x_i \sim \mathcal{N}(m_i, \sigma)$$

Gaussian Probability Density Function (PDF)

$$\frac{e^{-(x-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}$$

Probability Distribution Function: Normal-distributed with constant σ



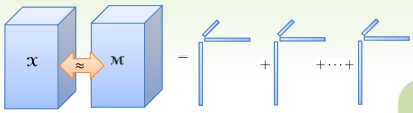
Minimize negative log likelihood with $\mu_i = m_i$ and σ constant for all entries:

$$-\log(\mathcal{L}(\mathcal{M})) = \sum_{ijk} \frac{(x_i - m_i)^2}{2\sigma^2} + \tfrac{1}{2}\log 2\pi\sigma^2$$

$$\min F(\mathcal{M}) = \sum_i (x_i - m_i)^2$$

Anderson-Bergman, Duersch, Hong, Kolda 2017

"Rayleigh CP" with Linear Link



What if the data is nonnegative ($x_i \geq 0$)?

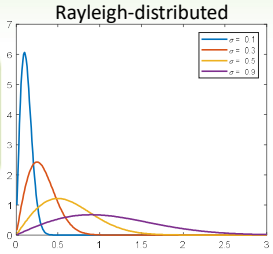
Assume data is Rayleigh-distributed.

$$x_i \sim \text{Rayleigh}(m_i)$$

Rayleigh Probability Density Function (PDF)

$$\frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}$$

Probability Distribution Function: Rayleigh-distributed



Minimize negative log likelihood with $\sigma_i = m_i$:

$$-\log(\mathcal{L}(\mathcal{M})) = \sum_i -\log x_i + 2\log m_i + \frac{x_i^2}{2m_i^2}$$

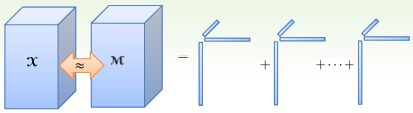
$$\min F(\mathcal{M}) = \sum_i 2\log m_i + \frac{x_i^2}{2m_i^2}$$

Requires $m_i \geq 0$

$$\mathbb{E}(x_i) = m_i\sqrt{\frac{\pi}{2}}$$

Anderson-Bergman, Duersch, Hong, Kolda 2017

Poisson CP: Identity Link

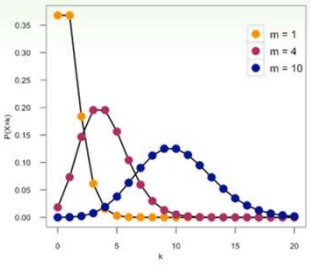


Consider data to be Poisson distributed with parameter m_i

$$x_i \sim \text{Poisson}(m_i)$$

Probability Mass Function (PMF):

$$\frac{e^{-\lambda}\lambda^x}{x!}$$



Minimizing negative log likelihood with $\lambda = m$:

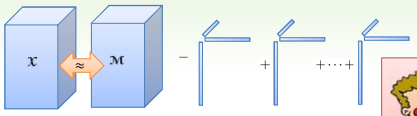
$$\min F(\mathcal{X}, \mathcal{M}) = \sum_i m_i - x_i \log(m_i) + \log(x_i!)$$

Requires $m_i \geq 0$

$$\mathbb{E}(X_i) = m_i$$

Chi & Kolda 2012, Anderson-Bergman, Duresch, Hong, and Kolda 2017

"Boolean CP" with Odds Link



Random Coin Flip: Probability versus Odds

$p \in [0,1]$: probability of 1

$r \geq 0$: odds ratio of 1

$$r = \frac{p}{1-p} \Leftrightarrow p = \frac{r}{1+r}$$

What if data is binary ($x_i \in \{0,1\}$)?

m_i= odds ratio of $x_i = 1$.

$$x_i \sim \text{Bernoulli}(m_i/(1+m_i))$$

$$\mathbb{E}(x_i) = \frac{m_i}{1+m_i}$$

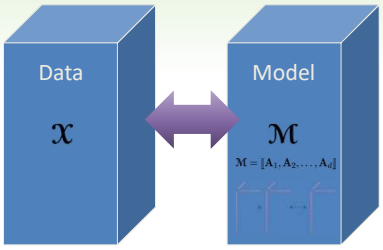
Requires $m_i \geq 0$

Probability Mass Distribution (PMF) $p^x(1-p)^{1-x} \Leftrightarrow \left(\frac{r}{1+r}\right)^x \left(\frac{1}{1+r}\right)^{1-x}$

$$\min F(\mathcal{M}) = \sum_i \log(m_i + 1) - x_i \log m_i$$

Anderson-Bergman, Duersch, Hong, Kolda 2017

Generalized CP



$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_i f(x_i, m_i)$$

$$\text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

"Normal" CP uses:

$$f(x,m) = (x-m)^2$$

"Nonneg" CP uses :

$$f(x,m) = 2\log m + x^2/(2m^2) \quad (x \geq 0, m \geq 0)$$

"Count" CP (Chi-Kolda 2012) uses:

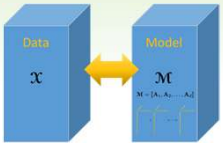
$$f(x,m) = m - x\log m \quad (x \in \mathbb{N}, m \geq 0)$$

"Binary" CP uses:

$$f(x,m) = \log(m+1) - x\log m \quad (x \in \{0,1\}, m \geq 0)$$

Anderson-Bergman, Duersch, Hong, Kolda 2017

Generalized CP Gradient



$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_i f(x_i, m_i)$$
$$\text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

Define a tensor $\mathcal{G}$ such that $g(i_1, \ldots, i_d) = g_i = \dfrac{\partial f}{\partial m}(x_i, m_i)$

Recall $m_i = \sum_{j=1}^{r} a_1(i_1, j)\, a_2(i_2, j) \cdots a_d(i_d, j)$

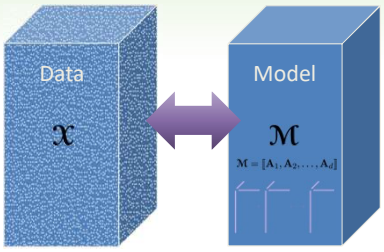
Then $$\frac{\partial F}{\partial a_k(i_k, j)} = \underbrace{g(i_1, \ldots, i_d)}_{\text{No dependency on model form (M)}} \underbrace{a_1(i_1, j) \cdots a_{k-1}(i_{k-1}, j)\, a_{k+1}(i_{k+1}, j) \cdots a_d(i_d, j)}_{\text{No dependency on function (f)}}$$

Matrix Version: $$\frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$$

MTTKRP!

Anderson-Bergman, Duersch, Hong, Kolda 2017

Generalized CP Gradient with Missing Data



Ω = Set of Known Entries in Data Tensor

$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \Omega} f(x_i, m_i)$$
$$\text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

Define a tensor $\mathcal{G}$ such that

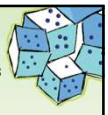
$$g_i = \begin{cases} \frac{\partial f}{\partial m}(x_i, m_i) & \text{if } i \in \Omega, \\ 0 & \text{if } i \notin \Omega. \end{cases}$$

Then (no change except G):

$$\frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{G}_{(k)}(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1)$$

Anderson-Bergman, Duersch, Hong, and Kolda 2017, Acar, Dunlavy, Kolda, Morup 2011

Observations on Generalized CP



- Can use any optimization method to solve the optimization problem

$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \Omega} f(x_i, m_i) \qquad \text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

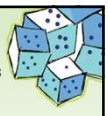
- Can easily add regularization

$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \Omega} f(x_i, m_i) + \sum_{k} \rho_k(\mathbf{A}_k) \quad \text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

- If data tensor (X) is sparse, no guarantee gradient will be efficient
  - Standard CP is a special case that has exploitable structure
  - Scalability is potentially a major problem
- If known data is sparse (Ω), then gradient will be efficient
  - Doesn't require any sparsity in data tensor
  - Perhaps this can be exploited? Yes!

Anderson-Bergman, Duersch, Hong, Kolda 2017

GCP Efficiency for Large Data



- If tensor is dense…
  - Just leave out data at random!
  - Ω represents the data we keep, usually only 10% of the data or less

$$F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \Omega} f(x_i, m_i)$$

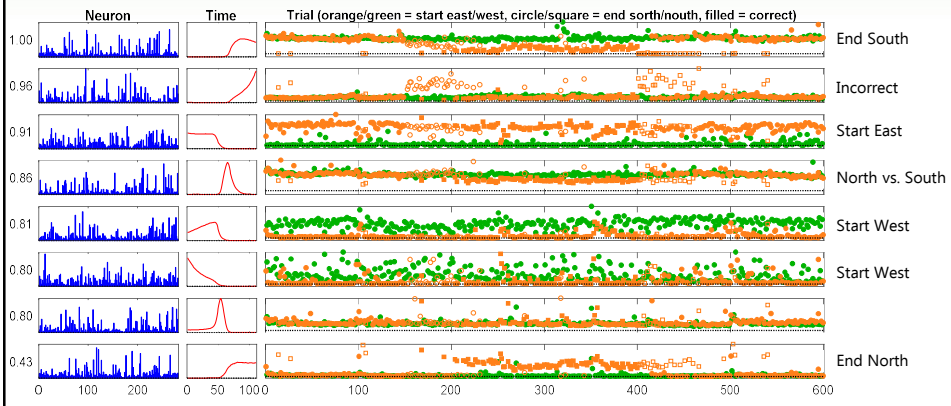
- If tensor is sparse…
  - Keep all the nonzeros but leave out all or most of the zeros
  - Add a penalty for the left out zero terms to encourage the model to stay small

$$F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \Omega} f(x_i, m_i) + \underbrace{\sum_{i \notin \Omega} \alpha m_i + \beta m_i^2}_{\text{nearly free to compute}}$$

  - Choices for α and β are up for debate

Anderson-Bergman, Duersch, Hong, Kolda 2017

Mouse Data using Rayleigh (Nonnegative)



Anderson-Bergman, Duersch, Hong, Kolda 2017

A Sparse Binary Dataset

- UC Irvine Chat Network
  - 4-way binary tensor
    - Sender (211)
    - Receiver (211)
    - Hour of Day (24)
    - Day (196)
  - 14,849 nonzeros (very sparse)

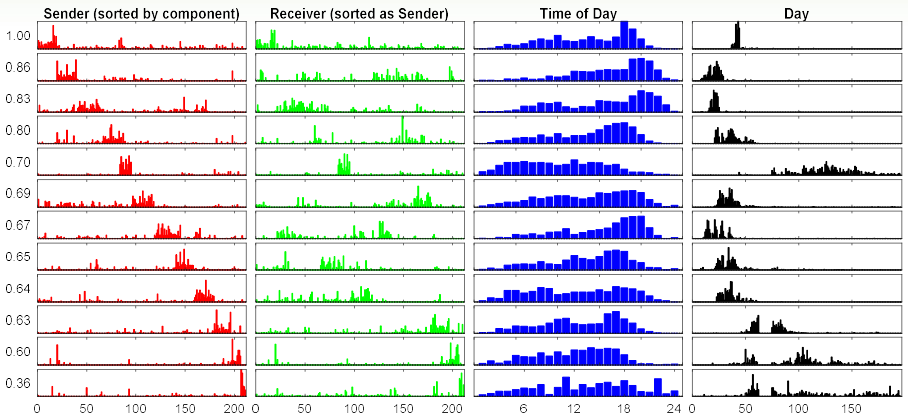
- Goodness-of-fit (Boolean-odds):

$$f(x,m) = \log(m+1) - x\log m$$

- Use GCP to compute rank-12 decomposition

Opsahl, T., Panzarasa, P., 2009. Clustering in weighted networks. Social Networks 31 (2), 155-163, doi: 10.1016/j.socnet.2009.02.002

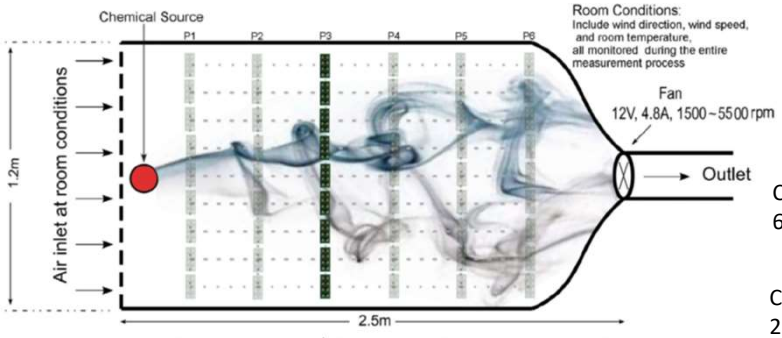
Binary Chat Data using Boolean CP



Anderson-Bergman, Duersch, Hong, Kolda 2017

Application to Hazardous Gas Dataset

71 Sensors × 5000 Timepoints × 5 Temperatures × 140 Experiments ≈ 2 GB



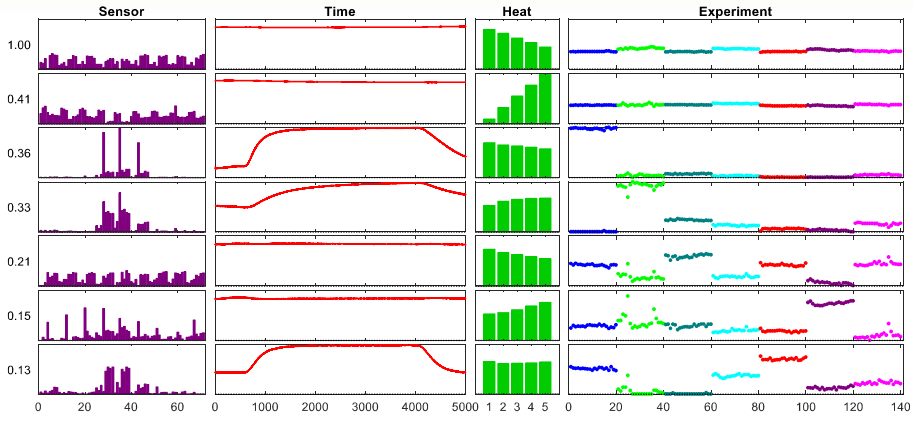
Position label	P 1	P 2	P 3	P 4	P 5	P 6
$x-axis\,distance\ (m)$	0.25	0.5	0.98	1.18	1.40	1.45

CP-ALS:
65 seconds

CP-ARLS:
27 seconds

A. Vergara, J. Fonollosa, J. Mahiques, M. Trincavelli, N. Rulkov and R. Huerta, ***On the performance of gas sensor arrays in open sampling systems using Inhibitory Support Vector Machines***, Sensors and Actuators B: Chemical, 2013, doi:10.1016/j.snb.2013.05.027

Gas Data Using Rayleigh

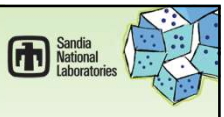


Anderson-Bergman, Duersch, Hong, Kolda 2017

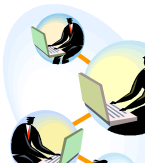
Poisson Tensor Factorization

Tammy Kolda, Eric Chi (NC State), Todd Plantenga (FireEye), Sammy Hansen (Spotify)

Sparse Tensor Computations

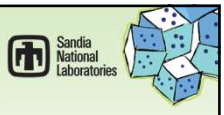


- Many real-world data analysis problems are naturally expressed as in terms of a *sparse* tensor
  - Computer traffic analysis
  - Term-document analysis
  - Email analysis
  - Link prediction
  - Web page analysis



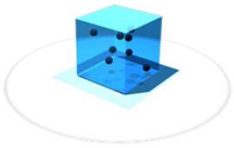
- But where do sparse tensors come from?
  - How are the entries generated?
  - How can we use that information in model fitting?

Generating Sparse Test Data

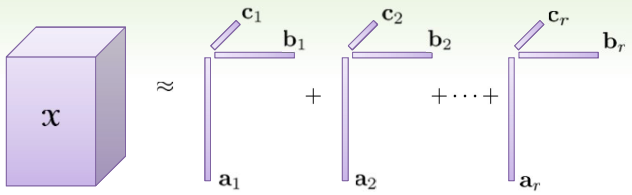


$$\mathcal{X} \sim \text{Poisson}\left(\lambda_1 \, \mathbf{c}_1 \, \mathbf{b}_1 \, \mathbf{a}_1 + \lambda_2 \, \mathbf{c}_2 \, \mathbf{b}_2 \, \mathbf{a}_2 + \cdots + \lambda_R \, \mathbf{c}_R \, \mathbf{b}_R \, \mathbf{a}_R \right)$$

- Each "occurrence" generated as follows
- Choose component j proportional to λ_j
- Given component j:
  - Choose index i_1 proportional to $\mathbf{a}_j$
  - Choose index i_2 proportional to $\mathbf{b}_j$
  - Choose index i_3 proportional to $\mathbf{c}_j$
- Increment $x(i_1, i_2, i_3)$ by one
- Repeat



Solving the Poisson Regression Problem



$$\min_{\mathcal{M}} \sum_i m_i - x_i \log m_i \text{ subject to } \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

- Highly nonconvex problem!
  - Assume r is given
- Alternating Poisson regression
  - Assume $(d-1)$ factor matrices are known and solve for the remaining one
  - Multiplicative updates like Lee & Seung (2000) for NMF, but improved
  - Typically assume data tensor $\mathcal{X}$ is sparse and have special methods for this
  - Newton or Quasi-Newton method

Chi & Kolda 2012; Hansen, Plantenga, & Kolda 2015

Alternating Poisson Regression (CP-APR)

Repeat until converged…

1. $\bar{\mathbf{A}} \leftarrow \arg\min_{\bar{\mathbf{A}} \geq 0} \sum_i m_i - x_i \log m_i \text{ s.t. } \mathcal{M} = \sum_j \bar{\mathbf{a}}_j \circ \mathbf{b}_j \circ \mathbf{c}_j$
2. $\boldsymbol{\lambda} \leftarrow \mathbf{e}^T \bar{\mathbf{A}}$; $\mathbf{A} \leftarrow \bar{\mathbf{A}} \cdot \text{diag}(1/\boldsymbol{\lambda})$

Fix **B**,**C**; solve for **A**

3. $\bar{\mathbf{B}} \leftarrow \arg\min_{\bar{\mathbf{B}} \geq 0} \sum_i m_i - x_i \log m_i \text{ s.t. } \mathcal{M} = \sum_j \mathbf{a}_j \circ \bar{\mathbf{b}}_j \circ \mathbf{c}_j$
4. $\boldsymbol{\lambda} \leftarrow \mathbf{e}^T \bar{\mathbf{B}}$; $\mathbf{B} \leftarrow \bar{\mathbf{B}} \cdot \text{diag}(1/\boldsymbol{\lambda})$

Fix **A**,**C**; solve for **B**

5. $\bar{\mathbf{C}} \leftarrow \arg\min_{\bar{\mathbf{C}} \geq 0} \sum_i m_i - x_i \log m_i \text{ s.t. } \mathcal{M} = \sum_j \mathbf{a}_j \circ \mathbf{b}_j \circ \bar{\mathbf{c}}_j$
6. $\boldsymbol{\lambda} \leftarrow \mathbf{e}^T \bar{\mathbf{C}}$; $\mathbf{C} \leftarrow \bar{\mathbf{C}} \cdot \text{diag}(1/\boldsymbol{\lambda})$

Fix **A**,**B**; solve for **C**

Convergence Theory

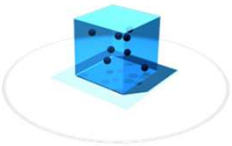
Theorem: The CP-APR algorithm will **converge to a constrained stationary point** if the subproblems are strictly convex and solved exactly at each iteration.

Chi & Kolda 2012

Solving the Subproblem

$$\min_{\bar{\mathbf{A}} \geq 0} \sum_i m_i - x_i \log m_i \text{ s.t. } \mathcal{M} = \sum_j \bar{\mathbf{a}}_j \circ \mathbf{b}_j \circ \mathbf{c}_j$$

$$\min_{\bar{\mathbf{A}} \geq 0} \sum_i \left(\sum_j \bar{a}_{i_1 j} b_{i_2 j} c_{i_3 j} \right) - x_i \log \left(\sum_j \bar{a}_{i_1 j} b_{i_2 j} c_{i_3 j} \right)$$

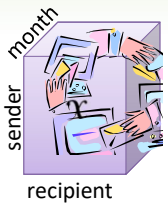


Lemma: The subproblems are **strictly convex** under mild conditions.

Chi & Kolda 2012

Motivating Example: Enron Email

- Emails from Enron FERC investigation
  - 8540 Messages
  - 28 Months (from Dec 1999 to Mar 2002)
  - 105 People (sent and received at least one email every month)
  - $x(i_1, i_2, i_3)$ = # emails from sender i_1 to recipient i_2 in month i_3
  - 105 x 105 x 28 = 308,700 possible entries
  - 8,500 nonzero counts
  - **3% dense**

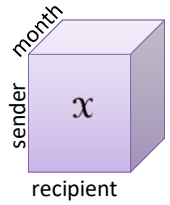


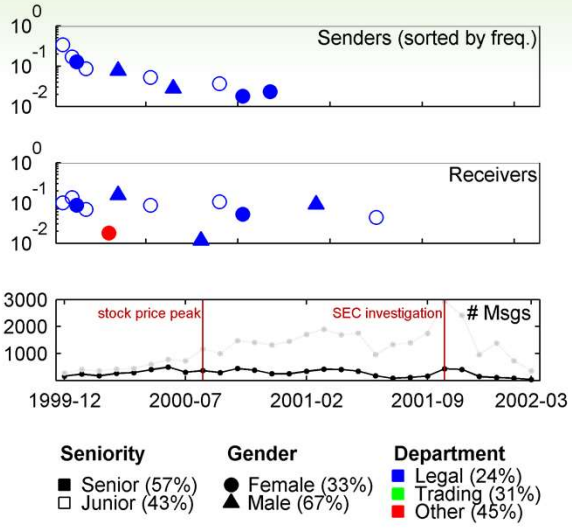
- Questions: What can we learn about this data?
  - Each person labeled by Zhou et al. (2007); see also Owen and Perry (2010)
    - Seniority: 57% senior, 43% junior
    - Gender: 67% male, 33% female
    - Department: 24% legal, 31% trading, 45% other

This information is not part of the tensor factorization

Enron Email Data (Component 1)

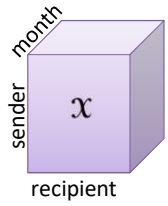
Legal Dept;
Mostly Female

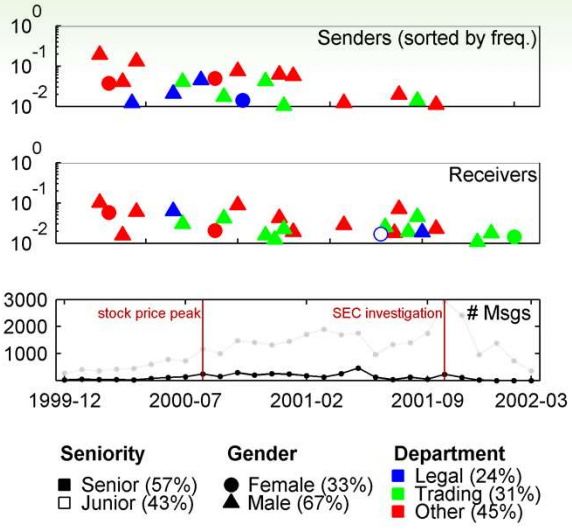




Enron Email Data (Component 3)

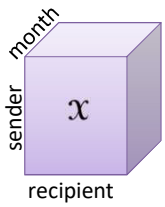
Senior;
Mostly Male

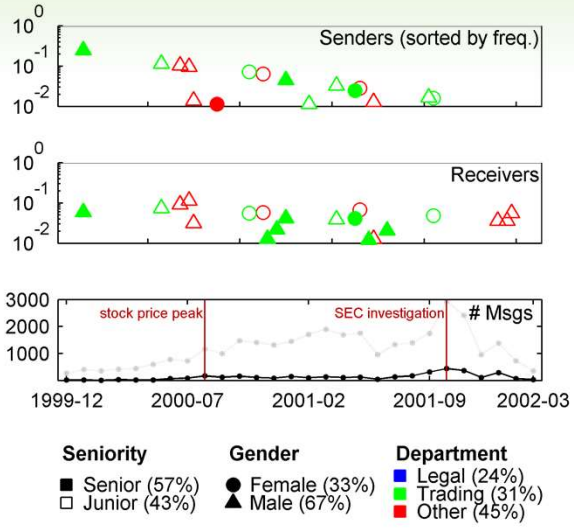




Enron Email Data (Component 4)

Not Legal





Randomized Least Squares for CP Decomposition

Tammy Kolda, Casey Battaglino (GA Tech)
and Grey Ballard (Wake Forest)

CP-ALS: Fitting CP Model via Alternating Least Squares

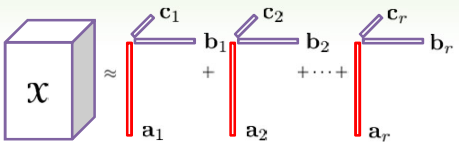
Sandia National Laboratories

Repeat until fit stops changing...

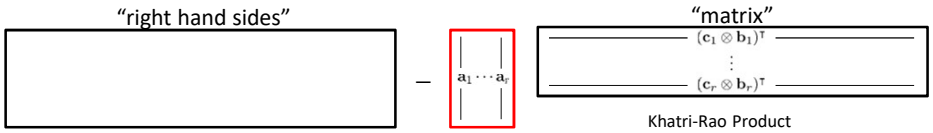
$$\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\intercal\|^2$$

$$\min_{\mathbf{B}} \|\mathbf{X}_{(2)} - \mathbf{B}(\mathbf{C} \odot \mathbf{A})^\intercal\|^2$$

$$\min_{\mathbf{C}} \|\mathbf{X}_{(3)} - \mathbf{C}(\mathbf{B} \odot \mathbf{A})^\intercal\|^2$$



$$\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\intercal\|^2$$



$$\|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\intercal\|_F^2$$

$n_k \times n^d/n_k$ $\quad$ $n_k \times r$ $\quad$ $r \times n^d/n_k$

Battaglino, Ballard, & Kolda 2017

Recall: Special Structure of the Least Squares Problem

Sandia National Laboratories

$$\min_{\mathbf{A}} \|\underbrace{\mathbf{X}_{(1)}}_{n_1 \times (n_2 n_3)} - \underbrace{\mathbf{A}}_{n_1 \times r}\underbrace{(\mathbf{C} \odot \mathbf{B})^\intercal}_{r \times (n_2 n_3)}\|^2$$

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

$$\mathbf{A}^\intercal = (\mathbf{C} \odot \mathbf{B})^\dagger \mathbf{X}_{(1)}^\intercal$$

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

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

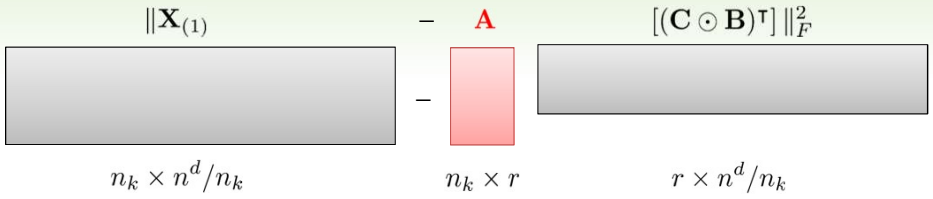
The most expensive step is *not* the backsolve.

Rather, it's the formation of the Khatri-Rao product!

So, how will randomized methods help?

Battaglino, Ballard, & Kolda 2017

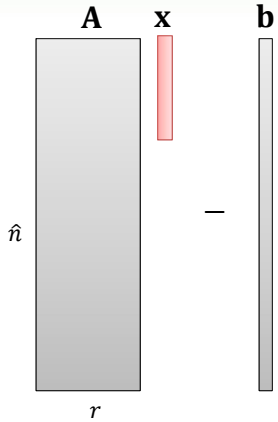
CP Least Squares Problem



How to randomize this?

Aside: Sketching for Standard Least Squares

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$$



$$\mathbf{x} \leftarrow \mathbf{A} \backslash \mathbf{b}$$

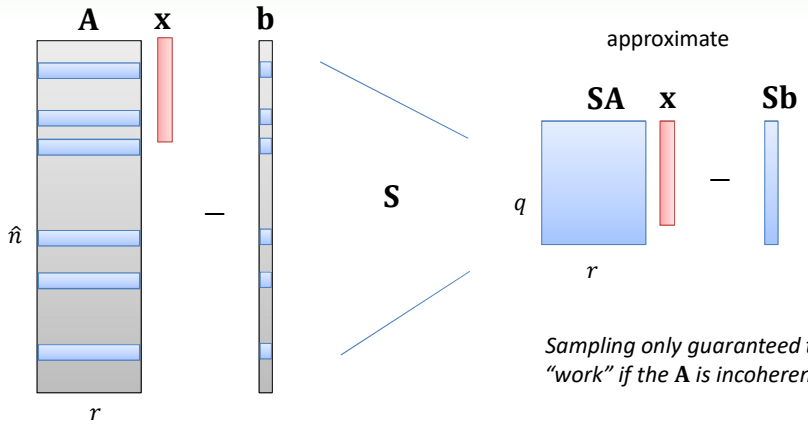
Backslash causes MATLAB to automatically call the best solver (qr, etc.)

$\mathcal{O}(\hat{n}r^2)$

Sarlós 2006, Woodruff 2014

Sampled Least Squares

Choose q rows, uniformly at random

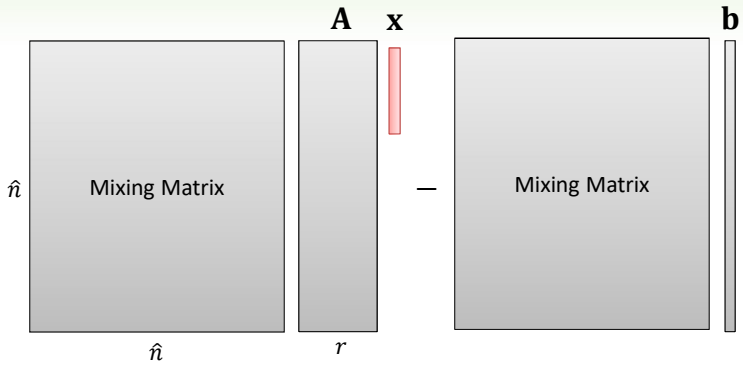


Sampling only guaranteed to "work" if the **A** *is incoherent.*

$\mathcal{O}(\hat{n}r^2)$ $\mathcal{O}(qr^2)$

Sarlós 2006, Woodruff 2014

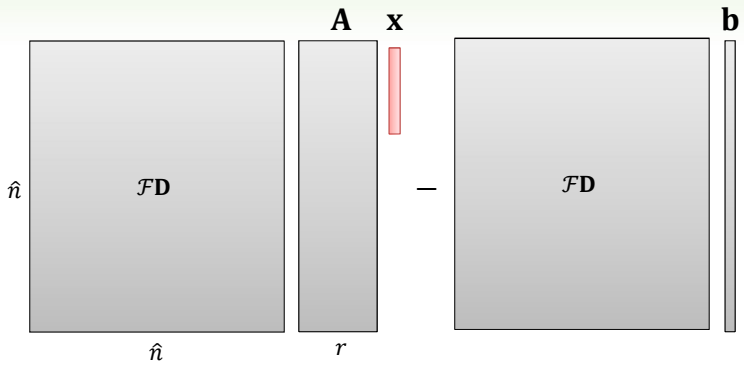
Enforcing Incoherence



- Many good choices of mixing matrix, such as a matrix with entries chosen from a uniform random distribution.
- But no reduction in cost!

Sarlós 2006, Woodruff 2014

Fast Johnson-Lindenstrauss Transform (FJLT)



- Instead, use FFT ($\mathcal{F}$) followed by random diagonal with +/- 1 entries ($\mathbf{D}$).
- Costs only $r \log \hat{n}$ to apply
- Practical application in Blendenpik, yielding ~4X speedup versus LAPACK

Sarlós 2006, Woodruff 2014, Ailon & Chazelle 2006, Avron, Maymounkov, & Toledo 2010

Sampled/Mixed Least Squares

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$$

$$\min_{\mathbf{x}} \|\mathbf{S}\mathbf{A}\mathbf{x} - \mathbf{S}\mathbf{b}\|$$

Sampling only, No mixing.

$$\min_{\mathbf{x}} \|\mathbf{S}\mathcal{F}\mathbf{D}\mathbf{A}\mathbf{x} - \mathbf{S}\mathcal{F}\mathbf{D}\mathbf{b}\|$$

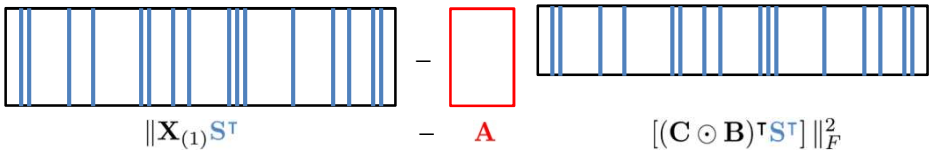
Sampling + Mixing

Ailon & Chazelle 2006; Avron, Maymounkov, & Toledo 2010

CP-ARLS

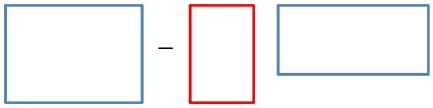
$$\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\intercal\|^2$$

$$\mathbf{A} \leftarrow \mathbf{X}_{(1)}/(\mathbf{C} \odot \mathbf{B})^\intercal$$



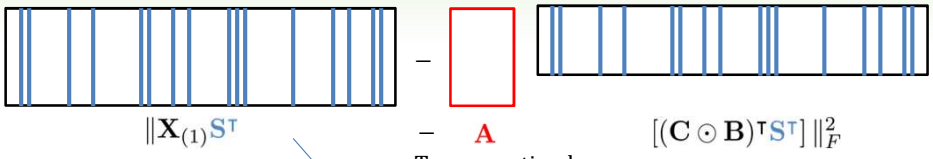
$\|\mathbf{X}_{(1)}\mathbf{S}^\intercal$ – $\mathbf{A}$ $[(\mathbf{C} \odot \mathbf{B})^\intercal\mathbf{S}^\intercal]\|_F^2$

$$\mathbf{A} \leftarrow \mathbf{X}_{(1)}\mathbf{S}^\intercal/(\mathbf{C} \odot \mathbf{B})^\intercal\mathbf{S}^\intercal$$



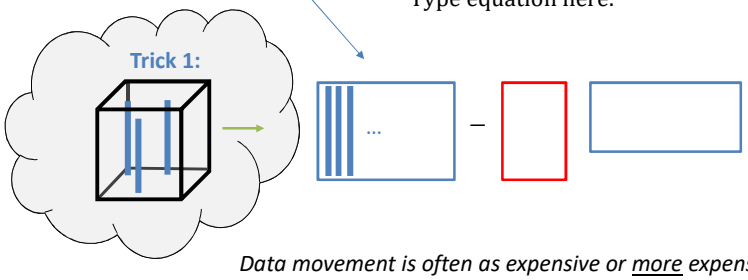
Battaglino, Ballard, & Kolda 2017

CP-ARLS Trick #1: Avoid unfolding data tensor



$\|\mathbf{X}_{(1)}\mathbf{S}^\intercal$ – $\mathbf{A}$ $[(\mathbf{C} \odot \mathbf{B})^\intercal\mathbf{S}^\intercal]\|_F^2$

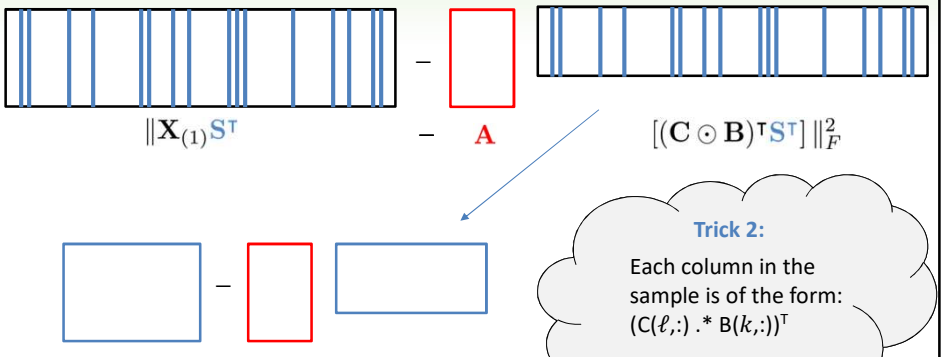
Type equation here.



Data movement is often as expensive or more expensive than FLOPS. Just move the minimum and no more.

Battaglino, Ballard, & Kolda 2017

CP-ARLS Trick #2: Don't form Khatri-Rao Product



The Khatri-Rao product is actually the most expensive part of CP-ALS. Skip this and save lotsa time.

Trick 2:

Each column in the sample is of the form: $(C(\ell,:) .* B(k,:))^\intercal$

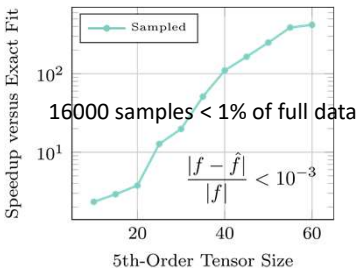
Mixing covered in the paper!

Battaglino, Ballard, & Kolda 2017

Randomizing the Convergence Check

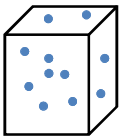
Repeat until fit stops changing…

$$\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^\intercal\|^2$$
$$\min_{\mathbf{B}} \|\mathbf{X}_{(2)} - \mathbf{B}(\mathbf{C}\odot\mathbf{A})^\intercal\|^2$$
$$\min_{\mathbf{C}} \|\mathbf{X}_{(3)} - \mathbf{C}(\mathbf{B}\odot\mathbf{A})^\intercal\|^2$$



16000 samples < 1% of full data

$$f(\mathbf{A},\mathbf{B},\mathbf{C}) = \sum_i (x_i - m_i)^2$$
$$\text{s.t. } \mathcal{M} = [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]$$

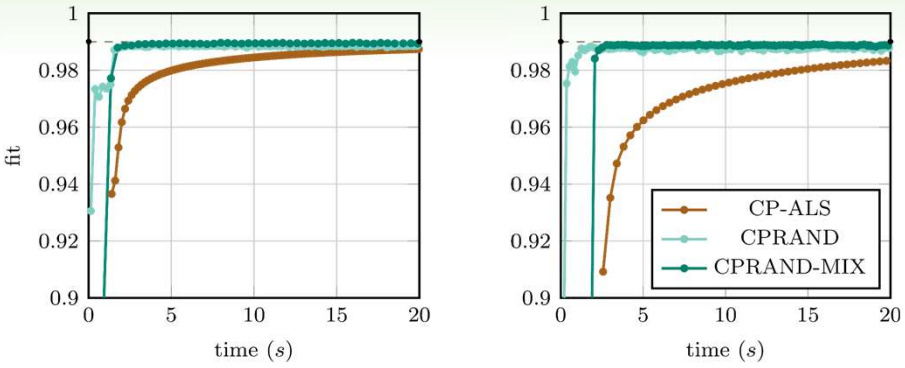


Estimate convergence of function values using small random subset of elements in function evaluation (use Chernoff-Hoeffding to bound accuracy)

$$f(\mathbf{A},\mathbf{B},\mathbf{C}) = \frac{n^d}{|\hat{\Omega}|}\sum_{i\in\hat{\Omega}}(x_i - m_i)^2$$

Battaglino, Ballard, & Kolda 2017

CPRAND faster than CP-ALS



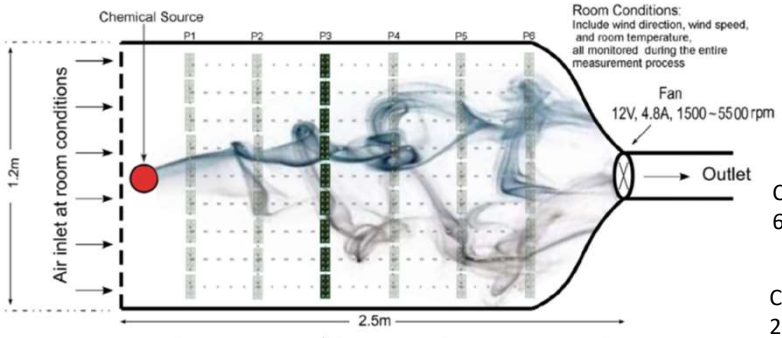
300 x 300 x 300 Random Rank-5 Tensor with 1% Noise

80 x 80 x 80 x 80 Random Rank-5 Tensor with 1% Noise

Battaglino, Ballard, & Kolda 2017

Application to Hazardous Gas Dataset

71 Sensors × 5000 Timepoints × 5 Temperatures × 140 Experiments ≈ 2 GB



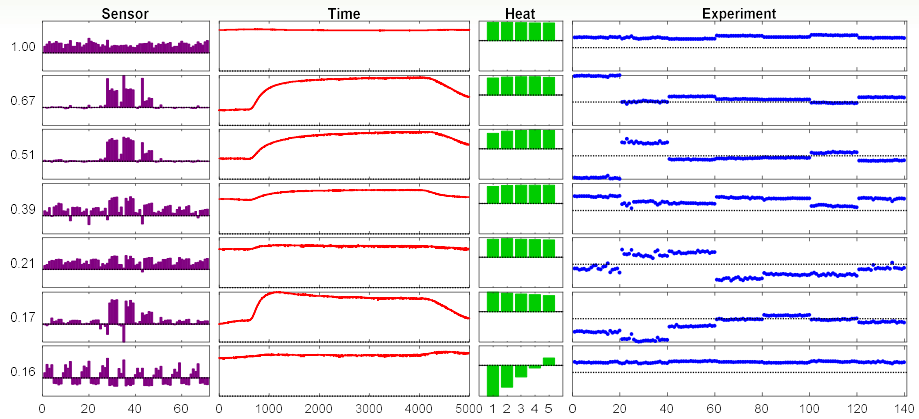
Position label	P 1	P 2	P 3	P 4	P 5	P 6
$x-axis\,distance\ (m)$	0.25	0.5	0.98	1.18	1.40	1.45

CP-ALS: 65 seconds

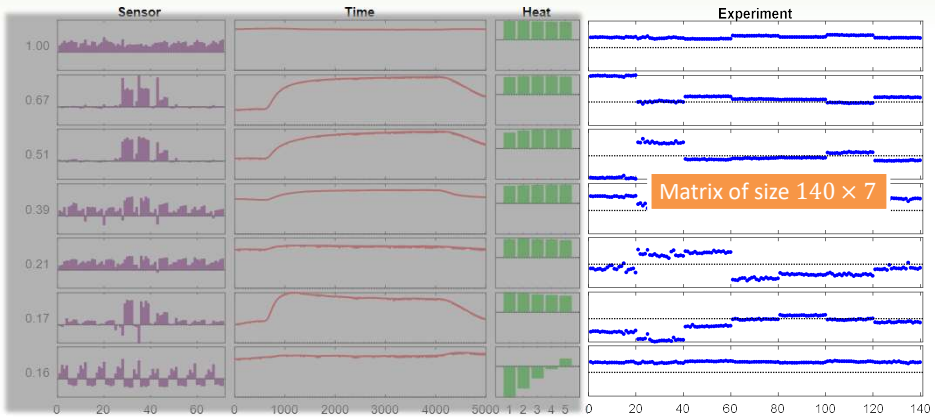
CP-ARLS: 27 seconds

A. Vergara, J. Fonollosa, J. Mahiques, M. Trincavelli, N. Rulkov and R. Huerta, ***On the performance of gas sensor arrays in open sampling systems using Inhibitory Support Vector Machines***, Sensors and Actuators B: Chemical, 2013, doi:10.1016/j.snb.2013.05.027

Factors from Gas Dataset



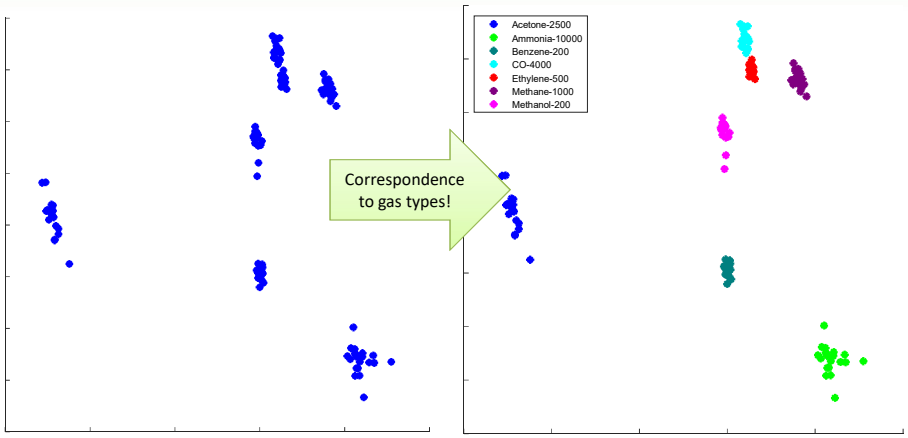
Factors from Gas Dataset



Viz of Experiment Factor Matrix Using PCA Projection

Sandia National Laboratories

Experiments (140) projected onto 2D space using PCA

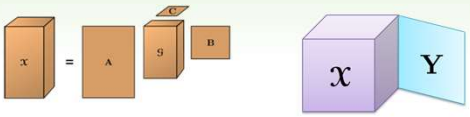


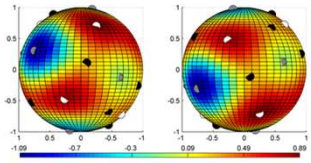
Sandia National Laboratories

Summary

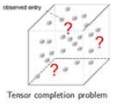
What We Didn't Cover

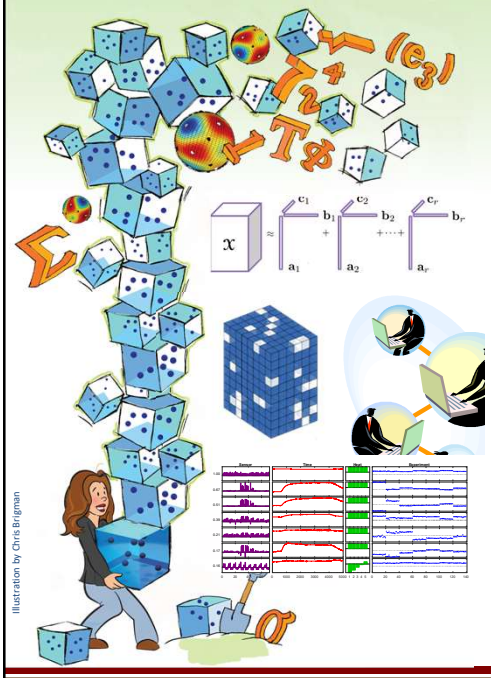
- Other tensor models
  - Tucker decomposition
  - Tensor train and hierarchical tensor decomposition
- Coupled factorizations
- Tensor eigenproblems
- Symmetric tensor decompositions
- Functional (i.e., continuous) tensor decomposition
- Tensor completion & tensor nuclear norm
- And much more!





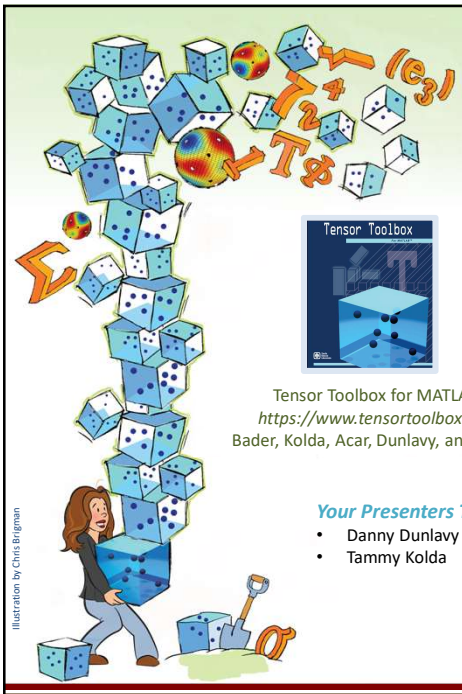
$$f(x_1, \ldots, x_d) = \sum_{j-1}^{r} \phi_j^{(1)}(x_1) \cdots \phi_j^{(d)}(x_d)$$





Takeaways

- CP has many applications
- Nonconvex, difficult optimization problem
- Determining number of components is NP-hard but critical for real-world applications
- How to handle missing data? Ignore it!
- Alternative objective functions
  - Poisson tensor factorization
  - Generalized CP
- MTTKRP is key kernel
- Matrix sketching speeds up least squares subproblems
- Many open problems!
- Applications
  - Sensor data
  - Neuroscience data
  - Network data



Tensor Toolbox for MATLAB:
https://www.tensortoolbox.org/
Bader, Kolda, Acar, Dunlavy, and others

Your Presenters Today
- Danny Dunlavy
- Tammy Kolda

Acknowledgements

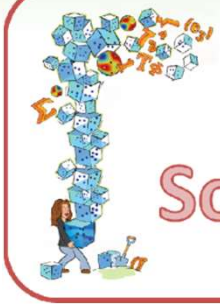
Current Collaborators
- Cliff Anderson-Bergman (Sandia)
- Gavin Baker (Sandia)
- Grey Ballard (Sandia/Wake Forest)
- Richard Barrett (Sandia)
- Casey Battaglino (Sandia/GA Tech)
- Jon Berry (Sandia)
- Karen Devine (Sandia)
- Jed Duersch (Sandia)
- Alex Gorodetsky (Sandia/Michigan)
- David Hong (Sandia/Michigan)
- Alicia Klinvex (Sandia)
- Hemanth Kolla (Sandia)
- Rich Lehoucq (Sandia)
- Jiajia Li (GA Tech)
- Eric Phipps (Sandia)
- Prashant Rai (Sandia)
- Keita Teranishi (Sandia)
- Rich Vuduc (GA Tech)
- Alex Williams (Sandia/Stanford)
- Kina Winoto (Sandia)
- Jeff Young (GA Tech)
- Plus many more previous collaborators!

References

- **Overall review:** T. G. Kolda and B. W. Bader, ***Tensor Decompositions and Applications***, SIAM Review, 2009, doi:10.1137/07070111X
- **Original MATLAB paper:** B. W. Bader and T. G. Kolda, ***Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping***, ACM Transactions on Mathematical Software, 2006, doi:10.1145/1186785.1186794
- **Special MATLAB classes:** B. W. Bader and T. G. Kolda, ***Efficient MATLAB Computations with Sparse and Factored Tensors***, SIAM Journal on Scientific Computing, 2007, doi:10.1137/060676489
- **Application to Mouse Data:** A. Williams, T. G. Kolda, S. Ganguli, et al., ***Unsupervised discovery of low-dimensional neural dynamics both within and across single trials through tensor analysis***, bioRxiv, 2017 (to appear in Neuron)
- **Application to Link Prediction:** D. M. Dunlavy, T. G. Kolda and E. Acar, ***Temporal Link Prediction using Matrix and Tensor Factorizations***, ACM Transactions on Knowledge Discovery from Data, 2011, doi:10.1145/1921632.1921636
- **All-at-once Optimization Approach:** E. Acar, D. M. Dunlavy and T. G. Kolda, ***A Scalable Optimization Approach for Fitting Canonical Tensor Decompositions***, Journal of Chemometrics, 2011, doi:10.1002/cem.1335
- **Missing data:** E. Acar, D. M. Dunlavy, T. G. Kolda, M. Mø and rup, ***Scalable Tensor Factorizations for Incomplete Data***, Chemometrics and Intelligent Laboratory Systems, 2011, doi:10.1016/j.chemolab.2010.08.004
- **More missing data:** E. Acar, D. M. Dunlavy, T. G. Kolda, M. Mø and rup, ***Scalable Tensor Factorizations with Missing Data***, SDM10: Proceedings of the 2010 SIAM International Conference on Data Mining, 2010, doi:10.1137/1.9781611972801.61
- **Randomized CP-ALS:** C. Battaglino, G. Ballard and T. G. Kolda, ***A Practical Randomized CP Tensor Decomposition***, January 2017, arXiv:1701.06600 (to appear in SIAM J. Matrix Analysis and Applications)
- **Poisson tensor factorization:** E. C. Chi and T. G. Kolda, ***On Tensors, Sparsity, and Nonnegative Factorizations***, SIAM Journal on Matrix Analysis and Applications, 2012, doi:10.1137/110859063
- **More Poisson tensor factorization:** S. Hansen, T. Plantenga and T. G. Kolda, ***Newton-Based Optimization for Kullback-Leibler Nonnegative Tensor Factorizations***, Optimization Methods and Software, 2015, doi:10.1080/10556788.2015.1009977
- **Generalized CP:** Cliff Anderson-Bergman, J. Duersch, D. Hong, T. G. Kolda, ***Generalized Canonical Polyadic Tensor Decomposition***, 2017 (coming soon)

Contact: Tammy Kolda, tgkolda@sandia.gov, Danny Dunlay, dmdunla@sandia.gov

Lab 2 (20 minutes)



Try it out.
Solutions Below.

- `lab_missing_data.mlx`
- `lab_nonnegative_cp.mlx`
- `lab_boolean_cp.mlx`
- `lab_cross_validation.mlx`
- `lab_possion.mlx`
- `lab_randomization.mlx`