Sandia National Laboratories

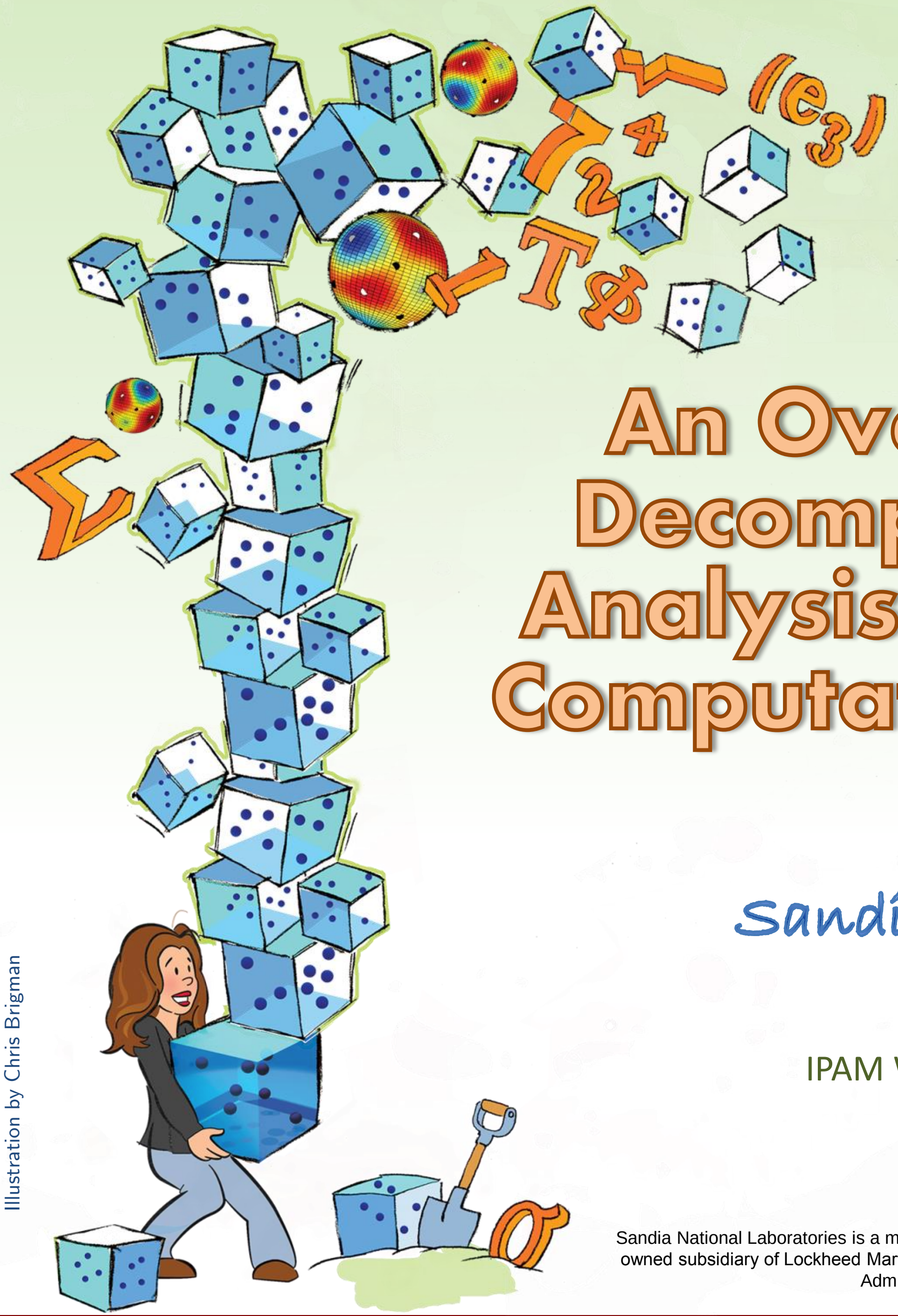


Illustration by Chris Brigman

An Overview of Tensor Decompositions for Data Analysis, with Emphasis on Computation and Scalability

Tamara G. Kolda

Sandia National Laboratories

Livermore, CA

IPAM Workshop: Big Data Meets Computation

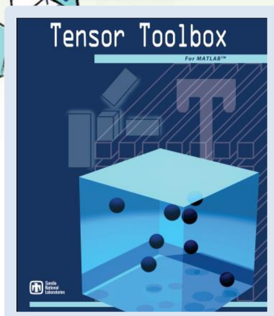
February 2, 2017

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

U.S. DEPARTMENT OF ENERGY

Acknowledgements

Kolda and Bader, Tensor Decompositions and Applications, *SIAM Review* '09



Tensor Toolbox for MATLAB: Bader, Kolda, Acar, Dunlavy, and others

TUCKERMPI

https://gitlab.com/tensors/TuckerMPI
Klinvex, Austin, Ballard, Kolda

- Evrim Acar (Univ. Copenhagen*)
- **Woody Austin (Univ. Texas Austin*)**
- Brett Bader (Digital Globe*)
- **Grey Ballard (Wake Forest*)**
- Robert Bassett (UC Davis*)
- **Casey Battaglino (GA Tech*)**
- **Eric Chi (NC State Univ.*)**
- Danny Dunlavy (Sandia)
- Sammy Hansen (IBM*)
- Joe Kenny (Sandia)
- **Alicia Klinvex (Sandia)**
- **Hemanth Kolla (Sandia)**
- Jackson Mayo (Sandia)
- Morten Mørup (Denmark Tech. Univ.)
- Todd Plantenga (FireEye*)
- Martin Schatz (Univ. Texas Austin*)
- Teresa Selee (GA Tech Research Inst.*)
- Jimeng Sun (GA Tech)
- **Alex Williams (Stanford*)**

Plus many more former students and colleagues, collaborators for workshops, tutorials, etc.

** = Long-term stint at Sandia*

Illustration by Chris Brigman

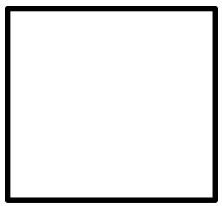
Intro

A Tensor is an d-Way Array

Vector
d = 1

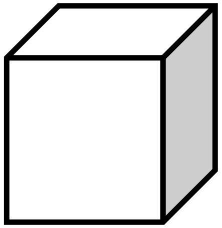
$\mathbf{a}$

Matrix
d = 2



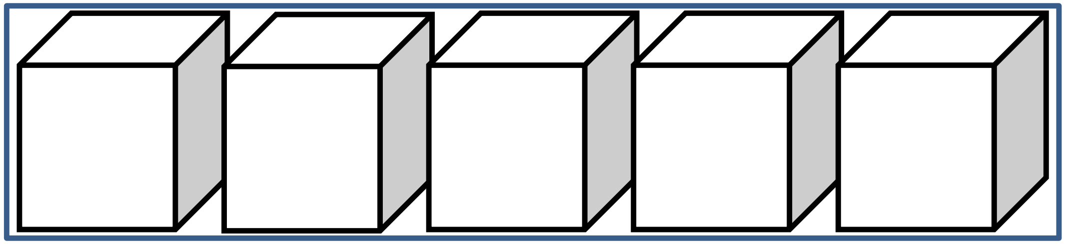
$\mathbf{A}$

3rd-Order Tensor
d = 3



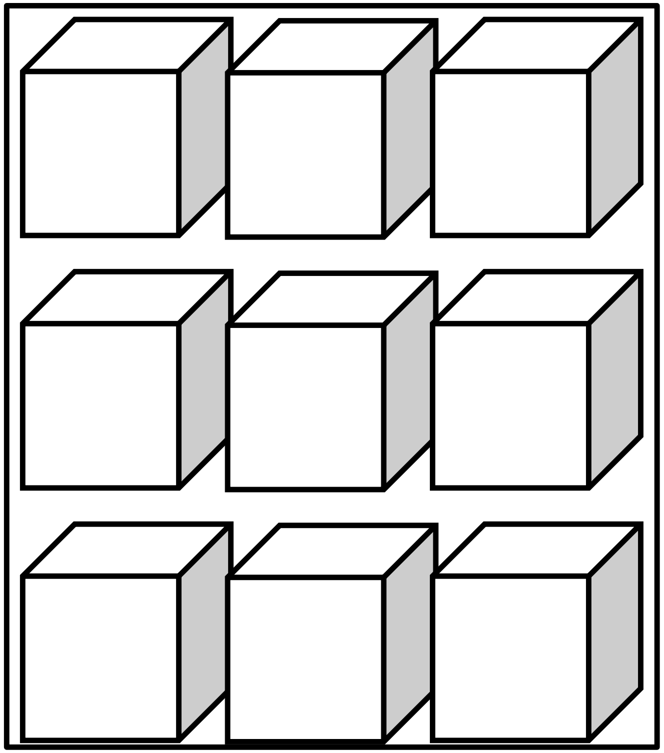
$\mathcal{A}$

4th-Order Tensor
d = 4



$\mathcal{A}$

5th-Order Tensor
d = 5

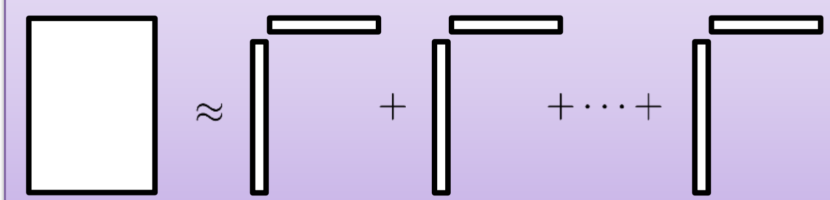


$\mathcal{A}$

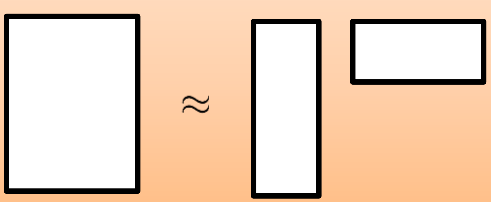
Tensor Decompositions are the New Matrix Decompositions

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

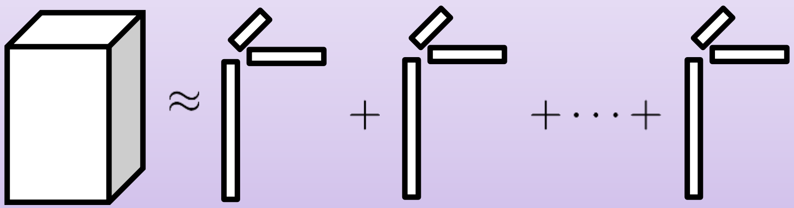
Viewpoint 1: Sum of outer products, useful for interpretation



Viewpoint 2: High-variance subspaces, useful for compression

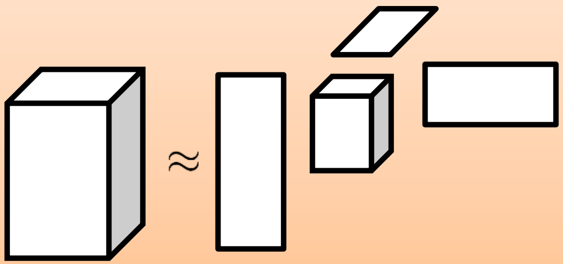


CP Model: Sum of d-way outer products, useful for interpretation



CANDECOMP, PARAFAC, Canonical Polyadic, CP

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



HO-SVD, Best Rank-(R1,R2,…,RN) decomposition

Other models for compression include hierarchical Tucker and tensor train.

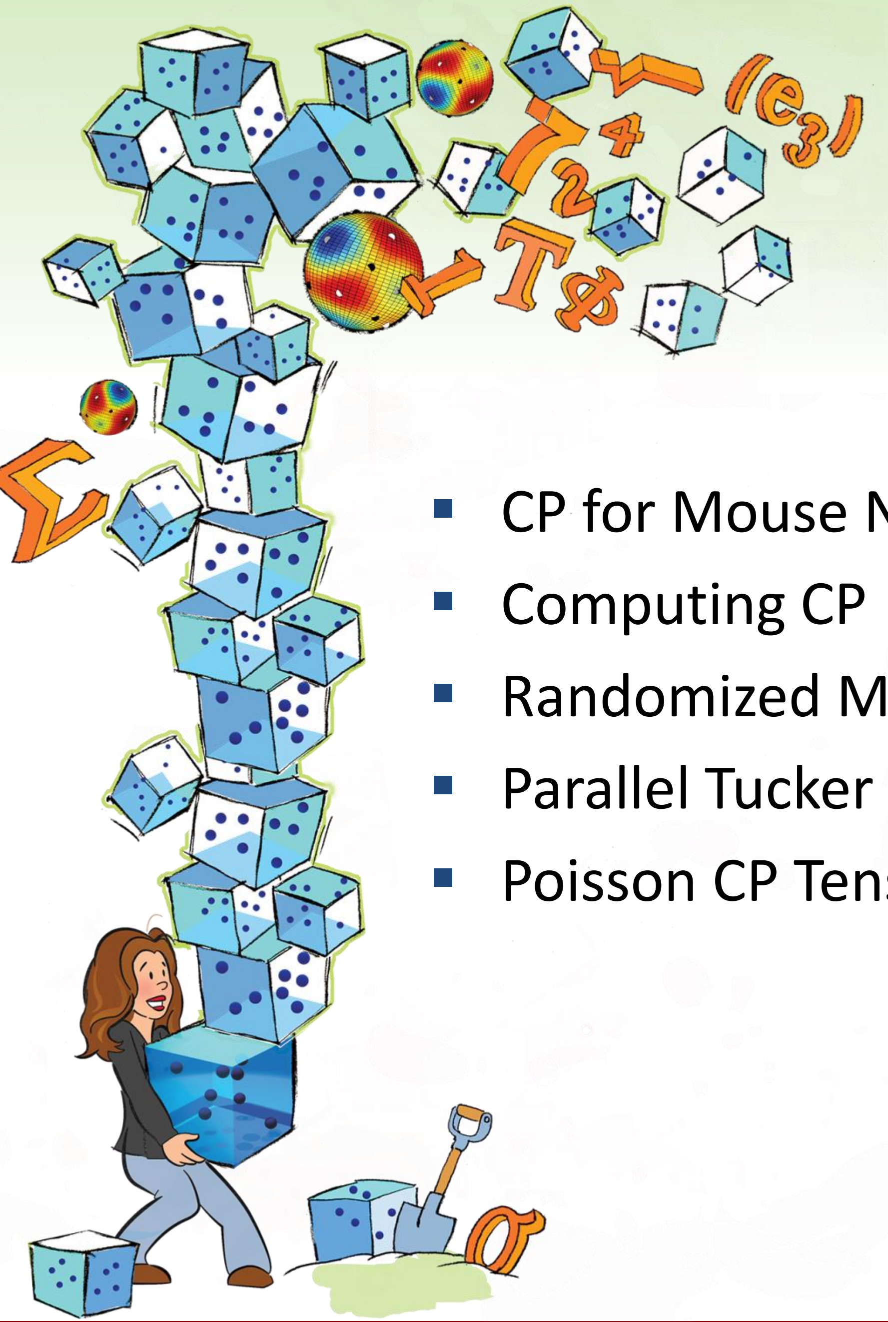


Illustration by Chris Brigman

Outline

- CP for Mouse Neural Activity
- Computing CP
- Randomized Method for Computing CP
- Parallel Tucker for Data Compression
- Poisson CP Tensor Decomposition (time permitting)

Motivation for CP Tensor Decomposition

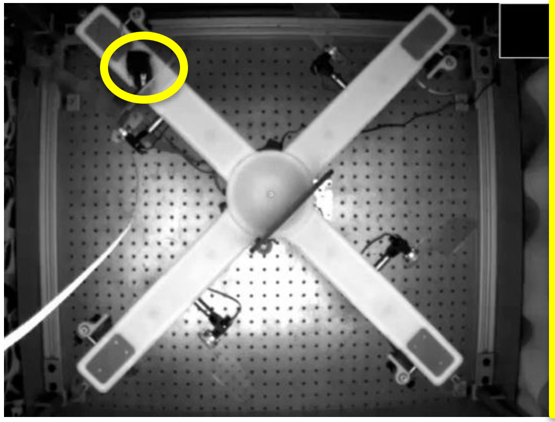
Featuring work of Alex Williams

Motivating Example: Neuron Activity in Learning

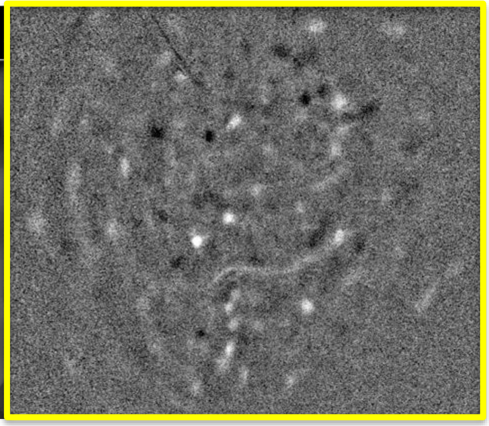
Thanks to Schnitzer Group @ Stanford
Mark Schnitzer, Fori Wang, Tony Kim

Microscope by Inscopix

mouse in "maze"

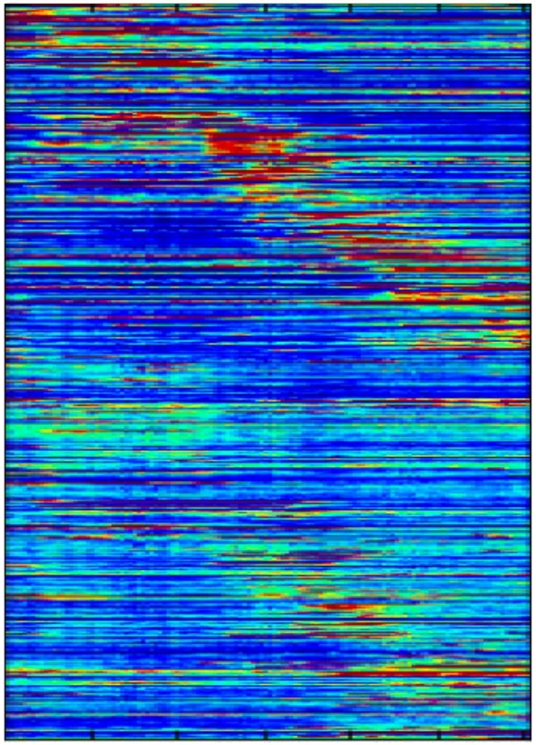


neural activity



One Column of Neuron x Time Matrix

One Trial
300 neurons × 120 time bins



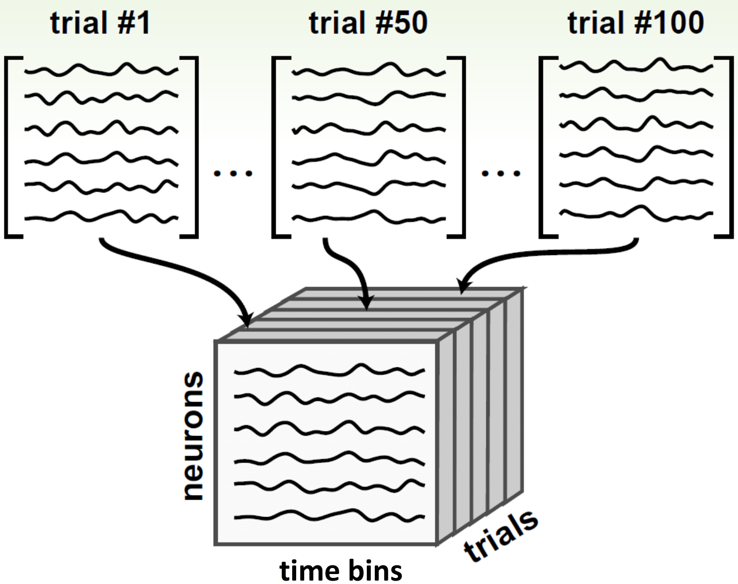
neurons

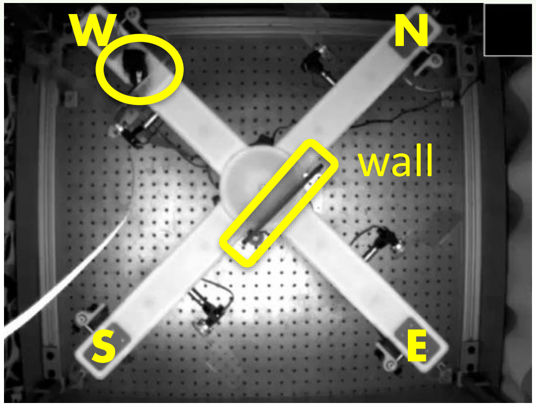
time ⟶

× 600 trials (over 5 days)

Coming soon: Williams, Wang, Kim, Schnitzer, Ganguli, Kolda, 2016

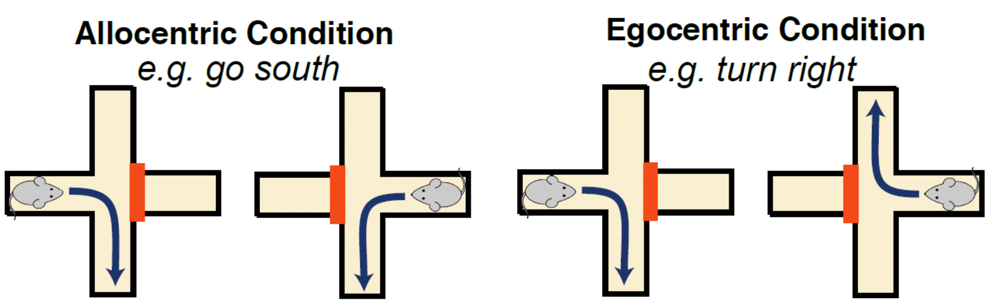
Trials Vary Start Position and Strategies





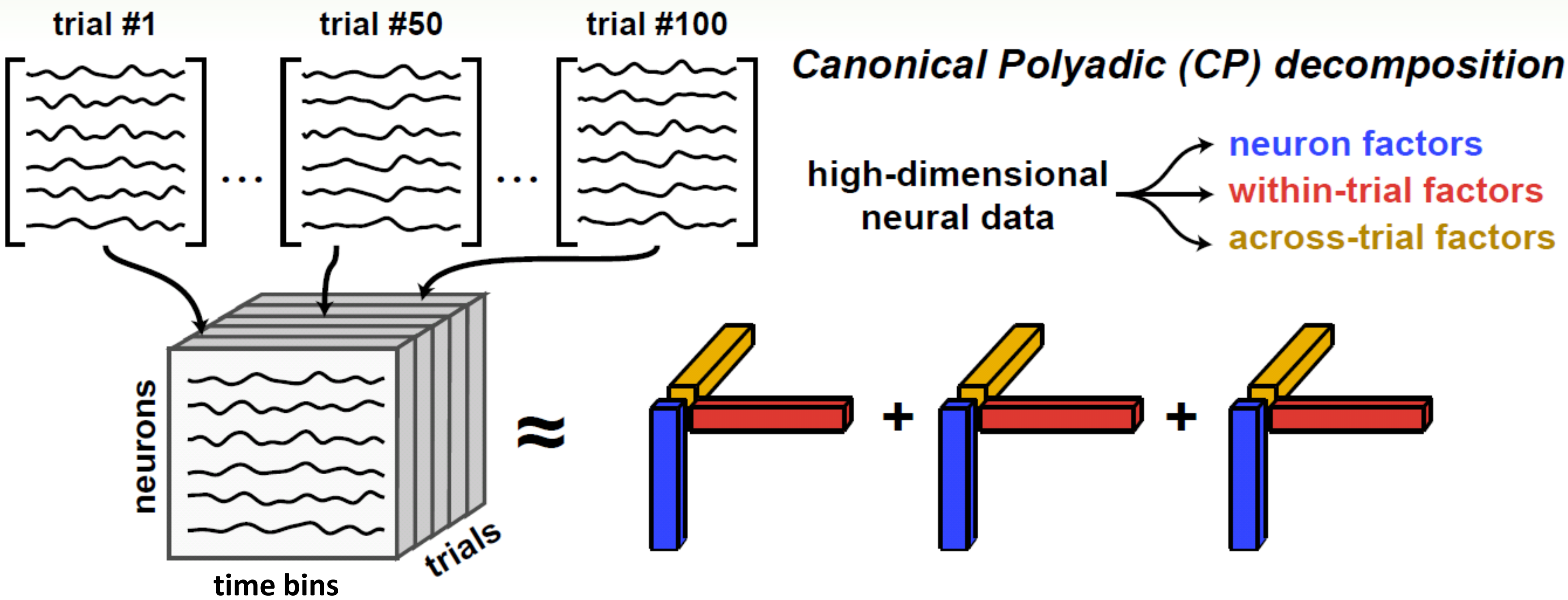
note different patterns on curtains

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



Coming soon: Williams, Wang, Kim, Schnitzer, Ganguli, Kolda, 2016

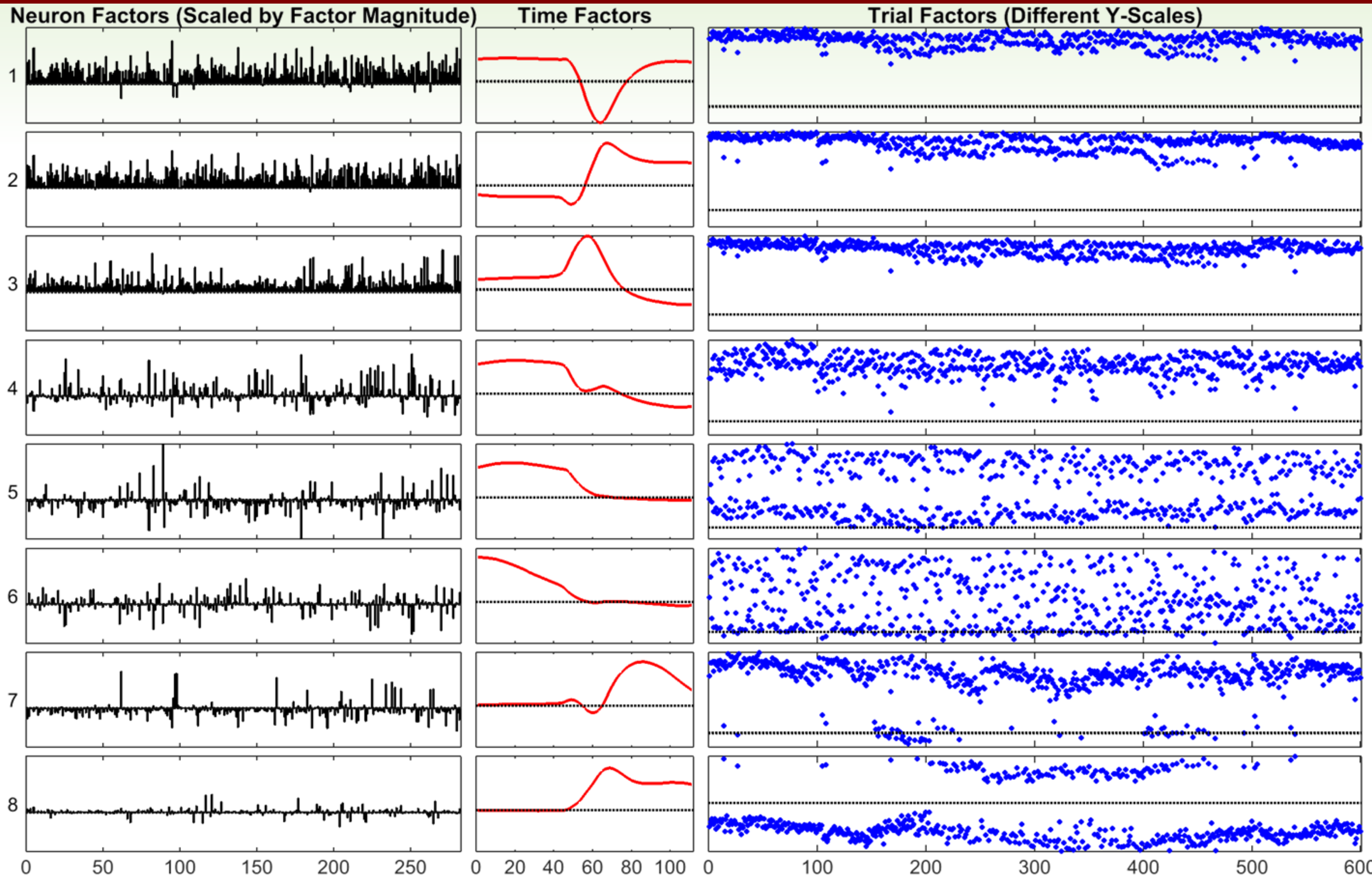
CP for Simultaneous Analysis of Neurons, Time, and Trial



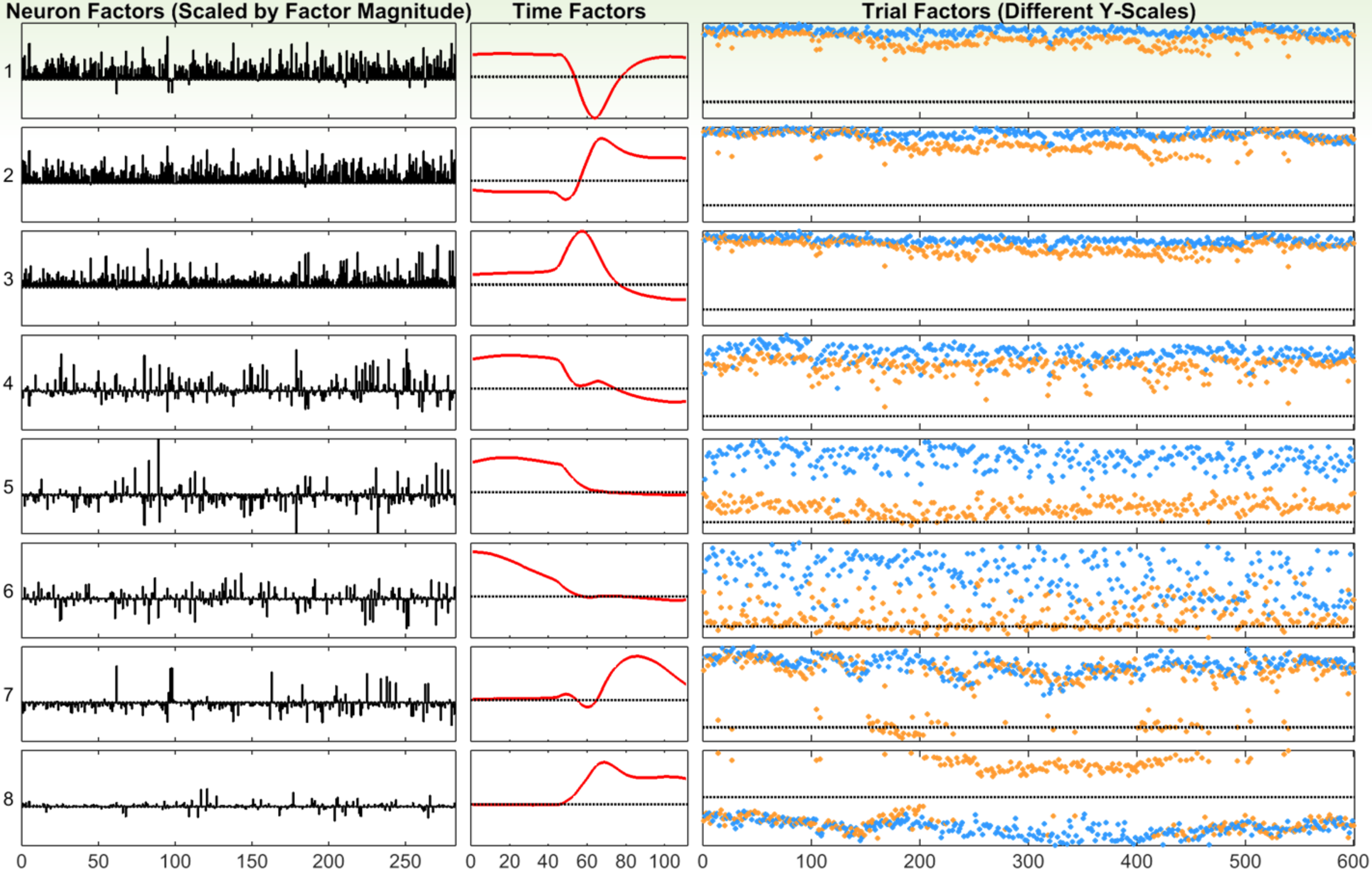
Past work could only look at 2 factors at once: Time x Neuron, Trial x Neuron, etc.

Coming soon: Williams, Wang, Kim, Schnitzer, Ganguli, Kolda, 2016

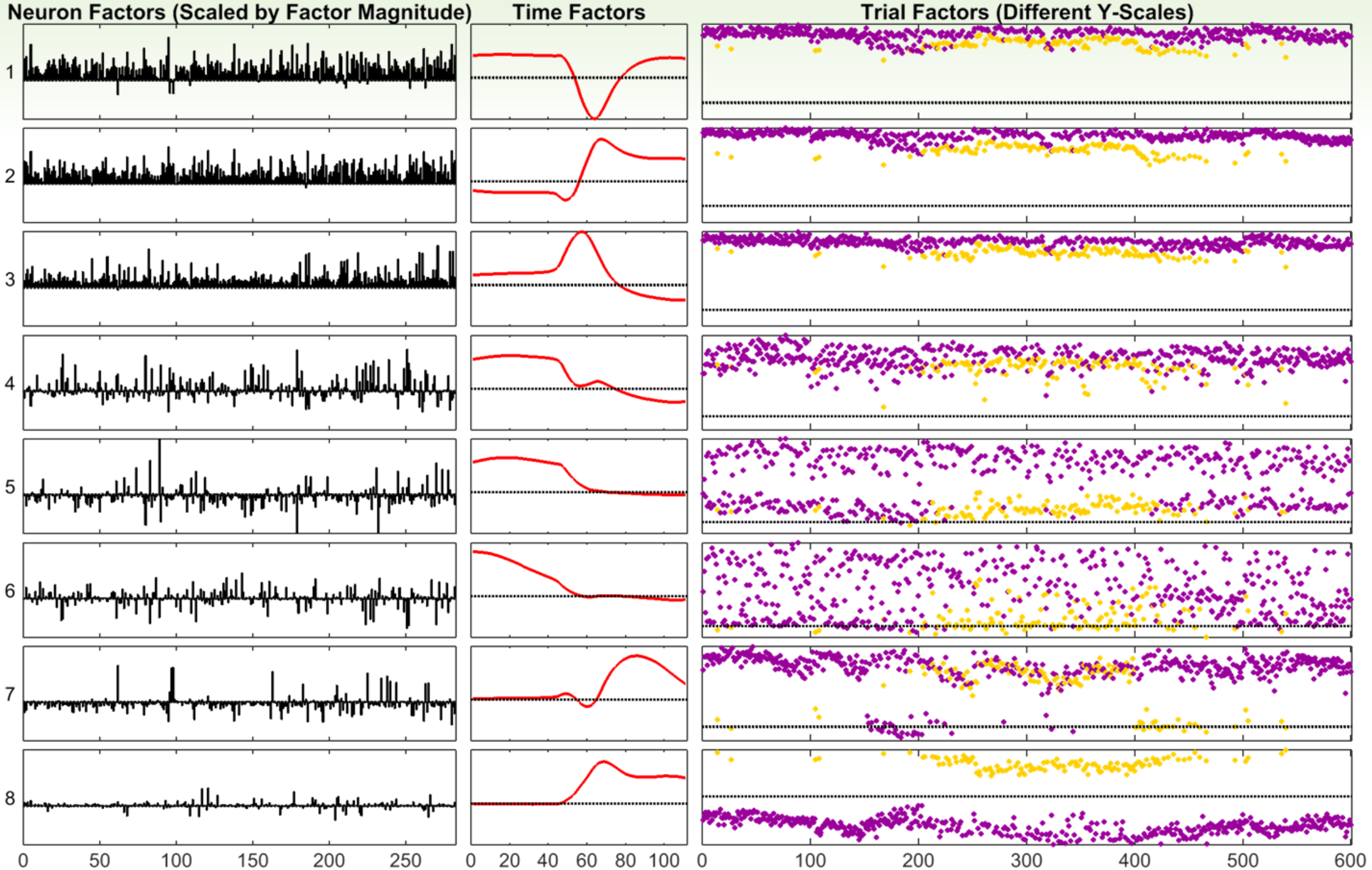
8-Component CP Tensor Decomposition



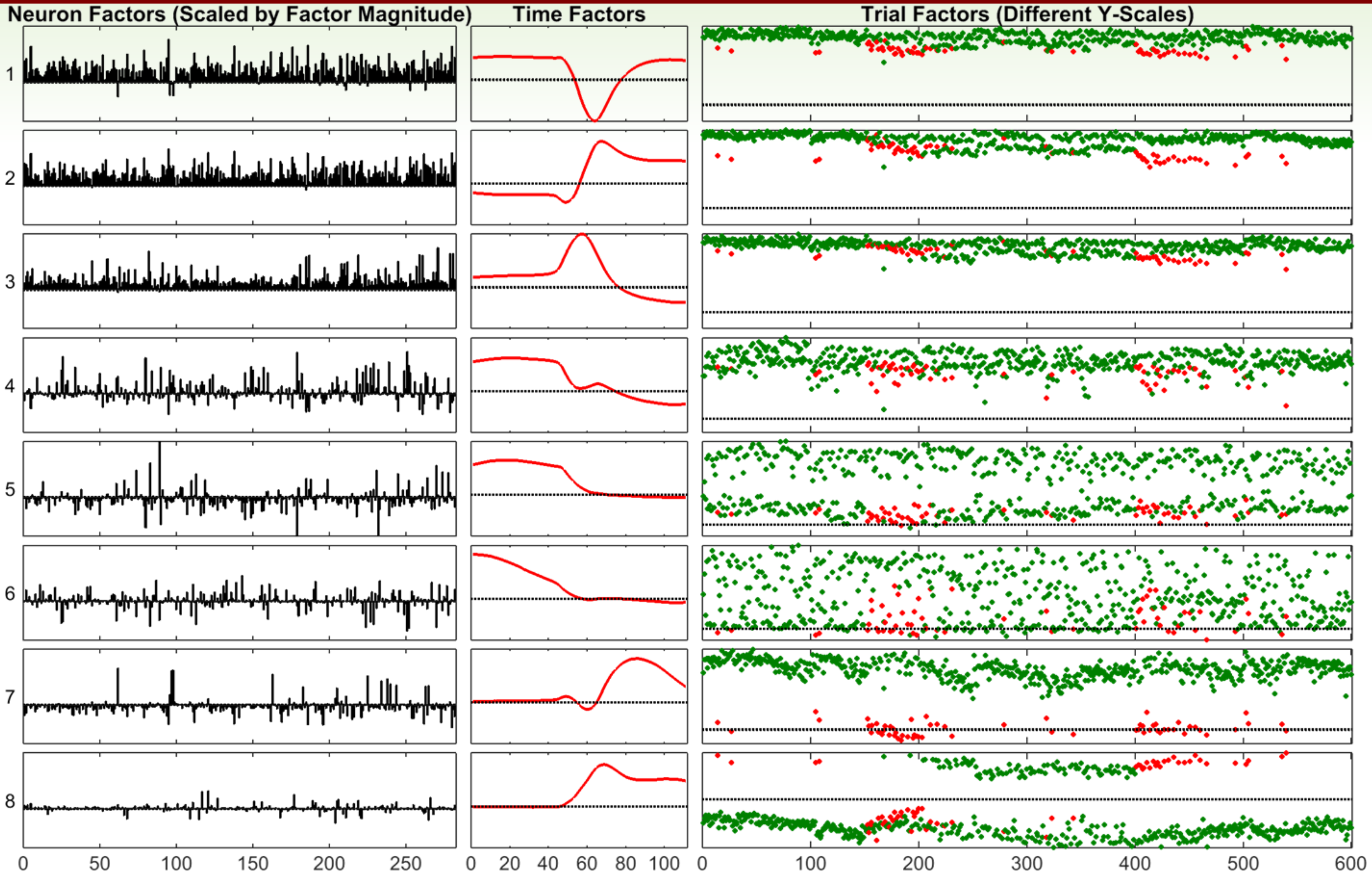
Component 5 Separates Starting Point (Orange = East, Blue = West)



Component 8 Separates End Point (Purple = South, Orange = North)



Component 7 Separates Reward (Green = Yes, Red = No)

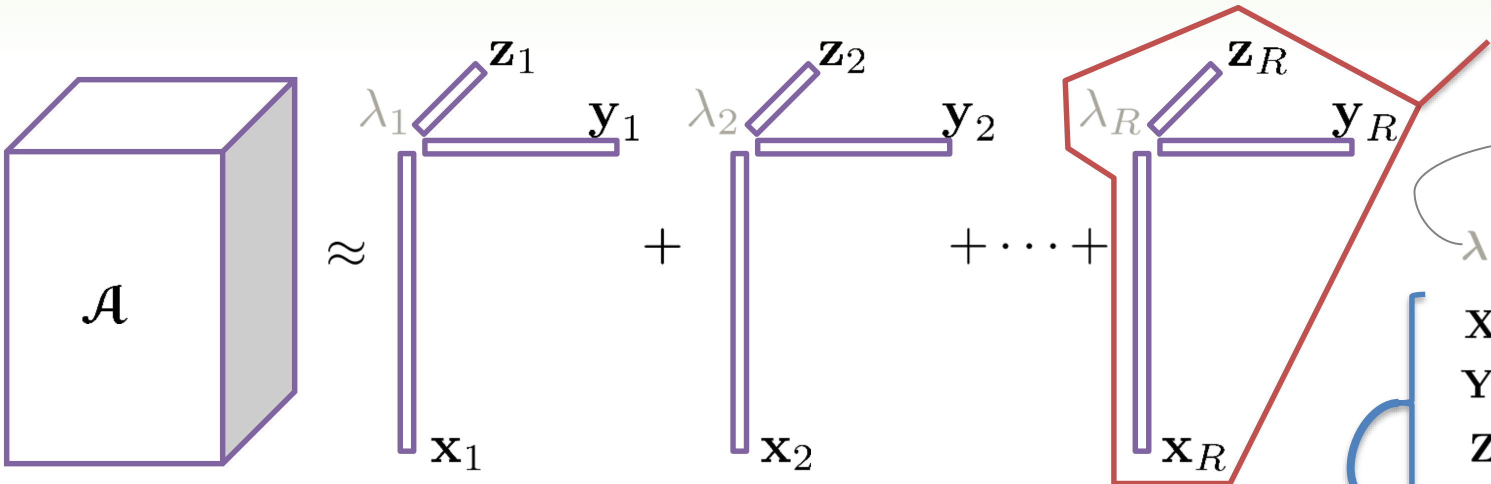


Fitting the CP Decomposition

CP Tensor Decomposition: Sum of Outer Products

Sandia National Laboratories

CANDECOMP/PARAFAC or canonical polyadic (CP) Model



Optional weights

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \cdots & \lambda_R \end{bmatrix}^{\mathsf{T}}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_R \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \cdots & \mathbf{y}_R \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_R \end{bmatrix}$$

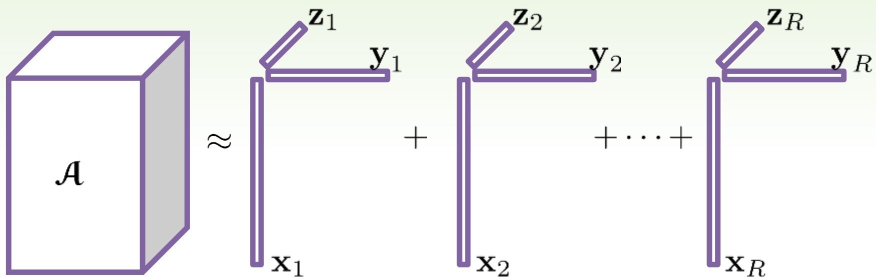
Factor Matrices

Model: $\mathcal{M} = \sum_r \lambda_r \, \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$

$$\boxed{\min_{\mathcal{M}} \sum_{ijk} (a_{ijk} - m_{ijk})^2 \quad \text{subject to} \quad m_{ijk} = \sum_r \lambda_r \, x_{ir} \, y_{jr} \, z_{kr}}$$

Key references: Hitchcock, 1927; Harshman, 1970; Carroll and Chang, 1970

CP-ALS: Fitting CP Model via Alternating Least Squares



Convex (linear least squares) subproblems can be solved exactly

+

Structure makes easy inversion

$$f(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \sum_{ijk}\Big(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr}\Big)^2$$

Repeat until convergence:

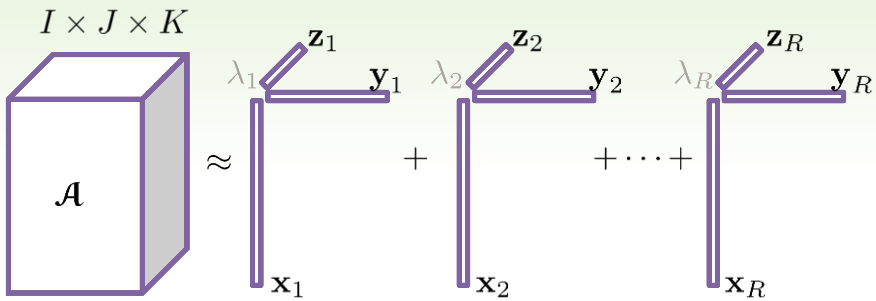
Step 1: $\min_{\mathbf{X}} \sum_{ijk}\Big(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr}\Big)^2$

Step 2: $\min_{\mathbf{Y}} \sum_{ijk}\Big(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr}\Big)^2$

Step 3: $\min_{\mathbf{Z}} \sum_{ijk}\Big(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr}\Big)^2$

Harshman, 1970; Carroll & Chang, 1970

Challenges for Fitting CP Model



variables = $R(I + J + K)$
data points = IJK

Rank = minimal R to exactly reproduce tensor

- **Nonconvex:** Polynomial optimization problem ⇒ *Initialization matters*
- **Permutation and scaling ambiguities:** Can reorder the r's and arbitrarily scale vectors within each component so long as the product of the scaling is 1 ⇒ *May need regularization*
- **Rank unknown:** Determining the "rank" R that yields exact fit is NP-hard (Håstad 1990, Hillar & Lim 2009) ⇒ *No easy solution, need to try many*
- **Low-rank?** Best "low-rank" factorization may not exist (Silva & Lim 2006) ⇒ *Need bounds on components* $\|\lambda_r \, \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r\| = |\lambda_r| \, \|\mathbf{x}_r\| \, \|\mathbf{y}_r\| \, \|\mathbf{z}_r\|$
- **Not nested:** Best rank-(R-1) factorization may not be part of best rank-R factorization (Kolda 2001) ⇒ *Cannot use greedy algorithm*

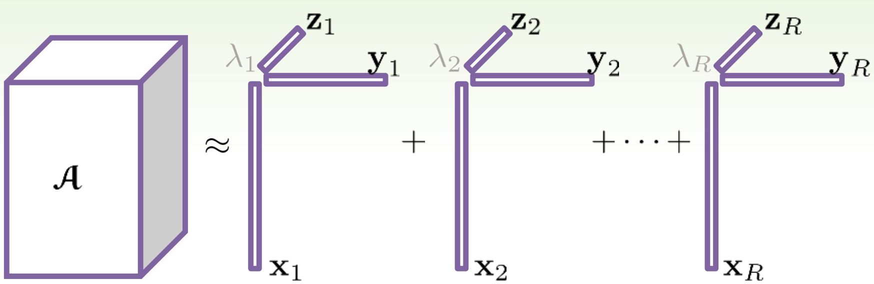
Example 9 x 9 x 9 Tensor of Unknown Rank

- 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 $\Rightarrow \leq$ 621 variables

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

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

Opportunities for Fitting CP Model



k-rank(**X**) = maximum value k such that *any* k columns of **X** are linearly independent

- Factorization is **essentially unique** (i.e., up to permutation and scaling) under the condition the the sum of the factor matrix k-rank values is $\geq$ 2R + d – 1 (Kruskal 1977)

$$\text{k-rank}(\mathbf{X}) + \text{k-rank}(\mathbf{Y}) + \text{k-rank}(\mathbf{Z}) \geq 2R + 2$$

- If $R \ll I,J,K$, then can use **compression** to reduce dimensionality before solving CP model (CANDELINC: Carroll, Pruzansky, and Kruskal 1980)
- Efficient **sparse kernels** exist (Bader & Kolda, SISC 2007)

A Randomized Method for Fitting the CP Decomposition

Featuring work of Casey Battaglino, Grey Ballard

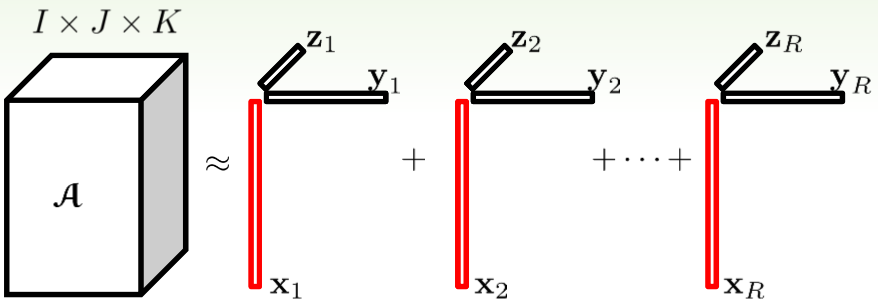

Rewriting CP-ALS Subproblem as Matrix Least Squares

Repeat until convergence:

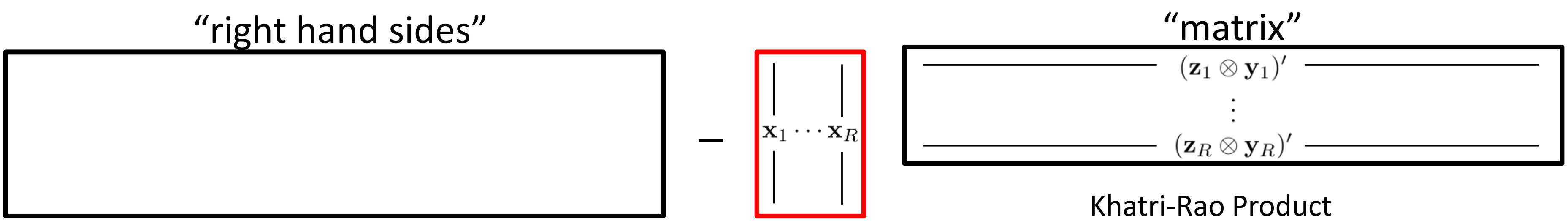
Step 1: $\min_{\mathbf{X}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir} \, y_{jr} \, z_{kr} \right)^2$

Step 2: $\min_{\mathbf{Y}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir} \, y_{jr} \, z_{kr} \right)^2$

Step 3: $\min_{\mathbf{Z}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir} \, y_{jr} \, z_{kr} \right)^2$



Rewrite as matrix least squares problem
(backwards and transposed matrix version!)



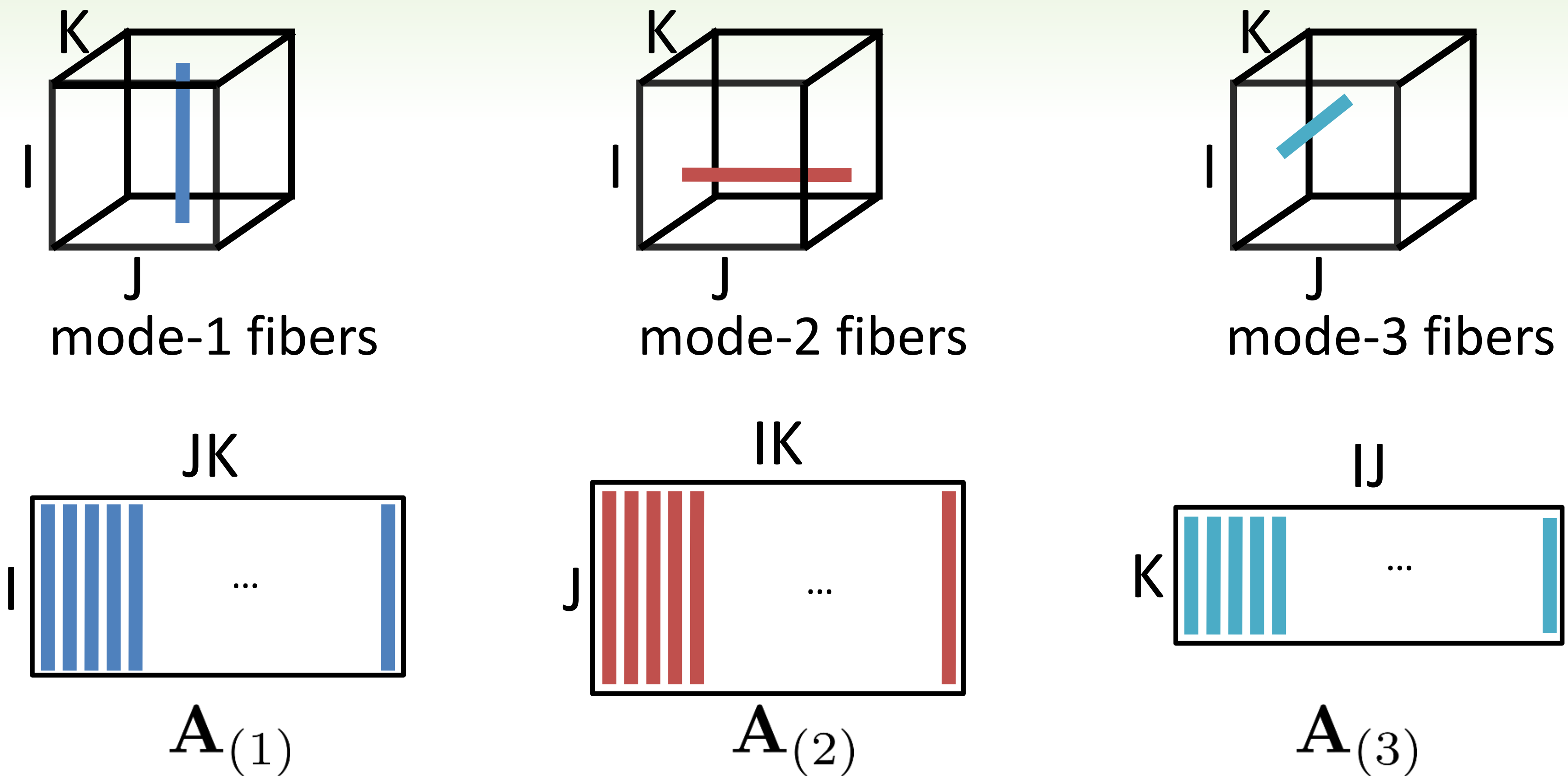
Khatri-Rao Product

$$\| \mathbf{A}_{(1)} - \mathbf{X} (\mathbf{Z} \odot \mathbf{Y})' \|_F^2$$

$I \times JK$ $\quad$ $I \times R$ $\quad$ $R \times JK$

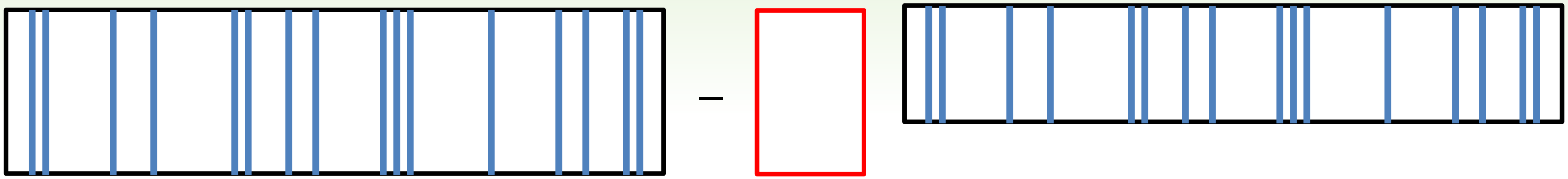
Coming soon: Battaglino, Ballard, Kolda, 2016

Matricization/Unfolding: Rearranging a Tensor as a Matrix

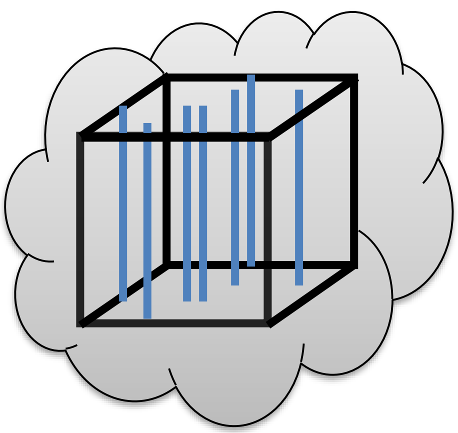


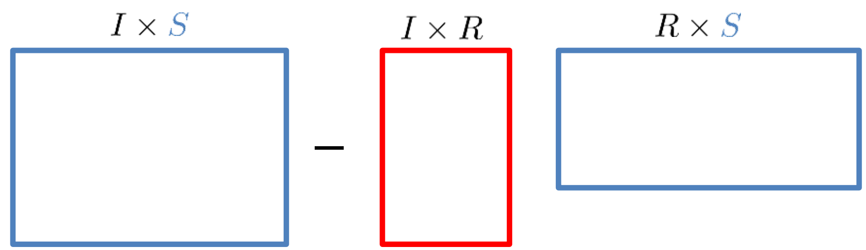
Mathematically convenient expression, but generally do not want to actually form the unfolded matrix due to the expense of the *data movement*.

Randomizing the Matrix Least Squares Subproblem



$$\| \mathbf{A}_{(1)} \mathbf{S} - \mathbf{X} \, [(\mathbf{Z} \odot \mathbf{Y})' \mathbf{S}] \|_F^2$$





Each column in the sample is of the form: (Z(:,k) .* Y(:,j))'

Two "tricks"

1. Never permute elements of tensor A intro $I \times JK$ matrix form
2. Never form full Khatri-Rao product of size $R \times JK$

Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

Mixing for Incoherence: One Time & Done

Need to apply an fast Johnson-Lindenstrauss Transform (FJLT) to $(\mathbf{Z} \odot \boldsymbol{Y})'$ to ensure *incoherence:*

$$\|\mathbf{A}_{(1)}\mathbf{S} - \mathbf{X}(\mathbf{Z} \odot \mathbf{Y})'\mathbf{S}\|_F^2 \Longrightarrow \|\mathbf{A}_{(1)}(\mathbf{DF})\mathbf{S} - \mathbf{X}(\mathbf{Z} \odot \mathbf{Y})'\underbrace{(\mathbf{DF})\mathbf{S}}_{\text{Diagonal Random Sign Flip \& FFT}}\|_F^2$$

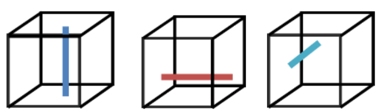
We use a Kronecker product of FJLTs and conjecture it's an FJLT:

$$\|\mathbf{A}_{(1)}(\mathbf{F}_K\mathbf{D}_K \otimes \mathbf{F}_J\mathbf{D}_J)'\mathbf{S} - \mathbf{X}(\mathbf{Z} \odot \mathbf{Y})'(\mathbf{F}_K\mathbf{D}_K \otimes \mathbf{F}_J\mathbf{D}_J)'\mathbf{S}\|_F^2$$

$$\|\mathbf{A}_{(1)}(\mathbf{F}_K\mathbf{D}_K \otimes \mathbf{F}_J\mathbf{D}_J)'\mathbf{S} - \mathbf{X}(\mathbf{F}_K\mathbf{D}_K\mathbf{Z} \odot \mathbf{F}_J\mathbf{D}_J\mathbf{Y})'\mathbf{S}\|_F^2$$

We ultimately transform the tensor <u>one time</u> in all modes and just apply the reverse transform in one mode.

$\mathbf{F}_I\mathbf{D}_I$ $\mathbf{F}_J\mathbf{D}_J$ $\mathbf{F}_K\mathbf{D}_K$



Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

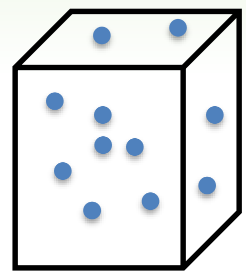
Randomizing the Convergence Check

Repeat until convergence:

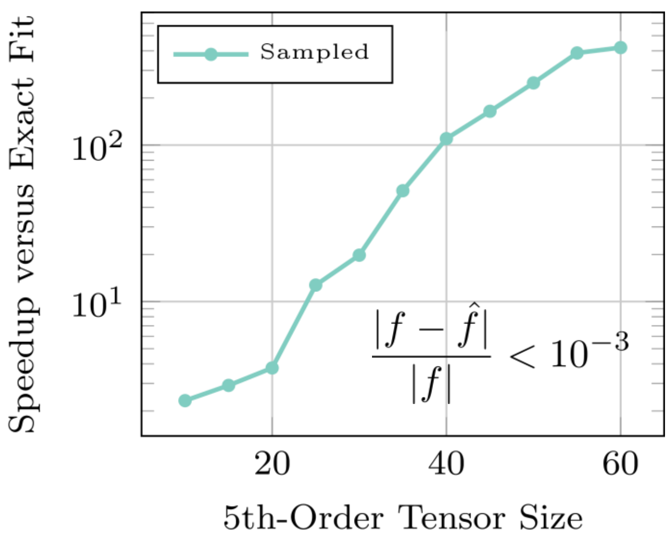
Step 1: $\min_{\mathbf{X}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr} \right)^2$

Step 2: $\min_{\mathbf{Y}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr} \right)^2$

Step 3: $\min_{\mathbf{Z}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr} \right)^2$



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

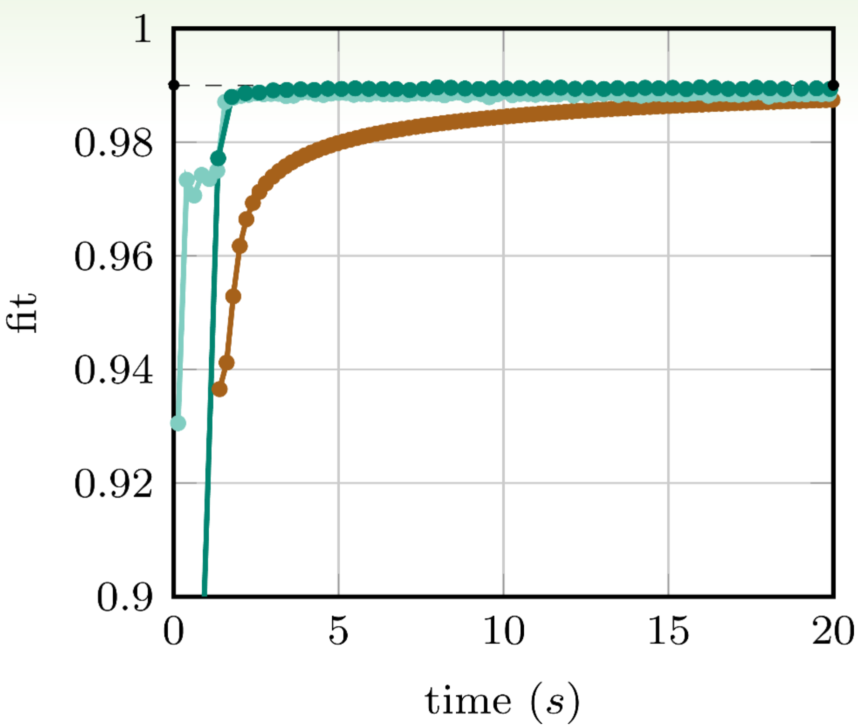


$$f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr} \right)^2$$

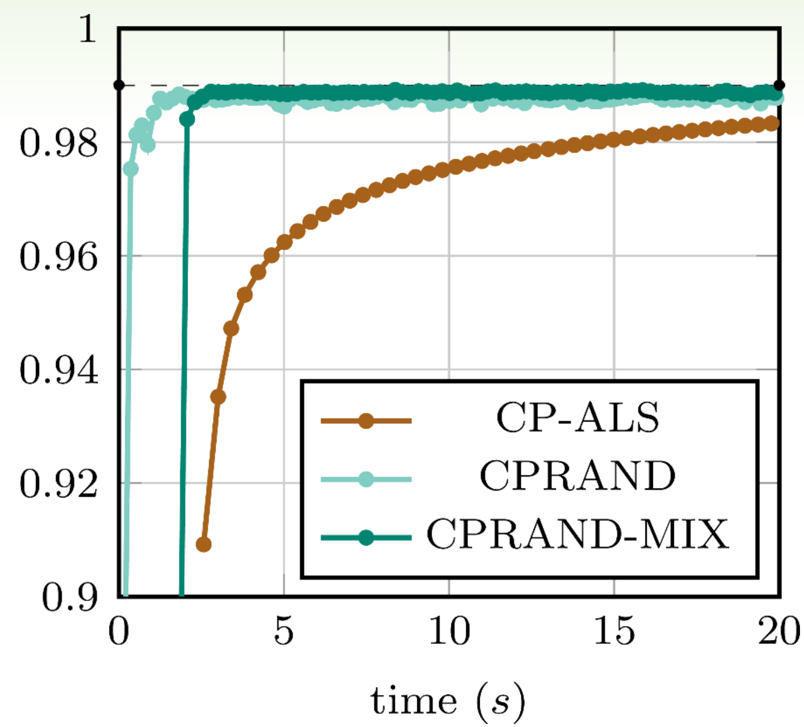
$$\hat{f}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \approx \frac{IJK}{|\Omega|} \sum_{ijk \in \Omega} \left(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr} \right)^2$$

Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

CPRAND faster than CP-ALS



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

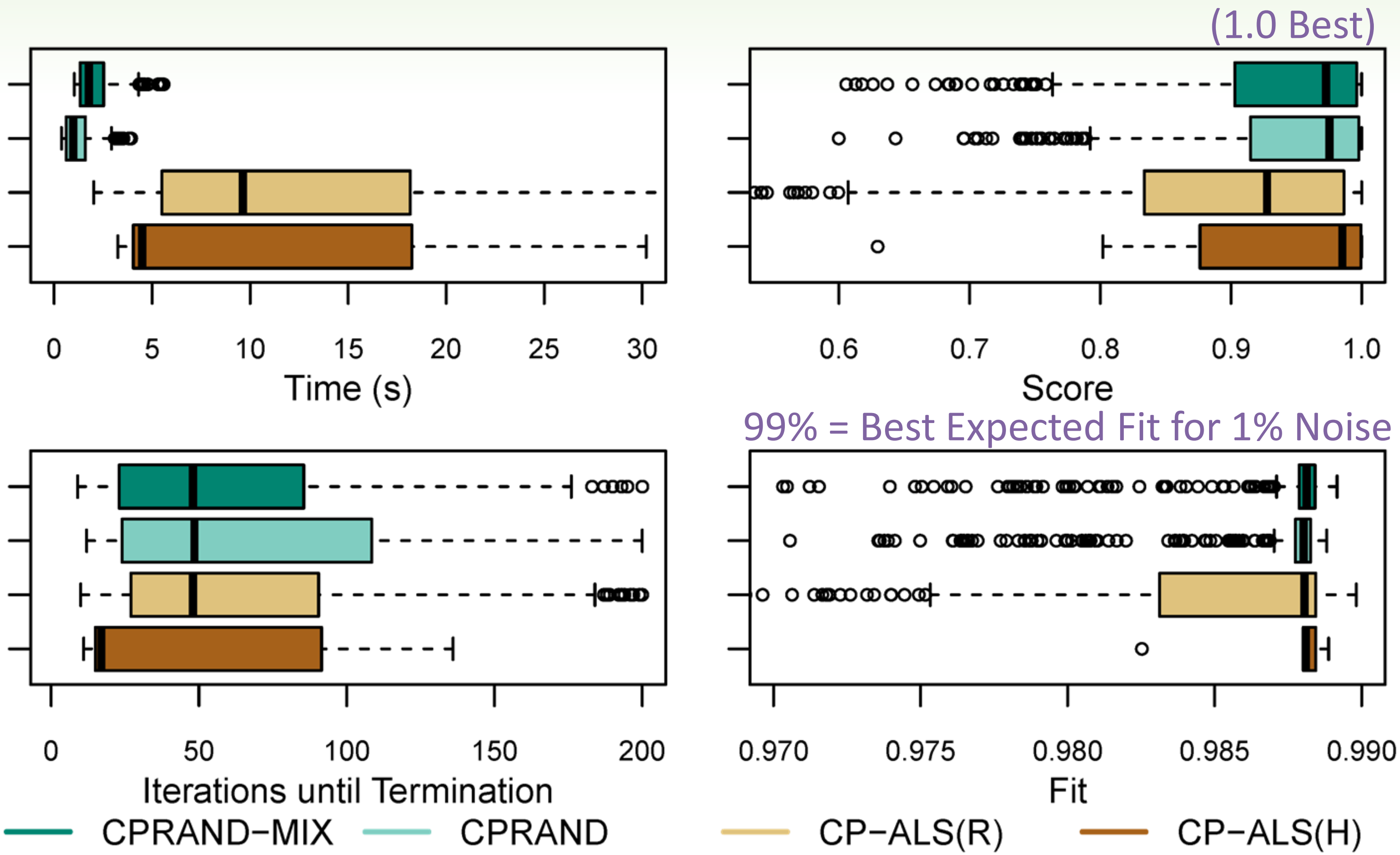


80 x 80 x 80 x 80 Random Rank-5 Tensor
with 1% Noise

Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

CPRAND Faster than CP-ALS in Low Noise Conditions

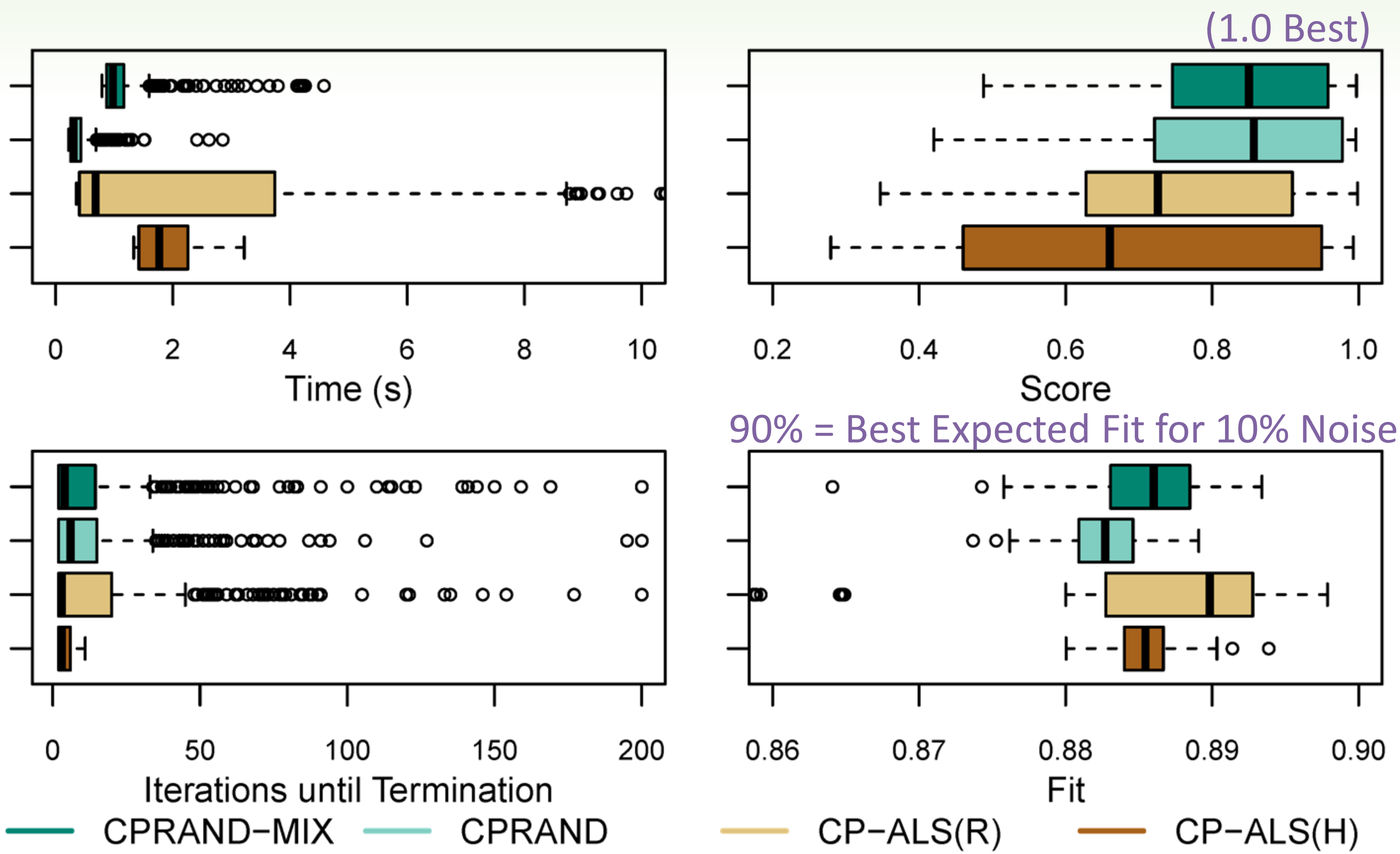
200 Tensors: 400 x 400 x 400, R = 5 or 6, Noise = 1%



Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

CPRAND Faster than CP-ALS in Low Noise Conditions

200 Tensors: 400 x 400 x 400, R = 5 or 6, Noise = 10%

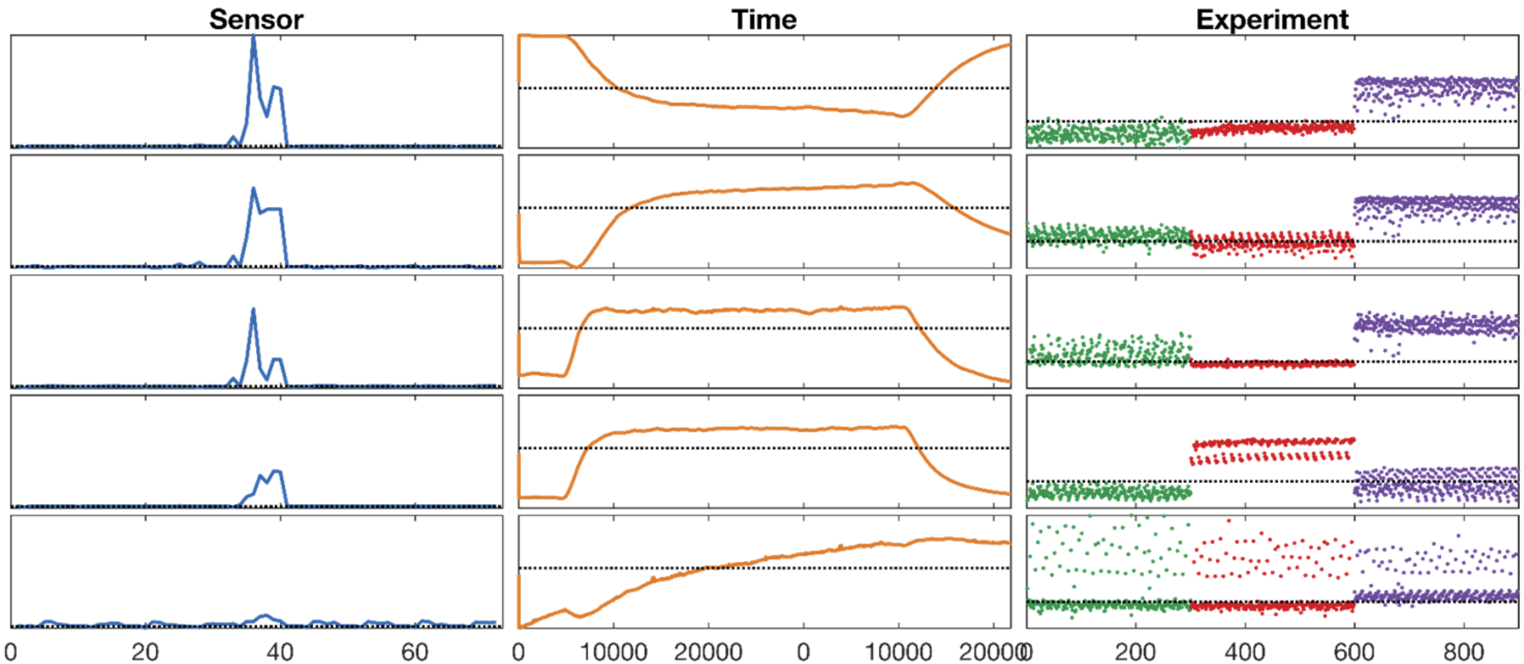


Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

Analysis of Hazardous Gas Data

900 experiments (with three different gas types) x 72 sensors x 25,900 time steps (13 GB)

Method	Median Time (s)	Median Fit	Median Classification Error
CPRAND	53.6	0.715	0.61%
CP-ALS (H)	578.4	0.724	0.67%
CP-ALS (R)	204.7	0.724	0.67%

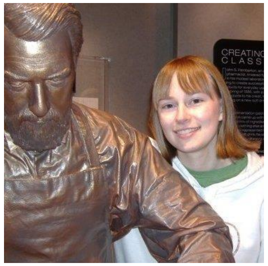


This mode scaled by component size

Color-coded by gas type

Data from Vergara et al. 2013; see also Vervliet and De Lathauwer (2016)

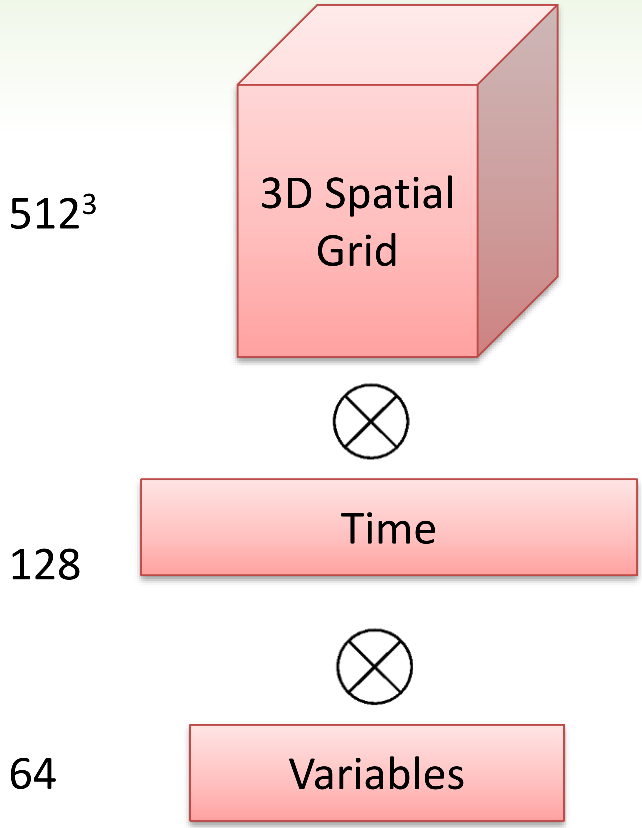
Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

Parallel Tucker Decomposition for Data Compression

Featuring work of Woody Austin, Grey Ballard, Alicia Klinvex, Hemanth Kolla

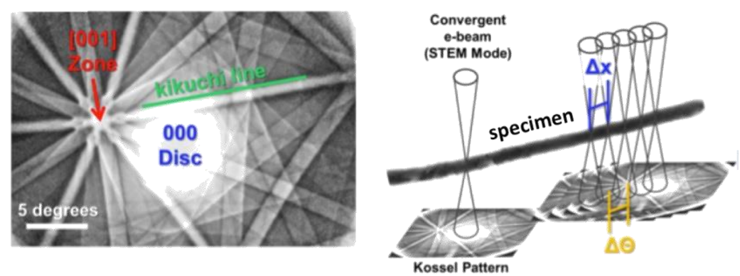
DOE Advanced Simulations and Experiments Deluged by Data



2^{40} elements

8TB (double precision)

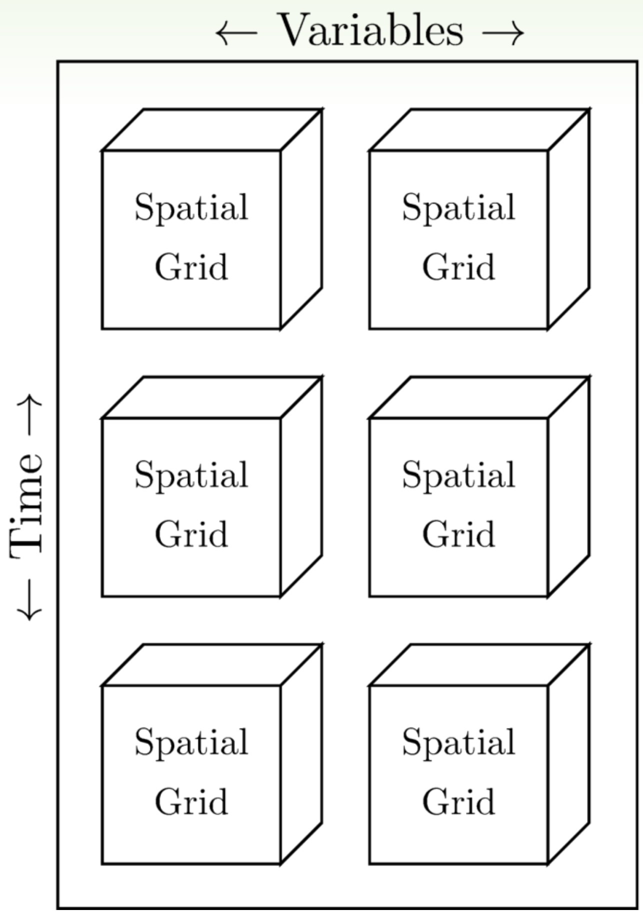
- Combustion simulations
  - S3D used direct numerical simulation
  - Gold standard for comparisons, but…
  - Single experiment produces terabytes of data
  - Storage limits spatial, temporal resolutions
  - Difficult to analyze or transfer data
- Electron microscopy
  - New technology produces 1-D spectra and 2-D diffraction patterns *at each pixel*
  - Single experiment produces terabytes of data
  - Usually 10-100 experiments per
  - Limited hardware utilization due to storage limits



- Other applications
  - Telemetry
  - Cosmology Simulations (with LBL)
  - Climate Modeling (with ORNL)

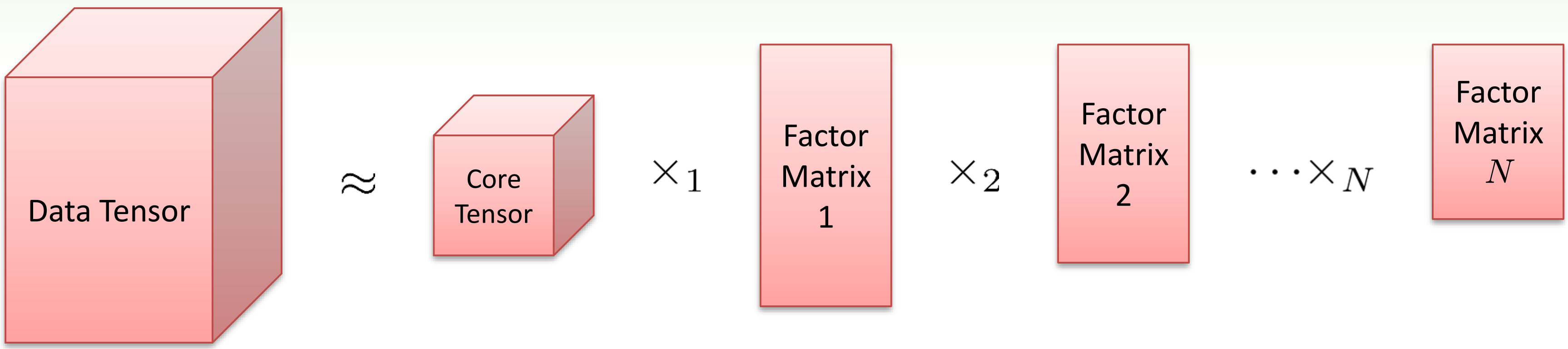
Goal: Reduce Storage Bottleneck via Multiway Compression

- Typical compression methods includes
  - Bzip
  - Wavelets
  - Interpolation
- Standard methods generally ignore multidimensional structure
  - Most science data sets have natural multidimensional structure
- Propose to use multiway compression
  - Always as good as and generally exceeds compression of lossy two-way methods
  - Initial experiments show 90-99% reductions
- Many scientific data sets have natural multiway (N-way) structure
  - 3-D spatial grid
  - Temporal data
  - Multiple measurements
- Parallel compression required to handle large data sets!



Reduction based on Tucker Tensor Decomposition

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}$$



$I_1 \times I_2 \times \cdots \times I_N$ $\quad R_1 \times R_2 \times \cdots \times R_N$ $\quad I_1 \times R_1$ $\quad I_2 \times R_2$ $\quad I_N \times R_N$

Storage: $\prod_{n=1}^{N} I_n$

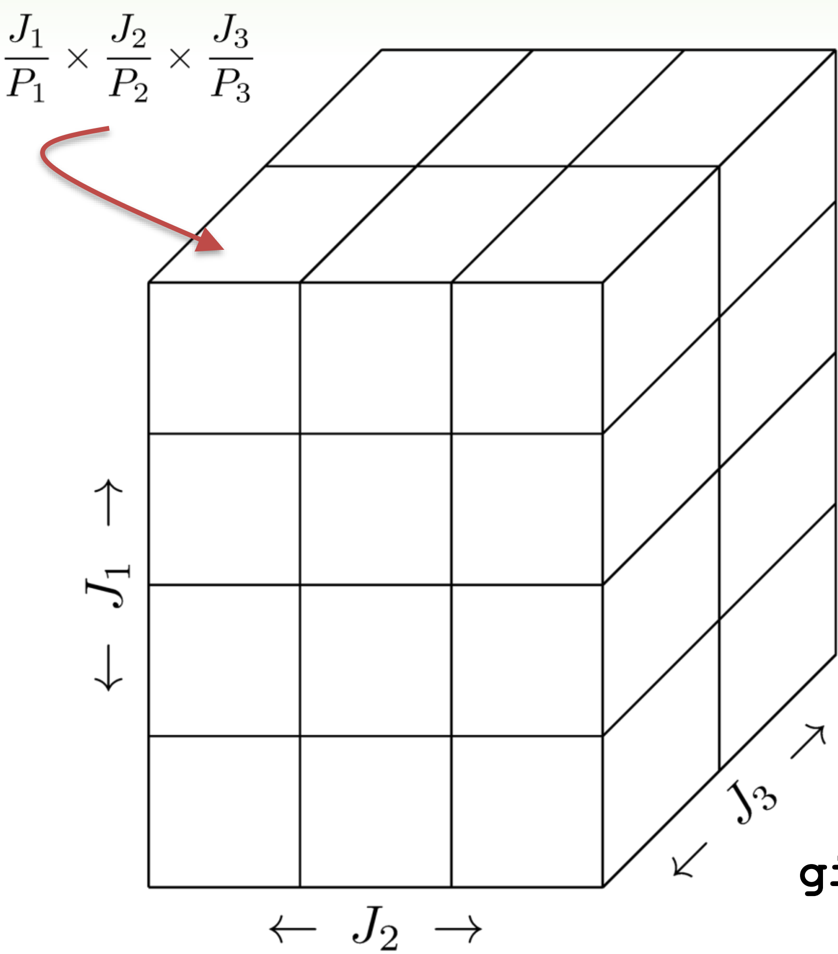
Storage: $\prod_{n=1}^{N} R_n + \sum_{n=1}^{N} R_n I_n$

History:

- Hitchcock (1927) – Mathematical definition
- Tucker (1966) – Algorithms and applications for 3-way data
- Kapteyn, Neudecker, Wansbeek (1986) – Algorithms for N-way data
- Vannieuwenhoven, Vandebril, and Meerbergen (2012) – Sequentially-truncated algorithm

New Contribution: Parallel Tucker Decomposition

Processor Grid: $P_1 \times P_2 \times P_3$



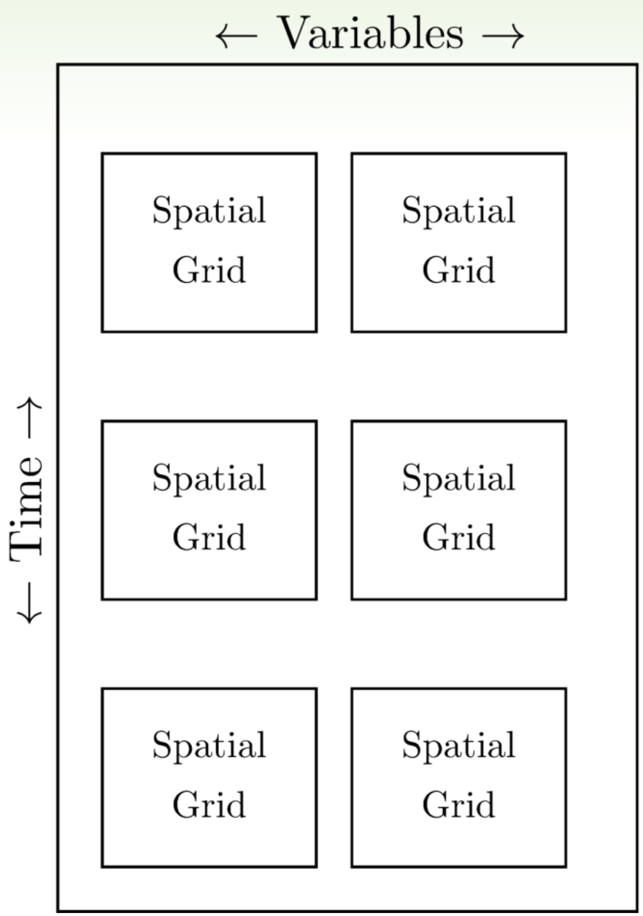
- Uses Cartesian block distribution
- Main parallel kernels
  - Gram matrix
  - Tensor-times-matrix
- Exploits clever local and global data layout
  - Communication avoiding
- Details in IPDPS'16 paper
- Code available (soon) at GitLab

TUCKERMPI

git@gitlab.com:tensors/TuckerMPI.git

Alicia Klinvex, Woody Austin, Grey Ballard

Combustion Simulation Results



4-way Tensor, 672 x 672 x 32 x 626
Storage: 74GB

Reduced size determined automatically to satisfy error criteria:

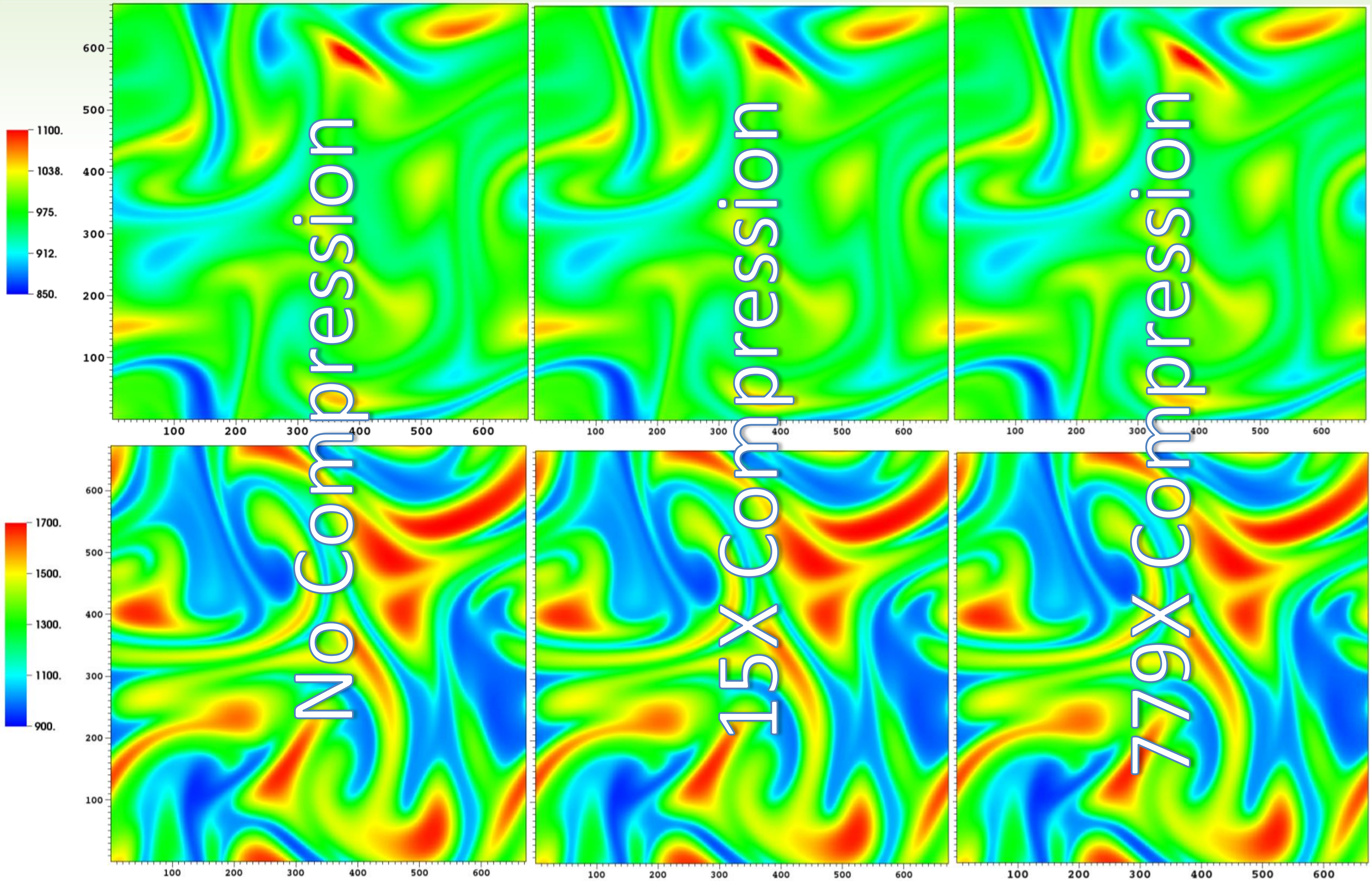
$$\|\mathcal{X} - \hat{\mathcal{X}}\| \leq \epsilon \|\mathcal{X}\|$$

Compression ratio:

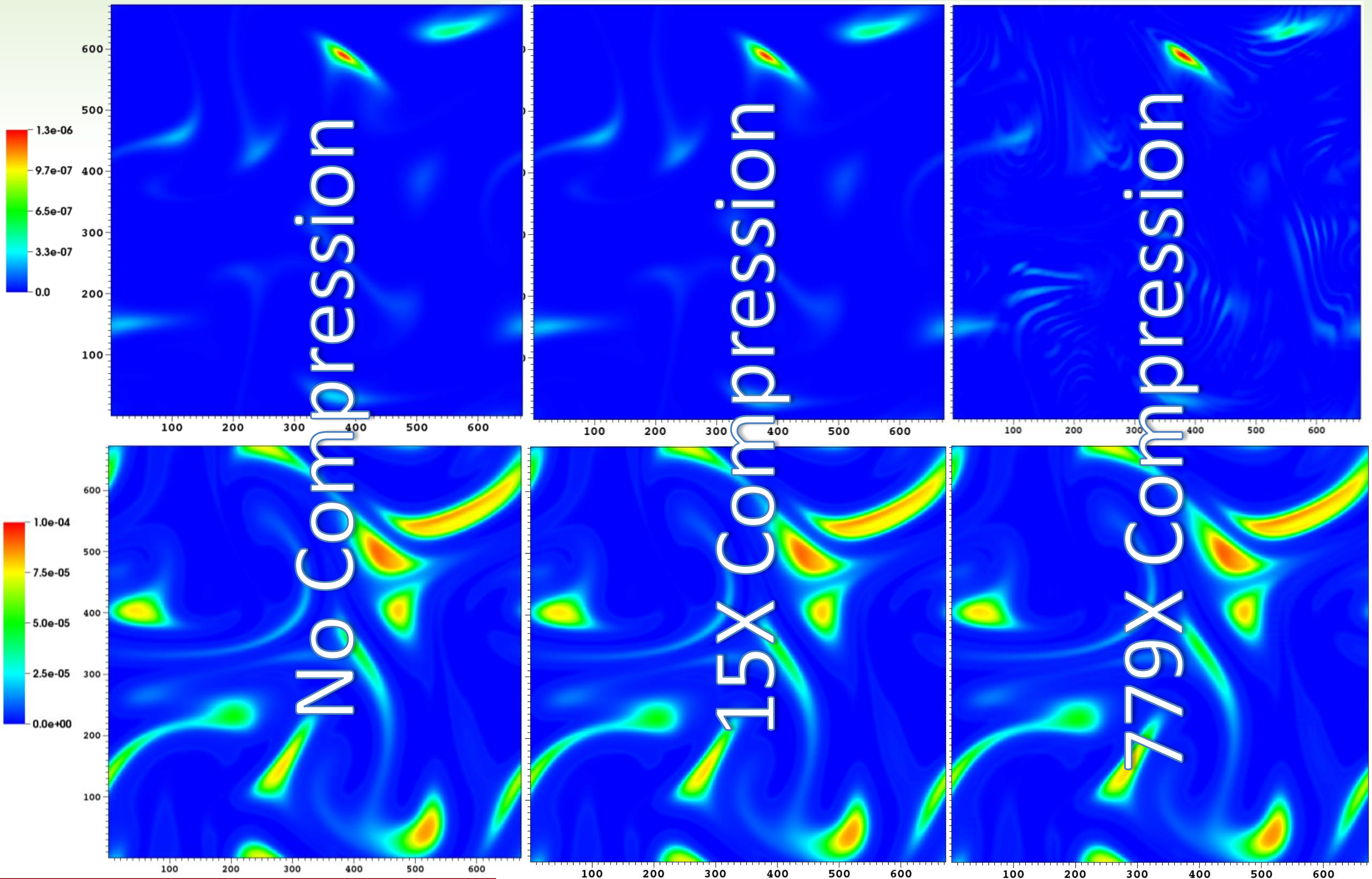
$$C = \prod_{k=1}^{N} I_n \Big/ \left(\prod_{k=1}^{N} R_n + \sum_{k=1}^{N} I_n R_n \right)$$

- Combustion simulation: HCCI Ethanol
  - 2D Spatial Grid: 672 x 672
  - Species/Variables: 32
  - Time steps: 626
- *Scaled each species by its max value*
- Experiment 1: $\epsilon = 10^{-2}$
  - New Core: 111 x 105 x 22 x 46 (<100MB)
  - Compression: 779X
- Experiment 2: $\epsilon = 10^{-4}$
  - New Core: 330 x 310 x 31 x 199 (640MB)
  - Compression: 15X

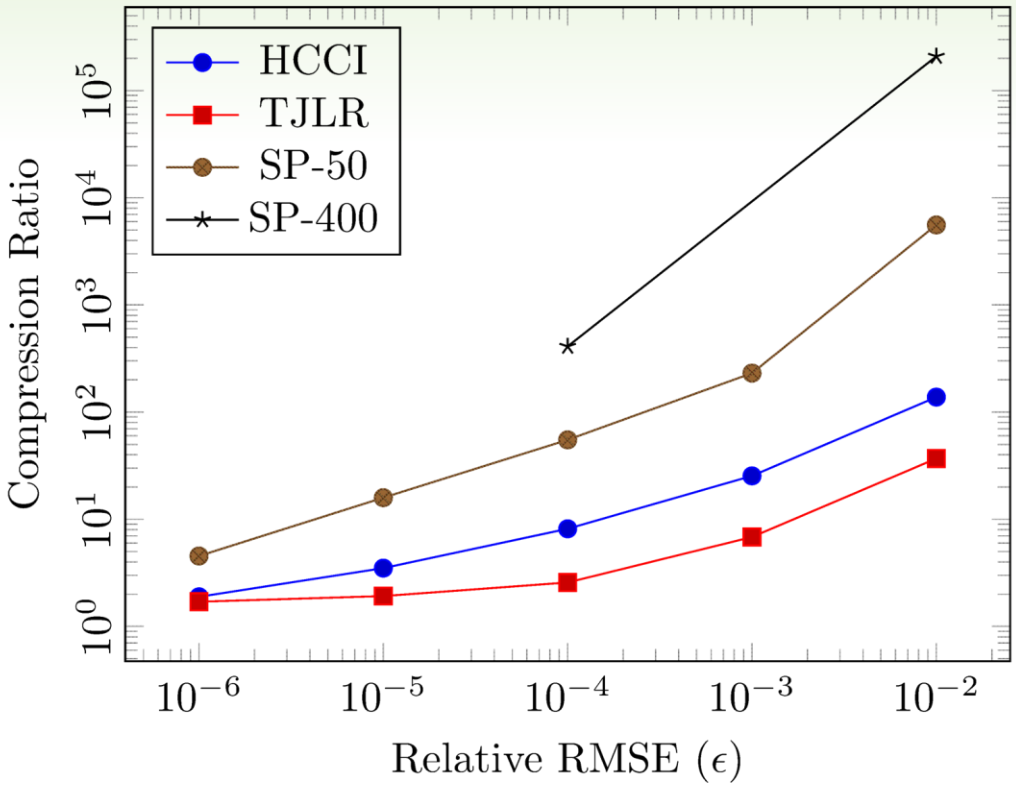
Combustion Simulation Results: Temperature @ Disparate Timesteps



Combustion Simulation Results: OH @ Disparate Timesteps



Compression on Different Data Sets



$$\|\mathcal{X} - \hat{\mathcal{X}}\| \leq \epsilon \|\mathcal{X}\|$$

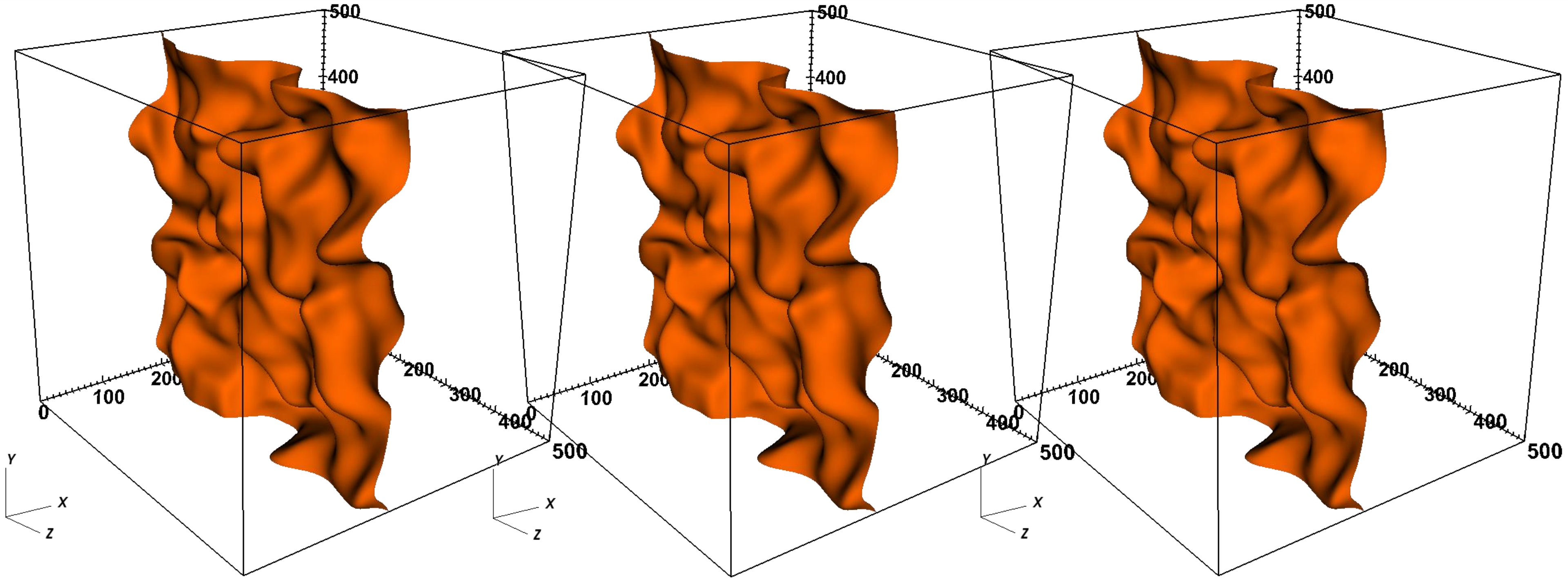
$$C = \prod_{k=1}^{d} N_k \Big/ \left(\prod_{k=1}^{d} R_k + \sum_{k=1}^{d} N_k R_k \right)$$

- HCCI: 672 x 672 x 33 x 627
- TJLR: 460 x 700 x 360 x 35 x 16
- SP-50: 500 x 500 x 500 x 11 x 50
- SP-400: 500 x 500 x 500 x 11 x 400

New results just in on 4.4 TB tensor for SP of size 500 x 500 x 500 x 11 x 400.
Achieved 410X compression at 1e-4 RMSE, reducing to just 10 GB!
Run time is 58 seconds on ORNL's Rhea platform, using 1104 processes.

SP-400 4.4TB Dataset: Can you guess which is the original?

Original | ϵ = 1e-4 | ϵ = 1e-2

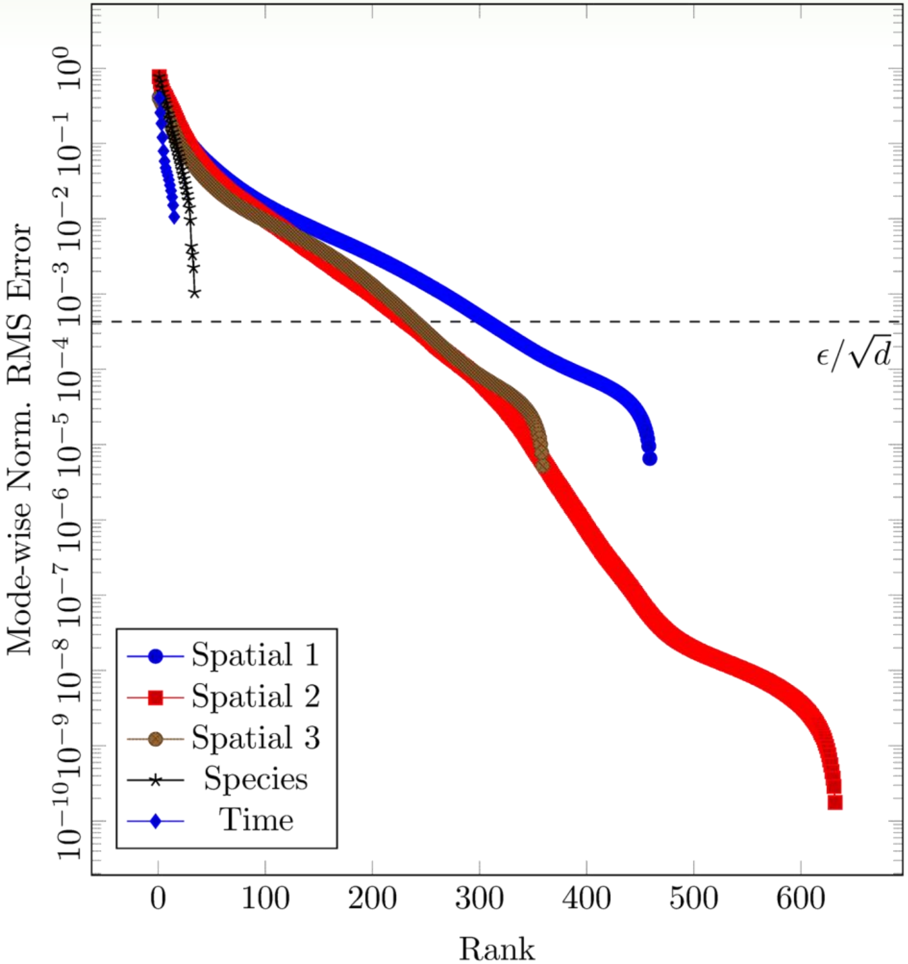


400X Compression | 20000X Compression

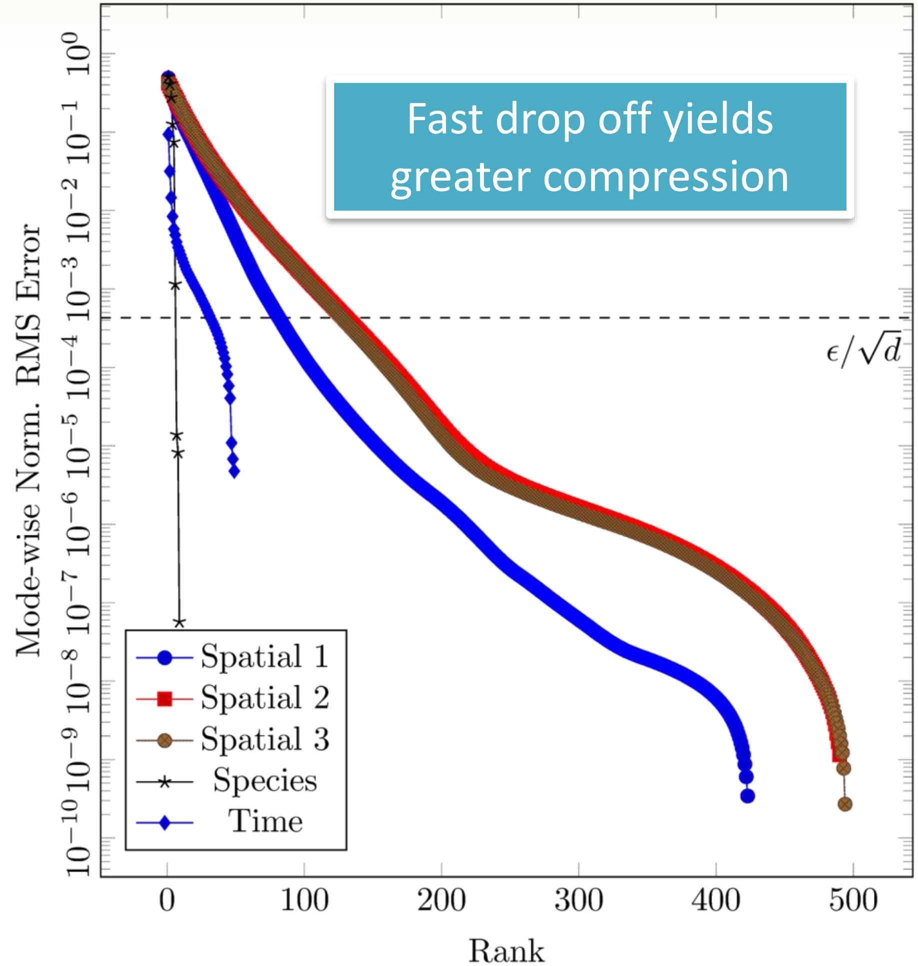
Flame surface at single time step. Using temperature variable (iso-value is 2/3 of max).

Compression Depends on Data Redundancy (Eigenvalues Don't Lie)

TJLR: 460 x 700 x 360 x 35 x 16



SP-50: 500 x 500 x 500 x 11 x 50



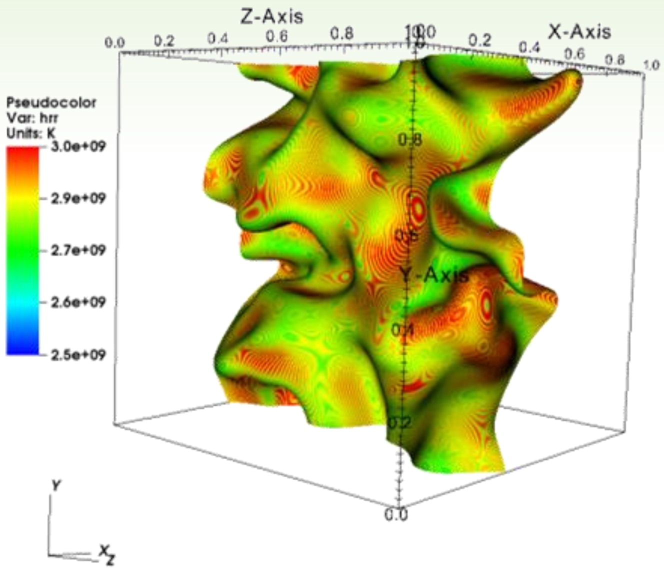
Fast drop off yields greater compression

Key Feature: Need Only Do Partial Reconstruction on Laptops, etc.

Reconstruction requires as much space as the original data!

$$\hat{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)} \times_5 \mathbf{U}^{(5)}$$

$N_1 \times N_2 \times N_3 \times N_4 \times N_5$



But we can just reconstruct the portion that we need at the moment:

$$\bar{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{C}^{(3)}\mathbf{U}^{(3)} \times_4 \mathbf{C}^{(4)}\mathbf{U}^{(4)} \times_5 \mathbf{C}^{(5)}\mathbf{U}^{(5)}$$

$N_1 \times N_2 \times \frac{N_3}{2} \times 1 \times 1$

$$\mathbf{C}^{(3)} = \begin{bmatrix} 1/2 & 0 & \cdots & 0 \\ 1/2 & 0 & \cdots & 0 \\ 0 & 1/2 & \cdots & 0 \\ 0 & 1/2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \quad \mathbf{C}^{(4)} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{bmatrix} \quad \mathbf{C}^{(5)} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

Downsample

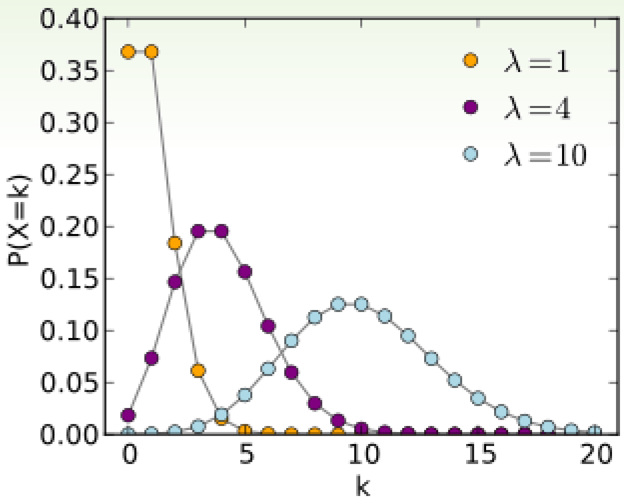
Pick single variable

Pick single time step

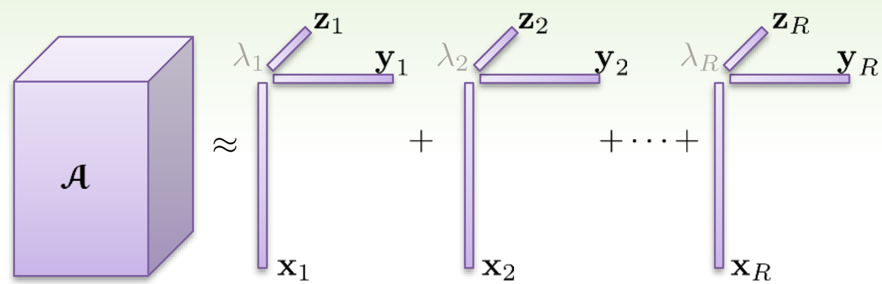
A New Type of CP Tensor Decomposition for Count Data

Featuring work of Eric Chi

New “Stable” Approach: Poisson Tensor Factorization (PTF)



$$P(X=x) = \frac{\exp(-\lambda)\lambda^x}{x!}$$



$$m_{ijk} = \sum_r \lambda_r \, x_{ir} \, y_{jr} \, z_{kr}$$

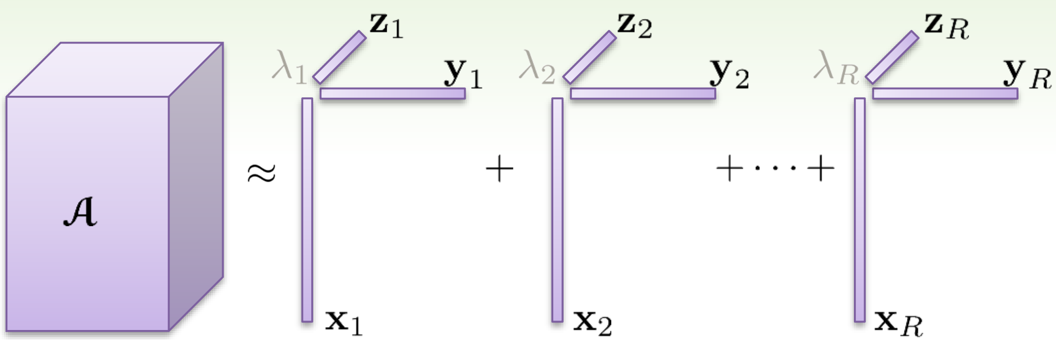
$$\boxed{a_{ijk} \sim \text{Poisson}(m_{ijk})}$$

Maximize this: $$\text{likelihood}(\mathcal{M}) = \prod_{ijk} \frac{\exp(-m_{ijk}) \, m_{ijk}^{a_{ijk}}}{a_{ijk}!}$$

By monotonicity of log, same as maximizing this: $$\text{log-likelihood}(\mathcal{M}) = c - \sum_{ijk} m_{ijk} - a_{ijk} \log(m_{ijk})$$

This objective function is also known as Kullback-Liebler (KL) divergence.
The factorization is automatically nonnegative.

Solving the Poisson Regression Problem



$$\mathcal{M} = \sum_r \lambda_r \, \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$$

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \cdots & \lambda_R \end{bmatrix}^{\mathsf{T}}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_R \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \cdots & \mathbf{y}_R \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_R \end{bmatrix}$$

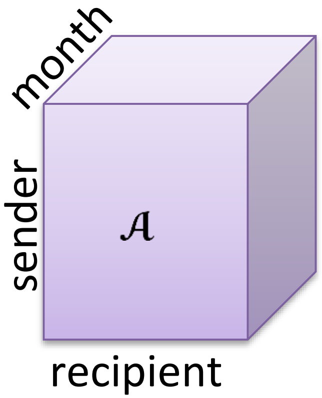
$$\min_{\mathcal{M}} \sum_{ijk} m_{ijk} - a_{ijk} \log m_{ijk} \quad \text{subject to} \quad m_{ijk} = \sum_r \lambda_r \, x_{ir} \, y_{jr} \, z_{kr}$$

- 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 A is sparse and have special methods for this
  - Newton or Quasi-Newton method

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

PTF for Time-Evolving Social Network

Enron email data from FERC investigation.



Data	8540 Email Messages
# Months	28 (Dec'99 – Mar'02)
# Senders/Recipients	108 (>10 messages each)
Links	8500 (3% dense)

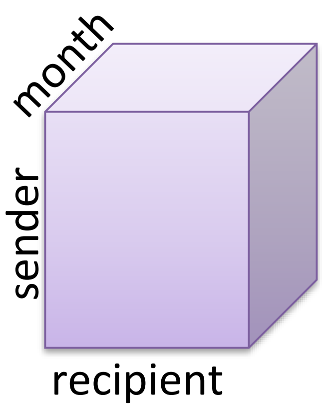
a_{ijk} = # emails from sender i to recipient j in month k

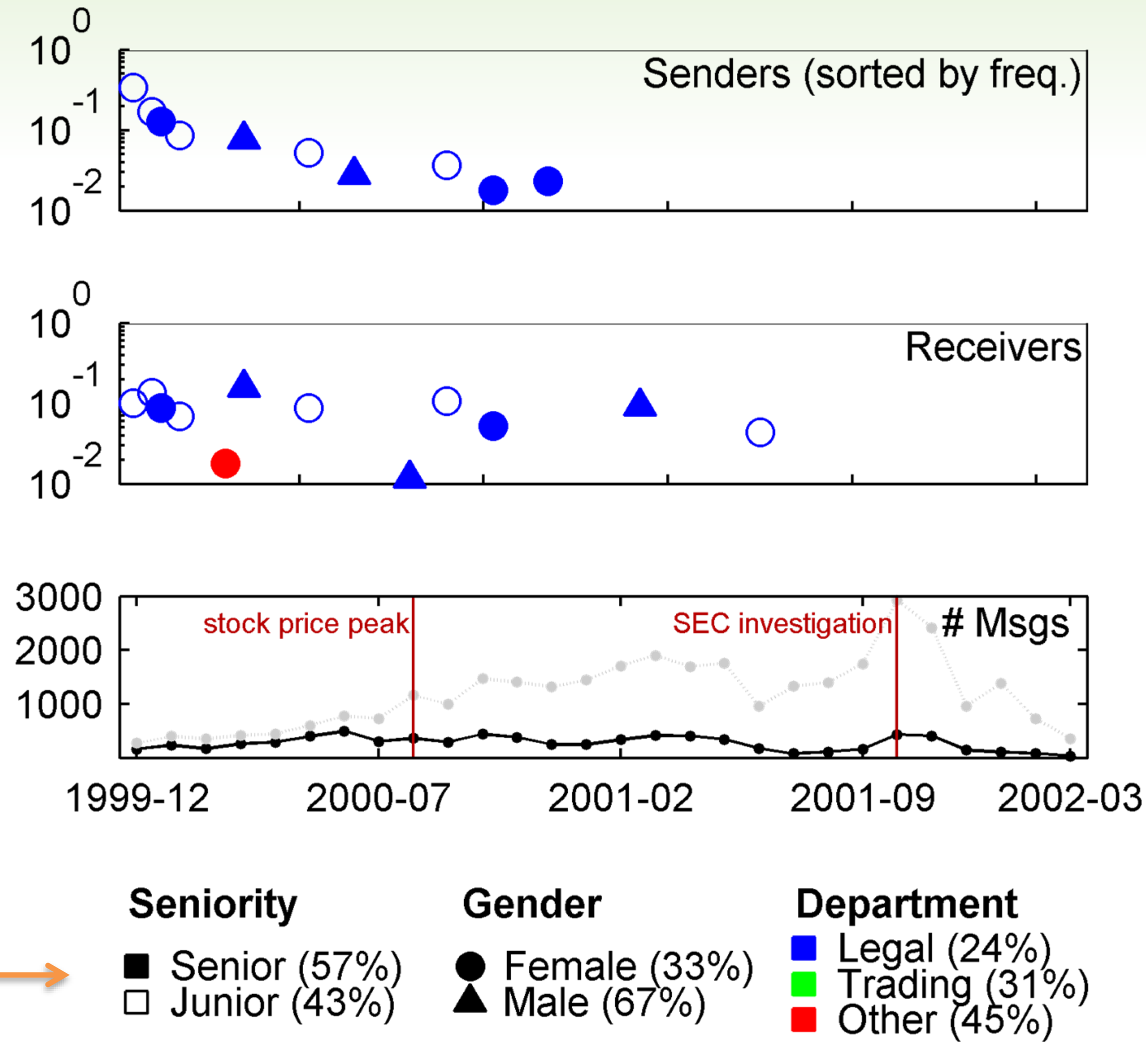
Let's look at some components from a 10-component (R=10) factorization, sorted by size…

Chi & Kolda, On Tensors, Sparsity, and Nonnegative Factorizations, SIMAX 2012

Enron Email Data (Component 1)

Legal Dept;
Mostly Female



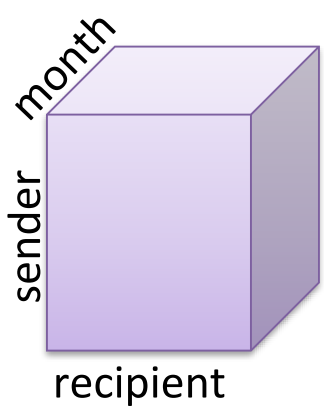


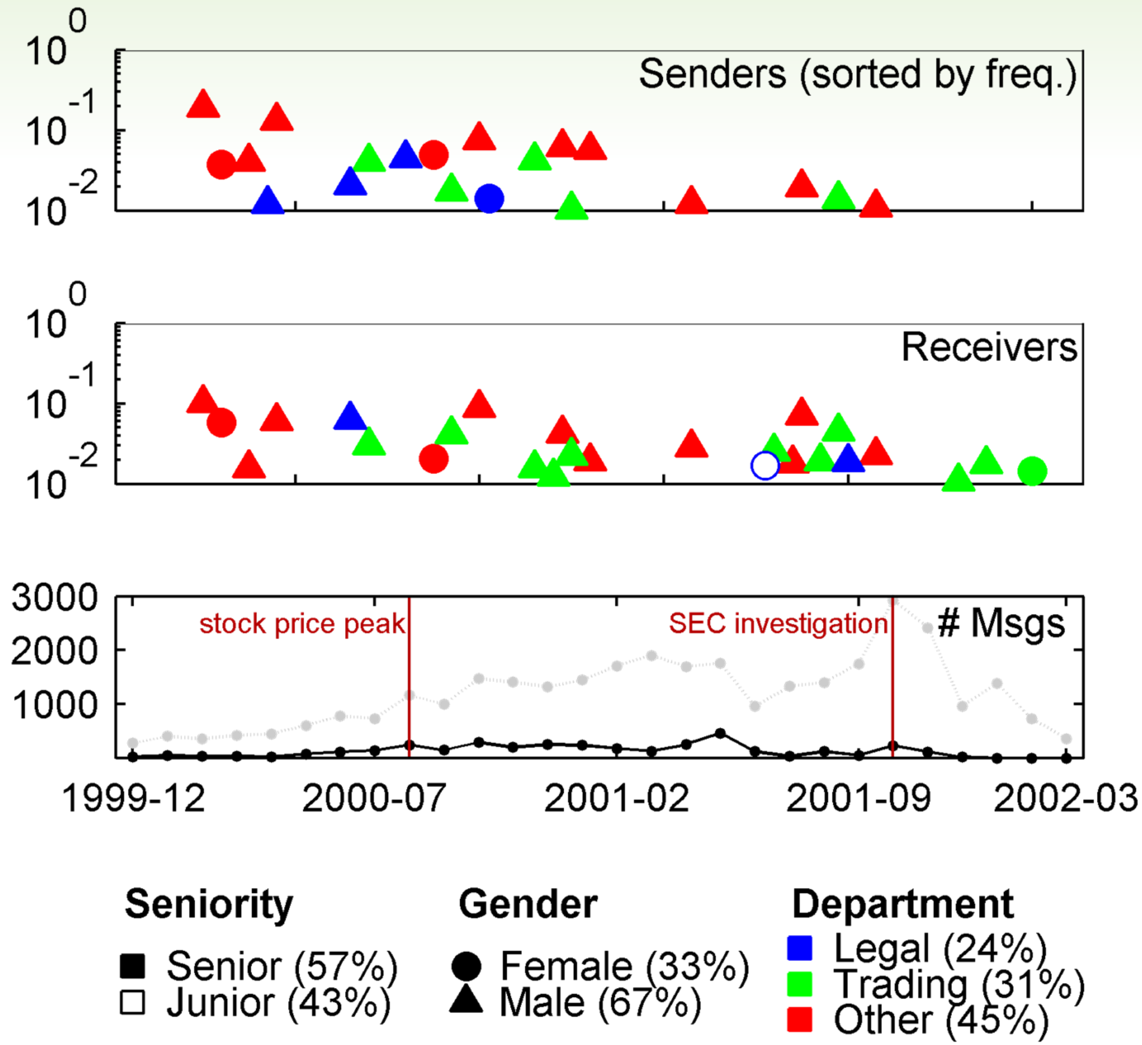
Each person labeled by Zhou et al. (2007)

Chi & Kolda, On Tensors, Sparsity, and Nonnegative Factorizations, SIMAX 2012

Enron Email Data (Component 3)

Senior;
Mostly Male

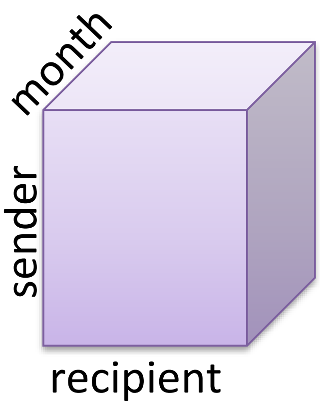


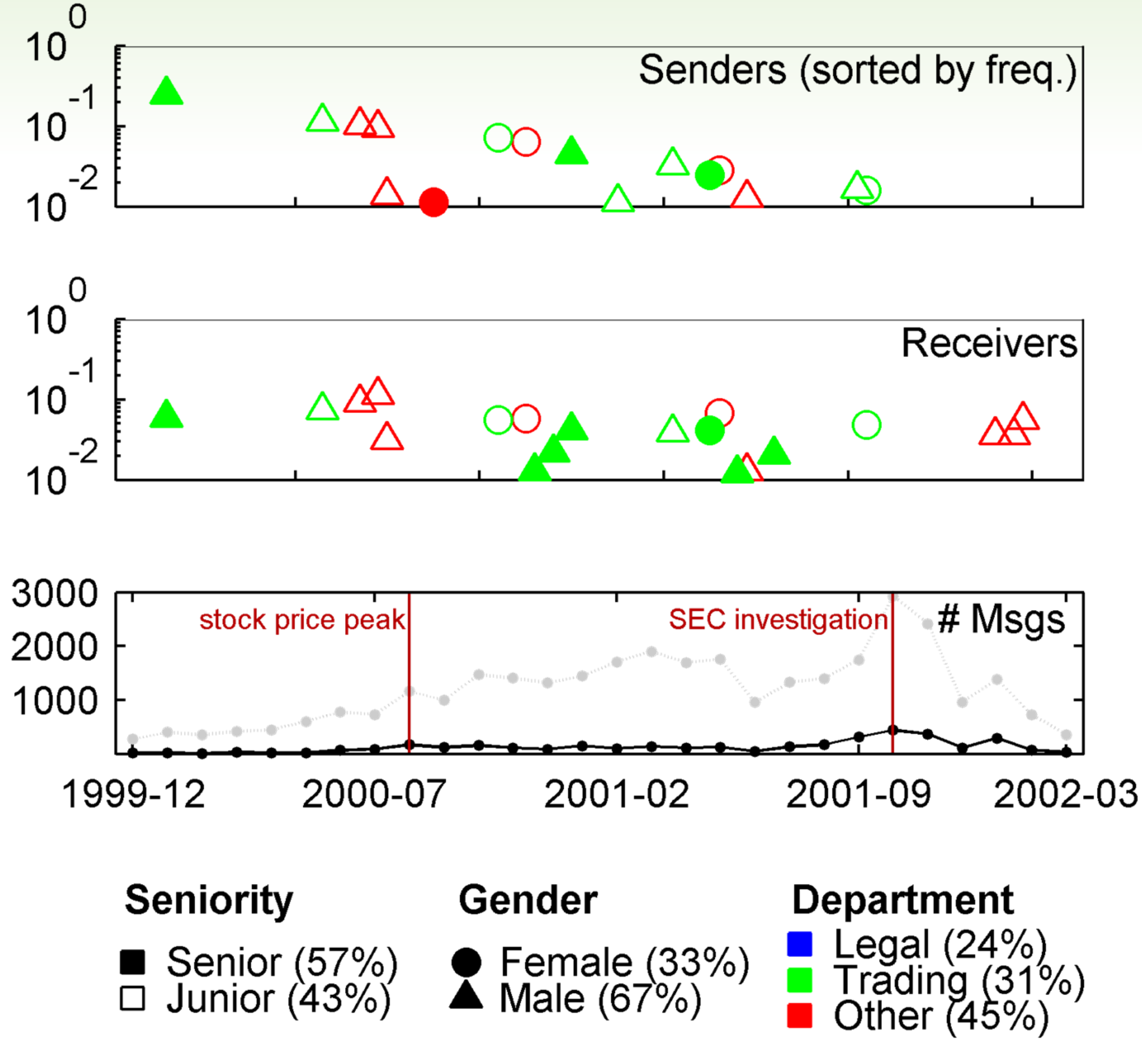


Chi & Kolda, On Tensors, Sparsity, and Nonnegative Factorizations, SIMAX 2012

Enron Email Data (Component 4)

Not Legal

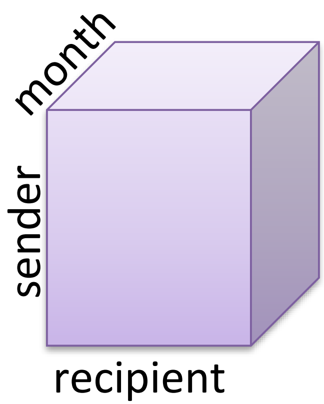


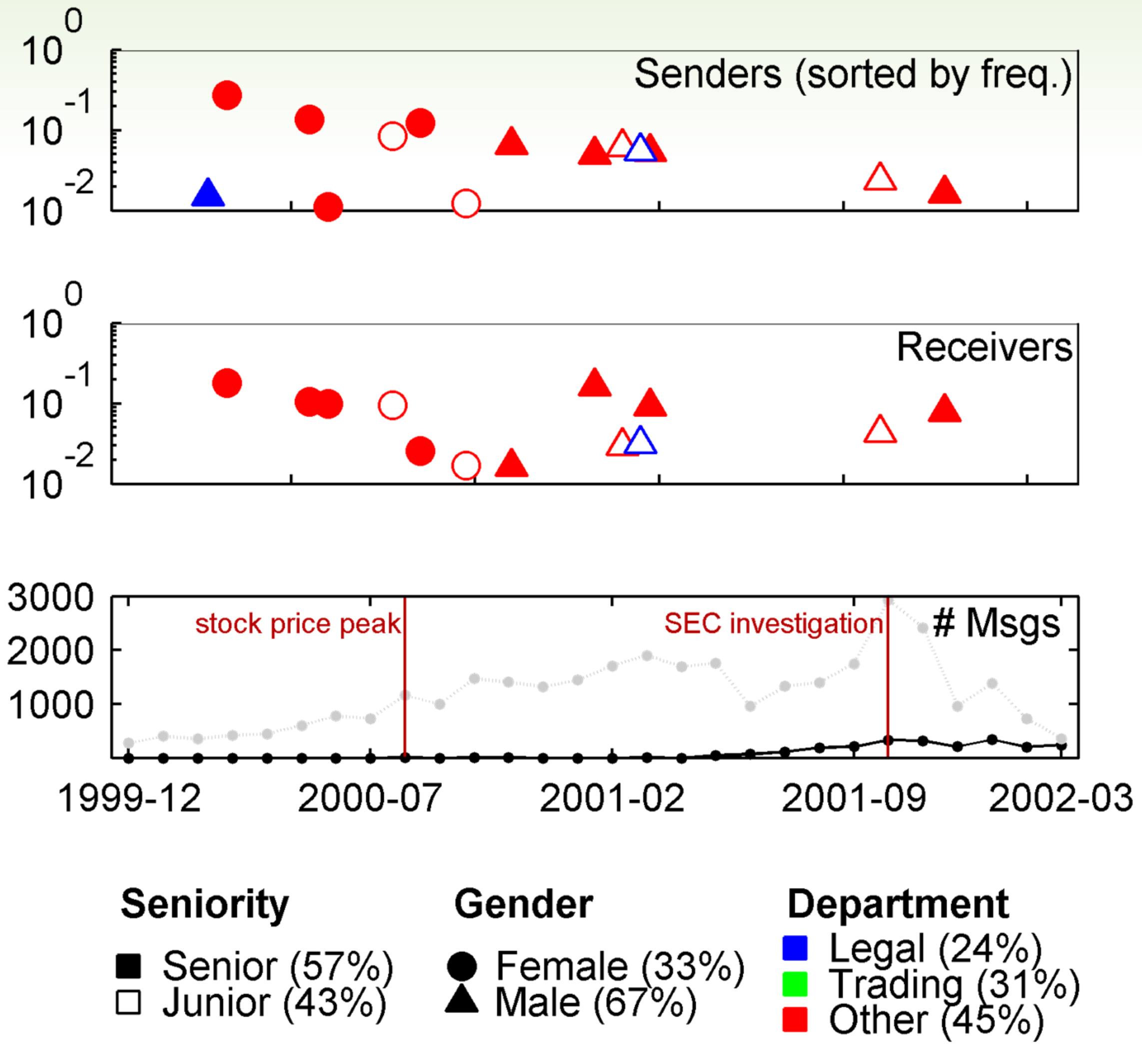


Chi & Kolda, On Tensors, Sparsity, and Nonnegative Factorizations, SIMAX 2012

Enron Email Data (Component 5)

Other;
Mostly Female



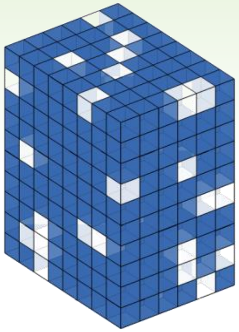


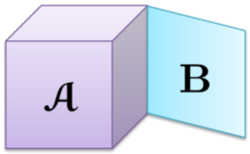
Chi & Kolda, On Tensors, Sparsity, and Nonnegative Factorizations, SIMAX 2012

A Few More Things I Didn’t Have Time to Talk About…

Quick Pointers to Other Work

- **All-at-once Optimization (not alternating):** Acar, Dunlavy, Kolda, ***A Scalable Optimization Approach for Fitting Canonical Tensor Decompositions***, J. Chemometrics, 2011, doi:10.1002/cem.1335
- **Missing Data:** Acar, Dunlavy, Kolda, Mørup, ***Scalable Tensor Factorizations for Incomplete Data***, Chemometrics and Intelligent Laboratory Systems, 2011, doi:10.1016/j.chemolab.2010.08.004
- **Coupled Factorizations**: Acar, Kolda, Dunlavy, ***All-at-once Optimization for Coupled Matrix and Tensor Factorizations***, MLG'11, arXiv:1105.3422
- **Tensor Symmetry:** Schatz, Low, van de Geijn, Kolda, ***Exploiting Symmetry in Tensors for High Performance***, SIAM J. Scientific Computing 2014, doi:10.1137/130907215
- **Symmetric Factorizations:** Kolda, ***Numerical Optimization for Symmetric Tensor Decomposition***, Mathematical Programming B2015, doi:10.1007/s10107-015-0895-0
- **Tensor Eigenvalue Problem**
  - Kolda, Mayo, ***An Adaptive Shifted Power Method for Computing Generalized Tensor Eigenpairs***, SIAM J. Matrix Analysis and Applications 2014, doi:10.1137/140951758
  - Ballard, Kolda, Plantenga, ***Efficiently Computing Tensor Eigenvalues on a GPU***, IPDPSW'11 2011, doi:10.1109/IPDPS.2011.287
  - Kolda, Mayo, ***Shifted Power Method for Computing Tensor Eigenpairs***, SIAM J Matrix Analysis and Applications 2011, doi:10.1137/100801482
- **Link Prediction using Tensors:** Dunlavy, Kolda, Acar, ***Temporal Link Prediction using Matrix and Tensor Factorizations***, ACM Trans. Knowledge Discovery from Data, 2011, doi:10.1145/1921632.1921636
- **Tucker for Sparse Data:** Kolda, Sun, ***Scalable Tensor Decompositions for Multi-aspect Data Mining***, ICDM 2008, doi:10.1109/ICDM.2008.89

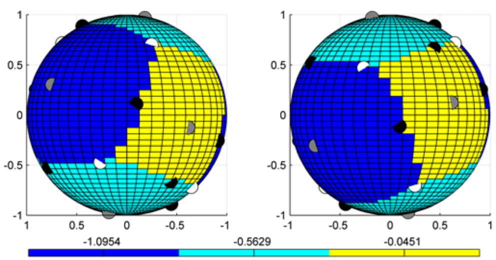




$$\mathcal{M} \approx \sum_r \lambda_r \, \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$$

$$\mathbf{B} \approx \mathbf{X}\mathbf{W}^{\top}$$

Positive Stable Basins of Attraction for 3x3x3x3 Tensors



Wrapping Up…

References, Codes, Etc.

- **Overview**
  - T. G. Kolda and B. W. Bader, ***Tensor Decompositions and Applications***, SIAM Review, Vol. 51, No. 3, pp. 455-500, September 2009, doi:10.1137/07070111X
  - B. W. Bader and T. G. Kolda, ***Efficient MATLAB Computations with Sparse and Factored Tensors***, SIAM Journal on Scientific Computing, Vol. 30, No. 1, pp. 205-231, December 2007, doi:10.1137/060676489
- **Randomization for CP**
  - C. Battaglino, G. Ballard, and T. G. Kolda, ***A Practical Randomized CP Tensor Decomposition***, January 2017, arXiv:1701.06600
- **Parallel Tucker for Compression**
  - W. Austin, G. Ballard and T. G. Kolda, ***Parallel Tensor Compression for Large-Scale Scientific Data***, IPDPS'16: Proceedings of the 30th IEEE International Parallel and Distributed Processing Symposium, pp. 912-922, May 2016, doi:10.1109/IPDPS.2016.67, arXiv:1510.06689
- **Poisson Tensor Factorization (Count Data)**
  - E. C. Chi and T. G. Kolda, ***On Tensors, Sparsity, and Nonnegative Factorizations***, SIAM Journal on Matrix Analysis and Applications, Vol. 33, No. 4, pp. 1272-1299, December 2012, doi:10.1137/110859063
  - S. Hansen, T. Plantenga and T. G. Kolda, ***Newton-Based Optimization for Kullback-Leibler Nonnegative Tensor Factorizations***, Optimization Methods and Software, Vol. 30, No. 5, pp. 1002-1029, April 2015, doi:10.1080/10556788.2015.1009977, arXiv:1304.4964
- **Codes**
  - Tensor Toolbox for MATLAB: http://www.sandia.gov/~tgkolda/TensorToolbox/
  - TuckerMPI: https://gitlab.com/tensors/TuckerMPI

Tensors are a Foundational Tool for Data Analysis

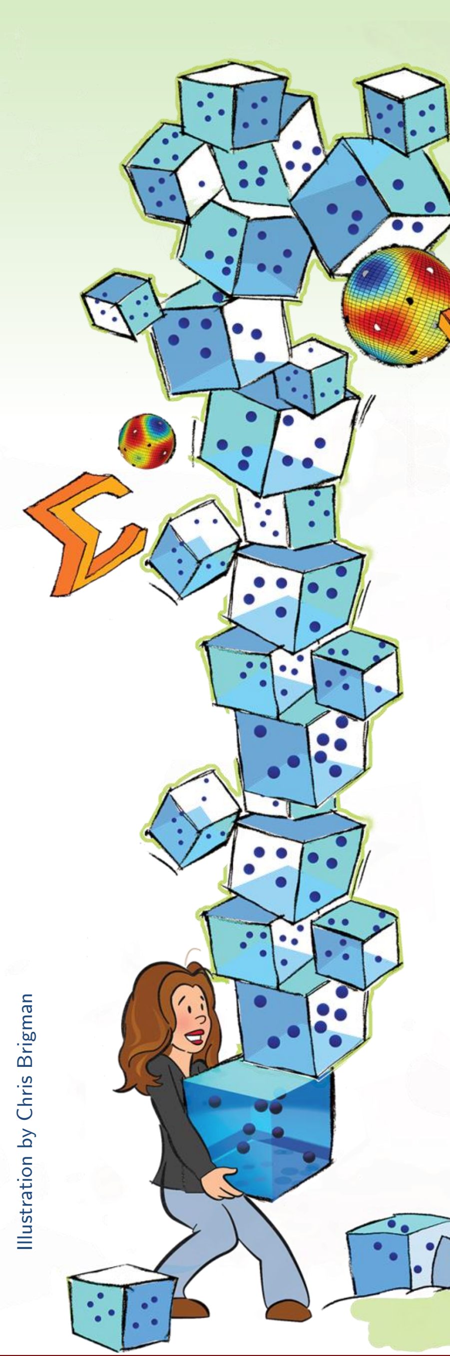
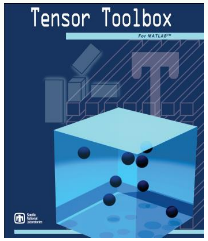
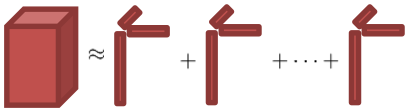


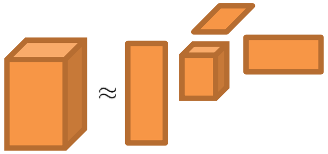
Illustration by Chris Brigman

- Tensor decomposition extends matrix factorization
  - SVD, PCA, NMF, etc.
- Useful for data analysis
  - Mouse neuron activity
  - CFD simulation data compression
  - Enron emails
- Choice of decompositions
  - Canonical polyadic (CP) decomposition
  - Tucker decomposition
  - Poisson tensor factorization
- Computational advances
  - Randomization
  - Parallelization





TUCKERMPI



Tamara G. Kolda
tgkolda@sandia.gov
http://www.sandia.gov/~tgkolda/