Gene Golub SIAM Summer School

Tensor Decompositions

Tammy Kolda
Sandia National Labs, Livermore, CA

June 2019

Part 1

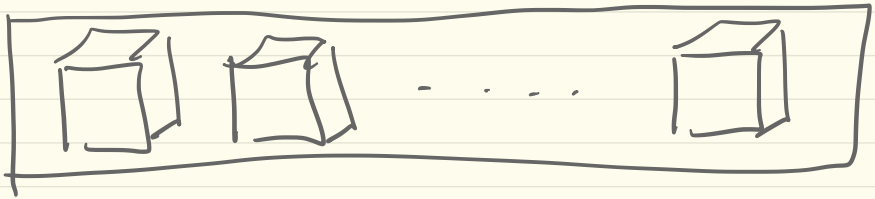
Tensor Basics

A tensor is a d-way array. d = order of the tensor

- Vector is a tensor with $d = 1$
- Matrix " " " " $d = 2$
- Higher-order tensor have $d \geq 3$

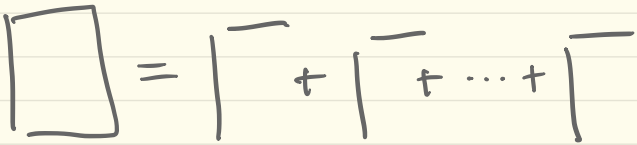
- Three-way tensor

- Four-way tensor



Tensor Factorization

- Matrix Factorization - breaks matrix into sum of rank-1 parts

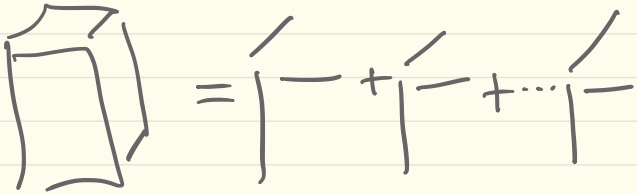


ex: SVD, EVD, NMF, sparse SVD

- matrix factorization finds high variance subspaces

ex: SVD, CUR, PCA/ICA

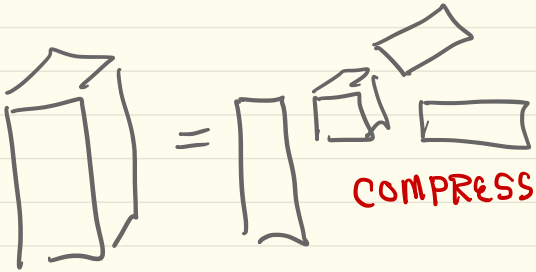
- Tensor factorization breaks into rank-1 parts



CP: CANDECOMP/PARAFAC

Interpretation

- Tensor factorization into high variance subspaces



COMPRESSION

ex: HOSVD, Tucker

Not discussed: Tensor Train, T-SVD, hierarchical Tucker

Example of CP from Neuroscience

Example Tensor from Neuroscience

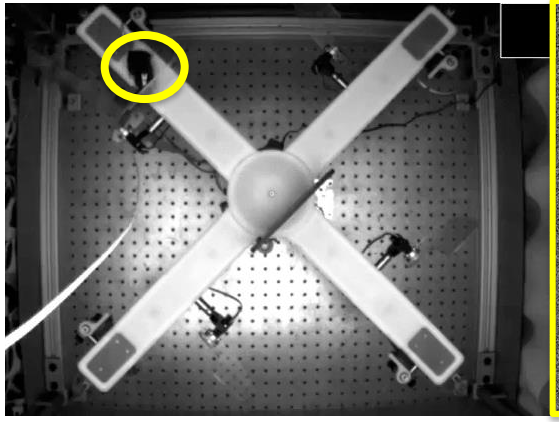
Source: Williams, et al. Unsupervised Discovery of Demixed, Low-dimensional Neural Dynamics across Multiple Timescales through Tensor Components Analysis. Neuron, 2018. https://doi.org/10.1016/j.neuron.2018.05.015

Activity of Single Neuron Measured Over Time Produces Vector Data

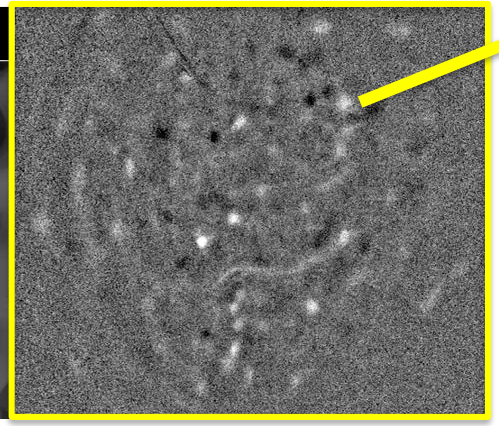
Thanks to Schnitzer Group @ Stanford
Mark Schnitzer, Fori Wang, Tony Kim

Microscope by Inscopix

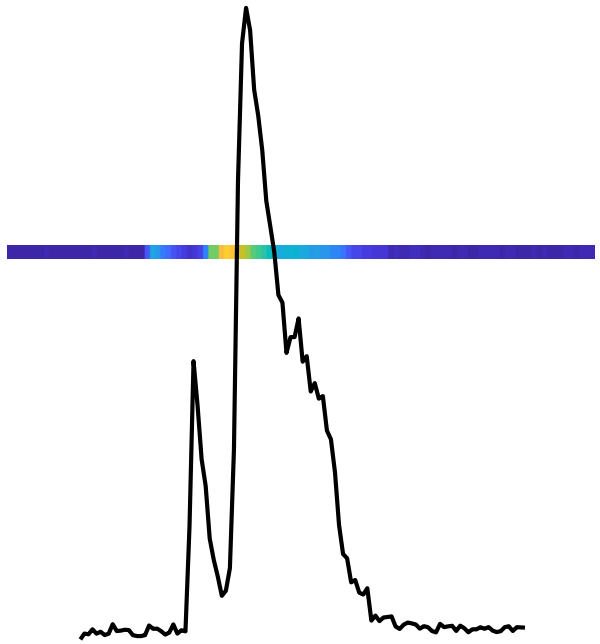
mouse in maze



neural activity via calcium imaging



111 time bins



Williams et al., Neuron, 2018

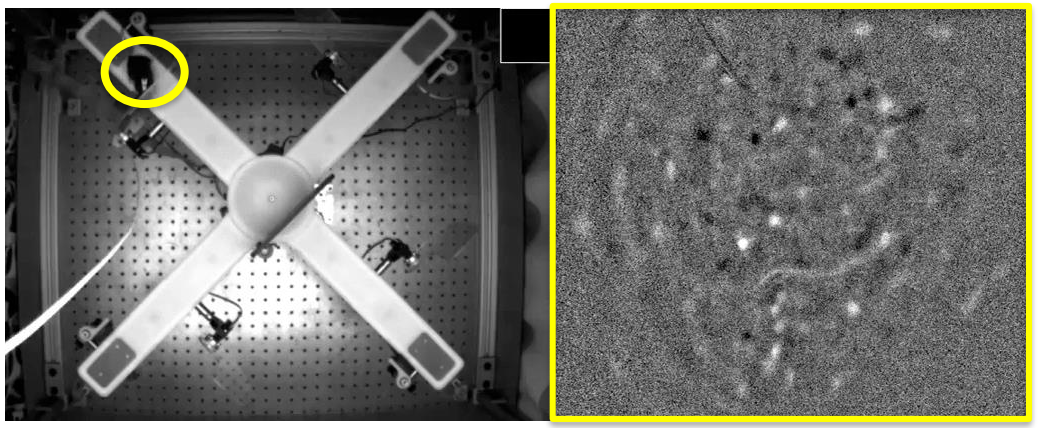
Multiple Neurons Measured Over Time Produces Matrix

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

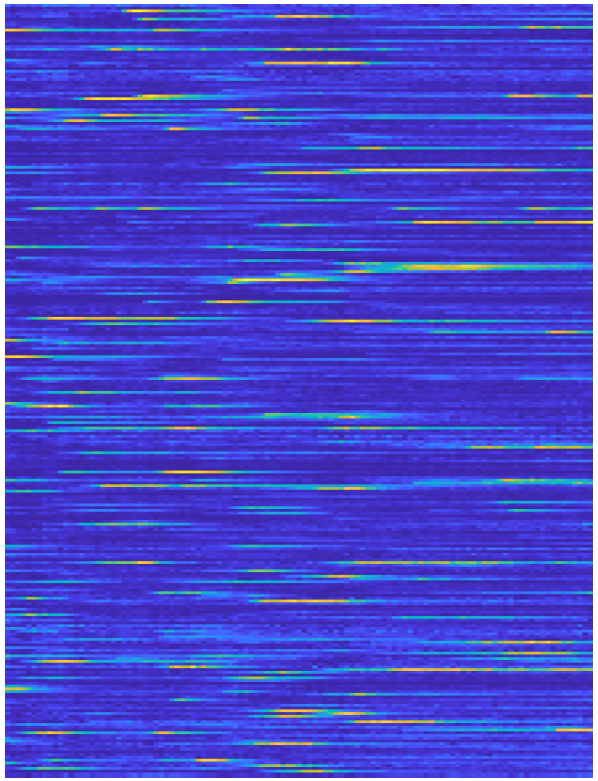
Microscope by Inscopix

mouse in "maze"

neural activity

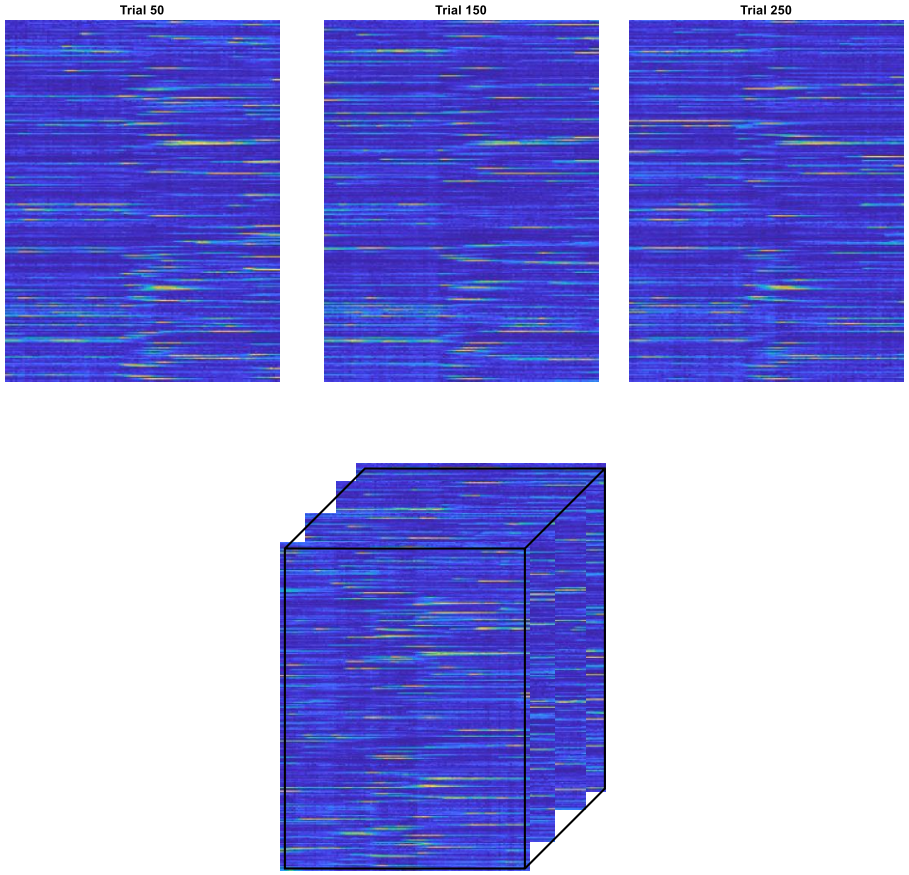


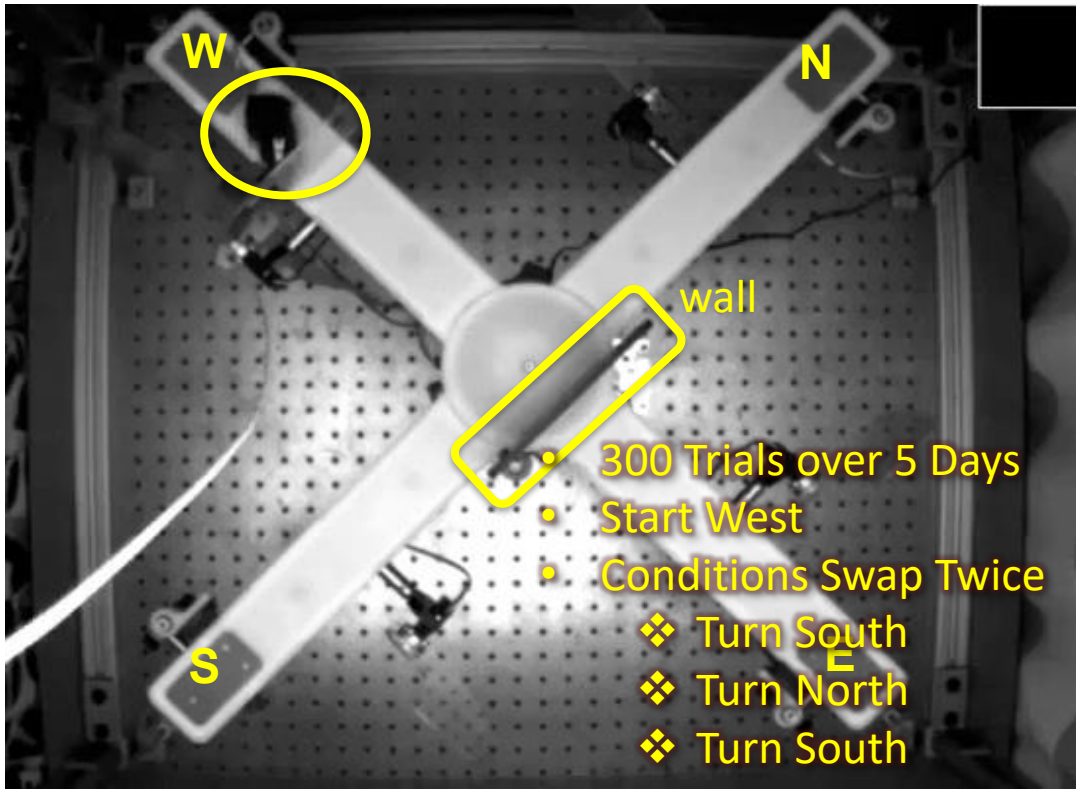
282 neurons × 111 time bins



Williams et al., Neuron, 2018

Multiple Trials Produces 3-way Tensor





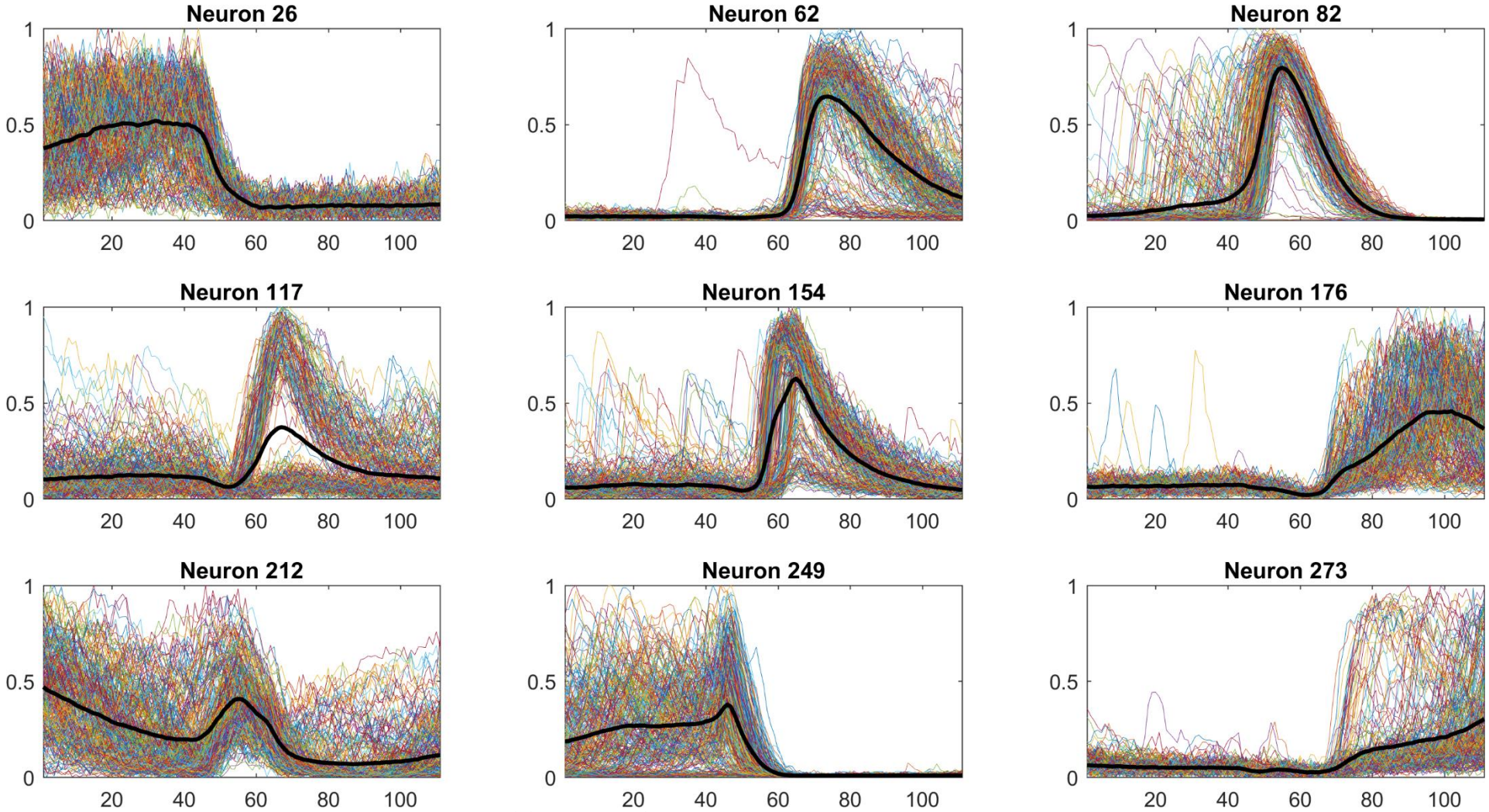
282 neurons × 111 time bins × 300 trials

Williams et al., Neuron, 2018

Example Neuron Activity

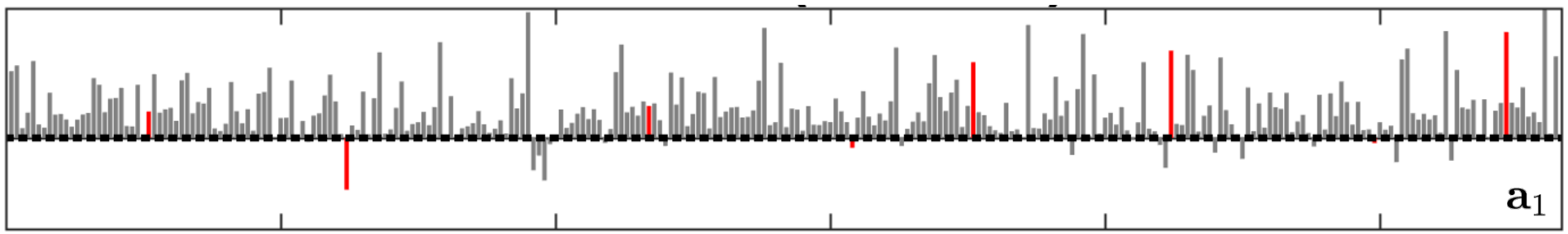
Thin lines show 300 individual trials

Thick line is average

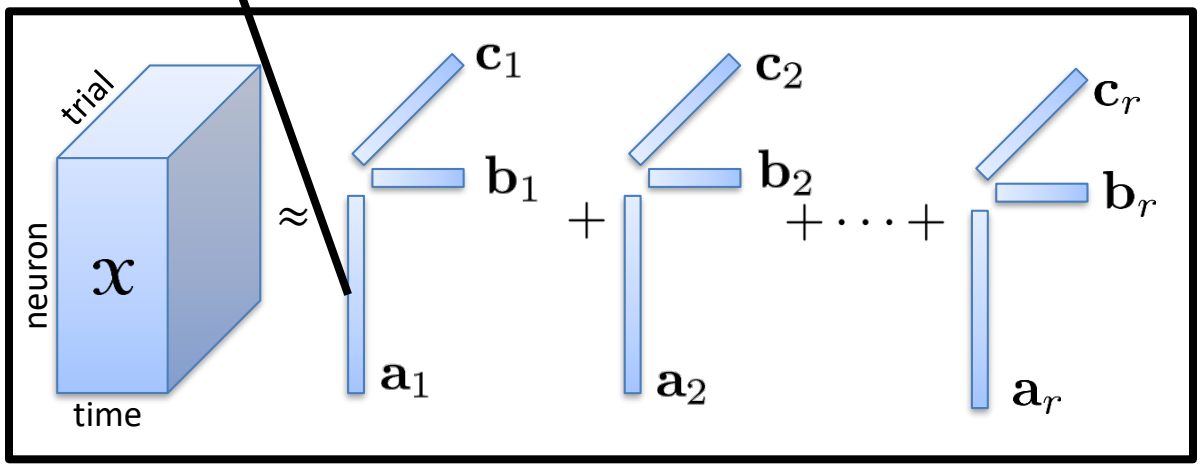


Hong, Kolda, Duersch, SIAM Review, 2019

Neuron Factor Vector Visualized as Bar Chart

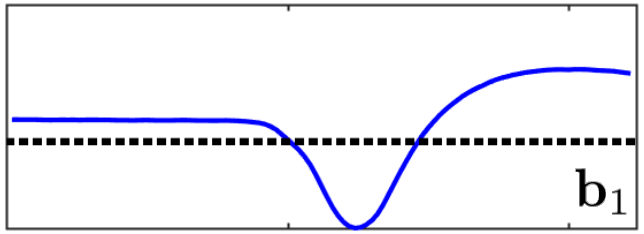


Neuron Modes Plotted as a Bar Chart
(Red Lines Correspond to Examples in Previous Slide)

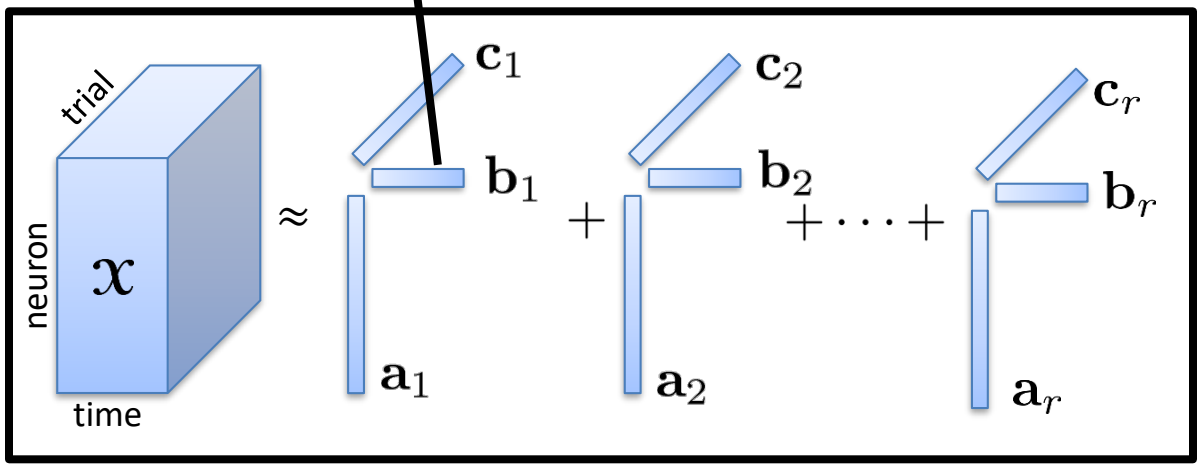


Hong, Kolda, Duersch, SIAM Review, 2019

Time Factor Vector Visualized as Line

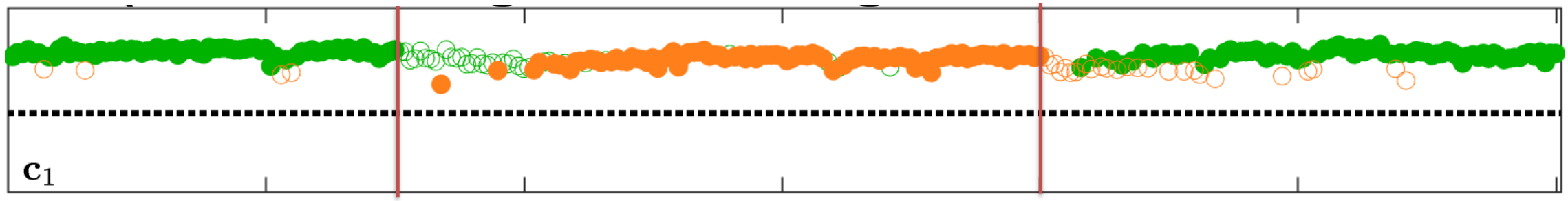


Time (within trial) Plotted as a Line
(Dashed Line is Zero)



Hong, Kolda, Duersch, SIAM Review, 2019

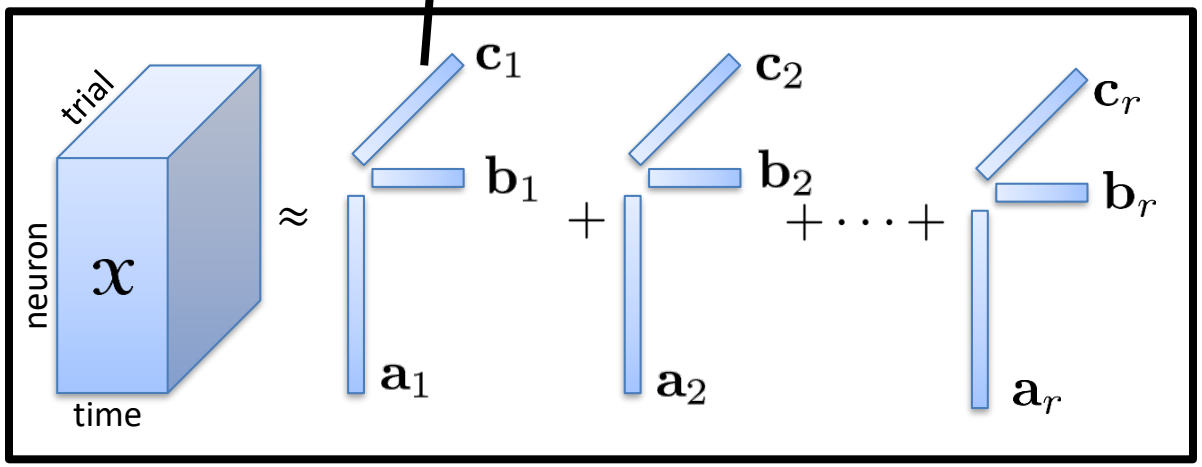
Trial Factor Vector Visualized as Color-Coded Scatter Plot



Rule Change

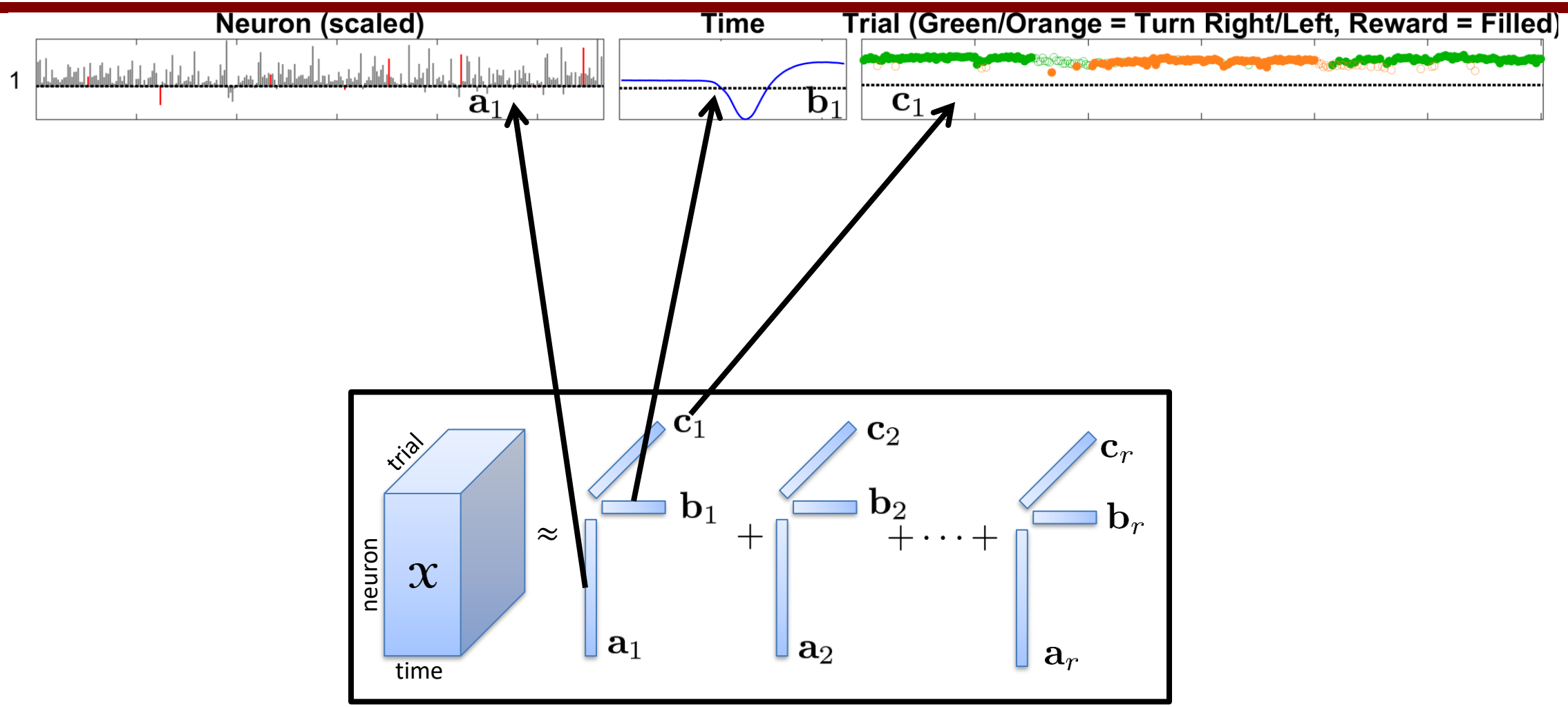
Trial Plotted as Scatter Graph
Right turn = Green
Left turn = Orange
Filled = Reward

Rule Change



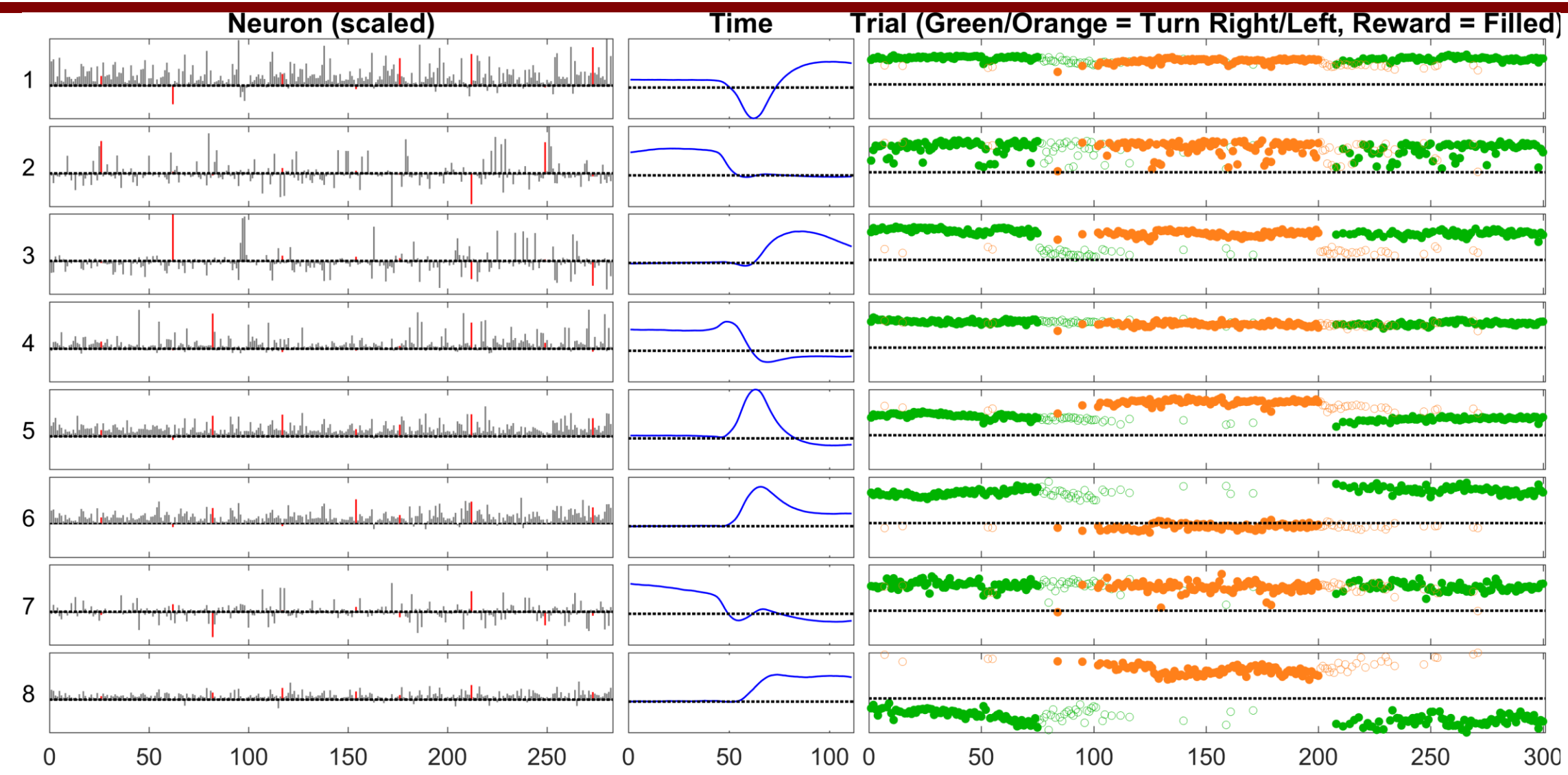
Hong, Kolda, Duersch, SIAM Review, 2019

Visualization of CP Tensor Decomposition Shows the Factors (Vectors)

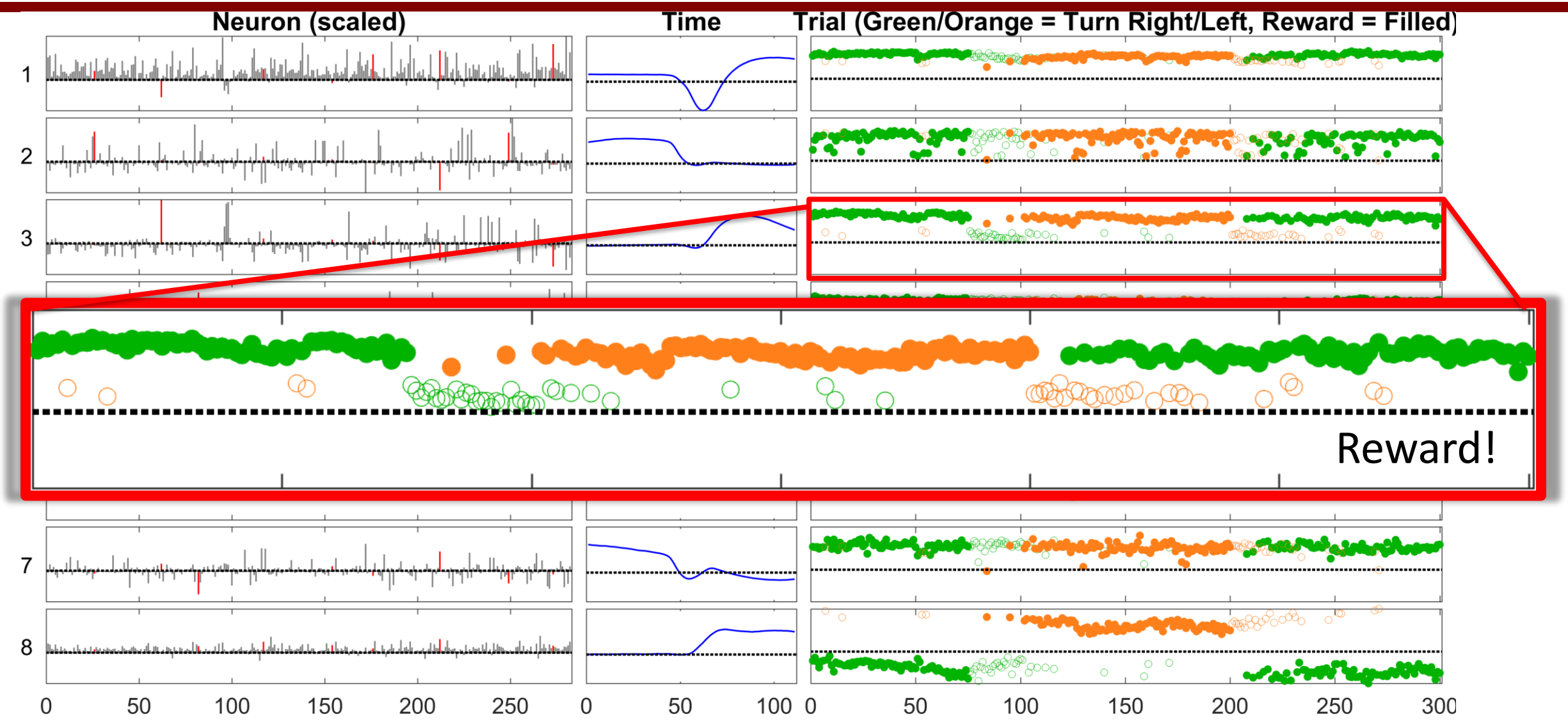


Hong, Kolda, Duersch, SIAM Review, 2019

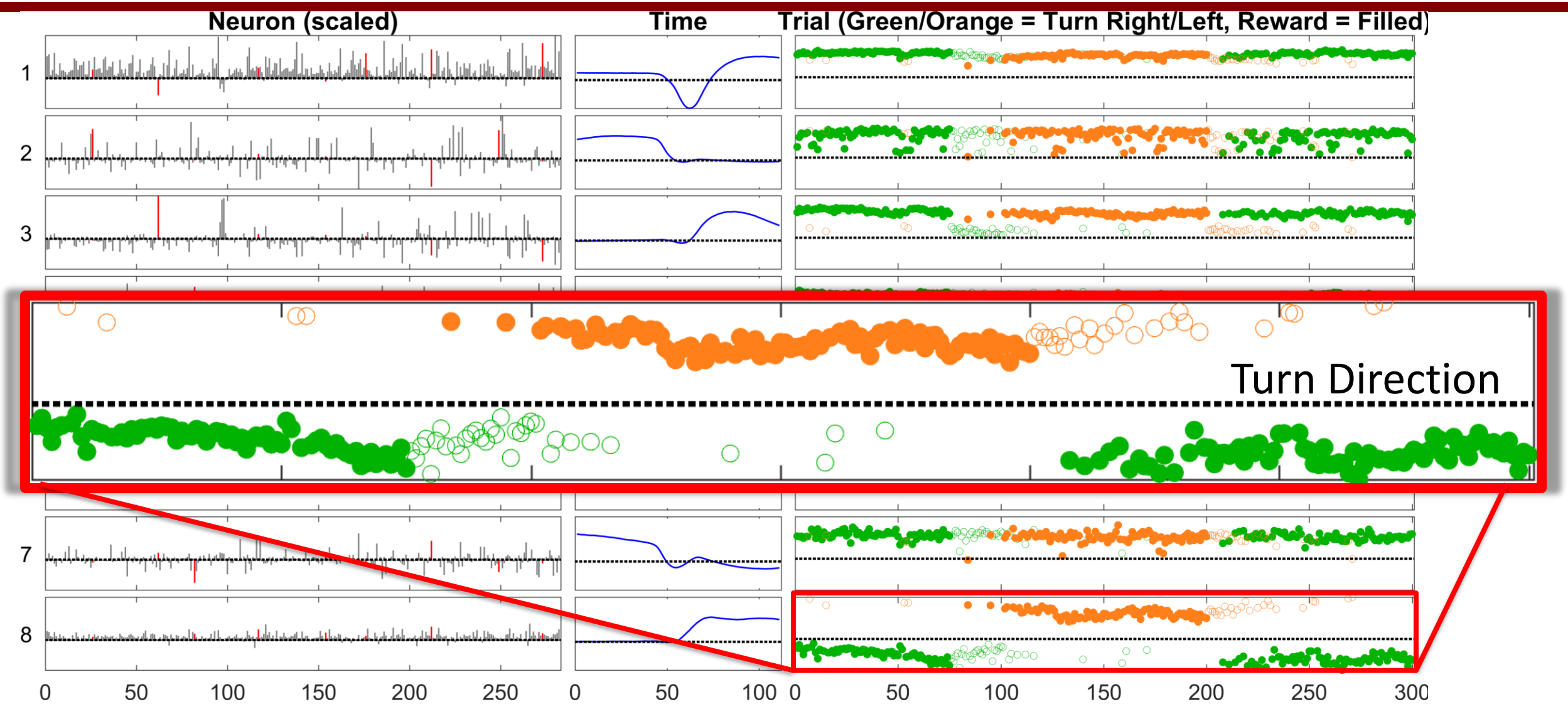
“Standard” CP Decomposition of Mouse Data, aka Gaussian ($f(x,m) = (x-m)^2$)



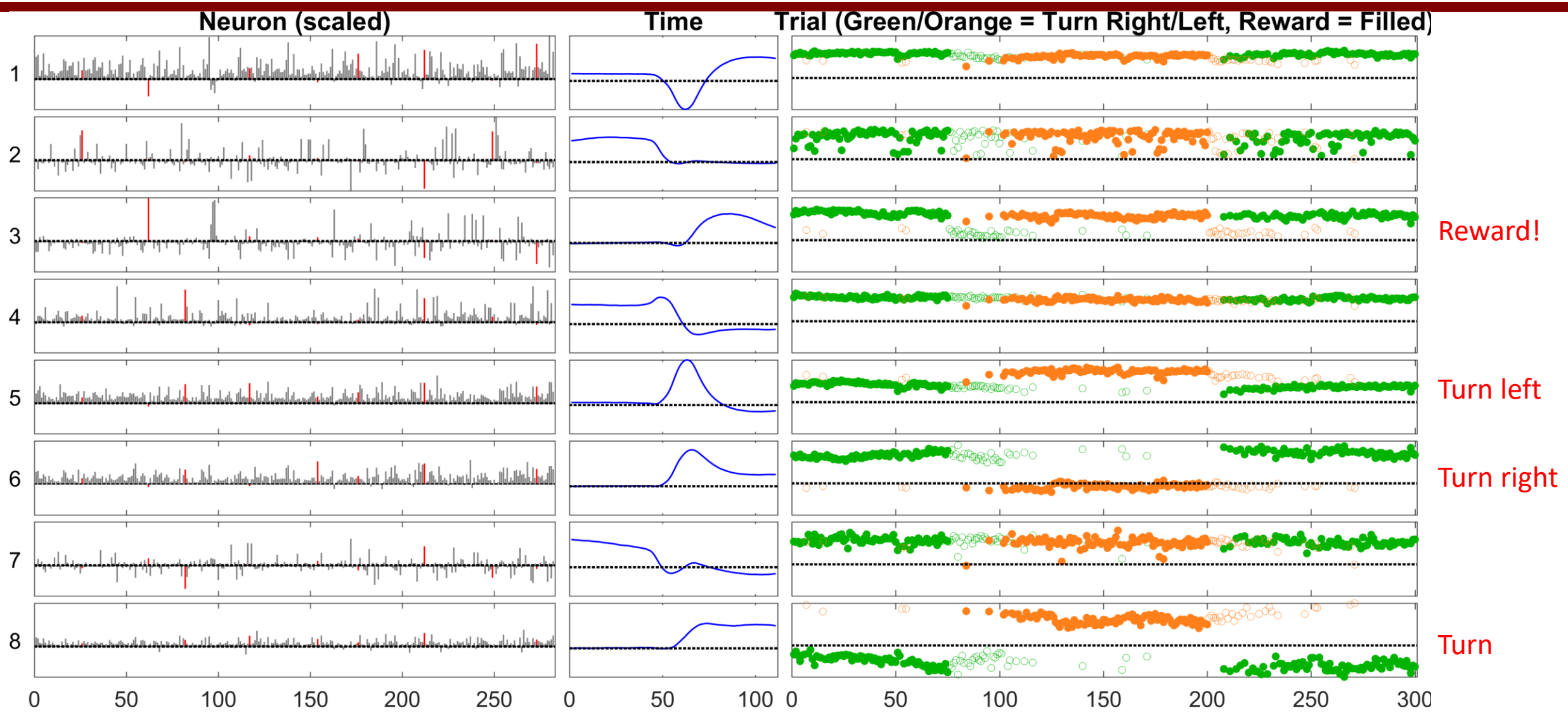
CP Tensor Decomposition "Sees" Reward



CP Tensor Decomposition "Sees" Turn Direction

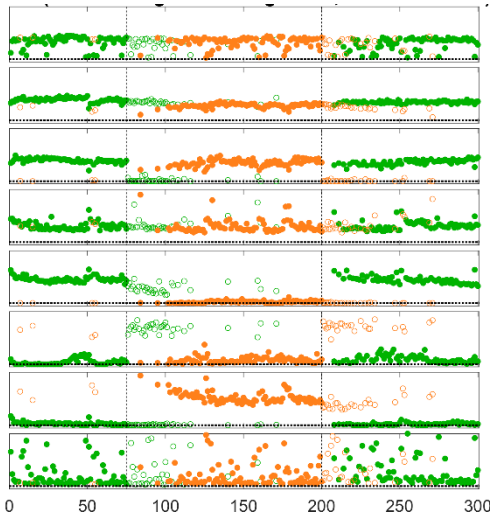


CP Tensor Decomposition Can be Tough to Interpret due to Negative Entries



Regression Using GCP Factors on Trial Mode

Trial Factor Matrix is 300×8

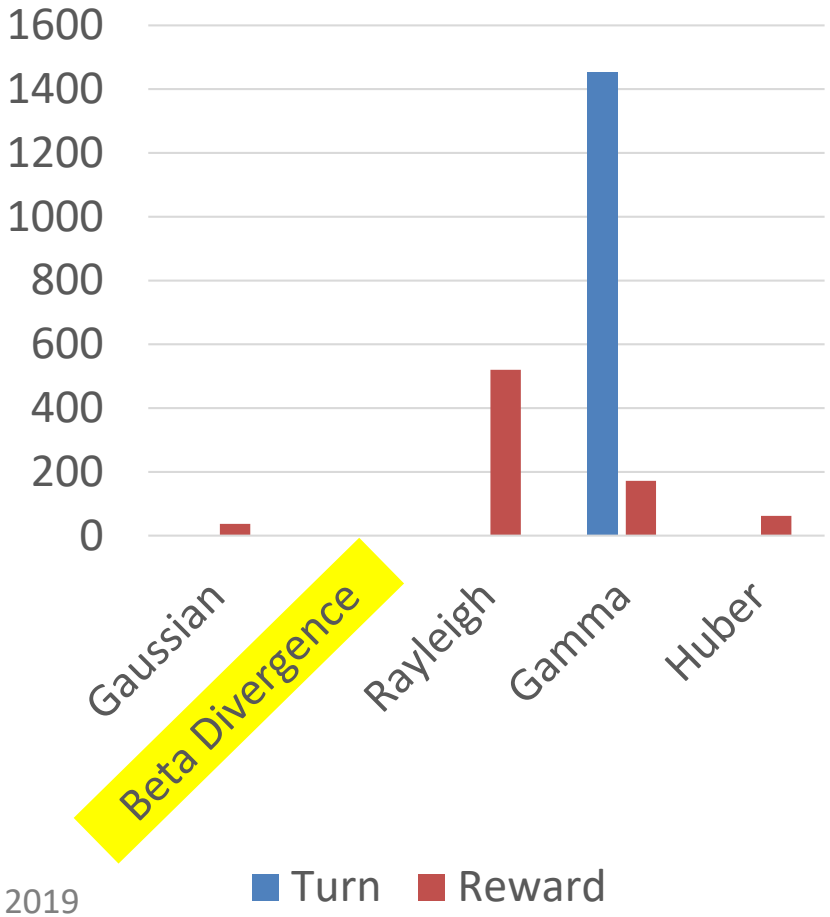


$$\min_{\boldsymbol{\beta}} \|\mathbf{A}_3^{\text{train}}\boldsymbol{\beta} - \mathbf{y}^{\text{train}}\|$$

$$\hat{\mathbf{y}}^{\text{test}} = [\mathbf{A}_3^{\text{test}}\boldsymbol{\beta} \geq 0.5]$$

Look at predicting turn and reward.
Split into two groups of 150 trials.
Train regression model with 1st group.
Test with 2nd group.
Repeat 100 times.

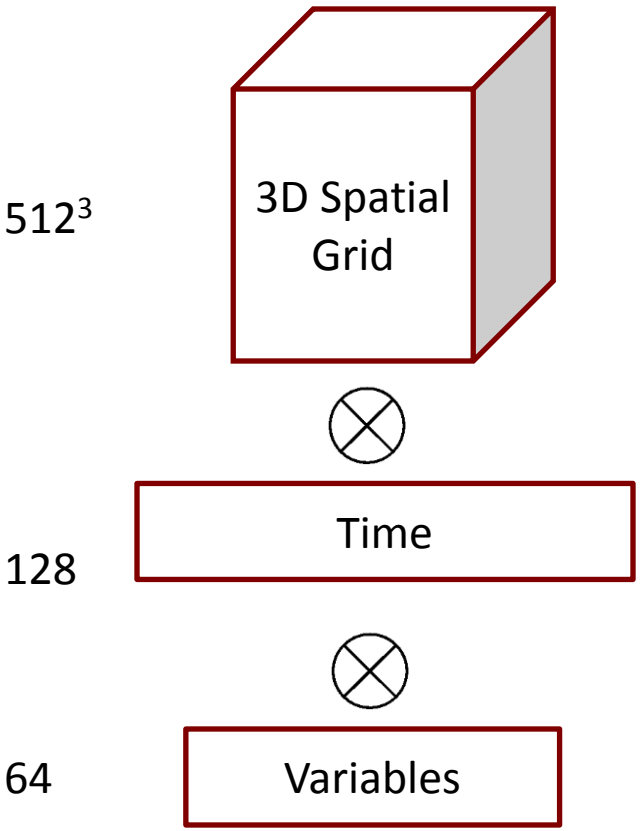
Regression Errors in 100 Trials (15000 predictions)



Hong, Kolda, Duersch, SIAM Review, 2019

Example of Tucker for Compression of Scientific Data

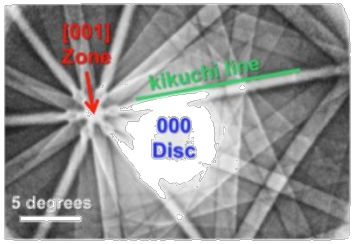
Motivation: Advanced Simulations and Experiments Deluged by Data

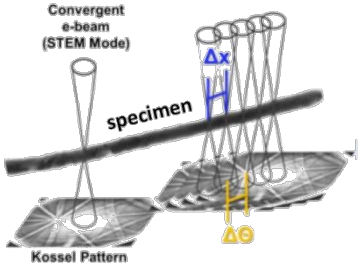


2^{40} elements

8TB (double precision)

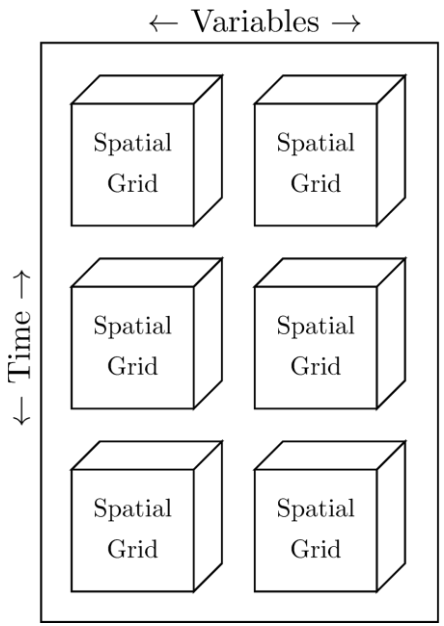
- Combustion simulations
  - S3D code uses direct numerical simulation
  - Gold standard for comparisons, but…
  - Single experiment produces terabytes of data
  - Storage limits spatial, temporal resolutions
  - Difficult to analyze or transfer data
- Electron microscopy
  - New technology produces 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



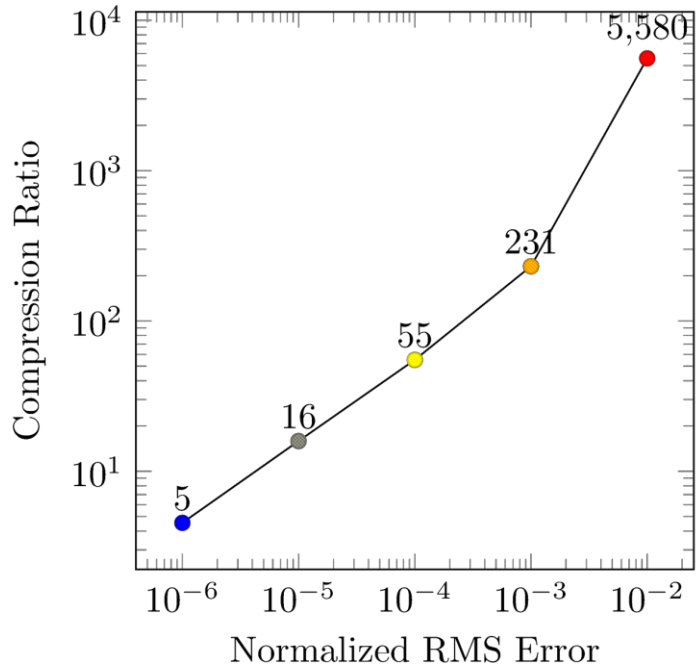


- Other applications
  - Telemetry
  - Cosmology Simulations
  - Climate Modeling

Claim: Tensor Tucker Compression Yields Up to 5000X Data Reduction



Natural five-way multiway structure of scientific data



Compression rates as fidelity varies for 550GB simulation dataset

Our contributions:

- Distributed implementation of Tucker compression
- Demonstration of good compression on combustion data sets

Recent TuckerMPI results

See Ballard, Klinvex, Kolda (2019-arXiv) for further context + details.

Table 4. Experimental setup

(a) Data sets to be compressed and the number of parallel nodes to be used in the compression experiments.

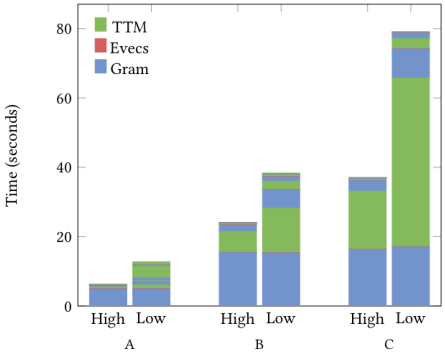
Dataset Name	Overall Tensor Size	Total Storage	Number of Nodes	Number of Processes	Storage per Node	Storage per Process
SP	$500 \times 500 \times 500 \times 11 \times 400$	4.4 TB	250	4000	17.6 GB	1.1 GB
JICF	$1500 \times 2080 \times 1500 \times 18 \times 10$	6.7 TB	350	5600	19.3 GB	1.2 GB

(b) Three different processor grid configurations per dataset, to test the efficiency of the compression. The local tensor size may vary, but here we list the largest local size in each dimension.

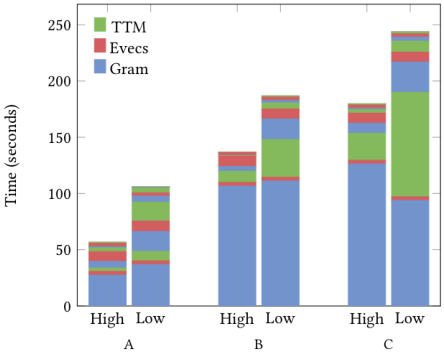
Dataset	Proc. Config. Name	Processor Grid	Local Tensor Size
SP	A	$1 \times 1 \times 40 \times 1 \times 100$	$500 \times 500 \times 13 \times 11 \times 4$
	B	$10 \times 8 \times 5 \times 1 \times 10$	$50 \times 63 \times 100 \times 11 \times 40$
	C	$40 \times 10 \times 1 \times 1 \times 10$	$13 \times 50 \times 500 \times 11 \times 40$
JICF	A	$1 \times 16 \times 35 \times 1 \times 10$	$1500 \times 130 \times 43 \times 18 \times 1$
	B	$10 \times 8 \times 7 \times 1 \times 10$	$150 \times 260 \times 215 \times 18 \times 1$
	C	$35 \times 16 \times 1 \times 1 \times 10$	$43 \times 130 \times 1500 \times 18 \times 1$

Dataset	Scenario Name	Relative Error	Compressed Core Size	Total Storage	Compression Ratio
SP	High	1e-2	$30 \times 38 \times 35 \times 6 \times 11$	21.5 MB	2×10^5
	Low	1e-4	$95 \times 129 \times 125 \times 7 \times 125$	10.7 GB	4×10^2
JICF	High	1e-2	$90 \times 61 \times 48 \times 13 \times 6$	167 MB	4×10^4
	Low	1e-4	$424 \times 387 \times 261 \times 18 \times 10$	45.7 GB	1×10^2

(a) ST-HOSVD compression results (independent of processor grid)



(b) ST-HOSVD on SP dataset with different processor grids



(c) ST-HOSVD on JICF dataset with different processor grids

Proc. Config.	Compression Scenario	SP dataset Memory Usage (GB)	SP dataset I/O Time (s) Input	SP dataset I/O Time (s) Output	SP dataset Comp. Time (s)	JICF dataset Memory Usage (GB)	JICF dataset I/O Time (s) Input	JICF dataset I/O Time (s) Output	JICF dataset Comp. Time (s)
A	High ϵ=1e-2	1.22	370	0.2	6	1.43	2308	2	57
	Low ϵ=1e-4	1.42		22	13	2.24		424	106
B	High ϵ=1e-2	3.33	877	0.9	24	3.62	2187	4	137
	Low ϵ=1e-4	3.33		983	38	3.62		18517	187
C	High ϵ=1e-2	3.34	861	0.8	37	3.61	2077	5	180
	Low ϵ=1e-4	3.34		13470	79	3.61		DNC	244

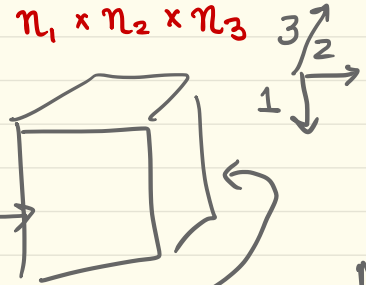
(d) STHOSVD per-core memory usage, I/O time, and computation time

Fig. 8. ST-HOSVD compression, run time breakdown, memory usage, and I/O cost for different choices of ϵ (in in Alg. 1) and processor grids. The different processor grids (A/B/C) are given in Tab. 4b. In the breakdowns, since these are 5-way datasets, there are five iterations of ST-HOSVD, each of which calls Gram, Evecs, and TTM. These are stacked from mode 0 (bottom) to mode 4 (top). The mode-0 calls are the most expensive because the tensor is reduced in size for subsequent iterations.

Tensor Notation for 3D Tensors

$\mathcal{X}$ is a tensor of size $3 \times 3 \times 2$

$n_1 \times n_2 \times n_3$



$$\mathcal{X}(:,:,1) = \begin{bmatrix} 3 & 9 & 1 \\ 8 & 2 & 1 \\ 4 & 3 & 9 \end{bmatrix}$$

$$\mathcal{X}(:,:,2) = \begin{bmatrix} 6 & 9 & 5 \\ 5 & 6 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

"Frontal Slices"

Tensor Fibers

Mode-1 ||||

Mode-2 ≡

Mode-3 ///

Examples

$\mathcal{X}(1,1,1) = 3$

$\mathcal{X}(2,1,1) = 8$

$\mathcal{X}(1,1,2) = 6$

$\mathcal{X}(3,2,1) = 3$

mode-1

$$\mathcal{X}(:,2,2) = \begin{bmatrix} 9 \\ 6 \\ 4 \end{bmatrix}$$

$$\mathcal{X}(3,2,:) = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

mode

$$\mathcal{X}(1,:,1) = \begin{bmatrix} 3 \\ 9 \\ 1 \end{bmatrix}$$

Mode-k fiber is of length n_k

Vectorizing a Tensor

$$\mathcal{X}(:,:,1) = \begin{array}{c} \xrightarrow{2} \\ \downarrow 1 \end{array} \begin{bmatrix} 3 & 9 & 1 \\ 8 & 2 & 1 \\ 4 & 3 & 9 \end{bmatrix}$$

$\downarrow 3$

$$\mathcal{X}(:,:,2) = \begin{bmatrix} 6 & 9 & 5 \\ 5 & 6 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ 8 \\ 4 \\ 9 \\ 2 \\ 3 \\ 1 \\ 1 \\ 9 \\ 6 \\ 5 \\ 1 \\ 9 \\ 6 \\ 4 \\ 5 \\ 4 \\ 1 \end{bmatrix} = \mathrm{vec}(\mathcal{X})$$

$\leftarrow$ How tensor is stored in memory.

size $= n_1 n_2 n_3 = 18$

General Formula

$$(i_1, i_2, i_3) \rightarrow (i_3 - 1) \cdot n_2 n_1 + (i_2 - 1) \cdot n_1 + i_1$$

Matricization / Unfolding a 3D Tensor

$$X(:,:,1) = \begin{bmatrix} 3 & 9 & 1 \\ 8 & 2 & 1 \\ 4 & 3 & 9 \end{bmatrix}$$

$$X(:,:,2) = \begin{bmatrix} 6 & 9 & 5 \\ 5 & 6 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

Mode-k unfolding arranges mode-k fibers as matrix columns

$$X_{(1)} = \begin{bmatrix} 3 & 9 & 1 & 6 & 9 & 5 \\ 8 & 2 & 1 & 5 & 6 & 4 \\ 4 & 3 & 9 & 1 & 4 & 1 \end{bmatrix} \begin{matrix} n_1 \times n_2 n_3 \\ 3 \times 6 \end{matrix}$$

$$X_{(2)} = \begin{bmatrix} 3 & 8 & 4 & 6 & 5 & 1 \\ 9 & 2 & 3 & 9 & 6 & 4 \\ 1 & 1 & 9 & 5 & 4 & 1 \end{bmatrix} \begin{matrix} n_2 \times n_1 n_3 \\ 3 \times 6 \end{matrix}$$

$$X_{(3)} = \begin{bmatrix} 3 & 8 & 4 & 9 & 2 & 3 & 1 & 1 & 9 \\ 6 & 5 & 1 & 9 & 6 & 4 & 5 & 4 & 1 \end{bmatrix} \begin{matrix} n_3 \times n_1 n_2 \\ 2 \times 9 \end{matrix}$$

$$(i_1, i_2, i_3) \rightarrow (i_1, i_1')$$
$$\rightarrow (i_2, i_2')$$
$$\rightarrow (i_3, i_3')$$

$$i_1' = (i_3 - 1) n_2 + i_2$$

$$i_2' = (i_3 - 1) n_1 + i_1$$

$$i_3' = (i_2 - 1) n_1 + i_1$$

Notation for d-way Tensors

Notation for index set
$[n] = \{1, 2, \ldots, n\}$

- $\mathcal{X}$ is a d-way array
- order $(\mathcal{X}) = d$ ndims $(\mathcal{X})$
- size $(\mathcal{X}) = n_1 \times n_2 \times \cdots \times n_d$
- element of $\mathcal{X}$: $\mathcal{X}(i_1, i_2, \ldots, i_d)$, $\mathcal{X}_{i_1 i_2 \cdots i_d}$, $\mathcal{X}_{\mathbf{i}}$ ← multiindex
- $\mathcal{I} = \{(i_1, \ldots, i_d) : i_1 \in [n_1], \ldots i_d \in [n_d]\}$
- Define

$$n = \sqrt[d]{\prod_{k=1}^{d} n_k} \quad \left(\begin{matrix}\text{geometric}\\ \text{mean}\end{matrix}\right) \Rightarrow |\mathcal{I}| = n^d$$

$$\bar{n} = \frac{1}{d} \sum_{k=1}^{d} n_k \quad \left(\begin{matrix}\text{arithmetic}\\ \text{mean}\end{matrix}\right)$$

$$n \leq \bar{n}$$

Storage $(\mathcal{X}) = n^d$

Vectorizing a d-way Tensor

"subscript" "linear index"

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

$$i' = \sum_{k=1}^{d} (i_k - 1)\, n'_k \qquad n'_k = \begin{cases} \prod_{\ell=1}^{k-1} n_\ell & \text{for } k \geq 2 \\ 1 & \text{for } k = 1 \end{cases}$$

$$\left.\begin{array}{l} \#_\text{sub2ind} \\ \#_\text{ind2sub} \end{array}\right\} \text{ for } n^d < 2^{32}$$

$$\left.\begin{array}{l} \#_\text{sub2ind64} \\ \#_\text{ind2sub64} \end{array}\right\} \text{ for } n^d < 2^{64}$$

Unfolding a d-way Tensor

$n_1 \times \cdots \times n_d$ $\qquad$ $n_k \times n^d / n_k$

$$(i_1, \ldots, i_d) \to (i_k, i_k')$$

$$i_k' = 1 + \sum_{l=1}^{k-1} (i_l - 1)\, n_l' + \sum_{l=k+1}^{d} (i_l - 1)\left(\frac{n_l'}{n_k}\right)$$

from before

$$i_k' \in \left[n^d / n_k \right]$$

4D Tensor

$3 \times 4 \times 3 \times 2$

$$\mathrm{vec}(\mathcal{X}) = \begin{bmatrix} 1 \\ \vdots \\ 72 \end{bmatrix}$$

$$\mathcal{X}(:,:,1,1) = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}$$

$$\mathcal{X}(:,:,2,1) = \begin{bmatrix} 13 & \vdots & \vdots & \vdots \\ 14 & \vdots & \vdots & \vdots \\ 15 & \cdots & \cdots & 24 \end{bmatrix}$$

$$\mathcal{X}(:,:,3,1) = \begin{bmatrix} 25 & | & | & | \\ | & | & | & 36 \end{bmatrix}$$

$$\mathcal{X}(:,:,1,2) = \begin{bmatrix} 37 & | & | & | \\ | & | & | & 48 \end{bmatrix}$$

$$\mathcal{X}(:,:,2,2) = \begin{bmatrix} 49 & | & | & | \\ | & | & | & 60 \end{bmatrix}$$

$$\mathcal{X}(:,:,3,2) = \begin{bmatrix} 61 & | & | & | \\ | & | & | & 72 \end{bmatrix}$$

$$\mathcal{X}_{(1)} = \begin{bmatrix} 1 & 4 & & 10 \\ 2 & 5 & \cdots & 11 \\ 3 & 6 & & 12 \end{bmatrix} \quad 3 \times 24$$

$$\mathcal{X}_{(2)} = \begin{bmatrix} 1\ 2\ 3 & 13 & & 61 \\ 4\ 5\ 6 & — & \cdots & — \\ 7\ 8\ 9 & — & & — \\ 10\ 11\ 12 & — 24 & & — 72 \end{bmatrix} \quad 4 \times 18$$

$$\mathcal{X}_{(3)} = \begin{bmatrix} 1 & 2 & & 12 & 37 & & 48 \\ 13 & 14 & \cdots & 24 & 49 & \cdots & 60 \\ 25 & 26 & & 36 & 61 & & 72 \end{bmatrix} \quad 3 \times 24$$

$$\mathcal{X}_{(4)} = \begin{bmatrix} 1 & 2 & & 36 \\ 37 & 38 & \cdots & 72 \end{bmatrix} \quad 2 \times 36$$

Blocks of size $n_k \times n'_k$

$$\#\text{ of blocks} = \frac{n^d}{n_k \cdot n'_k}$$

Matrix Products

Hadamard

$$\overset{n}{\underset{m}{A}} * \overset{n}{\underset{m}{B}} = \overset{n}{\underset{m}{C}} \qquad c_{ij} = a_{ij} b_{ij}$$

Kronecker of 2 vectors

$$\underset{a}{\overset{m}{\begin{bmatrix} \\ \\ \end{bmatrix}}} \otimes \underset{b}{\overset{n}{\begin{bmatrix} \\ \\ \end{bmatrix}}} = \underset{c}{\overset{mn}{\begin{bmatrix} \\ \\ \end{bmatrix}}} \begin{bmatrix} a_1 b_1 \\ \vdots \\ a_1 b_n \\ \vdots \\ a_m b_1 \\ \vdots \\ a_m b_n \end{bmatrix}$$

Matrix Products

$A = [a_1 \cdots a_r]$ column vectors

Khatri - Rao Product

$$\underset{m}{\overset{r}{\boxed{A}}} \odot_n \overset{r}{\boxed{B}} = \overset{r}{\underset{mn}{\begin{bmatrix} a_1 \otimes b_1 & a_2 \otimes b_2 & \cdots & a_r \otimes b_r \\ & & & \\ & & & \end{bmatrix}}}$$

Kronecker Product

$$\underset{m}{\overset{n}{\boxed{A}}} \otimes_p \overset{q}{\boxed{B}} = \underset{nq}{\overset{mp}{\begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}}}$$

Properties

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

$$(A \otimes B)^{+} = A^{+} \otimes B^{+}$$

$$(A \odot B) \odot C = A \odot (B \odot C)$$

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

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

Moore - Penrose Psuedoinverse

$$A \in \mathbb{R}^{m \times n}$$

$$A = U \Sigma V^{T}$$

$$A^{+} = V \Sigma^{+} U^{T}$$

$$\Sigma^{+} = \begin{bmatrix} 1/\sigma_1 & & & & \\ & \ddots & & & \\ & & 1/\sigma_r & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix}$$

$$\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$$

$$A^{+} = (A^{T}A)^{-1}A^{T}$$

Part 2

★ Lecture 2

Tensor Inner Product

$\mathcal{X}, \mathcal{Y}$ both of size $n_1 \times n_2 \times \cdots \times n_d$

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} \mathcal{X}(i_1, i_2, \ldots, i_d) \cdot \mathcal{Y}(i_1, i_2, \ldots, i_d) = \sum_{i \in \mathcal{I}} x_i \cdot y_i$$

$O(n^d)$

Norm

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

$O(n^d)$

Tensor-times-Tensor (TTT) $Z_{jk} = \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} \mathcal{X}(j, i_2, i_3) \cdot \mathcal{Y}(i_3, i_2, k)$

$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

$\mathcal{Y} \in \mathbb{R}^{n_3 \times n_2 \times n_4}$

$O(n_1 n_2 n_3 n_4)$

$$\mathcal{Z} = \langle \mathcal{X}, \mathcal{Y} \rangle_{(2,3;2,1)} \in \mathbb{R}^{n_1 \times n_4}$$

Vector Outer Product (special case of TTT)

$a \in \mathbb{R}^m$, $b \in \mathbb{R}^n$, $X = a \circ b = \begin{bmatrix} a_1 b_1 & \cdots & a_1 b_n \\ \vdots & \ddots & \vdots \\ a_m b_1 & \cdots & a_m b_n \end{bmatrix}$

ab^T

$\mathrm{vec}(X) = b \otimes a$

$X_{ij} = a_i b_j$ $\quad O(mn)$

$a \in \mathbb{R}^m$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^p$, $X = a \circ b \circ c \in \mathbb{R}^{m \times n \times p}$

$X_{ijk} = a_i b_j c_k$ $\quad O(mnp)$

$\mathrm{vec}(X) = c \otimes b \otimes a$

$X_{(1)} = a(c \otimes b)^T$ $\quad X_{(2)} = b(c \otimes a)^T$ $\quad X_{(3)} = c(b \otimes a)^T$

$a_k \in \mathbb{R}^{n_k}$, $k \in [d]$, $X = a_1 \circ \cdots \circ a_d$, $X_{i_1 i_2 \cdots i_d} = \prod_{k=1}^{d} a_k(i_k)$

$\mathrm{vec}(X) = a_d \otimes \cdots \otimes a_1 = \bigotimes_{k=d}^{1} a_k$ $\quad O(n^d)$

$X_{(k)} = a_k \left(a_d \otimes \cdots \otimes a_{k+1} \otimes a_{k-1} \otimes \cdots \otimes a_1\right)^T = a_k \left[\bigotimes_{\substack{\ell=d \\ \ell \neq k}}^{1} a_\ell\right]^T$

Tensor-Times-Matrix (TTM)

$$\mathcal{Y} = \mathcal{X} \times_k A \qquad \mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}, \; A \in \mathbb{R}^{m \times n_k}$$

$$\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_{k-1} \times m \times n_{k+1} \times \cdots \times n_d}$$

$$\mathcal{Y}_{i_1 \cdots i_{k-1} j i_{k+1} \cdots i_d} = \sum_{i_k=1}^{n_k} \mathcal{X}_{i_1 \cdots i_d} \cdot a_{j i_k} \quad \underline{\text{or}} \quad \underset{m \times \frac{n^d}{n_k}}{Y_{(k)}} = \overset{m \times n_k}{A} \underset{n_k \times \frac{n^d}{n_k}}{X_{(k)}}$$

Cost: $O(n^d m)$

$$\mathcal{Y} = \mathcal{X} \times_1 A_1 \times_2 A_2 \cdots \times_d A_d, \quad A_k \in \mathbb{R}^{m_k \times n_k}$$

$\Rightarrow \mathcal{Y}$ is of size $m_1 \times m_2 \times \cdots \times m_d$

$$\mathcal{Y}_{(k)} = A_k^T X_{(k)} \left[\bigotimes_{\substack{l=d \\ l \neq k}}^{1} A_l \right]$$

Tensor-Times-Vector (TTV)

$$y = \mathcal{X} \; \bar{\times}_k \; a \qquad a \in \mathbb{R}^{n_k}, \quad y \in \mathbb{R}^{n_1 \times \cdots \times n_{k-1} \times n_{k+1} \times \cdots \times n_d}$$

order $d-1$

$$y_{i_1 \cdots i_{k-1} i_{k+1} \cdots i_d} = \sum_{i_k=1}^{n_k} \mathcal{X}_{i_1 \cdots i_d} \cdot a_{i_k}$$

$$\alpha = \mathcal{X} \; \bar{\times}_1 \; a_1 \; \bar{\times}_2 \; a_2 \; \cdots \; \bar{\times}_d \; a_d \Rightarrow \text{scalar}$$

$$\alpha = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathcal{X}_{i_1 \cdots i_d} \, a_1(i_1) \cdot a_2(i_2) \cdots a_d(i_d)$$

$$O(n^d)$$

Matricized Tensor Times Khatri-Rao Product (MTTKRP)

$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$

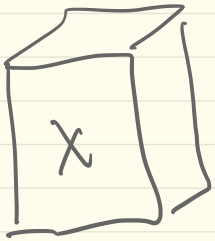
$A_k \in \mathbb{R}^{n_k \times r} \qquad k \in [d]$

MTTKRP in mode k is $\qquad O(n^d r)$

$$\underbrace{B}_{n_k \times r} = \underbrace{X_{(k)}}_{n_k \times \frac{n^d}{n_k}} \underbrace{\left(A_d \odot \cdots \odot A_{k+1} \odot A_{k-1} \odot \cdots \odot A_1\right)}_{\frac{n^d}{n_k} \times r}$$

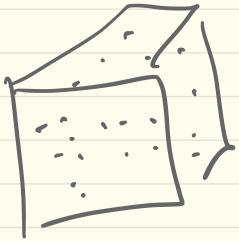
$$B(:,j) = \mathcal{X} \; \bar{\times}_1 A_1(:,j) \; \bar{\times}_2 A_2(:,j) \cdots \bar{\times}_d A_d(:,j)$$

Sparse Tensors



n^d storage

$$\frac{nnz(X)}{n^d} \approx < 1\%$$

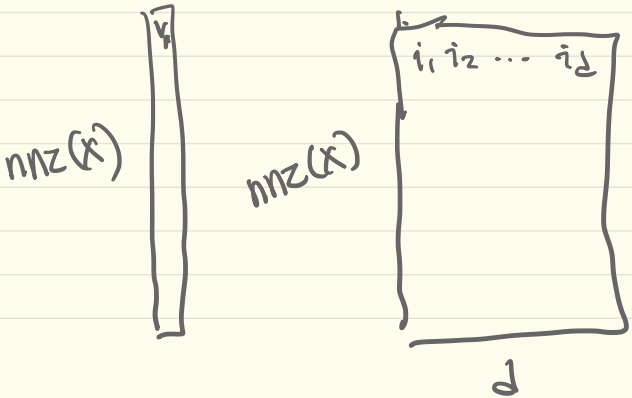


$nnz(X)$ = # nonzeros in X

value index (d-tuple)

COORDINATE FORMAT

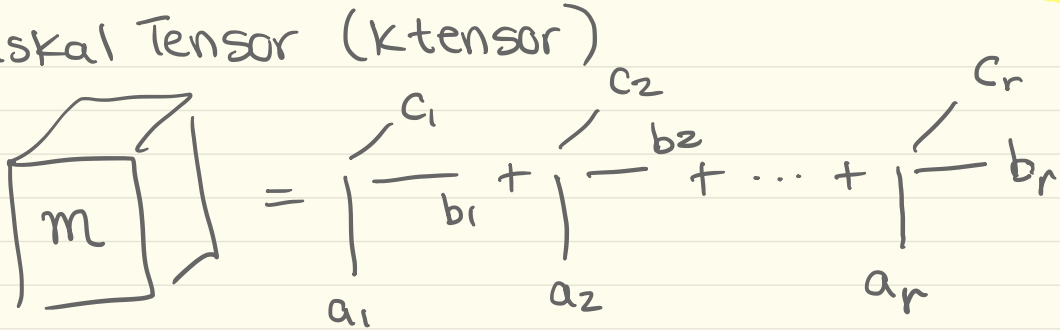
COO



$nnz(X)(d+1)$

FACTOR MATRICES

Kruskal Tensor (ktensor)



$$\mathcal{M} = \sum_{j=1}^{r} \tfrac{1}{2} a_j \circ 2 b_j \circ c_j$$

$$\mathcal{M}_{(1)} = \sum_{j=1}^{r} a_j (c_j \otimes b_j)^T$$

$$= [a_1 \cdots a_r][c_1 \otimes b_1 \cdots c_r \otimes b_r]^T$$

$$= A (C \odot B)^T$$

$n_1 \times r$

$A = [a_1 a_2 \cdots a_r]$

$B = [b_1 \cdots b_r]$

$n_2 \times r$

$C = [c_1 \cdots c_r]$

$n_3 \times r$

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

$$\mathcal{M} = [\![A_1, \ldots, A_d]\!]$$

d-way : $A_1, \ldots, A_d$ $\qquad A_k \in \mathbb{R}^{n_k \times r}$

$$\mathcal{M}_{(k)} = A_k (A_d \odot \cdots \odot A_{k+1} \odot A_{k-1} \odot \cdots \odot A_1)^T$$

Scaling Ambiguity

$$\lambda_j = \|a_j\|_2 \|b_j\|_2 \|c_j\|_2$$

$$a_j \leftarrow a_j / \|a_j\|, \quad b_j \leftarrow b_j / \|b_j\|, \quad c_j \leftarrow c_j / \|c_j\|$$

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

$$\mathcal{M} = \sum_{j=1}^{r} \lambda_j \, a_j \circ b_j \circ c_j$$

Tucker Tensor (ttensor)

$$\mathcal{M} = \mathcal{G} \times_1 U_1 \times_2 U_2 \cdots \times_d U_d$$

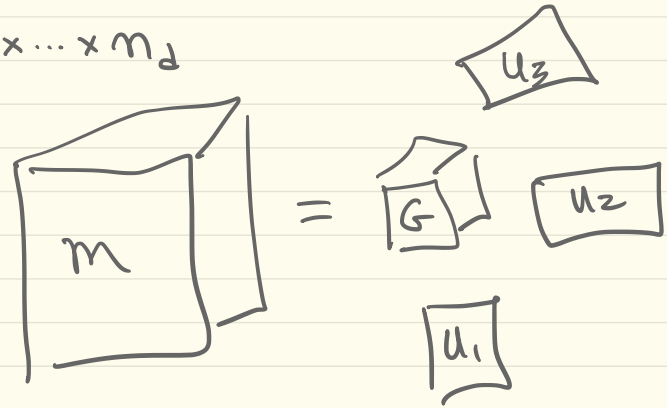
$\mathcal{M}$: $n_1 \times \cdots \times n_d$

$\mathcal{G}$: $r_1 \times \cdots \times r_d$

U_1: $n_1 \times r_1$

$$\mathcal{M}_{(1)} = U_1 G_{(1)} (U_3 \otimes U_2)^T$$

$$\mathcal{M}_{(k)} = U_k G_{(k)} \left(\bigotimes_{\substack{\ell=d \\ \ell \neq k}}^{1} U_\ell \right)^T$$



- U_k's have orthonormal columns
- $\|\mathcal{M}\| = \|\mathcal{G}\|$

CP vs Tucker

- CP factor matrices <u>not</u> orthogonal
  Tucker " " <u>are</u> orthogonal

- CP – interpretation
  Tucker – compression

- CP rank – NP-hard to determine
  Tucker rank – easy to compute
  ↳ easy to do ε-compression

- CP $\neq$ Tucker w/ diagonal core

History of CP

CP first invented in 1927

Frank Lauren Hitchcock
MIT Professor
(1875–1957)

THE EXPRESSION OF A TENSOR OR A POLYADIC AS A SUM OF PRODUCTS

By Frank L. Hitchcock

1. **Addition and Multiplication.**

Tensors are *added* by adding corresponding components. The *product* of a covariant tensor $A_{i_1 \cdots i_p}$ of order p *into* a covariant tensor $B_{i_{p+1} \cdots i_{p+q}}$ of order q is defined by writing

$$A_{i_1 \cdots i_p} B_{i_{p+1} \cdots i_{p+q}} = C_{i_1 \cdots i_{p+q}} \tag{1}$$

where the product $C_{i_1 \cdots i_{p+q}}$ is a covariant tensor of order $p+q$. When no confusion results indices may be omitted giving

$$\mathbf{AB} = \mathbf{C} \tag{1$_a$}$$

equivalent to the n^{p+q} equations (1). Boldface type is convenient for indicating that the letters do not denote merely numbers or scalars. Products of contravariant and of mixed tensors may be similarly defined.

A partial statement of the problem to be considered is as follows: to find under what conditions a given tensor can be expressed as a sum of products of assigned form. A more general statement of the problem will be given below.

2. **Polyadic form of a tensor.**

Any covariant tensor $A_{i_1 \cdots i_p}$ can be expressed as the sum of a finite number of tensors each of which is the product of p covariant vectors,

$$A_{i_1 \cdots i_p} = \sum_{j=1}^{j=h} a_{1j, i_1} a_{2j, i_2} \cdots a_{pj, i_p} \tag{2}$$

where a_{1j, i_1}, etc., are a set of hp covariant vectors. When the indices $i_1 \cdots i_p$ can be omitted this may be written

$$\mathbf{A} = \sum_{j=1}^{j=h} \mathbf{a}_{1j} \mathbf{a}_{2j} \cdots \mathbf{a}_{pj}. \tag{2$_a$}$$

The right member is now identical in appearance with a Gibbs

F. L. Hitchcock, ***The Expression of a Tensor or a Polyadic as a Sum of Products***, Journal of Mathematics and Physics, 1927

2. **Polyadic form of a tensor.**

Any covariant tensor $A_{i_1 \cdots i_p}$ can be expressed as the sum of a finite number of tensors each of which is the product of p covariant vectors,

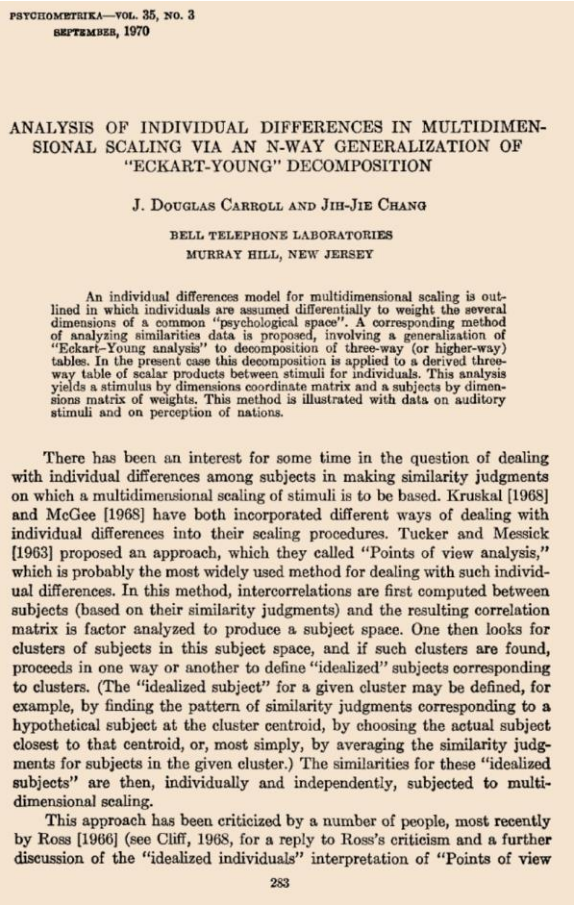
$$A_{i_1 \cdots i_p} = \sum_{j=1}^{j=h} a_{1j, i_1} a_{2j, i_2} \cdots a_{pj, i_p} \tag{2}$$

where a_{1j, i_1}, etc., are a set of hp covariant vectors. When the indices $i_1 \cdots i_p$ can be omitted this may be written

$$\mathbf{A} = \sum_{j=1}^{j=h} \mathbf{a}_{1j} \mathbf{a}_{2j} \cdots \mathbf{a}_{pj}. \tag{2$_a$}$$

CP Independently Reinvented (twice) in 1970

CANDECOMP: Canonical Decomposition



PSYCHOMETRIKA—VOL. 35, NO. 3
SEPTEMBER, 1970

ANALYSIS OF INDIVIDUAL DIFFERENCES IN MULTIDIMENSIONAL SCALING VIA AN N-WAY GENERALIZATION OF "ECKART-YOUNG" DECOMPOSITION

J. DOUGLAS CARROLL AND JIH-JIE CHANG

BELL TELEPHONE LABORATORIES
MURRAY HILL, NEW JERSEY

An individual differences model for multidimensional scaling is outlined in which individuals are assumed differentially to weight the several dimensions of a common "psychological space". A corresponding method of analyzing similarities data is proposed, involving a generalization of "Eckart–Young analysis" to decomposition of three-way (or higher-way) tables. In the present case this decomposition is applied to a derived three-way table of scalar products between stimuli for individuals. This analysis yields a stimulus by dimensions coordinate matrix and a subjects by dimensions matrix of weights. This method is illustrated with data on auditory stimuli and on perception of nations.

There has been an interest for some time in the question of dealing with individual differences among subjects in making similarity judgments on which a multidimensional scaling of stimuli is to be based. Kruskal [1968] and McGee [1968] have both incorporated different ways of dealing with individual differences into their scaling procedures. Tucker and Messick [1963] proposed an approach, which they called "Points of view analysis," which is probably the most widely used method for dealing with such individual differences. In this method, intercorrelations are first computed between subjects (based on their similarity judgments) and the resulting correlation matrix is factor analyzed to produce a subject space. One then looks for clusters of subjects in this subject space, and if such clusters are found, proceeds in one way or another to define "idealized" subjects corresponding to clusters. (The "idealized subject" for a given cluster may be defined, for example, by finding the pattern of similarity judgments corresponding to a hypothetical subject at the cluster centroid, by choosing the actual subject closest to that centroid, or, most simply, by averaging the similarity judgments for subjects in the given cluster.) The similarities for these "idealized subjects" are then, individually and independently, subjected to multidimensional scaling.

This approach has been criticized by a number of people, most recently by Ross [1966] (see Cliff, 1968, for a reply to Ross's criticism and a further discussion of the "idealized individuals" interpretation of "Points of view

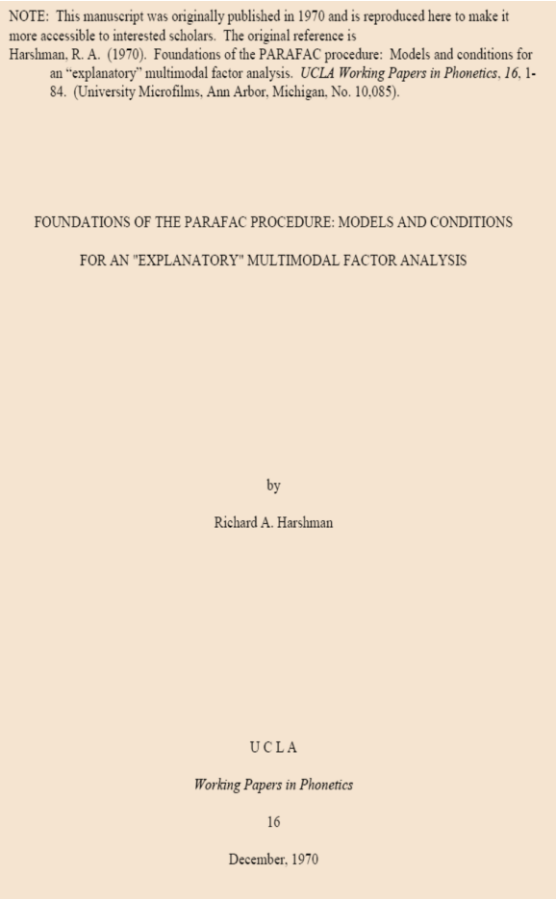
283

J. Douglas Carroll
Bell Labs
(1939-2011)

Jih-Jie Chang
Bell Labs
(1927-2007)

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

PARAFAC: Parallel Factors



NOTE: This manuscript was originally published in 1970 and is reproduced here to make it more accessible to interested scholars. The original reference is
Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, *16*, 1-84. (University Microfilms, Ann Arbor, Michigan, No. 10,085).

FOUNDATIONS OF THE PARAFAC PROCEDURE: MODELS AND CONDITIONS

FOR AN "EXPLANATORY" MULTIMODAL FACTOR ANALYSIS

by

Richard A. Harshman

UCLA

Working Papers in Phonetics

16

December, 1970

CP: CANDECOMP/PARAFAC

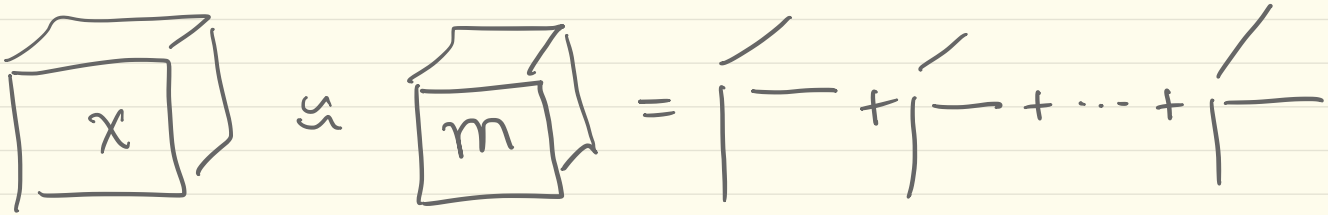
In 2000, Henk Kiers proposed this *compromise* name

CP: Canonical Polyadic

2010: Pierre Comon, Lieven DeLathauwer, and others reverse-engineered CP, revising some of Hitchcock's terminology

Many thanks to the following persons for helping me learn about Jih-Jie Chang: Fan Chung, Ron Graham, Shen Lin (husband), May Chang (niece), Lili Bruer (daughter).

CP-ALS (Alternating Least Squares)



data tensor

$n_1 \times n_2 \times \cdots \times n_d$

n^d data points

model tensor

$$m_i = \sum_{j=1}^{r} \prod_{k=1}^{d} a_k(i_k, j)$$

factor matrices

$A_k \in \mathbb{R}^{n_k \times r}$

$r \bar{n} d$ = # parameters

OPTIMIZATION PROBLEM

$$\min \quad F(\mathcal{X}, \mathcal{M}) = \sum_{i \in \mathcal{I}} (x_i - m_i)^2 \quad \text{s.t. } \mathcal{M} = [\![A_1, \ldots, A_d]\!]$$

$$A_k \in \mathbb{R}^{n_k \times r}$$

NONCONVEX

For now, assume r is given.

3-way Case

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

$$\|\mathcal{X} - \mathcal{M}\|^2 = \|X_{(1)} - \mathcal{M}_{(1)}\|^2$$

$$\min_{A} \|X_{(1)} - A(C \odot B)^T\|^2 \quad \text{Fix } B, C$$

$$\|(C \odot B)A^T - X_{(1)}^T\|$$

$$A^T = (C \odot B)^{\dagger} \cdot X_{(1)}^T$$

$$= (C^TC * B^TB)^{\dagger} (C \odot B)^T X_{(1)}^T$$

$$A = \underbrace{X_{(1)}(C \odot B)}_{\text{MTTKRP}} \underbrace{(C^TC * B^TB)^{\dagger}}_{r \times r}$$

Repeat until convergence..

$$A \leftarrow X_{(1)}(C \odot B)(C^TC * B^TB)^{\dagger}$$

$$B \leftarrow X_{(2)}(C \odot A)(C^TC * A^TA)^{\dagger}$$

$$C \leftarrow X_{(3)}(B \odot A)(B^TB * A^TA)^{\dagger}$$

convex subproblems

d-way case

$$\min_{A_k} \| X_{(k)} - M_{(k)} \|^2 \qquad M_{(k)} = A_k \left[\bigodot_{\substack{\ell=d \\ \ell \neq k}}^{1} A_\ell \right]^T$$

$$A_k = X_{(k)} \cdot Z_k \cdot V_k^{+}$$

$$Z_k = \left[\bigodot_{\substack{\ell=d \\ \ell \neq k}}^{1} A_k \right] \qquad V_k = \bigstar_{\substack{\ell=d \\ \ell \neq k}}^{1} \underbrace{(A_\ell^T A_\ell)}_{\text{Gram matrices}}$$

<u>Exercise</u>: Show cost (dense X) is $rn^d + r\left(\frac{n^d}{n_k}\right) + r^2(n+1) + r^3$

CP-ALS

$$\text{fit} = 1 - \|\mathcal{X} - \mathcal{M}\| / \|\mathcal{X}\|$$ proportion of data described by model

for $\ell = 1, 2, \ldots$ do

oldfit $\leftarrow$ fit

for $k = 1, \ldots, d$

$B \leftarrow X_k \bigodot_{\substack{\ell=d \\ \ell \neq k}} A_\ell$ MTTKRP

$V \leftarrow \bigstar_{\substack{\ell=d \\ \ell \neq k}} A_\ell^T A_\ell$

$A_k \leftarrow B / V$ $(A_k^T = V^+ B^T)$

$\lambda_j = \| A_k(:, j) \|$

$A_k(:, j) \leftarrow A_k(:, j) / \lambda_j$

end

fit $= 1 - \|\mathcal{X} - \mathcal{M}\| / \|\mathcal{X}\|$

end if $|\text{fit} - \text{oldfit}| < \tau$, then quit

Notes

- $\|X - M\|^2 = \|X\|^2 + \|M\|^2 - 2\langle X, M\rangle$
- $\langle X, M\rangle = e^T (B \star A_d)\lambda$
- Save $A_k^T A_k$ to reuse

Part 3

Workshop on Tensor Decompositions and Applications, CIRM, Luminy, Marseille, France, Aug. 29–Sep. 2, 2005

CP-OPT

$$\min \underbrace{\|X - \mathcal{M}\|^2}_{F} + \rho_1 \|A_1\|_F + \cdots + \rho_d \|A_d\|_F \quad \text{s.t.} \quad \mathcal{M} = [\![A_1, \ldots, A_d]\!], \; A_k \in \mathbb{R}^{n_k \times r}$$

- ALS advantage - convex subproblems
- "all-at-once" optimization, i.e., no alternating ...
  advantage - missing data, different loss functions, etc.

GRADIENT

$$\underbrace{G_k}_{n_k \times r} = \frac{\partial F}{\partial A_k} = \underbrace{X_{(k)} \bigodot_{\substack{\ell=d \\ \ell \neq k}}^{1} A_\ell}_{\text{MTTKRP}} - A_k \left[\bigstar_{\substack{\ell=d \\ \ell \neq k}}^{1} (A_\ell^T A_\ell) \right] \quad \text{for } k \in [d]$$

$O(rn^d)$

Note 1: Setting $G_k = 0$ + solving for A_k gives same equation as ALS

Note 2: Scaling ambiguity can be a nuisance, can be fixed with regularization

Plug into Optimization

- I like limited-memory BFGS with bound constraints

  L-BFGS

  - 1st order - requires gradients

$F(\theta)$ θ is a vector! of size p

$G(\theta) \in \mathbb{R}^p$

ktensor / tovec(m, ?)

$$\theta = \begin{bmatrix} \text{vec}(A_1) \\ \text{vec}(A_2) \\ \vdots \\ \text{vec}(A_d) \end{bmatrix} \quad r\bar{n}d = r\sum_{k=1}^{d} n_k$$

update

- Note: Can do Newton's method

Missing Data

What if some entries of X are missing?

<u>Option 1</u>: Fill in missing data w/ zero.
Solve as usual.

<u>Option 2</u>: "Expectation Maximation"

Iteration
- Use current $M = [\![A, B, C]\!]$ to fill in missing entries of $X \to \bar{X}$
- Recompute new M using $\bar{X}$ ★

<u>Option 3</u>: Ignore the missing values

$$F(X, M) = \sum_{i \in \mathcal{I}} w_i \cdot (x_i - m_i)^2 \qquad w_i = \begin{cases} 1 & \text{if entry } i \text{ known} \\ 0 & \text{otherwise (missing)} \end{cases}$$

$$G_k = \frac{\partial F}{\partial A_k} = \left[W_{(k)} \circledast (X_{(k)} - M_{(k)}) \right] \left(\bigodot_{\substack{\ell = d \\ \ell \neq k}}^{1} A_\ell \right)$$

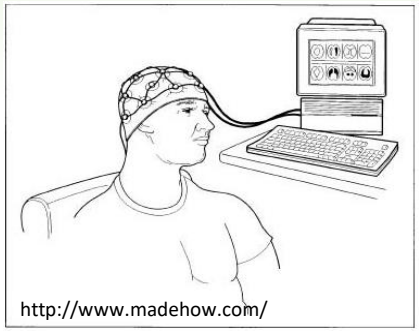
Complexity:
$O(rnd)$

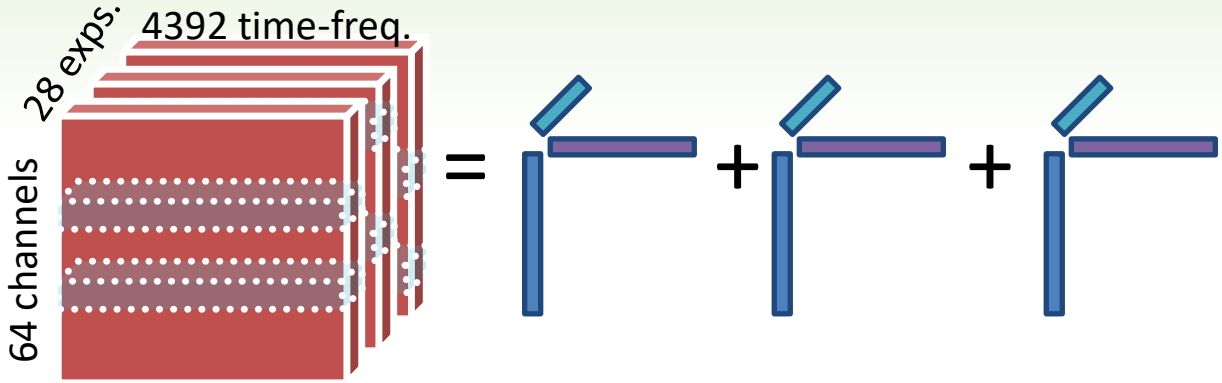
W sparse: $O(\mathrm{nnz}(W) \cdot r)$

sparse vs. scarce

EEG Results w/ Missing Data

Brain dynamics can be captured even extensive missing channels

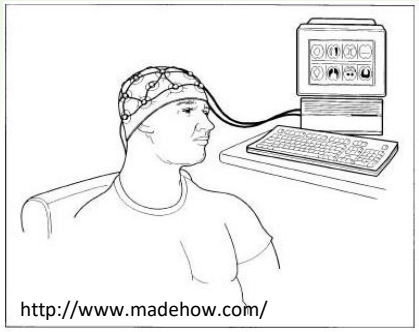


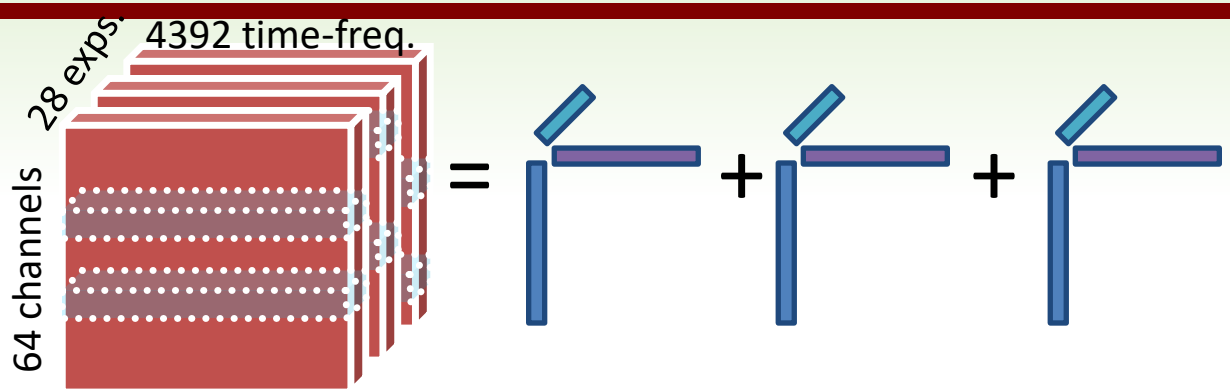


Number of Missing Channels	Replace Missing Entries with Mean	Ignore Missing Entries
1	0.98	1.00
10	0.82	0.98
20	0.67	0.95
30	0.45	0.89
40	0.24	0.65

Acar, Dunlavy, Kolda & Mørup 2010 & 2011

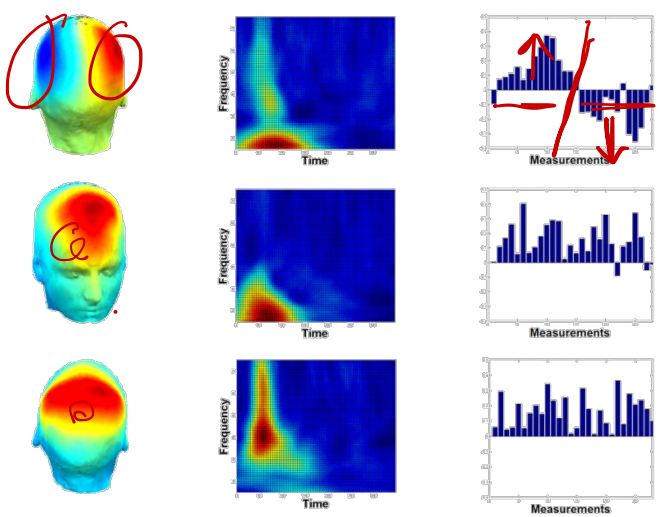
Brain dynamics can be captured even extensive missing channels





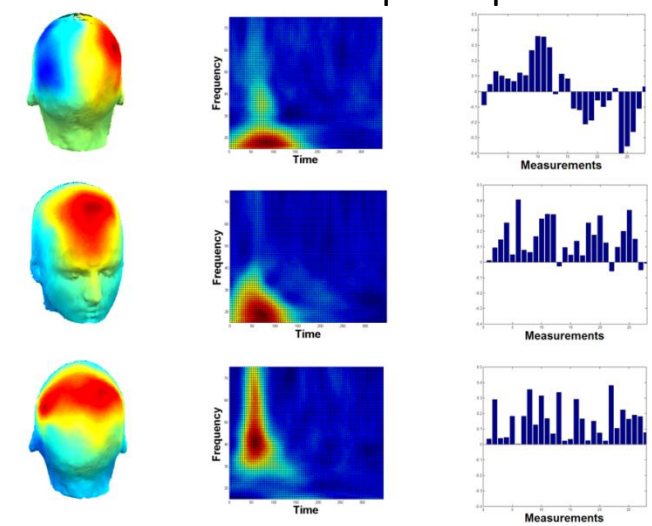
No Missing Data

channel | time-freq | experiments



30 Chan./Exp. Missing

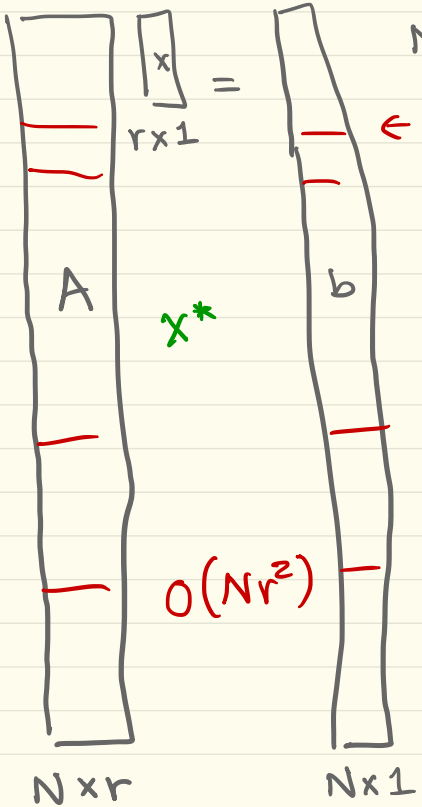
channel | time-freq | experiments



Acar, Dunlavy, Kolda & Mørup 2010 & 2011

CP-ARLS R = randomized

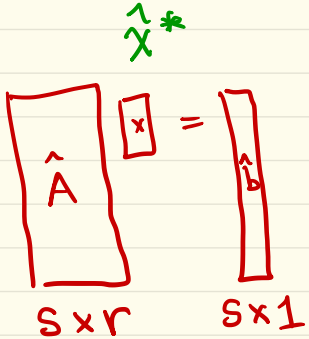
Randomized Least Squares



$N \gg r$

← pick s random rows of A + corresponding entries of b

$O(Nr^2)$



Will this work?

$$\|Ax^* - b\| \approx \|A\hat{x}^* - b\|$$

Works if matrix A is <u>incoherent</u> or if sample by <u>leverage scores</u>

<u>Defn</u> : Given $A \in \mathbb{R}^{N \times r}$, $N > r$. The leverage score of row i of A is

$$\ell_i(A) = \| U(i,:) \|^2$$

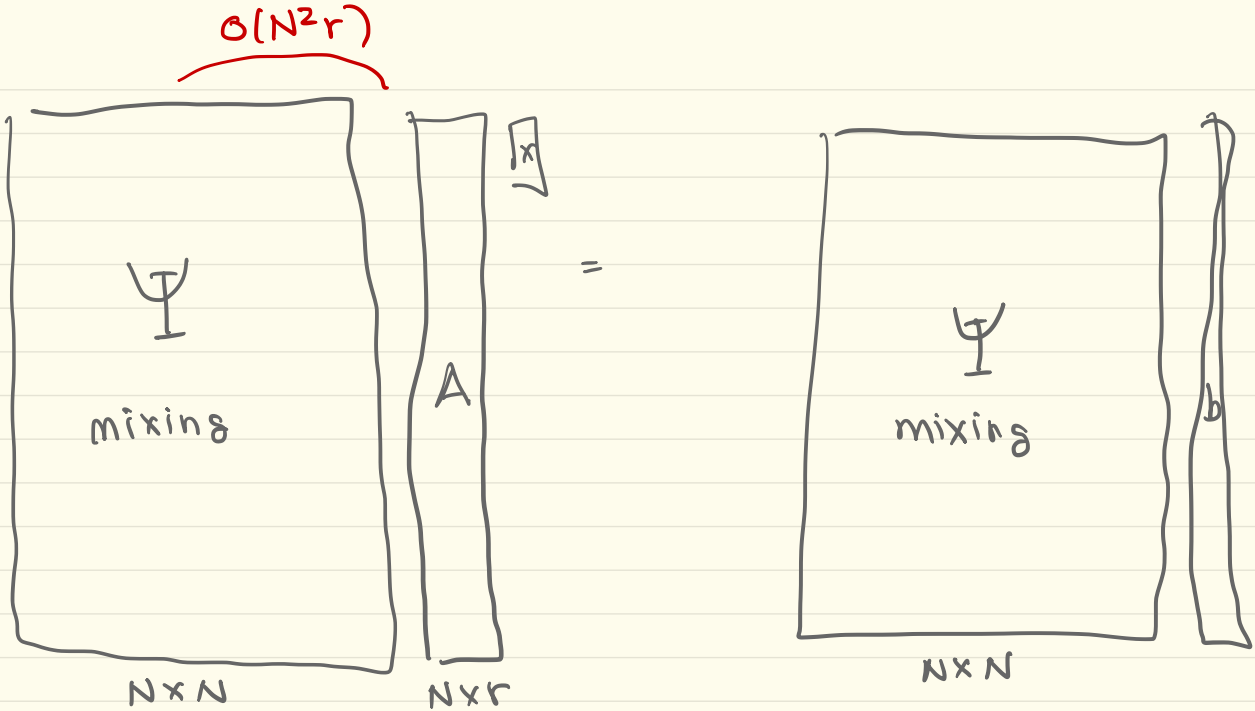
where U comes from a rank-r SVD of A. $(A = U \Sigma V^T)$

<u>Defn</u> : Coherence is the maximum leverage score of A:

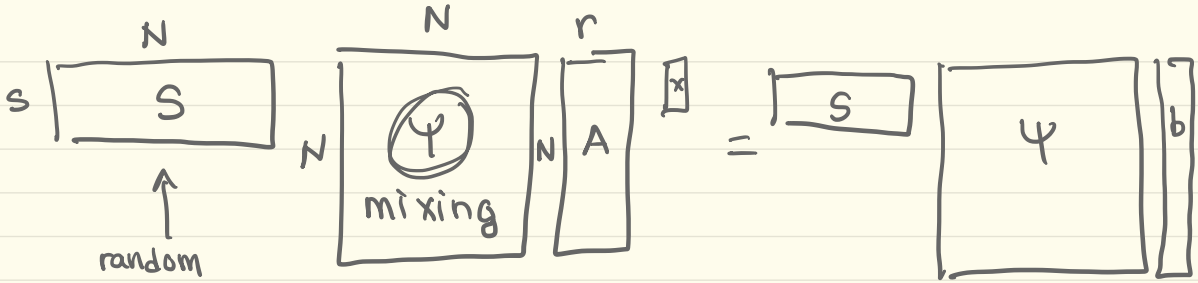
$$\mu(A) := \max_{i \in [N]} \ell_i(A)$$

$$\frac{1}{r} \leq \mu(A) \leq 1$$

<u>Defn</u> : A is incoherent if $\mu(A) \approx 1/r$.



$\Psi_{ij} \sim \mathcal{N}(0,1)$ (randn)



random
s rows
of N×N
identity
matrix

$O(sNr) \to O(Nr^2)$ $[s > r]$

cost to
compute
$(S\Psi)\cdot A$

Fast choice of Ψ:

$$\Psi = \mathcal{F} D$$

cost of $(\Psi A) = N \log N$

Matrix Sketching

Let $E \subset \mathbb{R}^N$ be a set of r points.
A Johnson-Lindenstrauss Transform (JLT) is a random linear map Φ such that ΦE has minimal distortion, i.e.,

$$(1-\varepsilon)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\varepsilon)\|x\|^2$$

with high probability.

Example: $\Phi \in \mathbb{R}^{s \times N}$ w/ $s = O(\varepsilon^{-2} \log r)$

$\phi_{ij} \sim N(0, 1/s)$ is a JLT

Gaussian Testing Matrix

$O(sN)$

Fast JLT (FJLT)

FJLT employs fast matrix-vector multiply to reduce the cost. For instance, choosing

$$\Phi = S \mathcal{F} D \in \mathbb{R}^{s \times N} \quad \text{with} \quad s = O(\varepsilon^{-2} \cdot \underline{\log N} \cdot \log r)$$

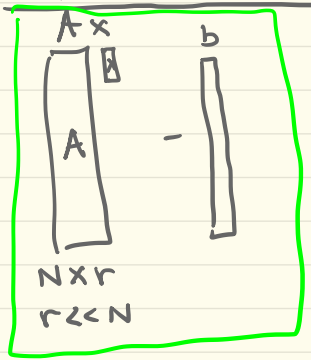
$S \in \mathbb{R}^{s \times N}$ = s random rows of $N \times N$ identity

$\mathcal{F} \in \mathbb{R}^{N \times N}$ = FFT of size N

$D \in \mathbb{R}^{N \times N}$ = diagonal matrix with random $\underbrace{+/- 1}_{\text{Rademacher}}$ on diagonal

Cost to compute Φx is $O(N \log N)$

Application Least Squares



Option ∅ : Direct solve $O(Nr^2)$

Option 1 : $SAx - Sb$ $\quad O(sr^2)$

$S = s$ rows of $N \times N$ identity

(But only works if A is incoherent.)

✗ **Option 2** : $\Phi A x - \Phi b$, Φ = random Gaussian testing matrix $\in \mathbb{R}^{s \times N}$ $\quad s = O(\varepsilon^{-2} \log r)$

Computing ΦA costs $O(Nsr)$ (no savings)

Option 3 : FJLT $\Phi = S \mathcal{F} D \in \mathbb{R}^{\bar{s} \times N}$ $\quad \bar{s} = O(\varepsilon^{-2} \log r \log N)$ $\quad O(rN\log N)$

Cost to compute $\Phi A = S(\mathcal{F}(\underbrace{DA}_{O(N)}))$

$O(rN\log N)$

$O(sr)$

Tensor Labs for Gene Golub SIAM Summer School, 2019

Tamara G. Kolda, Sandia National Labs

Many of this material is taken from prior labs that have been written with the help of co-presenters **Danny Dunlavy** and **Kina Winoto**. Many thanks also to various collagues and interns that have contributed as well.

This work was supported by the DOE ASCR Applied Mathematics program, and the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia National Laboratories is a multi-program 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-NA-0003525.

Order of Exercises

Lab 1

- [ ] MATLAB Basics
- [ ] Software Install
- [ ] CP via Alternating Least Squares (CP-ALS) - also introduces Amino Acids dataset and includes nonnegative least squares challenge
- [ ] Creating Artificial Test Problems - includes adding noise, creating high congruency

Lab 2

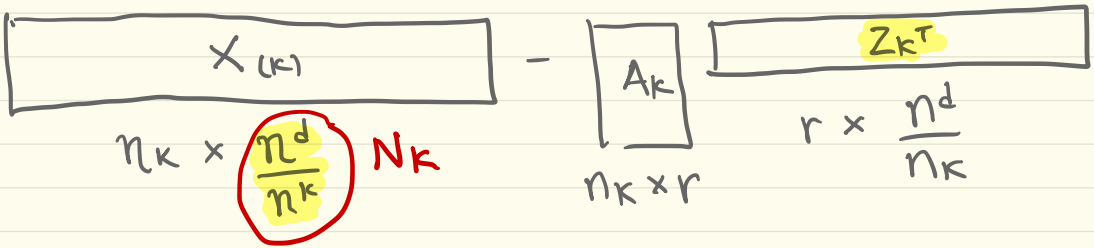
- [ ] CP via Alternating Randomized Least Squares (CP-ARLS, aka CP-RAND) - also introduces Hazardous Gas dataset
- [ ] CP via Direct Optimization (CP-OPT)
- [ ] CP with Missing Data (CP-WOPT)

Lab 3

- [ ] Generalized CP (GCP) via Direct Optimization - introduces Mouse Neuron dataset
- [ ] GCP via Stochastic Optimization - introduces Chicago Crime dataset
- [ ] Symmetric Tensor Factorization via Direct Optimzation - on data from a mixture of Gaussians

Part 4

CP-ALS : Inner iteration k : $\| X_{(k)} - A_k Z_k^T \|$ $\quad Z_k = \bigodot_{\substack{\ell=d \\ \ell \neq k}}^{1} A_\ell$



- Solve this least squares problem: $O(n_k \cdot \frac{n^d}{n_k} r^2) = O(n^d r^2)$
- Special structure from Khatri-Rao Product (KRP) $Z_k = O(n^d r)$
- Randomized least squares could reduce cost to $O(n_k \cdot s \cdot r^2)$

  BUT Forming KPP Z_k costs $O\left(\frac{n^d}{n_k} \cdot r\right)$

- Can matrix sketching help? Yes, with caveats

CP-ARLS (R = randomized) aka CPRAND

$\| X_{(k)} - A_k Z_k^T \|$ $\quad S$ = s random rows of $N_k \times N_k$ identity matrix

$$\| X_{(k)} S^T - A_k Z_k^T S^T \|$$

$$A_k = \underbrace{X_{(k)} S^T}_{\bar{X}^T} / \underbrace{Z_k^T S^T}_{\bar{Z}^T}$$

<u>Trick #1</u>: Don't unfold X & then select columns. Instead, just grab entries from X. Avoids data movement which is expensive.

<u>Trick #2</u>: Don't form KRP Z_k. Instead just grab needed rows. Index $\ell \to$ multi-index $(i_2, \ldots, i_d)$ if $k=1$.

Sampled Khatri Rao (SKR)

$$Z_k(\ell, :) = \bigstar_{\ell=2}^{k} A_\ell(i_\ell, :)$$

Cost (sdr)

doesn't depend on N_k!

SKR Algorithm

function $Z_S = SKR(S, \{A_k\}, k)$

- Retrieve $s \times (d-1)$ array of sampled indices from S = idxs
- Z_S = ones(s, r)
- for $\ell = 1, \ldots, k-1, k+1, \ldots d$

  $\hat{A}_k = A_k(\text{idxs}(:, \ell), :)$

  $Z_S = Z_S * \hat{A}_k$

  end
- Return Z_S

CP-ARLS (no mixing)

- initial guess for the factor matrices $\{A_k\}_{k=1}^{d}$
- Repeat until convergence...
  - For $k = 1, \ldots, d$
    - Random Sample of indices $S \in \mathbb{R}^{s \times N_k}$
    - $Z_S = SKR(S, \{A_k\}, k)$
    - $X_S^T = S X_{(k)}^T$
    - $A_k = X_S^T \backslash Z_S^T$
    - Normalize columns
  - End
- End

Check Convergence (Trick #5)

- Compute the loss at a $\boxed{\text{fixed subset}}$ of tensor entries (fixed across all iterations)
- Not checking every iteration
- Let $\Omega \subset \mathcal{I}$ be the subset of indices

$$\hat{F} = \frac{n^d}{|\Omega|} \sum_{i \in \Omega} (x_i - m_i)^2$$

Is Z_k incoherent? How can we even check it?

<u>Lemma</u> (Battaglino et al 2018)

Given $A \in \mathbb{R}^{I \times J}$, $B \in \mathbb{R}^{K \times J}$. $\mu(A \odot B) \leq \mu(A) \cdot \mu(B)$

$\Rightarrow$ Not mixing is okay <u>if</u> factor matrices are themselves incoherent.

Default is to do <u>mixing</u>.

* High preprocessing cost
* Low per iteration cost

$$\|Ax - b\|$$

$$\downarrow$$

$$\left\| \begin{bmatrix} re(A) \\ im(A) \end{bmatrix} x = \begin{bmatrix} re(b) \\ im(b) \end{bmatrix} \right\|$$

keeps x real

* We use FFT $\Rightarrow$ complex-valued equations, but want real solution.

Mixing on Z_k...

$$\Phi = S \mathcal{F} D$$

cost would be $O(r \cdot N_k \log N_k)$ to form ΦZ_k

Instead,

$$\Phi = S \left(\bigotimes_{\substack{\ell=d \\ \ell \neq k}}^{1} \mathcal{F}_\ell \right) \left(\bigotimes_{\substack{\ell=d \\ \ell \neq k}}^{1} D_\ell \right)$$

$\mathcal{F}_\ell = n_\ell \times n_\ell$ FFT

$D_\ell = n_\ell \times n_\ell$ ± 1 random diagonal

$$\Phi Z_k = S \left[\bigodot_{\substack{\ell=d \\ \ell \neq k}}^{1} \underbrace{\mathcal{F}_\ell D_\ell A_\ell}_{\substack{\text{costs} \\ n_\ell \log n_\ell \\ \text{for } \ell\text{th factor}}} \right]$$

Total cost $=$

$$O\left(r \cdot \sum_{\ell \neq k} n_\ell \log n_\ell\right)$$

$$\ll O(r \cdot N_k \log N_k)$$

If $n = n_1 = \cdots = n_d$

$$r \cdot (d-1) \cdot n \log n \quad \text{vs.} \quad r \cdot n^{(d-1)} \cdot (d-1) \log n.$$

What about the other side of the equation- $X_{(k)}$.

Need to compute $\Phi_k X_{(k)}^T$

$$\Phi_k = S\left(\bigotimes_{\ell \neq k} \mathcal{F}_\ell\right)\left(\bigotimes_{\ell \neq k} D_\ell\right)$$

Fixed through out all iterations

Key is to precompute $\hat{X} = X \times_1 \mathcal{F}_1 D_1 \times_2 \mathcal{F}_2 D_2 \cdots \times_d \mathcal{F}_d D_d$

Upfront cost: $O(n^d \log n^d)$

Observe $\Phi_k X_{(k)}^T = (S \hat{X}_{(k)}^T) \cdot \mathcal{F}_k^* D_k$

sample first

Per iteration work: $O(s n_k \log n_k)$

<u>Note</u>: S changes every inner iteration <u>but</u> D_k's are fixed through all iterations

<u>Note</u>: Is Φ_k an FJLT?

Ruhui Jin, R. Ward, K.

<u>Lemma</u>: For Φ as described, $s = O(\varepsilon^{-2} \log N_K \cdot \log^{2d-3} r)$.

Related Ideas

- Sketch entire tensor just <u>once</u>

$$X_s = X \times_1 \Phi_1 \times_2 \Phi_2 \cdots \times_d \Phi_d$$

- Samples blocks of tensor

Generalized CP (GCP) Decomposition

Standard CP : $\min \|X - M\|^2$ s.t. $\text{rank}(M) \leq r$

$$\sum_{i \in \mathcal{I}} \underbrace{(x_i - m_i)^2}$$

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

GCP : $\min \sum f(x_i, m_i)$ s.t. $\text{rank}(M) \leq r$

PDFs / PMFs

$p(x|\theta)$ = probability distribution function (PDF) where

$$\int_{-\infty}^{x} p(t|\theta)\,dt = \text{Prob}(X \le x \mid \theta)$$

x continuous

$p(x|\theta)$ = probability mass function (PMF) where

$$\sum_{t \le x} p(t|\theta) = \text{Prob}(X \le x \mid \theta)$$

"Likelihood" is about estimating parameter θ given observed data.

Given x, $p(x|\theta)$ is the likelihood of θ.

maximize
LOG-LIKELIHOOD

liklihood for multiple x_i, θ_i

$$\prod_i p(x_i|\theta_i) \Rightarrow \sum_i \log p(x_i|\theta_i)$$

minimize negative los likelihood (NLL)

From PDF / PMF to a Loss Function

Natural parameter

$$p(x \mid \theta) \quad \text{and} \quad \ell(\theta) = m$$

PDF
PMF

data

link

model parameter

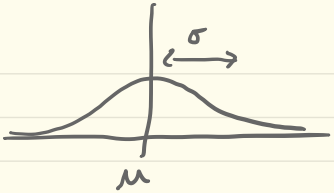
$$\min \; - \sum_i \log p(x_i \mid \theta_i)$$

$$\min \; - \sum_i \underbrace{\log p(x_i \mid \ell^{-1}(m_i))}_{f(x_i, m_i)}$$

Part 5

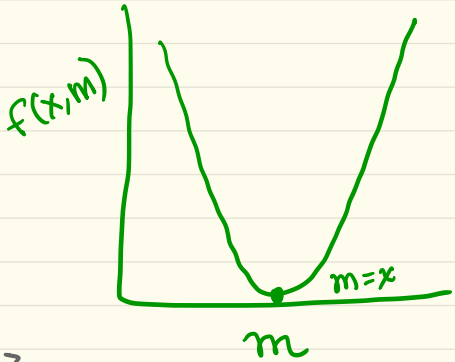
Gaussian Case

$$p(x \mid \mu, \underset{\text{constant}}{\sigma}) = \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}$$



$\ell(\mu) = \mu$ identity link

NLL : $-\log\left(\frac{e^{-(x-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}\right)$



$$\frac{(x-\mu)^2}{\cancel{2\sigma^2}} + \cancel{\frac{1}{2}\log 2\pi\sigma^2}$$

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

$$F(x,m) = \sum_i (x_i - m_i)^2$$

$x_i \sim N(m_i, \sigma)$

$x_i \sim m_i + N(0, \sigma)$

Bernoulli

$x \in \{0, 1\}$ Natural parameters: $\rho = \text{Prob}(x=1)$

"Coin Flip" PMF: $p(x \mid \rho) = \rho^x (1-\rho)^{(1-x)}$

Odds Link: $\ell(\rho) = \rho / 1-\rho$

Odds	Prob
1	1/2
.25	.2
4	.8

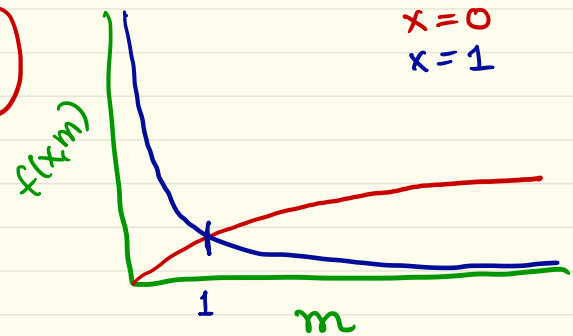
$$\text{NLL} = -x \log \rho - (1-x) \log(1-\rho)$$

$$= -\log(1-\rho) - x \log \rho + x \log(1-\rho)$$

$$= \log \underbrace{\frac{1}{(1-\rho)}}_{1+m} - x \log \underbrace{\frac{\rho}{1-\rho}}_{m}$$

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

$$m > 0$$



Bernoulli

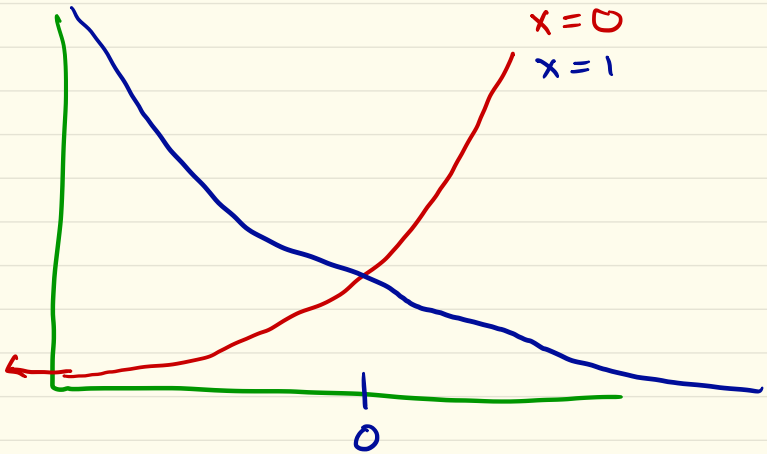
$x \in \{0, 1\}$ Natural parameters: $\rho = \text{Prob}(x=1)$

"Coin Flip" PMF: $p(x \mid \rho) = \rho^x (1-\rho)^{(1-x)}$

$\ell(\rho) = \log\left(\rho / 1-\rho\right)$ log-odds "logit"

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

Log Odds	Prob
0	.5
−1.39	.2
1.39	.8



Poisson

$x \in \{0, 1, 2 \dots\} = \mathbb{N}$ count data

λ = mean value $\in \mathbb{R}$, $\lambda \geq 0$

PMF $p(x \mid \lambda) = \dfrac{e^{-\lambda} \lambda^{x}}{x!}$, $\ell(\lambda) = \lambda$, $\ell(\lambda) = \log \lambda$

$f(x, m) = m - x \log m$ ..... $m > 0$

$f(x, m) = e^{m} - xm$ ..... $m \in \mathbb{R}$

Huber Loss "Robust"

$$f(x, m \mid \Delta) = \begin{cases} (x-m)^2 & \text{if } |x-m| \leq \Delta \\ 2\Delta |x-m| - \Delta^2 & \text{otherwise} \end{cases}$$

- Gamma)
- Rayleigh)
- Neg. Binomial)

(β) Beta Divergence

$$f(x, m; \beta) = \left(\frac{1}{\beta}\right) m^{\beta} - \left(\frac{1}{(\beta - 1)}\right) x m^{\beta - 1}$$

$\beta \in \mathbb{R} / \{0, 1\}$

Gradients

Lemma: Let $M = AB^T$, $A \in \mathbb{R}^{n \times r}$, $B \in \mathbb{R}^{p \times r}$. Then

$$\frac{\partial m_{ik}}{\partial a_{\tilde{i}j}} = \begin{cases} b_{\ell j} & \text{if } i = \tilde{i} \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } \begin{array}{l} i, \tilde{i} \in [n] \\ \ell \in [p] \\ j \in [r] \end{array}$$

Proof $\quad m_{i\ell} = \sum_{j=1}^{r} a_{ij} \cdot b_{\ell j}$

Lemma: $X \in \mathbb{R}^{n \times p}$, $W \in \mathbb{R}^{n \times p}$, $A \in \mathbb{R}^{n \times r}$, $B \in \mathbb{R}^{p \times r}$.

$$\tilde{F}(M, X, W) = \sum_{i=1}^{n} \sum_{\ell=1}^{p} w_{i\ell} \cdot f(x_{i\ell}, m_{i\ell}) \quad \text{s.t. } M = AB^T$$

Then $\frac{\partial \tilde{F}}{\partial A} = \underbrace{Y}_{n \times p} B \in \mathbb{R}^{n \times r}$ where $y_{i\ell} = w_{i\ell} \cdot \frac{\partial f}{\partial m}(x_{i\ell}, m_{i\ell})$ $\quad i \in [n], \ell \in [p]$

Proof: $\frac{\partial \tilde{F}}{\partial a_{ij}} = \sum_{\tilde{i}=1}^{n} \sum_{\ell=1}^{p} w_{\tilde{i}\ell} \cdot \underbrace{\frac{\partial f}{\partial a_{ij}}(x_{i\ell}, m_{i\ell})}_{\frac{\partial f}{\partial m_{\tilde{i}\ell}}(x_{\tilde{i}\ell}, m_{\tilde{i}\ell}) \cdot \frac{\partial m_{\tilde{i}\ell}}{\partial a_{ij}}}$

Thm: $F(X, M, W) = \sum_i w_i f(x_i, m_i)$ s.t. $\text{rank}(M) \le r$

$M = [\![A_1, \dots, A_d]\!]$

Then $\dfrac{\partial F}{\partial A_k} = Y_{(k)} \cdot Z_k$ where $y_i = w_i \cdot \dfrac{\partial f}{\partial m}(x_i, m_i)$

MTTKRP

$$Z_k = \bigodot_{\substack{\ell = d \\ \ell \neq k}}^{1} A_\ell .$$

Proof: $M_{(k)} = A_k Z_k$ use prev. le

Notes: Y is dense even if X is sparse. $O(rn^d)$

Y is sparse <u>if</u> W is sparse. $O(r \cdot nnz(W))$

$$\frac{\partial F}{\partial A_k} = Y_{(k)} Z_k$$

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

$$\frac{\partial f}{\partial m} = \cancel{2}(x-m)$$

$$= \left(X_{(k)} - M_{(k)}\right) Z_k$$

$$= X_{(k)} Z_k - \underbrace{M_{(k)}}_{A_k Z_k^T} Z_k$$

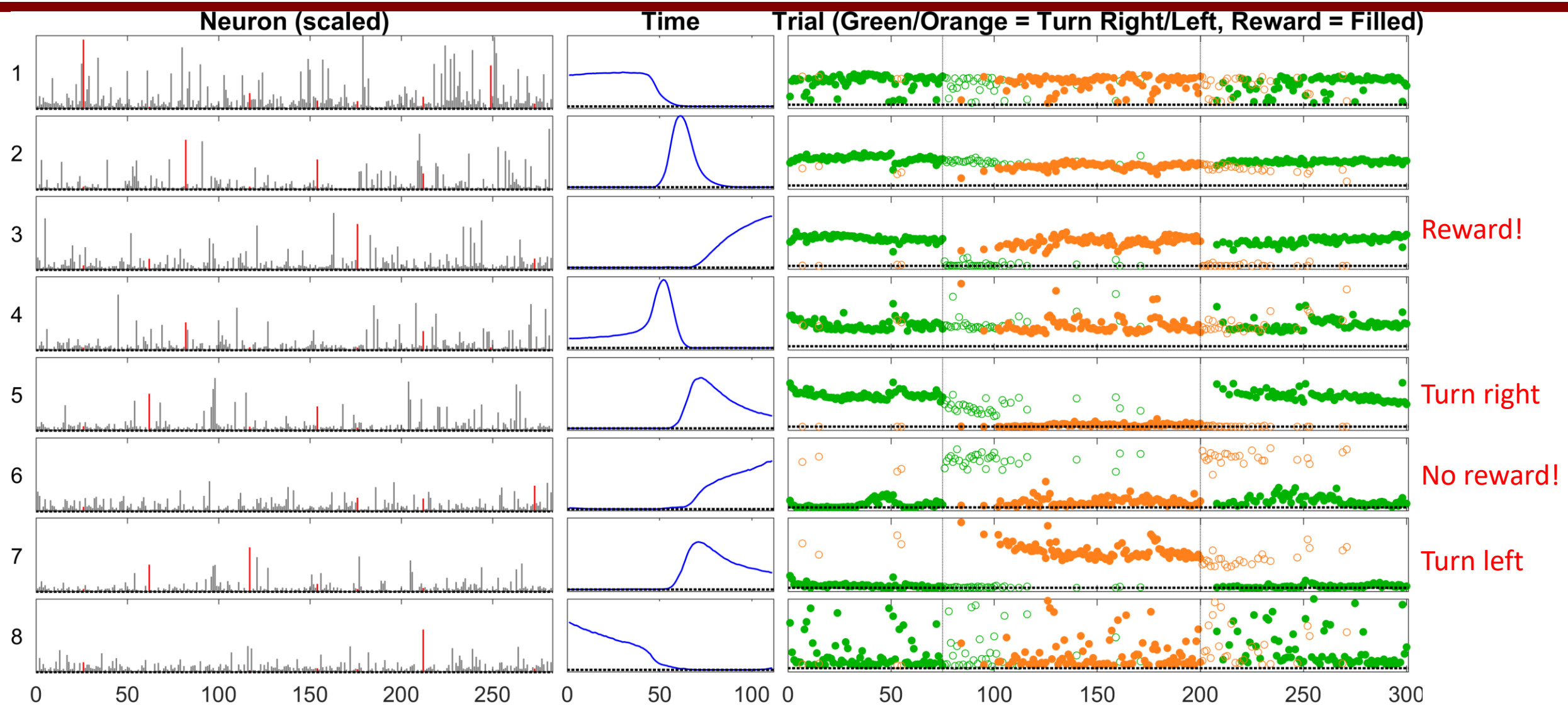
$$= \underline{X_{(k)} Z_k} - \underline{A_k} \underbrace{Z_k^T Z_k}_{\prod_{\substack{l=d \\ l \neq k}} A_l^T A_l}$$

sparse MTTKRP if X sparse

$O(r \cdot nnz(X))$

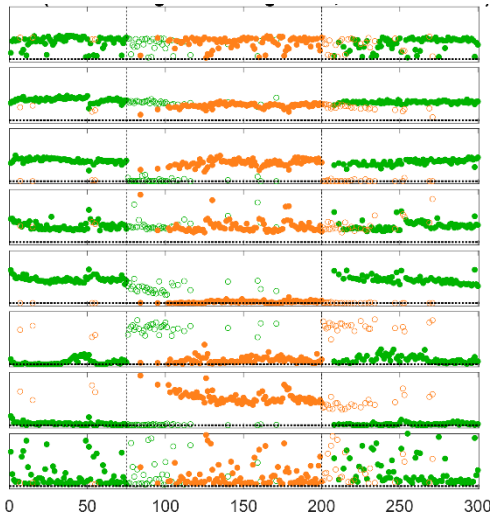
GCP Results

GCP Decomposition with Beta Divergence ($\beta = 0.5, f(x,m) = \sqrt{m} + x/\sqrt{m}$)



Regression Using GCP Factors on Trial Mode

Trial Factor Matrix is 300×8

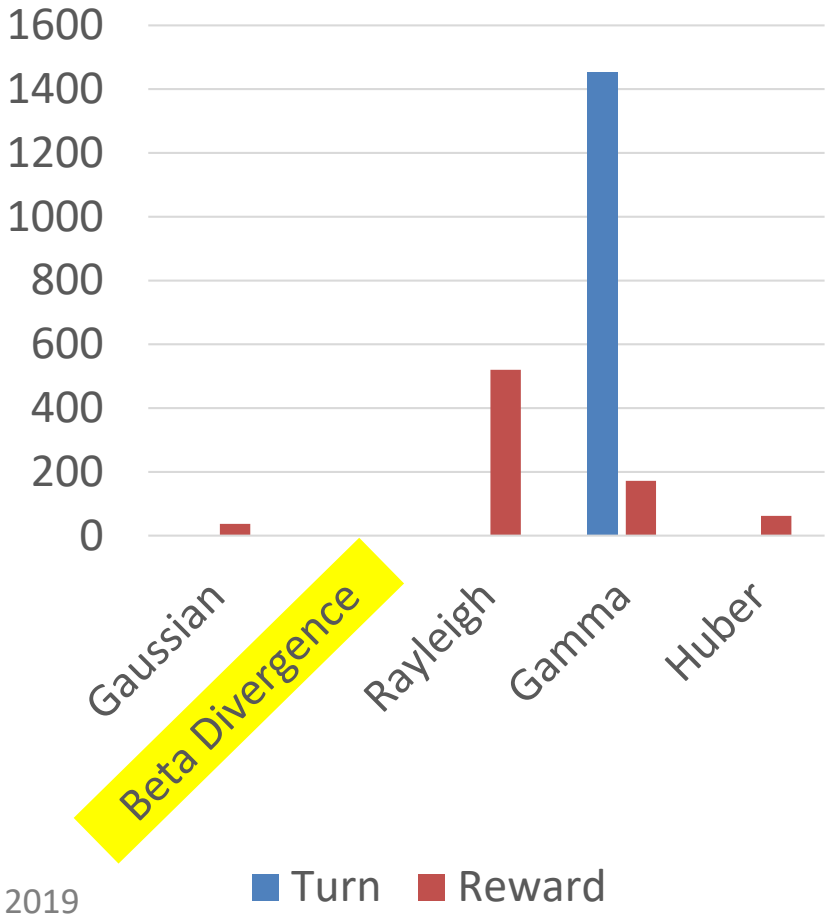


$$\min_{\boldsymbol{\beta}} \|\mathbf{A}_3^{\text{train}}\boldsymbol{\beta} - \mathbf{y}^{\text{train}}\|$$

$$\hat{\mathbf{y}}^{\text{test}} = [\mathbf{A}_3^{\text{test}}\boldsymbol{\beta} \geq 0.5]$$

Look at predicting turn and reward.
Split into two groups of 150 trials.
Train regression model with 1st group.
Test with 2nd group.
Repeat 100 times.

Regression Errors in 100 Trials (15000 predictions)



Hong, Kolda, Duersch, SIAM Review, 2019

Stochastic Optimization

$\min F(\theta)$

gradient descent (GD): $\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla F(\theta^{(t)})$

step length

stochastic GD: $\theta^{(t+1)} = \theta^{(t)} - \alpha \hat{G}(\theta^{(t)})$

small step size

$$\mathbb{E}(\hat{G}(\theta^{(t)})) = G(\theta^{(t)})$$

expectation

ADAM (Kingma + Ba)

Stochastic Gradient for GCP

Standard $\quad G_k = Y_{(k)} Z_k \qquad O(rn^d)$

Stochastic $\quad \hat{G}_k = \hat{Y}_{(k)} Z_k, \quad nnz(\hat{Y}) \le s \ll n^d \qquad O(r \cdot s)$

$\hookrightarrow O(rn)$

Option 1 : Uniform sample entries, s entries w/ replacement

- $\hat{s}_i$ = # times entry i is picked (usually $\emptyset$)
- $\hat{y}_i = \hat{s}_i \cdot \frac{n^d}{s} \cdot y_i$

Proof : $\mathbb{E}(\hat{y}_i) = \mathbb{E}(\hat{s}_i) \cdot \cancel{\frac{n^d}{s}} \cdot y_i$

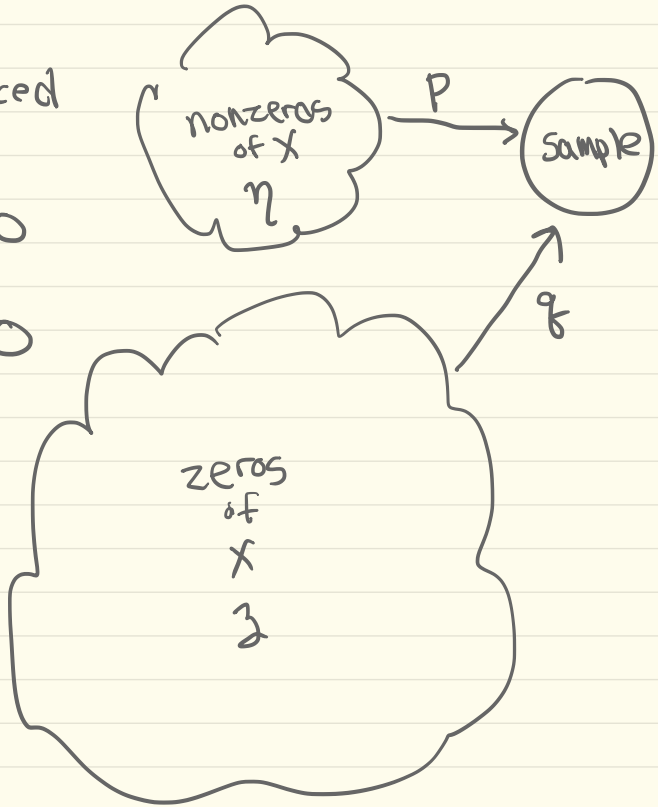
$\uparrow$

$\cancel{\frac{s}{n^d}}$

- How to pick s? $\quad s \approx O(\sum n_k) = O(n)$
- If X is sparse, then we might always pick zero's.

Option 2 : Stratified Sampling

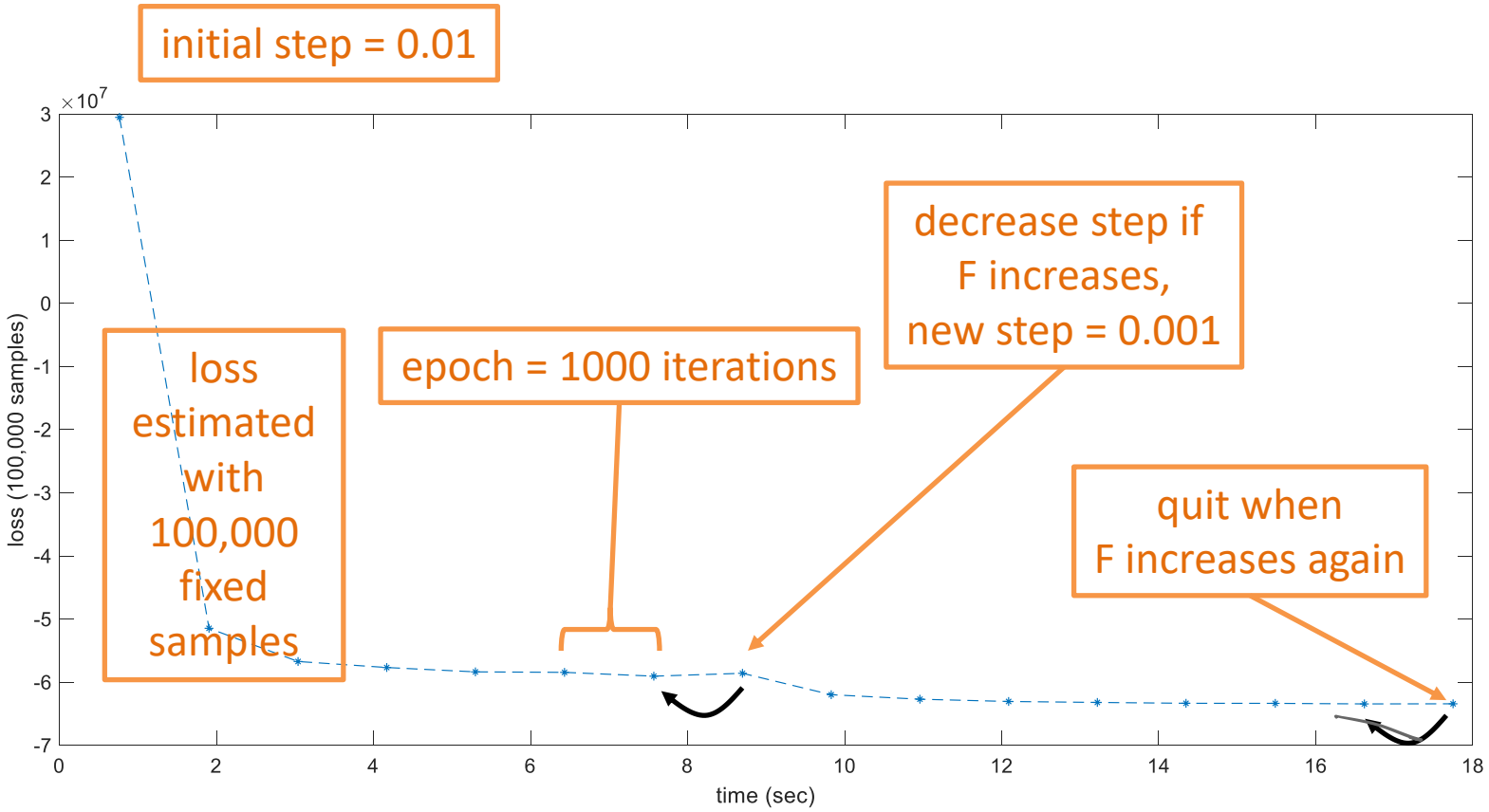
- $\tilde{p}_i$ = # times nonzero i picked
- $\tilde{q}_i$ = " " zero i "
- $\tilde{y}_i = \begin{cases} \tilde{p}_i \cdot \frac{\eta}{p} & \text{if } x_i \neq 0 \\ \tilde{q}_i \cdot \frac{z}{q} & \text{if } x_i = 0 \end{cases}$



Example Results

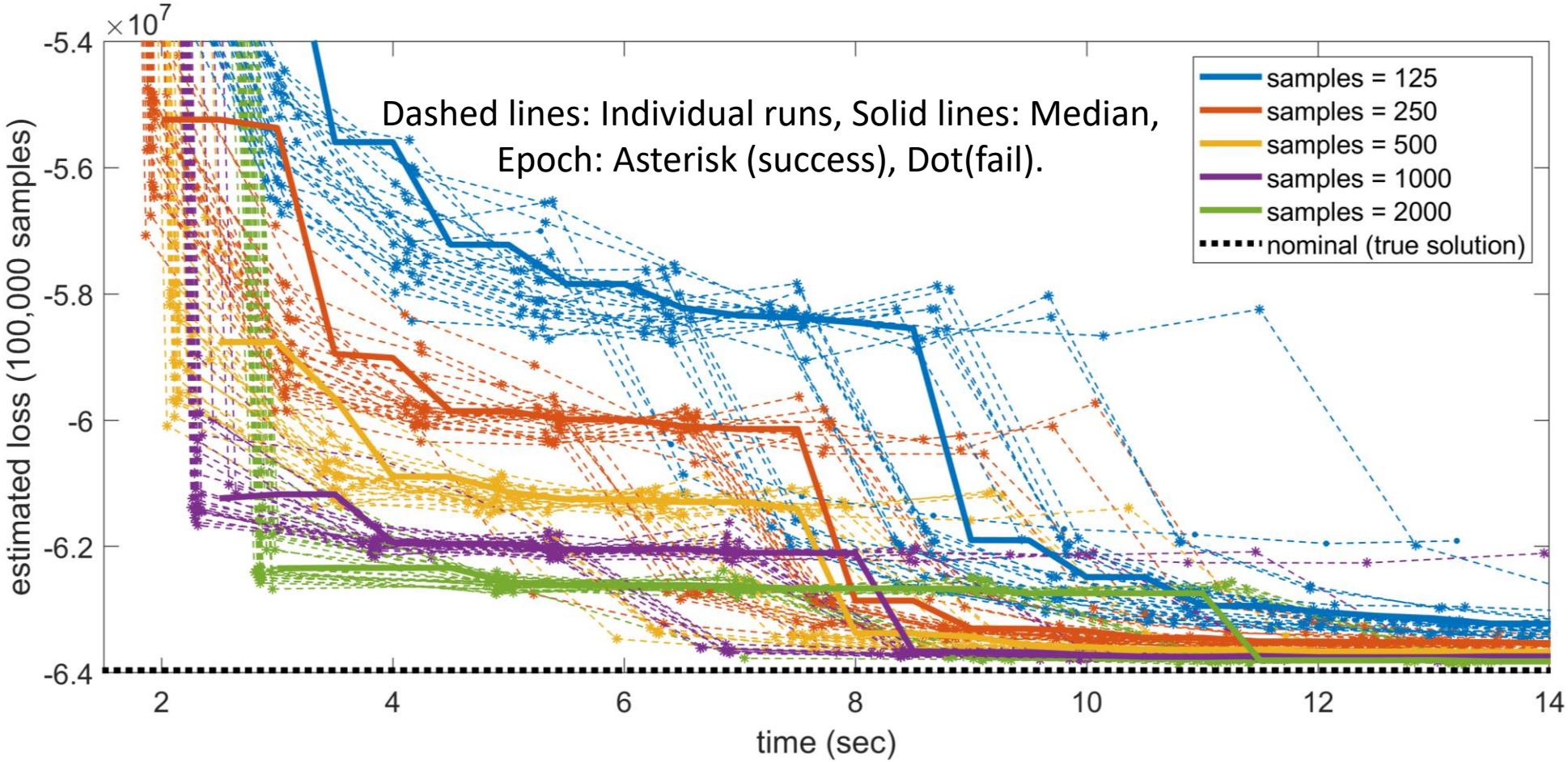
GCP with Stochastic Optimization

- Nonconvex problem
  - No guarantees of finding minimizer
- Using Adam (Kingma & Ba, 2015)
  - Default parameters
  - Some tweaks for checking convergence
- Past work on recommender systems uses SGD but ignores zeros
  - Gemulla, Nijkamp, Hass, Sismanis, KDD'11
  - Zhuang, Chin, Juan, and Lin, RecSys'13
- Past work on streaming uses SGD but data appears one slice at a time
  - Mardani, Mateos, Giannakis, IEEE TSP 2015
  - Maehara, Hayashi, Kawarabayashi,

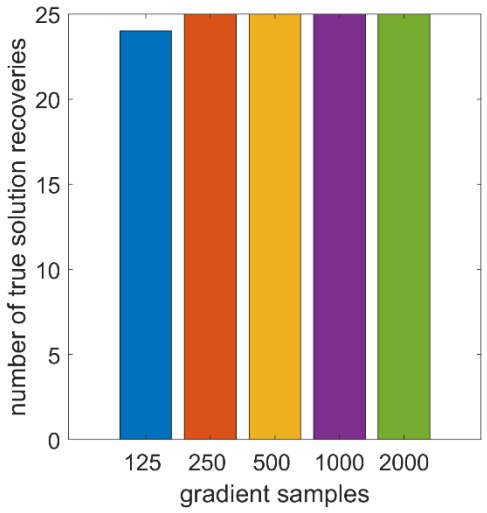


Example on Gamma-Distributed Data

$200 \times 150 \times 100 \times 50$ Tensor with low-rank ($r = 5$) structure based on Gamma distribution ($k = 1$, θ from model).
Gamma loss: $f(x, m) = \frac{x}{m} + \log m$. Running stochastic GCP with 25 random starts and varying numbers of samples.



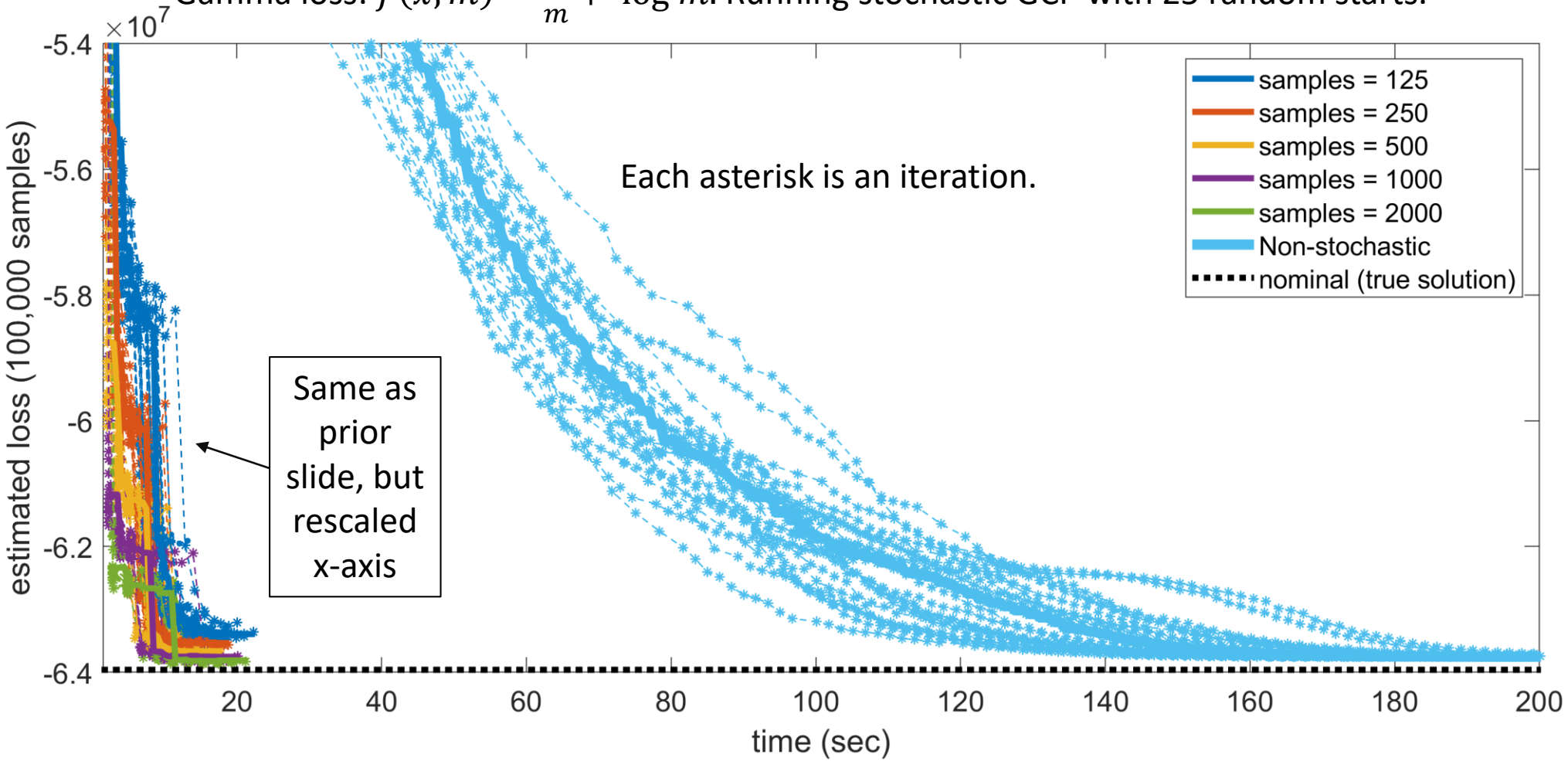
Success at Recovering Underlying Generative Factors



Stochastic vs. Non-Stochastic

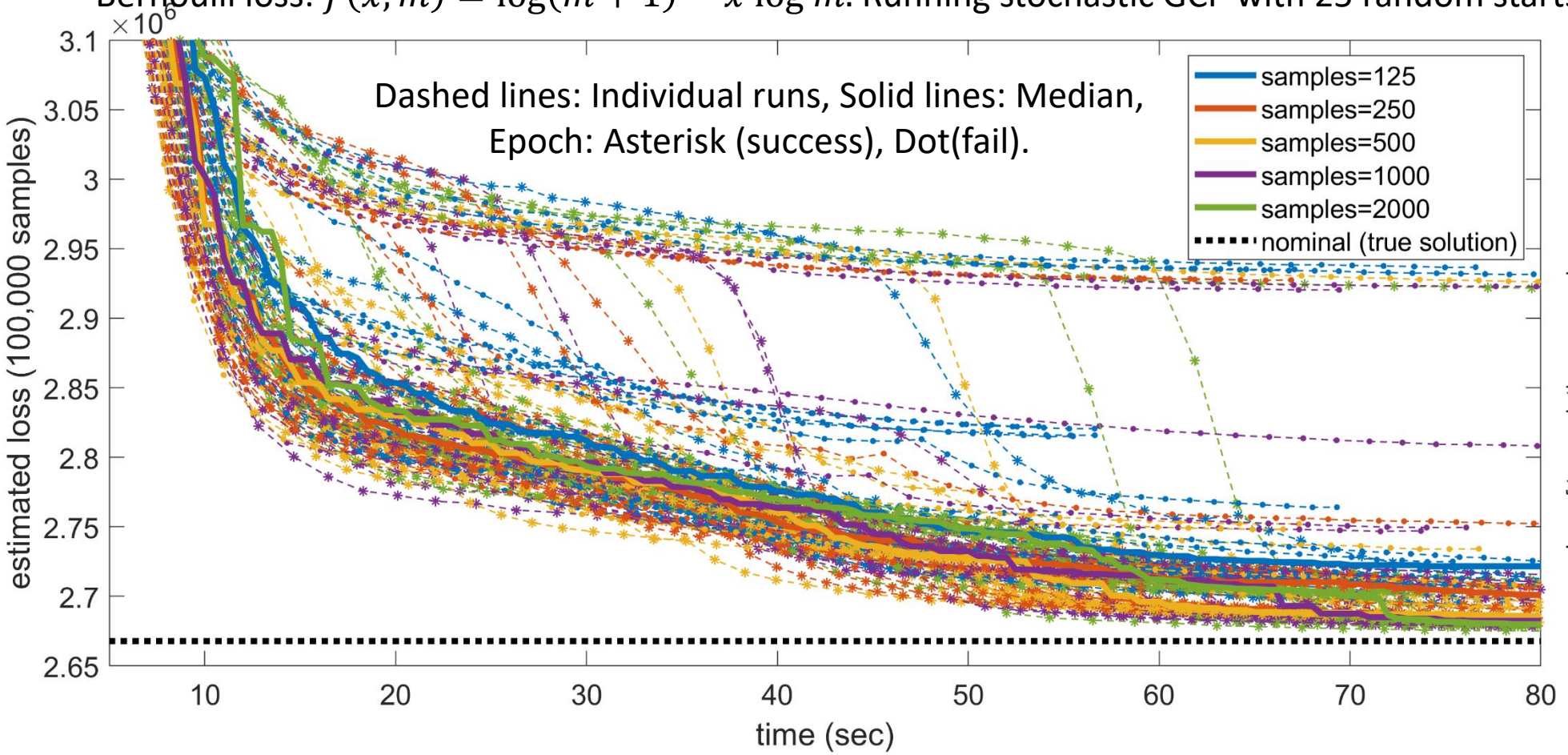
$200 \times 150 \times 100 \times 50$ Tensor with low-rank ($r = 5$) structure based on Gamma distribution ($k = 1$, θ from model).

Gamma loss: $f(x, m) = \frac{x}{m} + \log m$. Running stochastic GCP with 25 random starts.

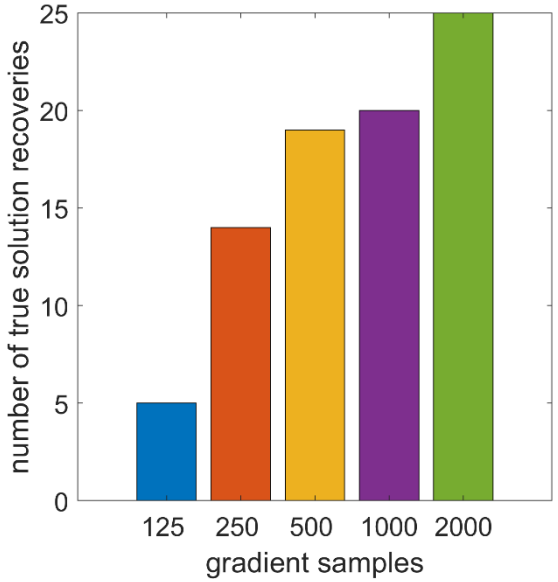


Example on Bernoulli-Distributed Data

$200 \times 150 \times 100 \times 50$ Tensor with low-rank ($r = 5$) structure based on Bernoulli distribution (odds from model).
Sparse tensor, less than 0.35% dense (~500K nonzeros).
Bernoulli loss: $f(x, m) = \log(m + 1) - x \log m$. Running stochastic GCP with 25 random starts, varying # of samples.

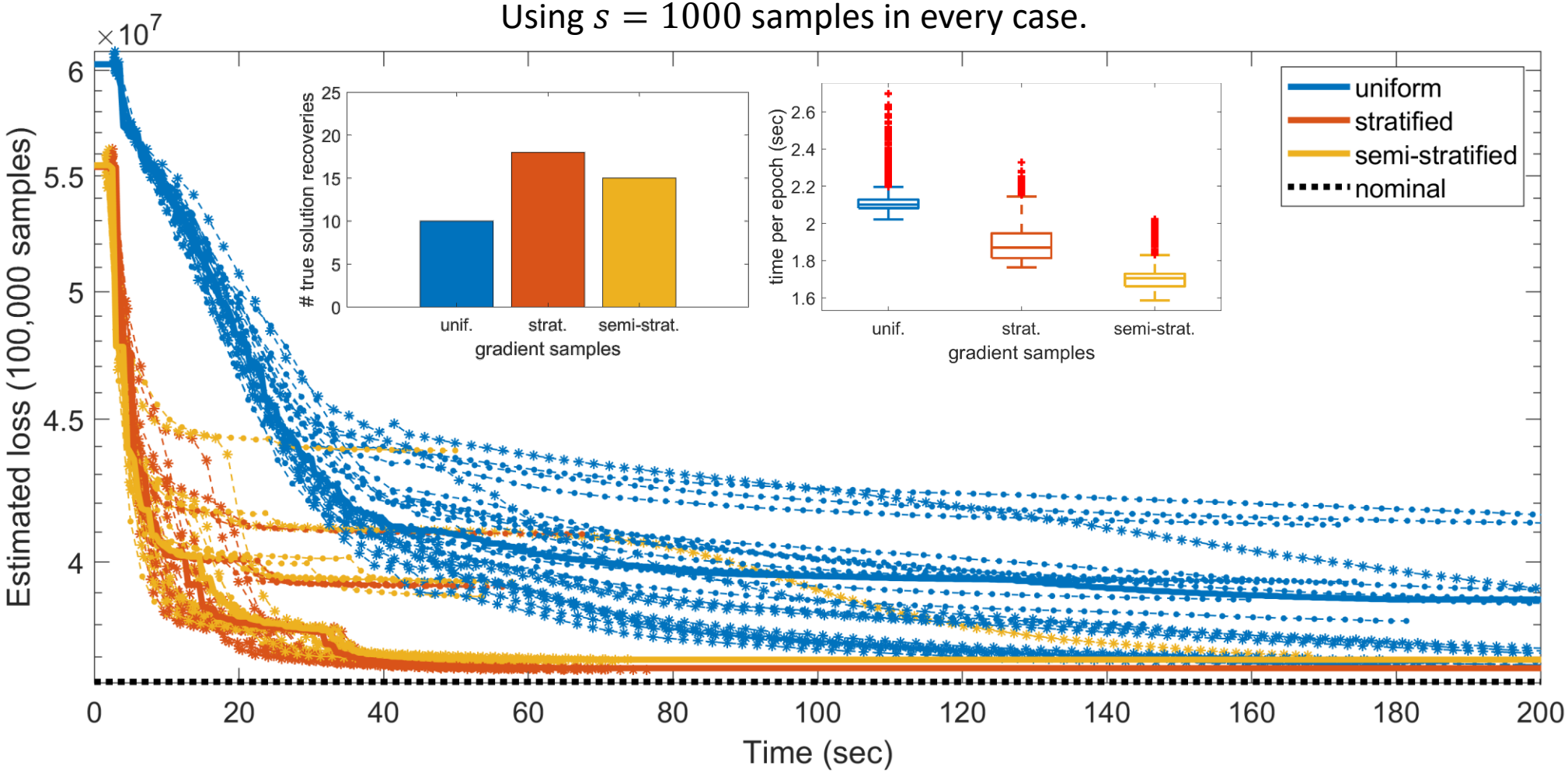


Success at Recovering Underlying Generative Factors



Uniform Sampling is Worse than Stratified for Sparse Tensors

Same set-up as binary experiments, but bigger tensor: $400 \times 300 \times 200 \times 100$, 0.38% dense (9M nonzeroes).
Using $s = 1000$ samples in every case.

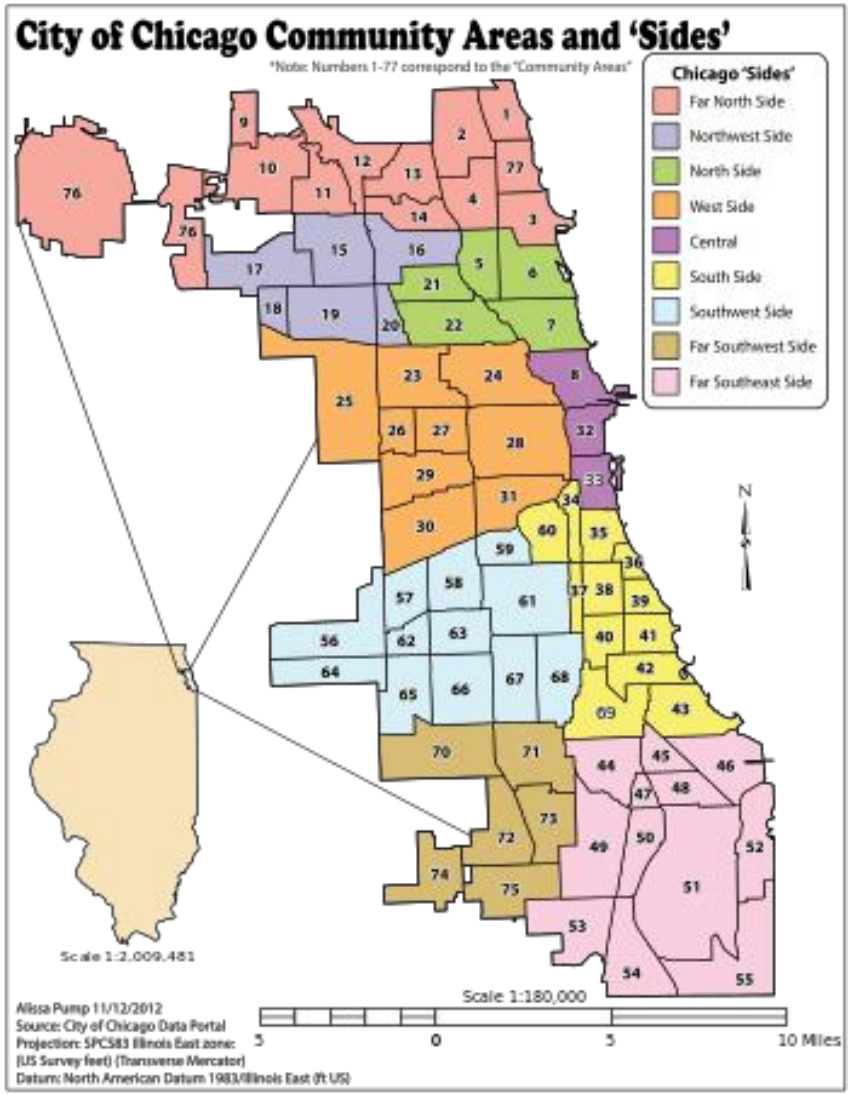


Chicago Crime Data

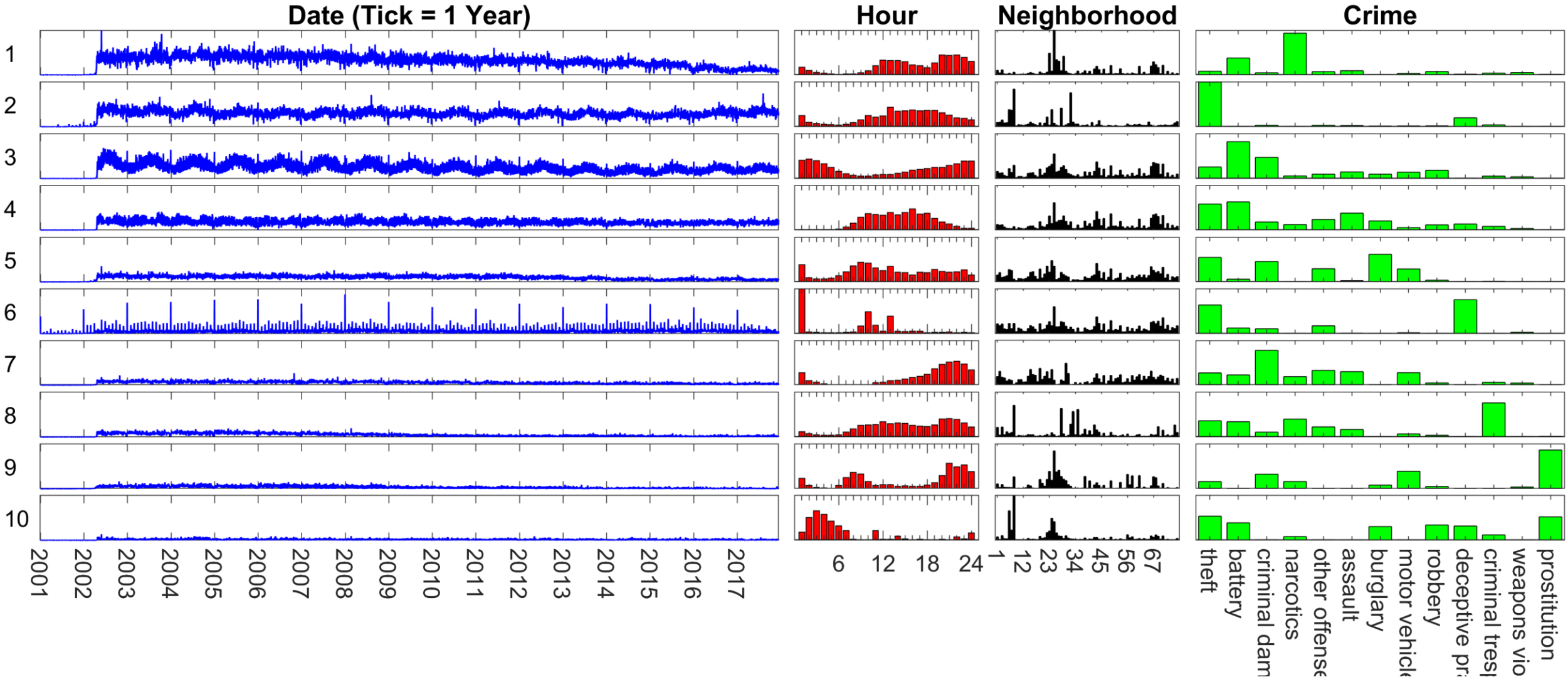
- 4-way count tensor
  - 6,186 Days
  - 24 Hours of the Day
  - 77 Community Areas
  - 32 Crime Types
- Non-zeros: 5,330,673
  - Storage: 0.21GB for sparse tensor
- Distribution of entries
  - 0: 98.54%
  - 1: 1.33%
  - ≥ 2: 0.12%
- Obtained from FROSTT (http://frostt.io/tensors/chicago-crime/)
- Data originally from Chicago Data Portal (https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2)

GCP-Count
Rank = 10
$s = 6{,}319$

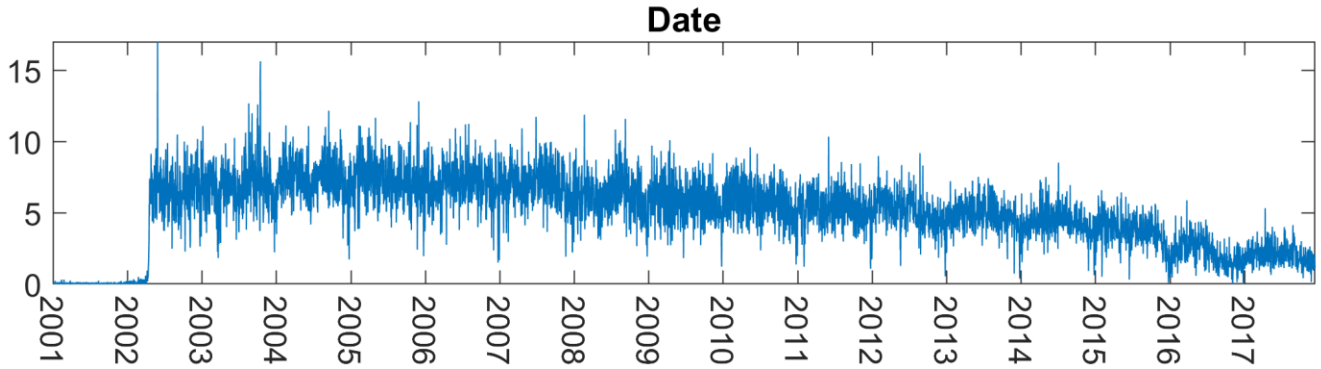
$$f(x, m) = m - x \log m$$

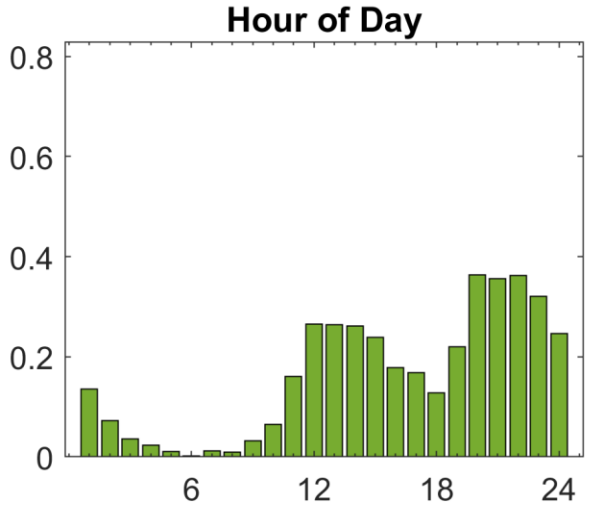


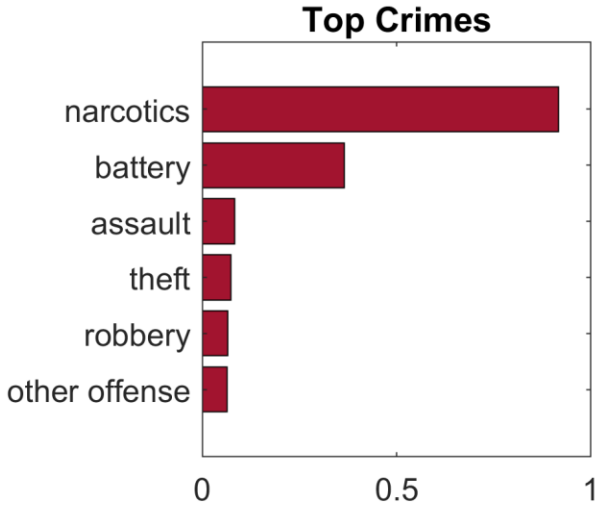
Application to Sparse Crime Binary Tensor (Semi-stratified results)

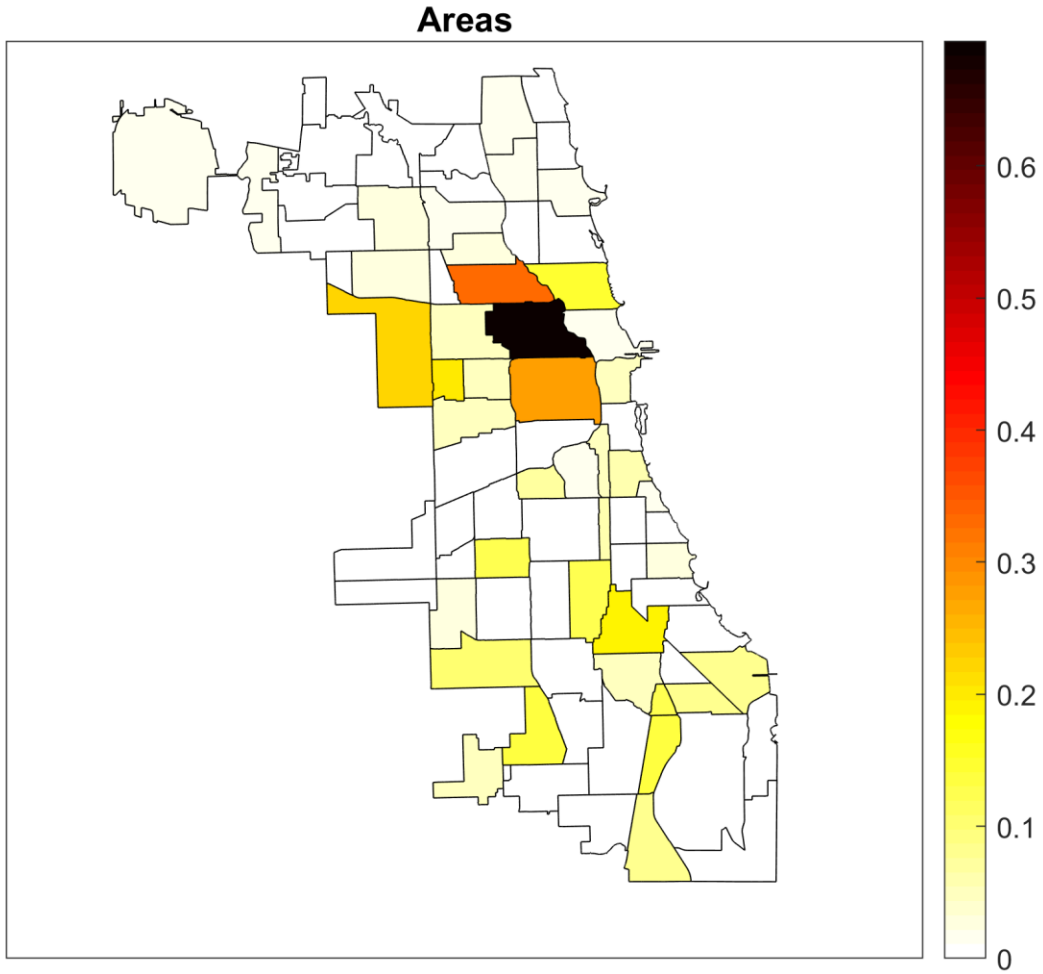


Component #1

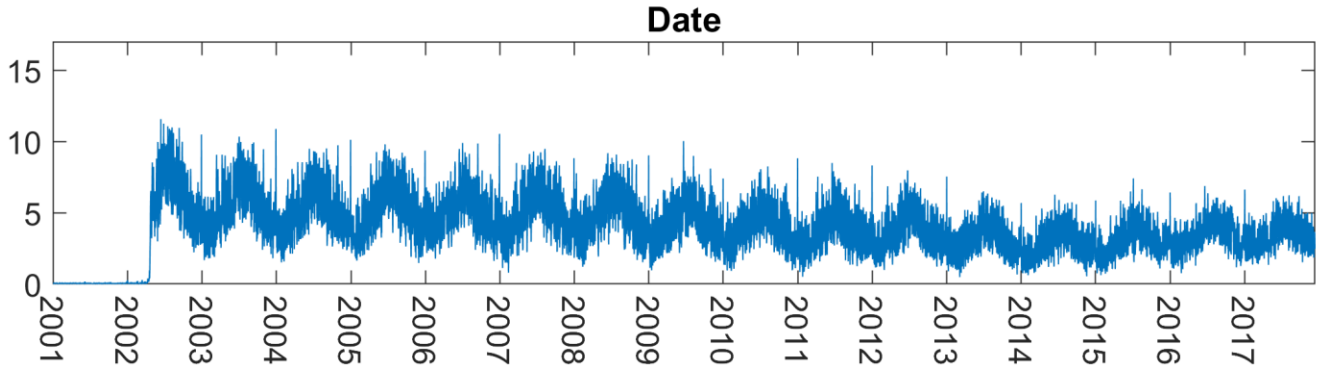

Date
15
10
5
0
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017

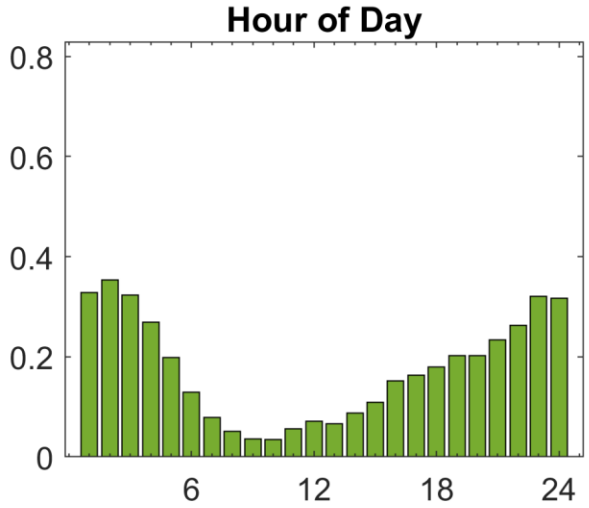

Hour of Day
0.8
0.6
0.4
0.2
0
6
12
18
24

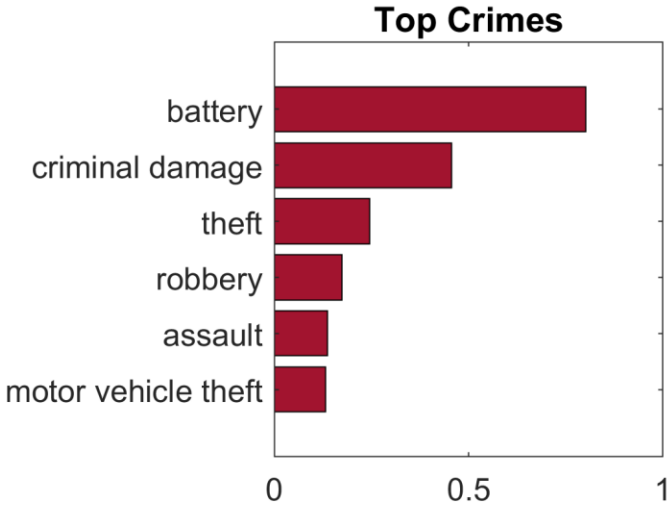

Top Crimes
narcotics
battery
assault
theft
robbery
other offense
0
0.5
1

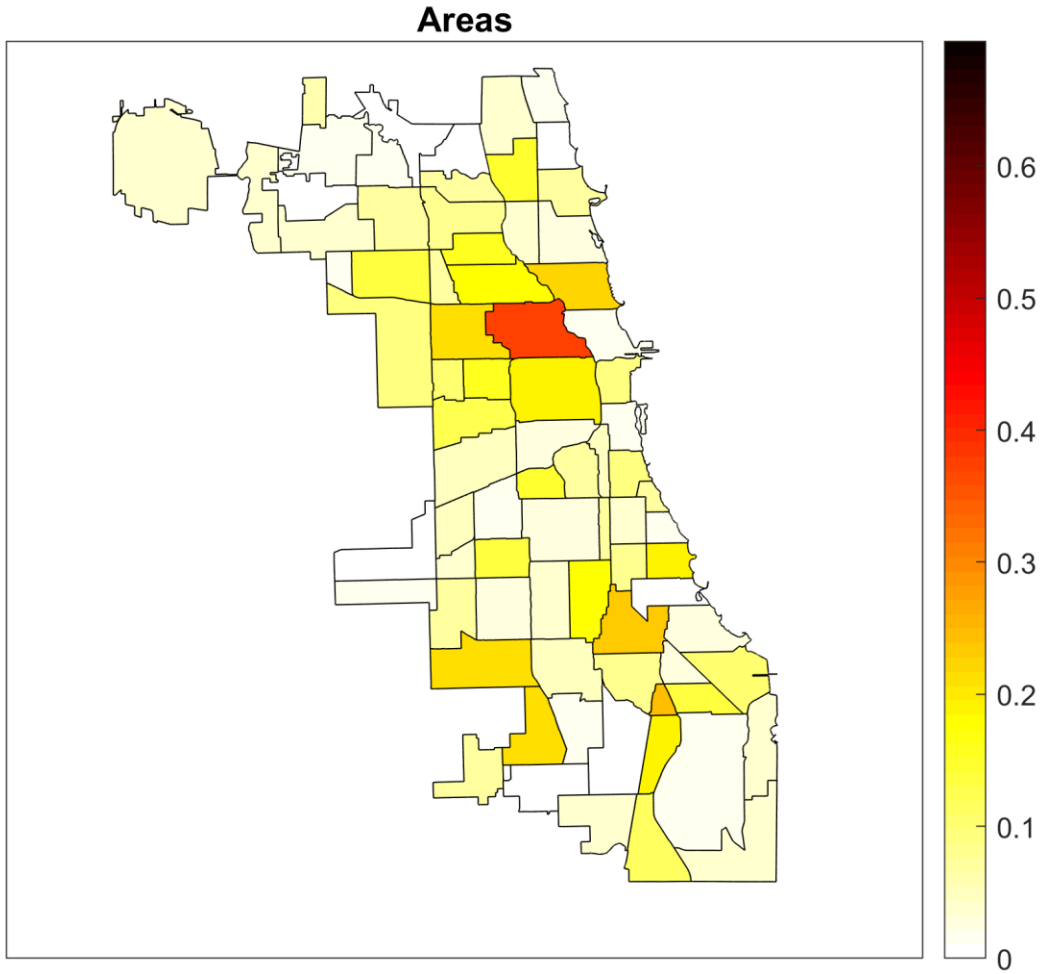

Areas
0.6
0.5
0.4
0.3
0.2
0.1
0

Component #3

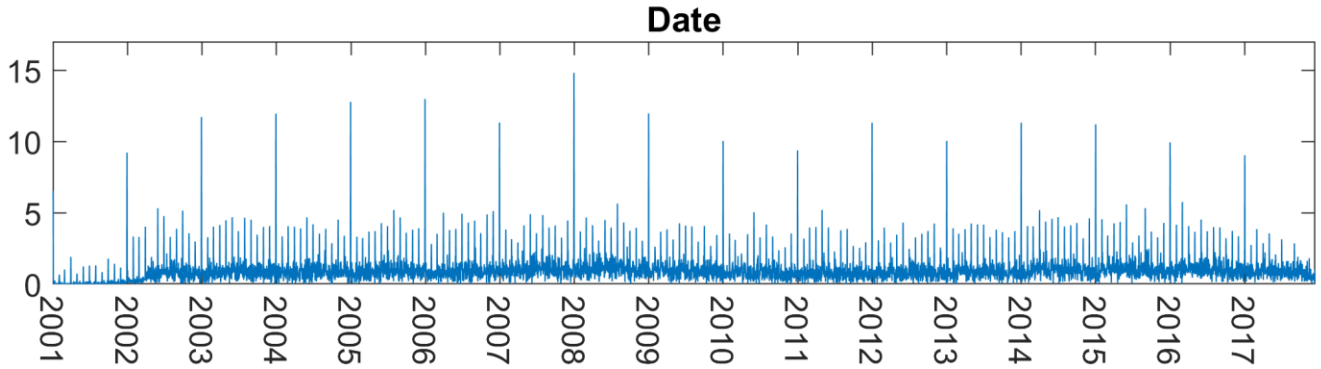

Date
15
10
5
0
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017

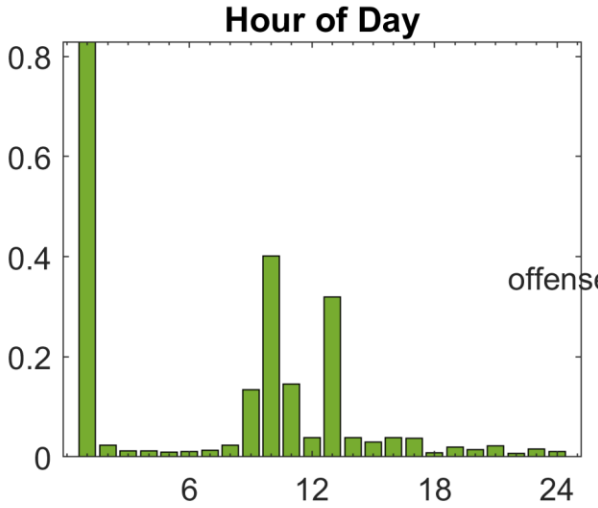

Hour of Day
0.8
0.6
0.4
0.2
0
6
12
18
24

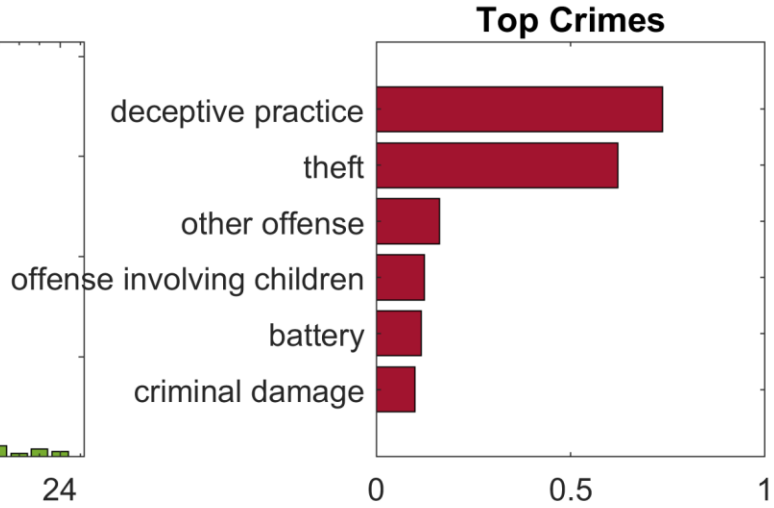

Top Crimes
battery
criminal damage
theft
robbery
assault
motor vehicle theft
0
0.5
1

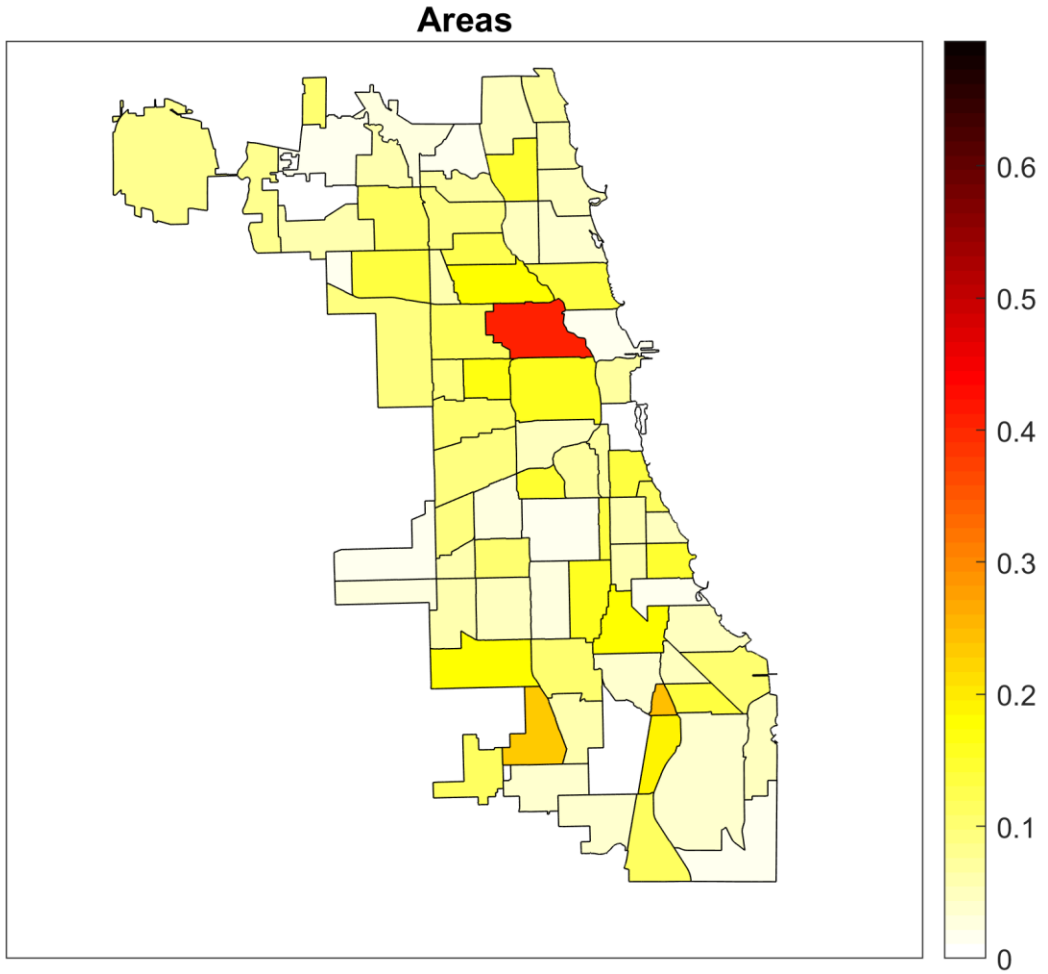

Areas
0.6
0.5
0.4
0.3
0.2
0.1
0

Component #6


Date
15
10
5
0
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017


Hour of Day
0.8
0.6
0.4
0.2
0
6
12
18
24


Top Crimes
deceptive practice
theft
other offense
offense involving children
battery
criminal damage
0
0.5
1


Areas
0.6
0.5
0.4
0.3
0.2
0.1
0