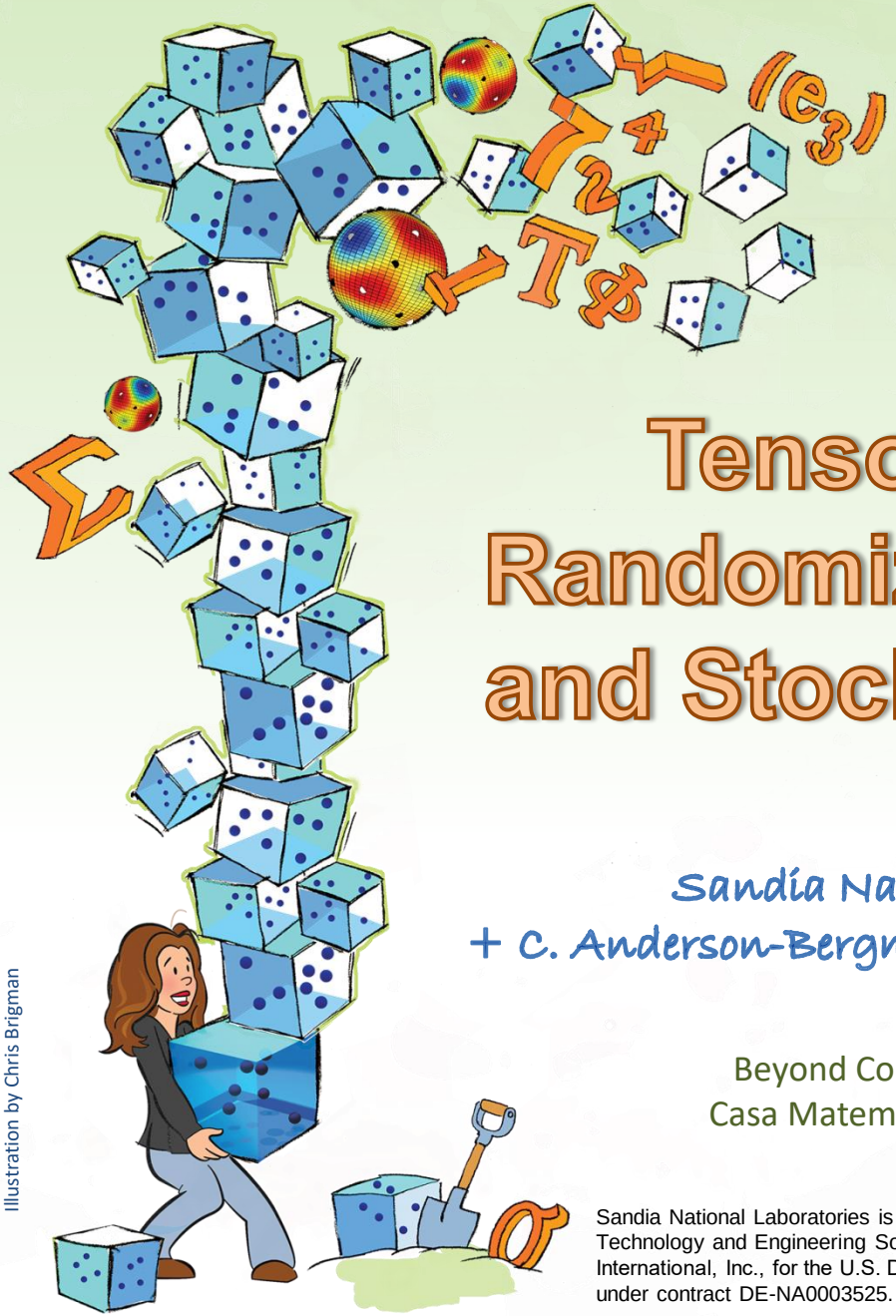


Illustration by Chris Brigman

Sandia National Laboratories

SAND2017-11485C

Tensor Factorization, Randomized Linear Algebra, and Stochastic Optimization

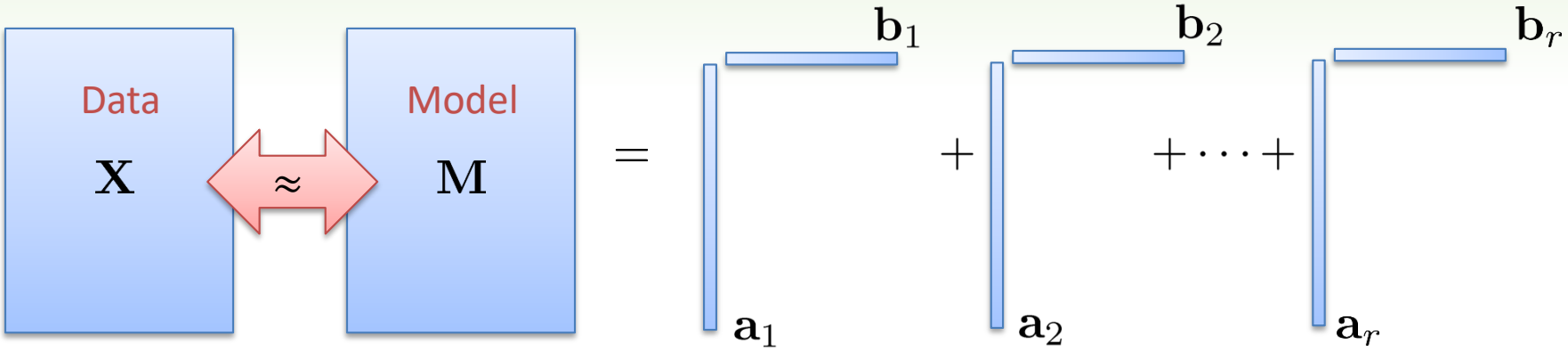
Tamara G. Kolda
Sandia National Laboratories, Livermore, CA
+ C. Anderson-Bergman, G. Ballard, C. Battaglino, and D. Hong

Beyond Convexity: Emerging Challenges in Data Science
Casa Matemática Oaxaca, Oaxaca, Mexico, Oct 22-27, 2017

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

U.S. DEPARTMENT OF ENERGY

Matrix Factorization



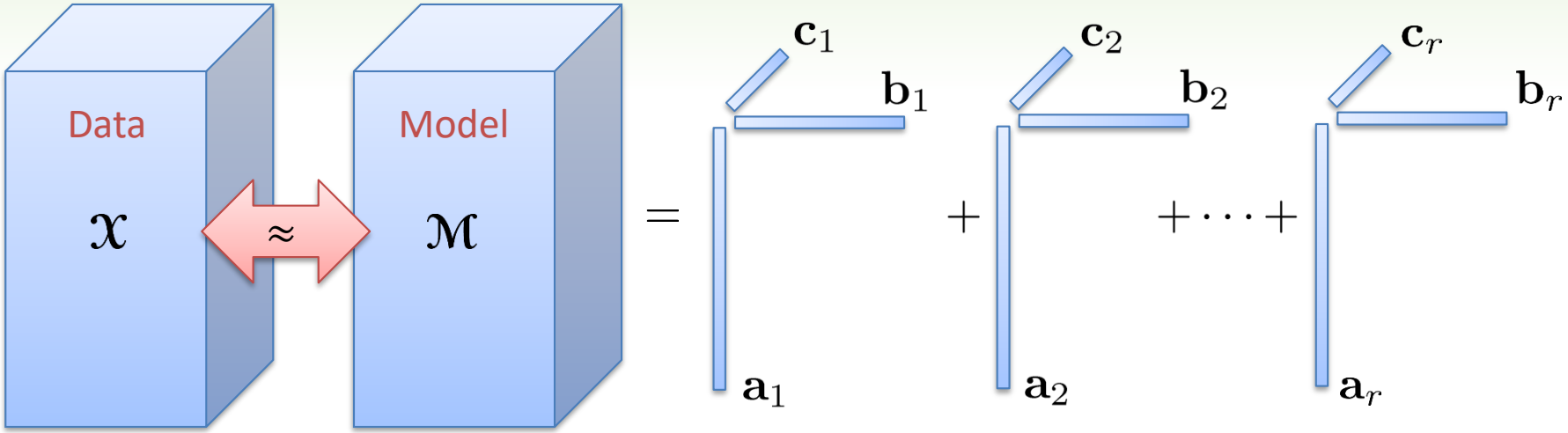
$$\mathbf{X} \approx \mathbf{M} = \sum_{j=1}^{r} \mathbf{a}_j \mathbf{b}_j' = \mathbf{A}\mathbf{B}'$$

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

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{M}\|^2 \equiv \sum_{ij} (x_{ij} - m_{ij})^2 \text{ s.t. } \mathbf{M} = [\![\mathbf{A}, \mathbf{B}]\!]$$

Tensor Factorization (3-way)

CP = CANDECOMP/PARAFAC or Canonical Polyadic

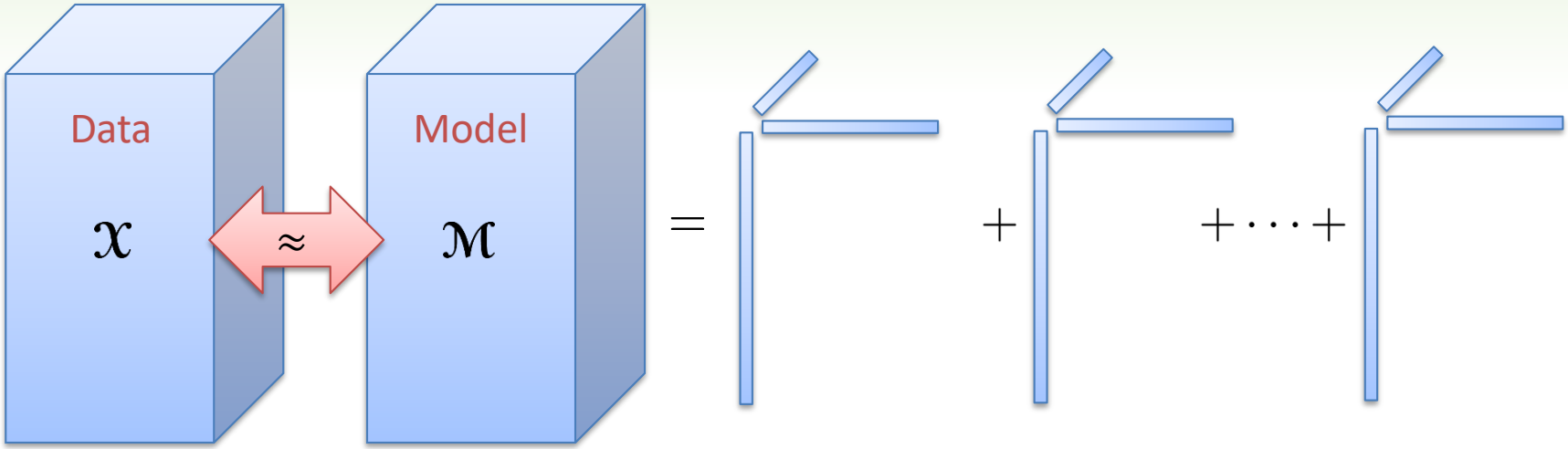


$$\mathcal{X} \approx \mathcal{M} = \sum_{j=1}^{r} \mathbf{a}_j \circ \mathbf{b}_j \circ \mathbf{c}_j = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathcal{X} - \mathcal{M}\|^2 \equiv \sum_{ijk} (x_{ijk} - m_{ijk})^2 \text{ s.t. } \mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

Hitchcock 1927, Harshman 1970, Carroll & Chang 1970

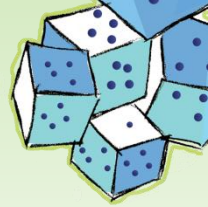
Tensor Factorization (d-way)



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

$$\min_{\mathbf{A}_1,\ldots,\mathbf{A}_d} \|\mathcal{X} - \mathcal{M}\|^2 \text{ s.t. } \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

Hitchcock 1927, Harshman 1970, Carroll & Chang 1970



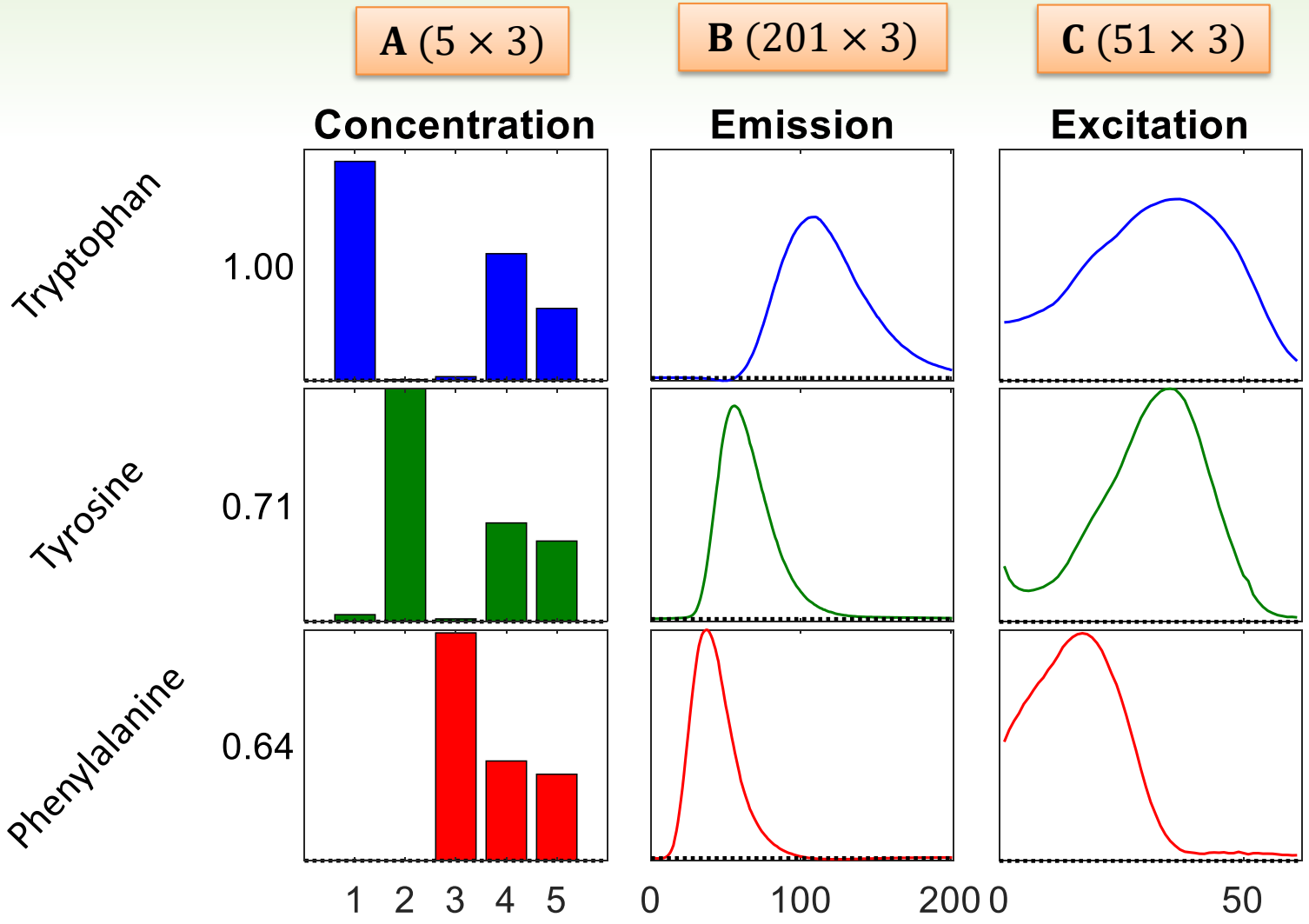
Two Motivating Examples

Amino Acids Fluroescence Dataset

- Fluorescence measurements of 5 samples containing 3 amino acids
  - Tryptophan
  - Tyrosine
  - Phenylalanine
- Tensor of size 5 x 51 x 201
  - 5 samples
  - 51 excitations
  - 201 emissions

R. Bro, ***PARAFAC: Tutorial and Applications***,
Chemometrics and Intelligent Laboratory Systems, 38:149-171, 1997

Rank-3 CP Factorization of Amino Acids Data

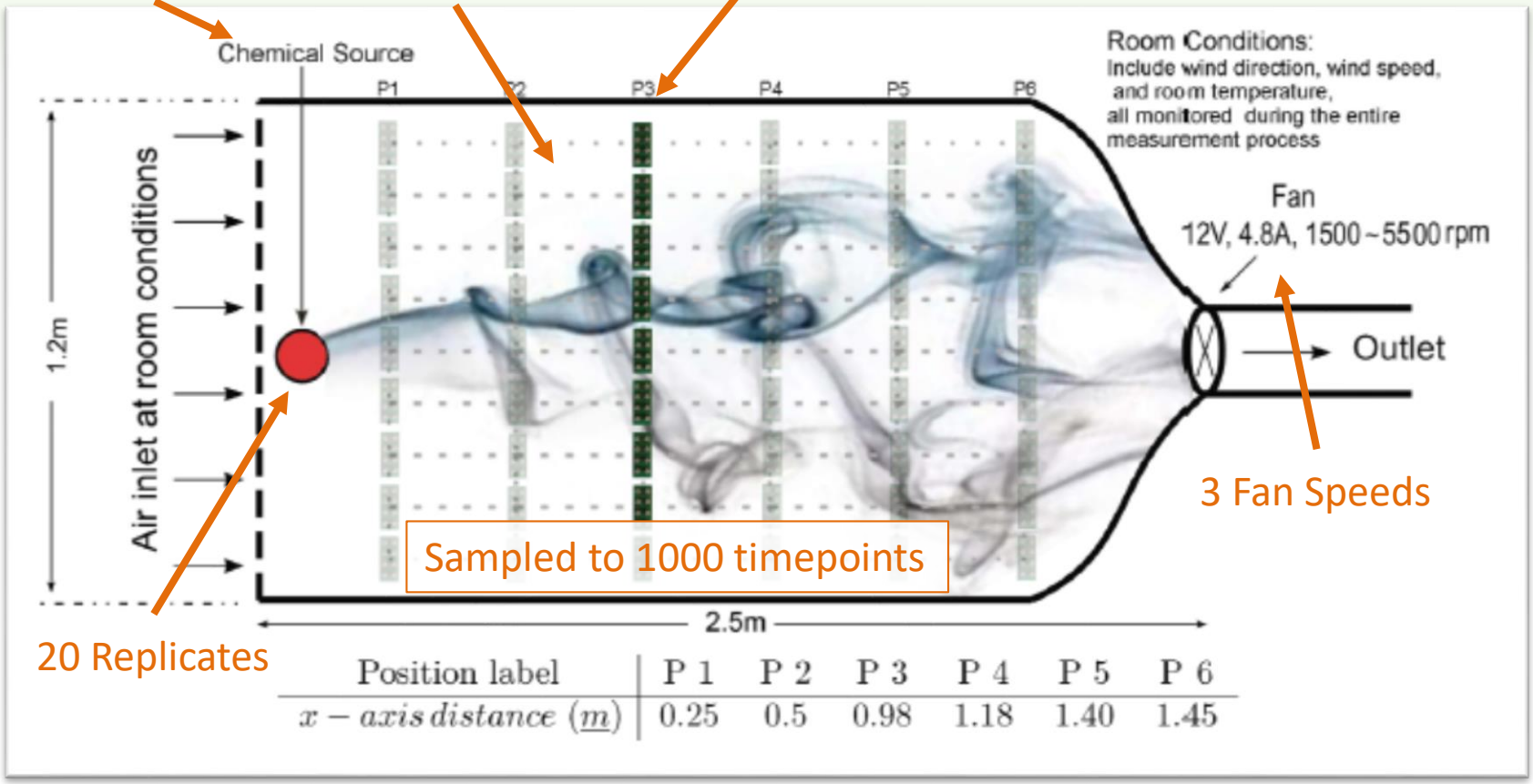


Hazardous Gases Sensor Dataset

Multiple Gases

5 Room Temperatures

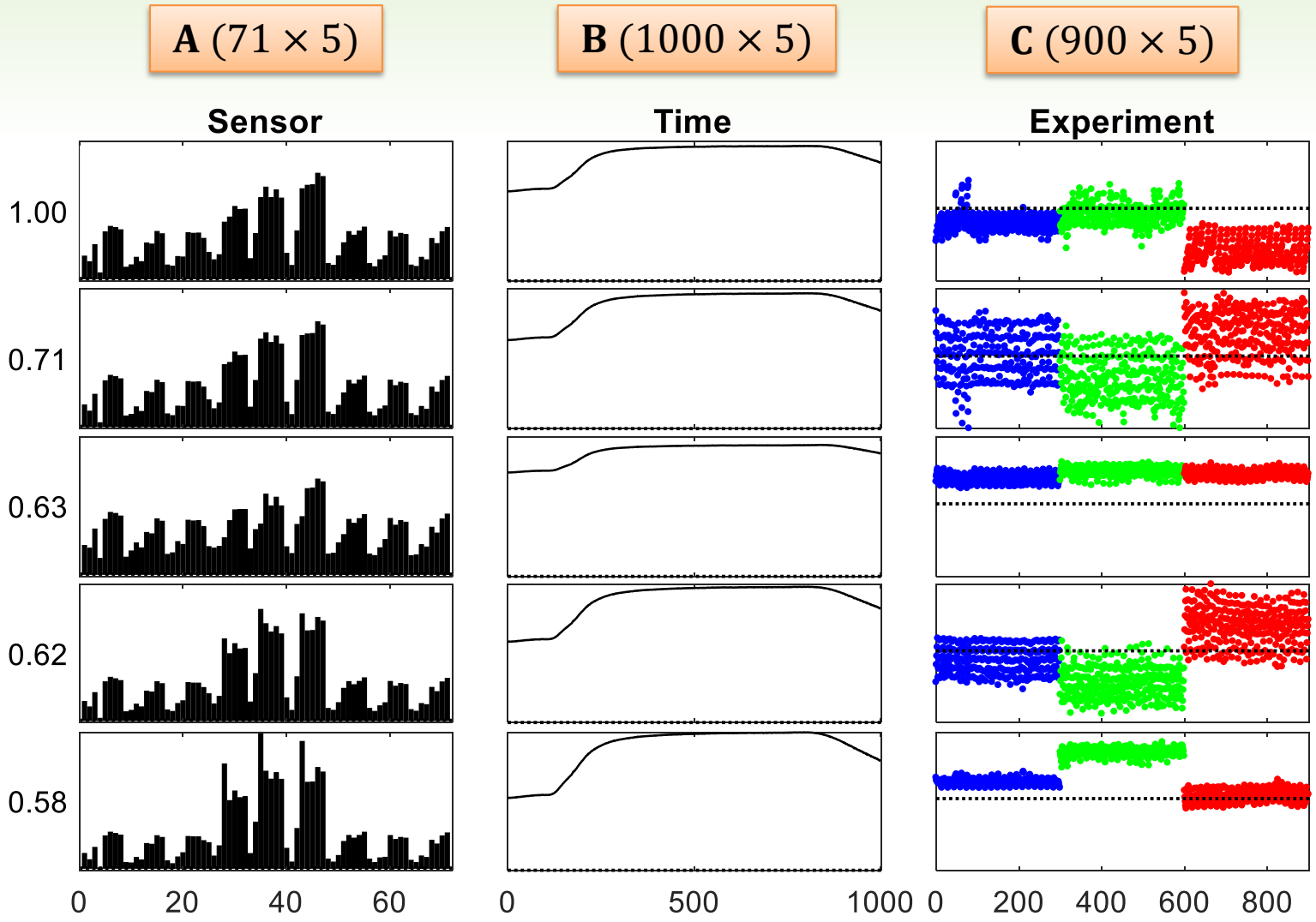
71 Sensors (9 x 4 x 2 grid with one missing)



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

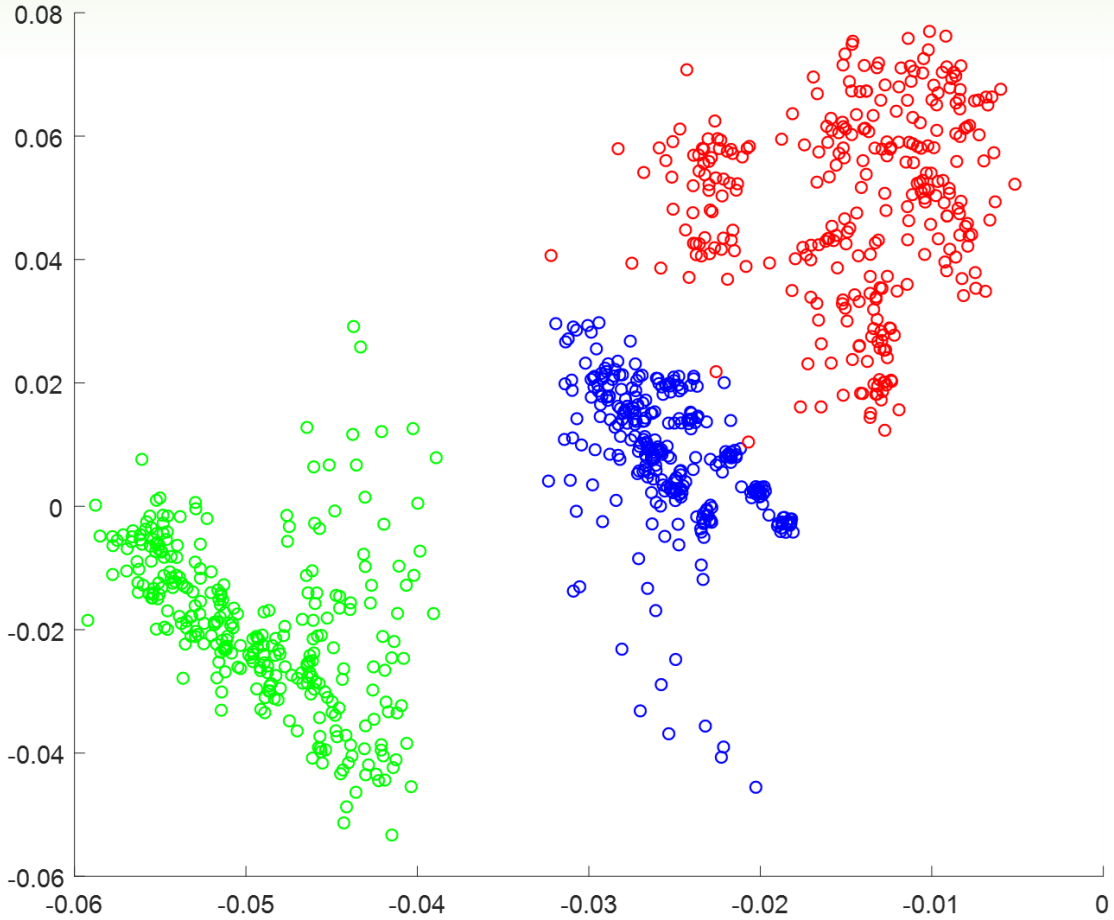
A. Vergara, J. Fonollosa, J. Mahiques, M. Trincavelli, N. Rulkov and R. Huerta, ***On the performance of gas sensor arrays in open sampling systems using Inhibitory Support Vector Machines***, Sensors and Actuators B: Chemical, 2013, doi:10.1016/j.snb.2013.05.027

Rank-5 CP Decomposition of Gas Data with 3 Gases

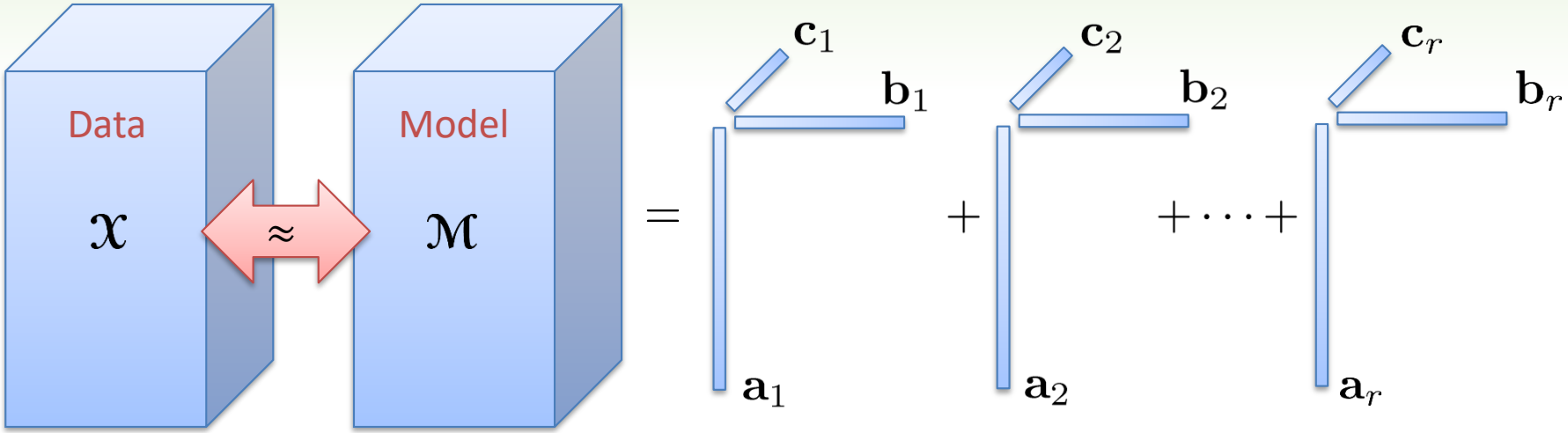


Viz of Experiments using C

Compute SVD of $\mathbf{C}$
(of size 900 x 5)
and use first two left
components to plot.



Tensor Factorization (3-way)



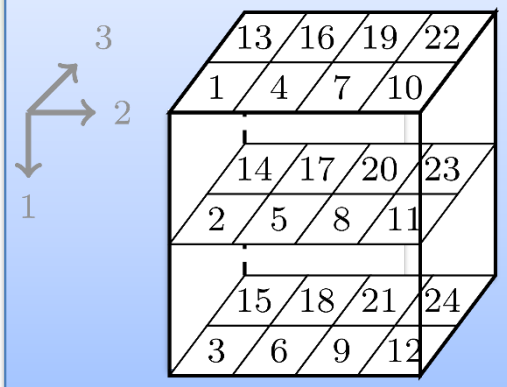
$$\mathcal{X} \approx \mathcal{M} = \sum_{j=1}^{r} \mathbf{a}_j \circ \mathbf{b}_j \circ \mathbf{c}_j = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathcal{X} - \mathcal{M}\|^2 \text{ s.t. } \mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

We can rewrite this as a matrix equation in **A**, **B**, or **C**.

Details: Unfolding, Khatri-Rao Products, and Connections to CP

Unfolding reorganizes the entries of the tensor into a matrix.



$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix} \quad \mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix}$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \end{bmatrix}$$

Kronecker Products, Unfolding, & Khatri-Rao Products

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1\mathbf{b} \\ \vdots \\ a_n\mathbf{b} \end{bmatrix}$$

$$(\mathbf{a} \circ \mathbf{b} \circ \mathbf{c})_{(1)} = \mathbf{a}(\mathbf{c} \otimes \mathbf{b})'$$

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 \cdots \mathbf{a}_r \otimes \mathbf{b}_r \end{bmatrix}$$

Connecting back to CP Model

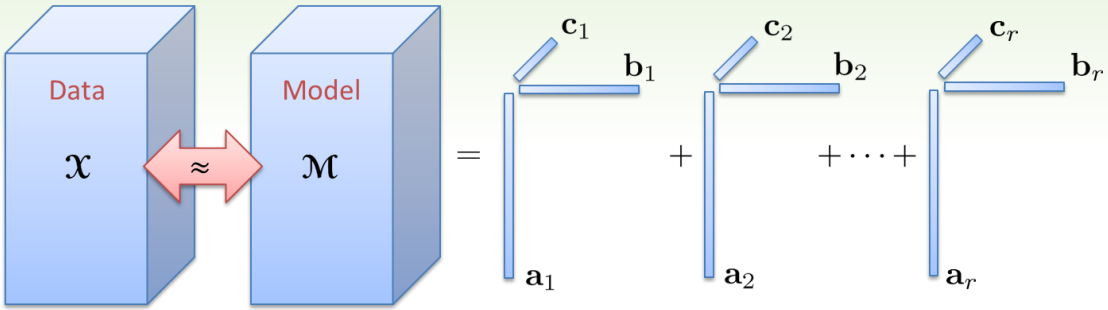
$$\mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_j \mathbf{a}_j \circ \mathbf{b}_j \circ \mathbf{c}_j$$

$$\mathbf{M}_{(1)} = \sum_j \mathbf{a}_j (\mathbf{c}_j \otimes \mathbf{b}_j)'$$

$$\mathbf{M}_{(1)} = \begin{bmatrix} \mathbf{a}_1 \cdots \mathbf{a}_r \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \otimes \mathbf{b}_1 \cdots \mathbf{c}_r \otimes \mathbf{b}_r \end{bmatrix}'$$

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

CP-ALS: Fitting CP Model via Alternating Least Squares



Convex (linear least squares) subproblems can be solved exactly

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathcal{X} - \mathcal{M}\|^2 \text{ s.t. } \mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

Repeat until convergence…

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

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

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

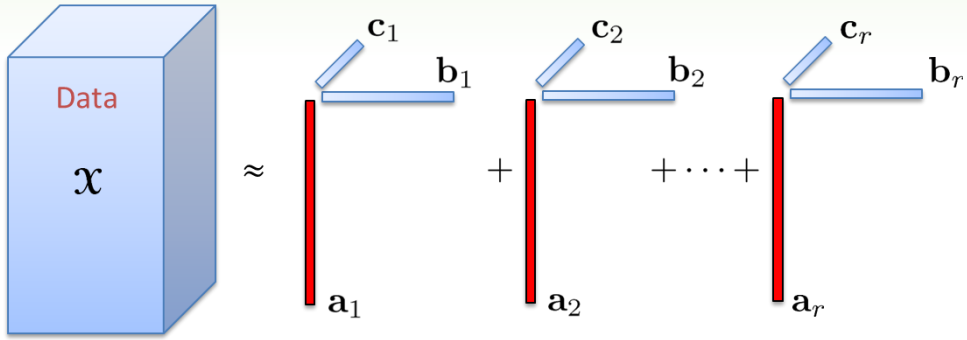
CP-ALS Least Squares Problem

Repeat until convergence…

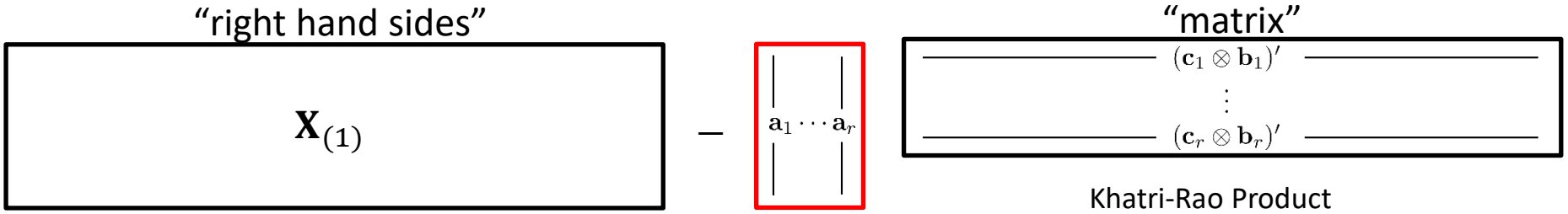
$$\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})'\|^2$$

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

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



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



Details: Special Structure of the Least Squares Problem

$$\min_{\mathbf{A}} \|\underbrace{\mathbf{X}_{(1)}}_{n \times n^{d-1}} - \underbrace{\mathbf{A}}_{n \times r}\underbrace{(\mathbf{C} \odot \mathbf{B})'}_{r \times n^{d-1}}\|^2$$

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

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

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

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

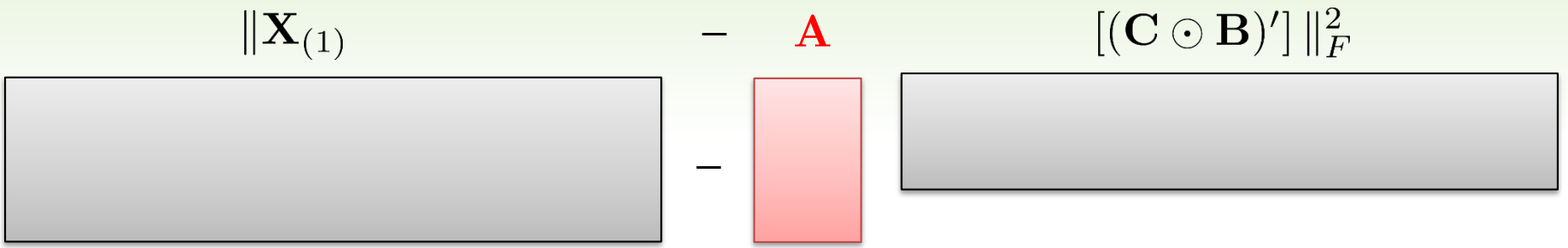
The most expensive step is *not* the backsolve.

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

So, how will randomized methods help?

Battaglino, Ballard, & Kolda 2017

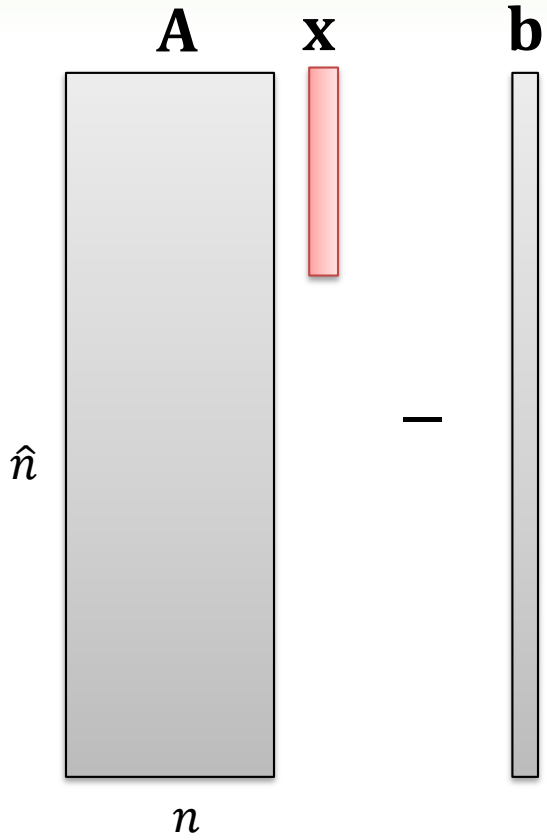
CP Least Squares Problem



How to randomize this?

Aside: Sketching for Standard Least Squares

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



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

$$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{b}$$

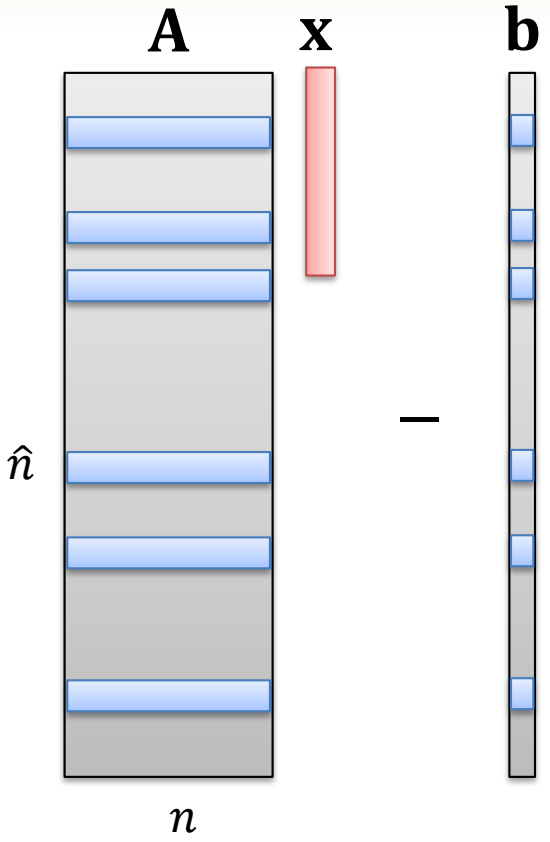
$$\mathbf{x} \leftarrow \mathbf{A} \backslash \mathbf{b}$$

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

Sarlós 2006, Woodruff 2014

Sampled Least Squares

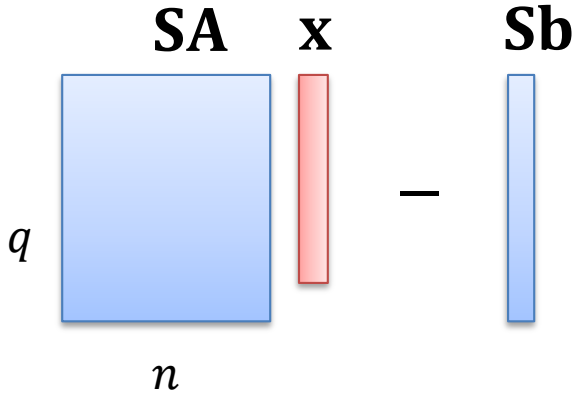
Choose q rows, uniformly at random



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

S

approximate

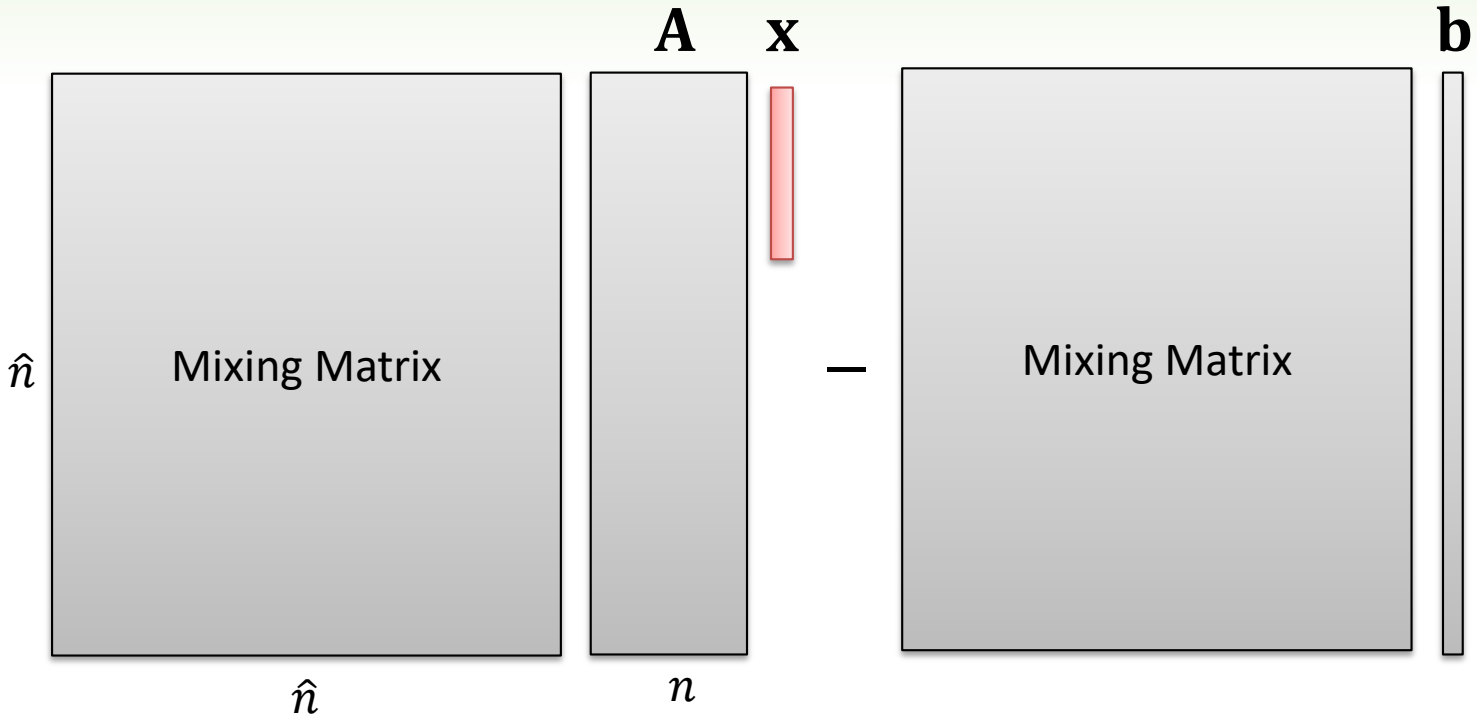


Sampling only guaranteed to "work" if the **A** *is incoherent.*

$\mathcal{O}(qn^2)$

Sarlós 2006, Woodruff 2014

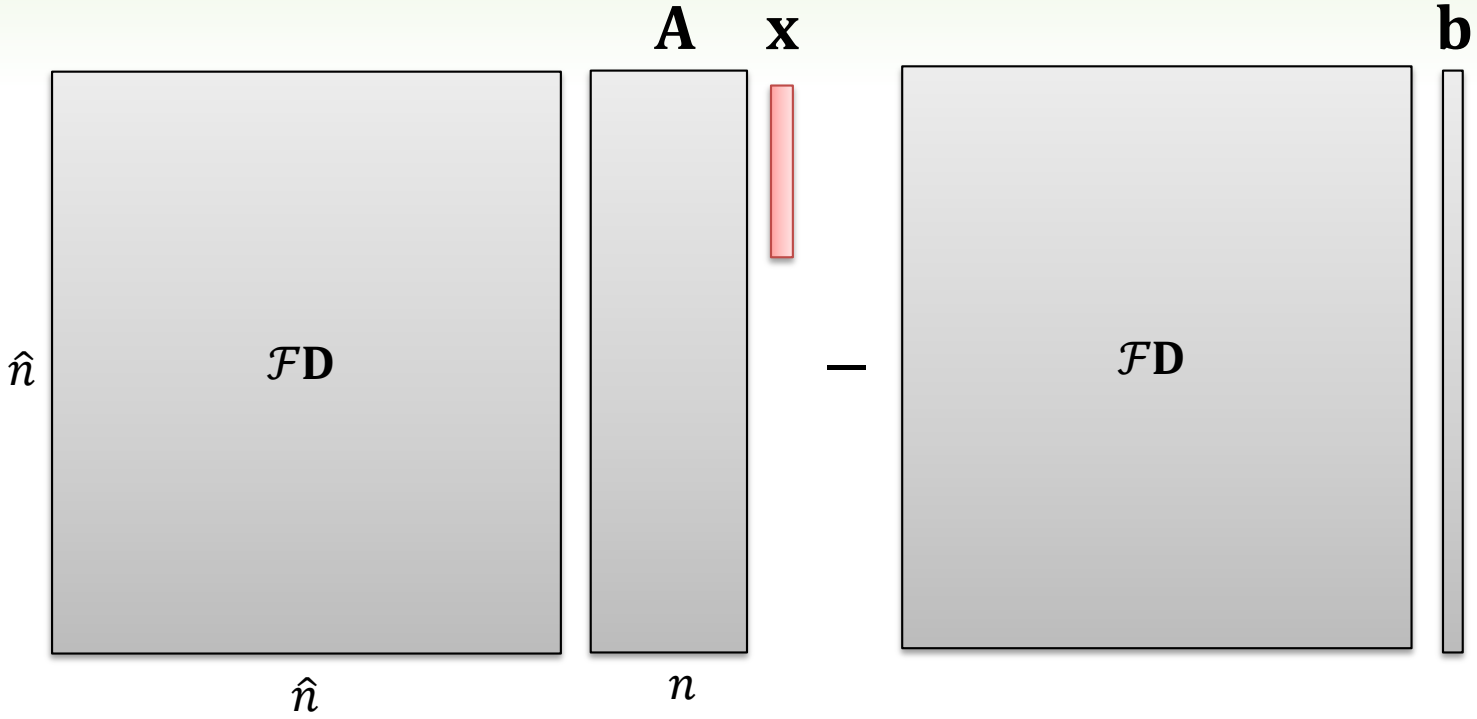
Enforcing Incoherence



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

Sarlós 2006, Woodruff 2014

Fast Johnson-Lindenstrauss Transform (FJLT)



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

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

Sampled/Mixed Least Squares

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

$$\min_{\mathbf{x}} \|\mathbf{SAx} - \mathbf{Sb}\|$$

Sampling only, No mixing.

$$\min_{\mathbf{x}} \|\mathbf{S}\mathcal{F}\mathbf{DAx} - \mathbf{S}\mathcal{F}\mathbf{Db}\|$$

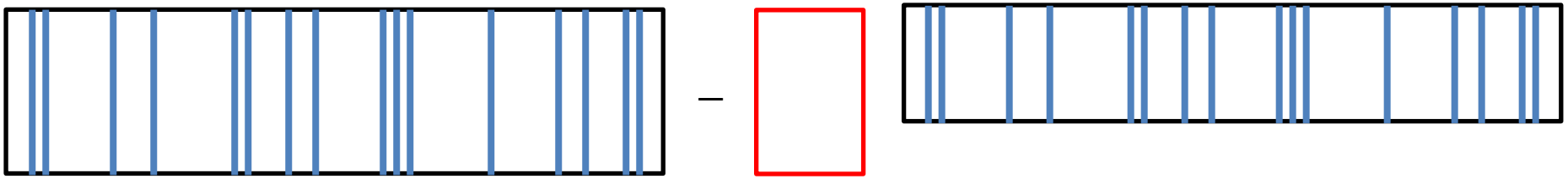
Sampling + Mixing

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

CP-ALS-RAND

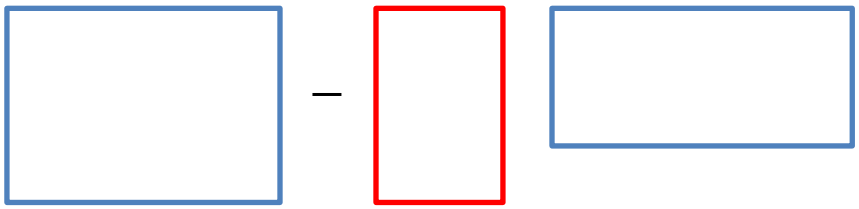
$$\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})'\|^2$$

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



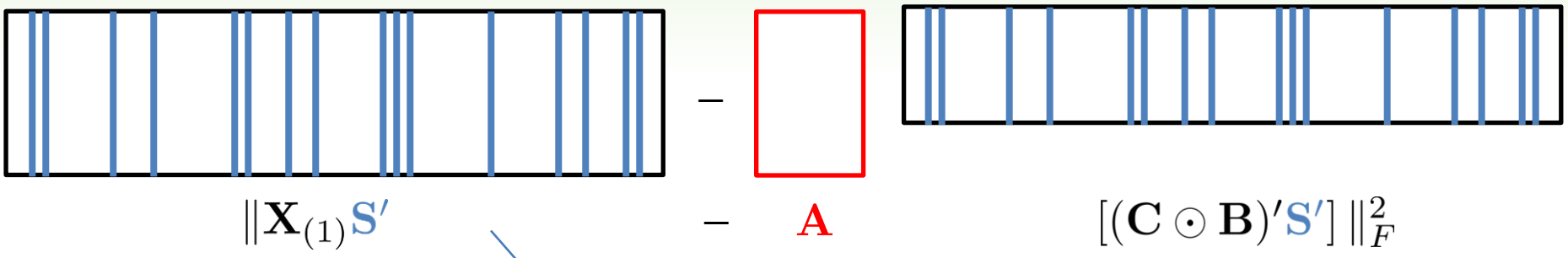
$\|\mathbf{X}_{(1)}\mathbf{S}'$ $-$ $\mathbf{A}$ $[(\mathbf{C} \odot \mathbf{B})'\mathbf{S}']\|_F^2$

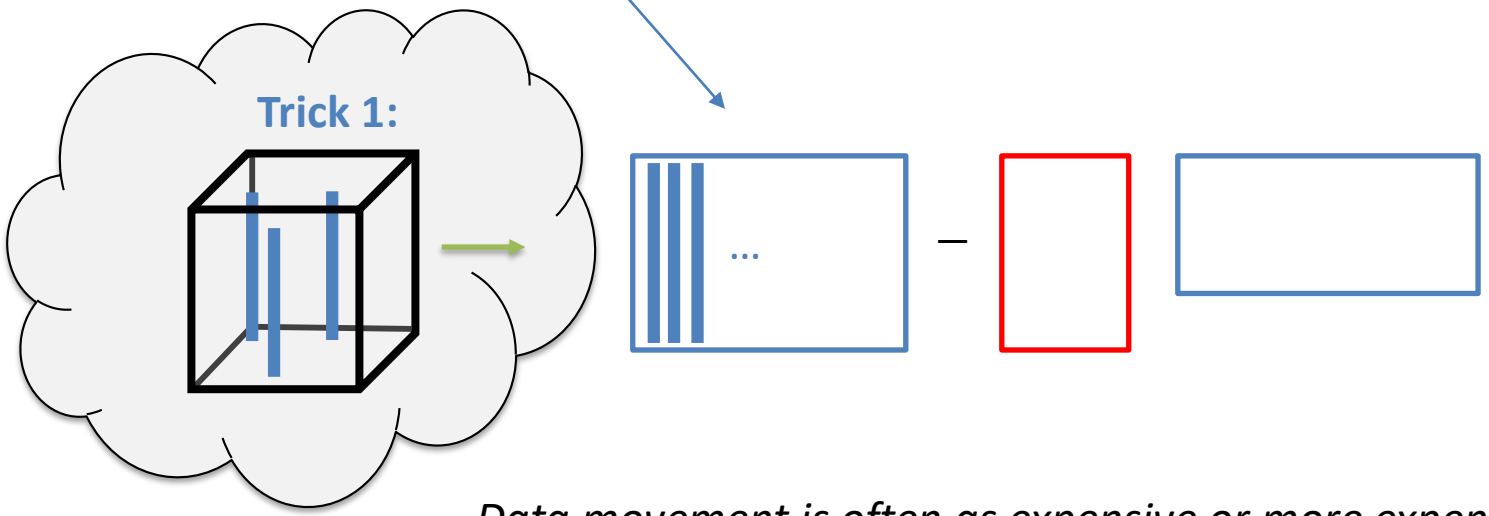
$$\mathbf{A} \leftarrow \mathbf{X}_{(1)}\mathbf{S}'/(\mathbf{C} \odot \mathbf{B})'\mathbf{S}'$$



Battaglino, Ballard, & Kolda 2017

CP-ALS-RAND Trick #1: Avoid unfolding data tensor

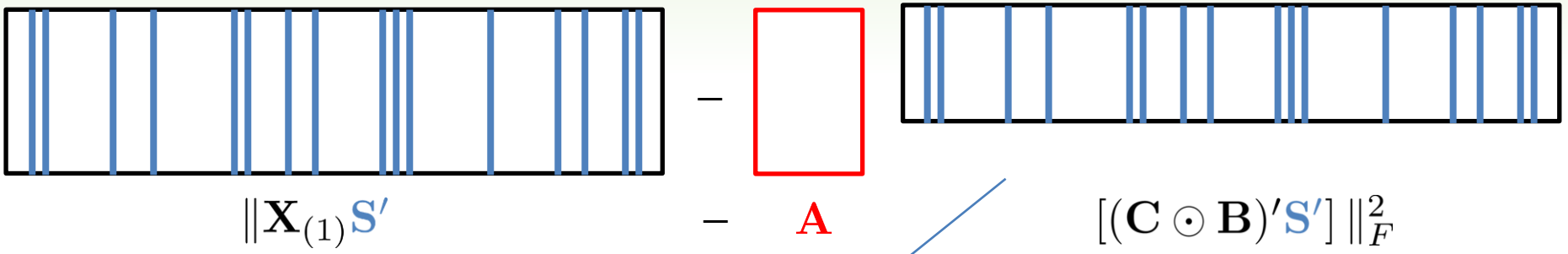


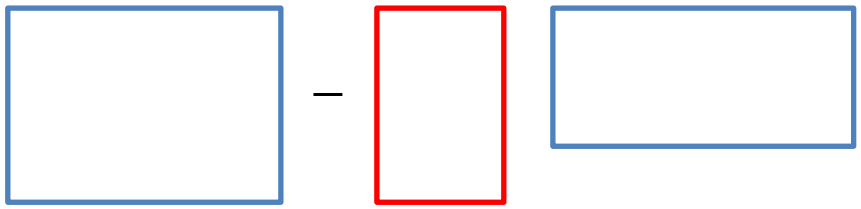


Data movement is often as expensive or <u>more</u> expensive than FLOPS.
Just move the minimum and no more.

Battaglino, Ballard, & Kolda 2017

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





Trick 2:

Each column in the sample is of the form: (C(ℓ,:) .* B(k,:))'

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

Battaglino, Ballard, & Kolda 2017

CP-ALS-RAND (No Mixing)

1: **while** not converged **do**
2: $\quad \left[\mathbf{A} \leftarrow \mathbf{X}_{(1)}\mathbf{S}_1{}'\right] / \left[(\mathbf{C} \odot \mathbf{B})'\mathbf{S}_1{}'\right]$
3: $\quad \left[\mathbf{B} \leftarrow \mathbf{X}_{(2)}\mathbf{S}_2{}'\right] / \left[(\mathbf{C} \odot \mathbf{A})'\mathbf{S}_2{}'\right]$
4: $\quad \left[\mathbf{C} \leftarrow \mathbf{X}_{(3)}\mathbf{S}_3{}'\right] / \left[(\mathbf{B} \odot \mathbf{A})'\mathbf{S}_3{}'\right]$
5: **end while**

No mixing performed here, yet converges in many cases!

Type equation here.Theorem: Khatri-Rao product can do some mixing (see paper).

$$\mu(\mathbf{A} \odot \mathbf{B}) \leq \mu(\mathbf{A})\mu(\mathbf{B})$$

Nevertheless, some problems still require mixing to converge.
Skipping details but can talk offline.

Battaglino, Ballard, & Kolda 2017

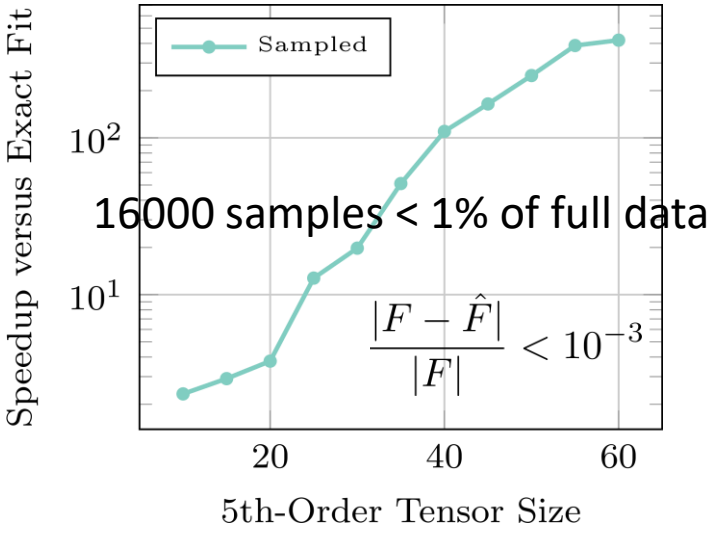
Randomizing the Convergence Check

Repeat until convergence…

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

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

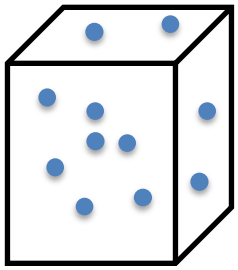
$$\min_{\mathbf{C}} \|\mathbf{X}_{(3)} - \mathbf{C}(\mathbf{B} \odot \mathbf{A})'\|^2$$



16000 samples < 1% of full data

$$F(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{ijk} (x_{ijk} - m_{ijk})^2$$

$$\text{s.t. } \mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$



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

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

Battaglino, Ballard, & Kolda 2017

CP-ALS-RAND

Given data tensor $\mathcal{X}$ and initial guesses for $\mathbf{A}, \mathbf{B}, \mathbf{C}$.
Fix $\hat{\Omega}$ with $|\hat{\Omega}| = p$.

1: $\hat{F}_0 \leftarrow \sum_{ijk \in \hat{\Omega}} (x_{ijk} - m_{ijk})^2$
2: **for** $\ell = 1, 2, \ldots$ **do**
3: **for** $t = 1, \ldots, T$ **do**
4: $\mathbf{A} \leftarrow \arg\min_{\mathbf{A}} \|\mathbf{X}_{(1)}\mathbf{S}^{\mathbf{A}}_{\ell,t} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})'\mathbf{S}^{\mathbf{A}}_{\ell,t}\|$
5: $\mathbf{B} \leftarrow \arg\min_{\mathbf{B}} \|\mathbf{X}_{(2)}\mathbf{S}^{\mathbf{B}}_{\ell,t} - \mathbf{B}(\mathbf{C} \odot \mathbf{A})'\mathbf{S}^{\mathbf{B}}_{\ell,t}\|$
6: $\mathbf{C} \leftarrow \arg\min_{\mathbf{C}} \|\mathbf{X}_{(3)}\mathbf{S}^{\mathbf{C}}_{\ell,t} - \mathbf{C}(\mathbf{B} \odot \mathbf{A})'\mathbf{S}^{\mathbf{C}}_{\ell,t}\|$
7: **end for**
8: $\hat{F}_\ell \leftarrow \sum_{ijk \in \hat{\Omega}} (x_{ijk} - m_{ijk})^2$
9: Check convergence
10: **end for**

Epoch

Same sample every time

Different sampling matrices every time

Randomized Least Squares

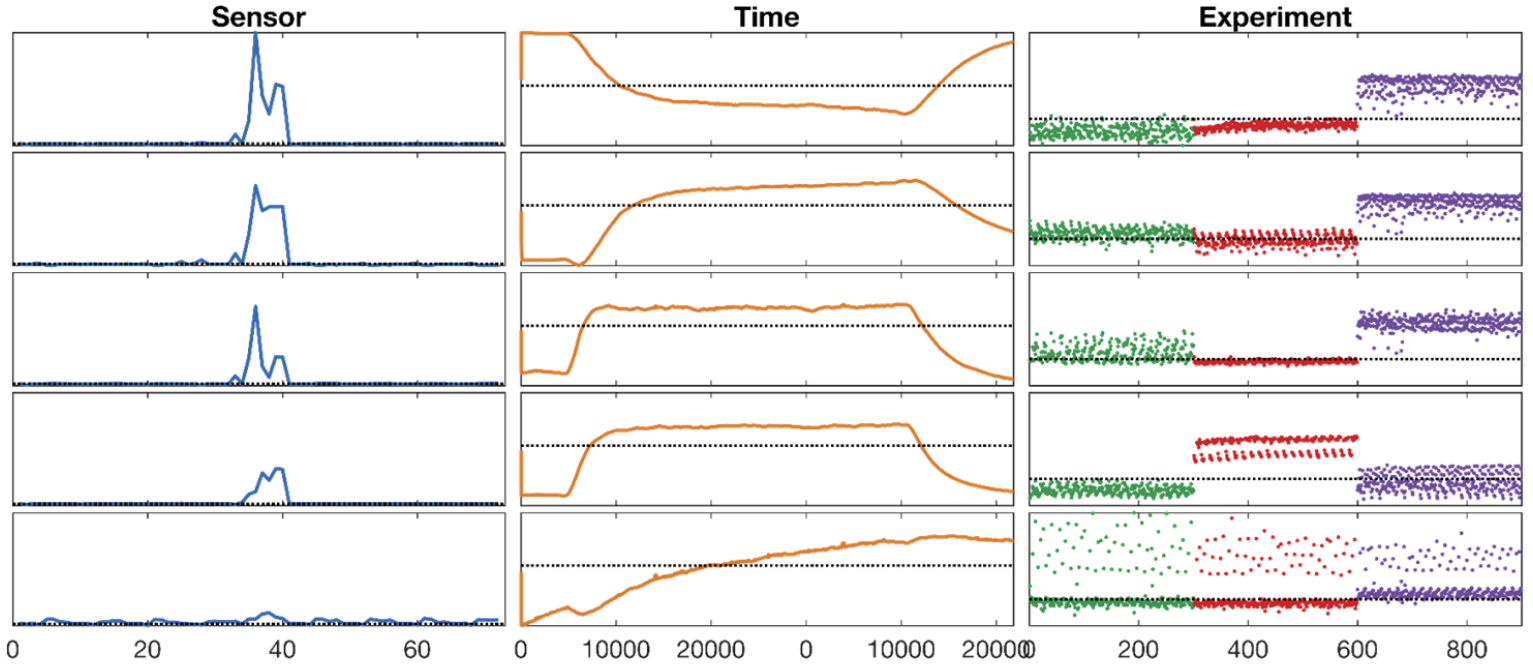
Use *approximate* function value

Acar, Dunlavy & Kolda, 2011

Speed Advantage: Analysis of Hazardous Gas Data

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

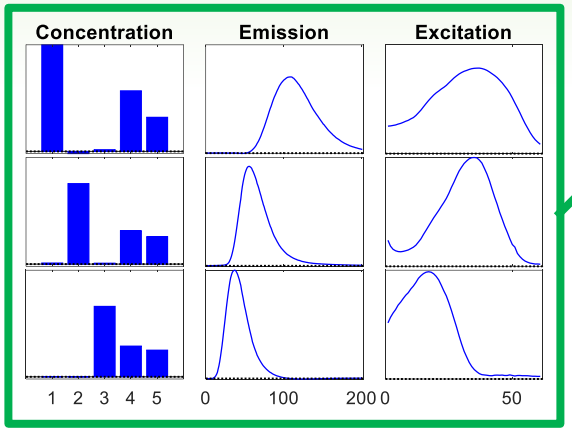
Method	Median Time (s)	Median Fit	Median Classification Error
CPRAND	53.6	0.715	0.61%
CP-ALS (H)	578.4	0.724	0.67%
CP-ALS (R)	204.7	0.724	0.67%



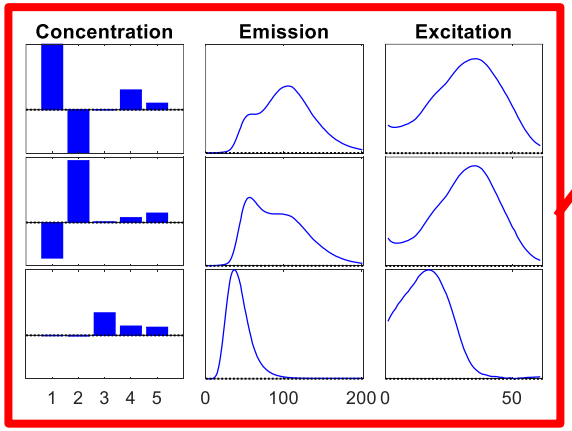
Data from Vergara et al. 2013; see also Vervliet and De Lathauwer (2016)

Battaglino, Ballard, & Kolda 2017

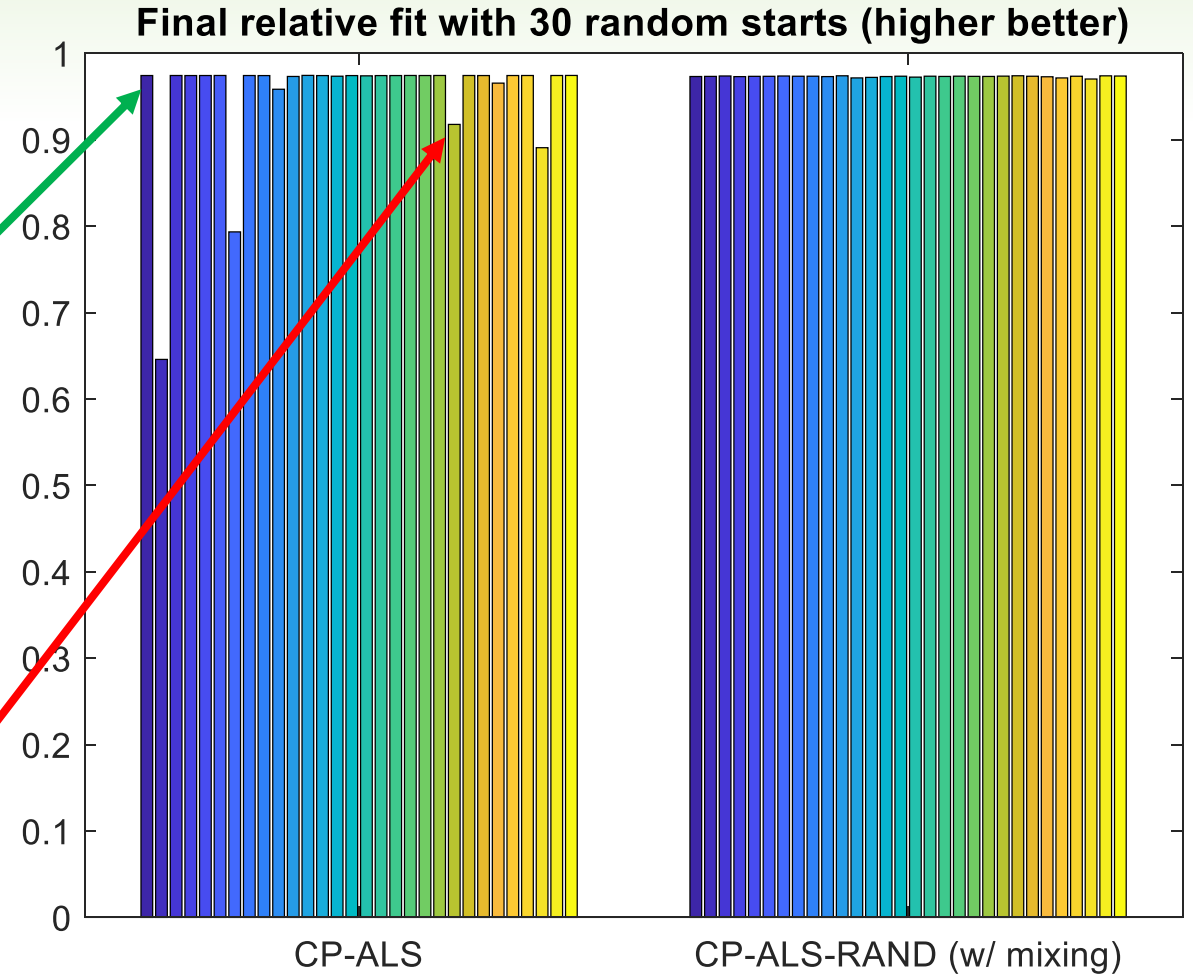
Globalization Advantage? Amino Acids Data



Fit = 0.97

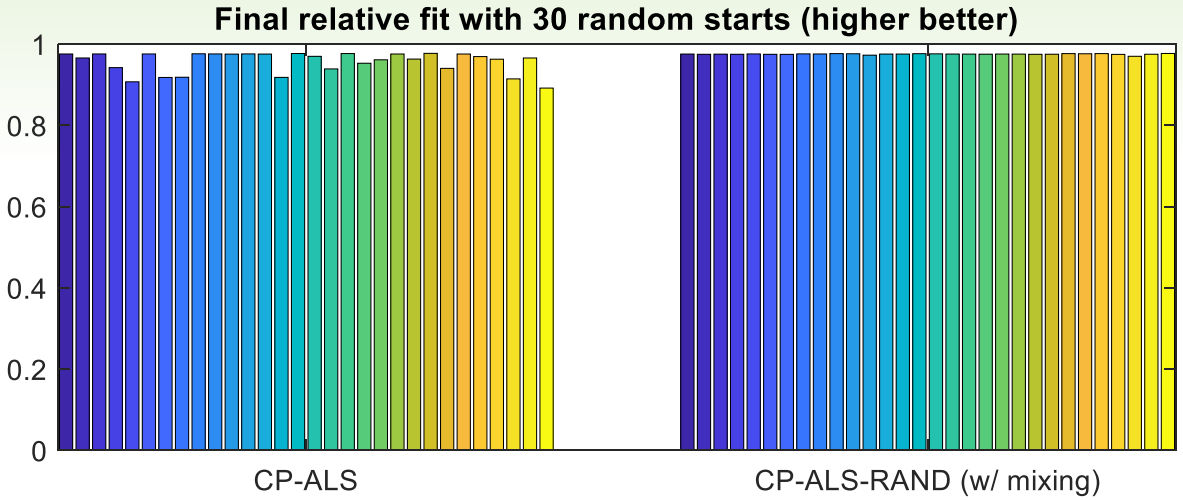


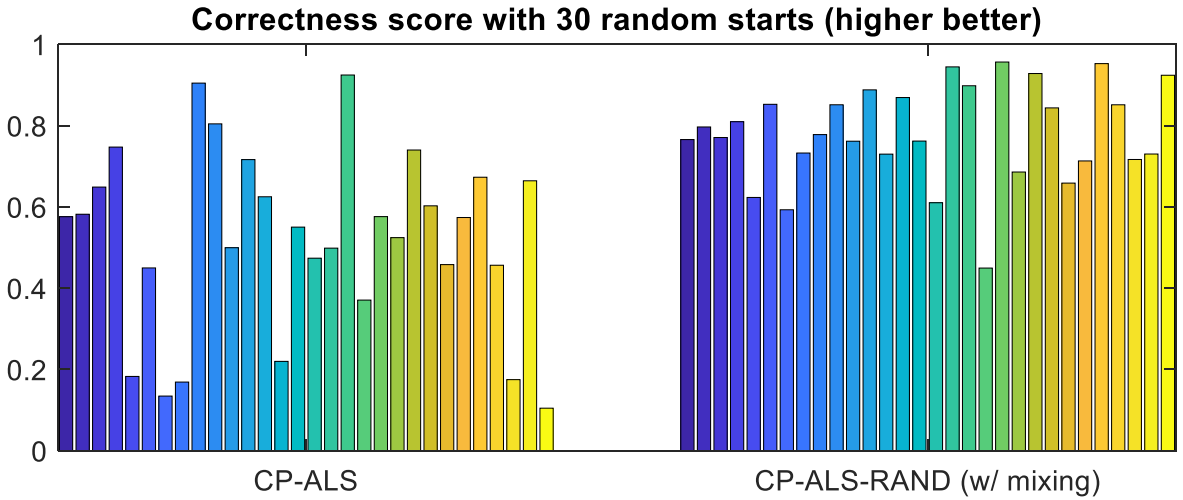
Fit = 0.92



Benefits are not as clear without mixing.

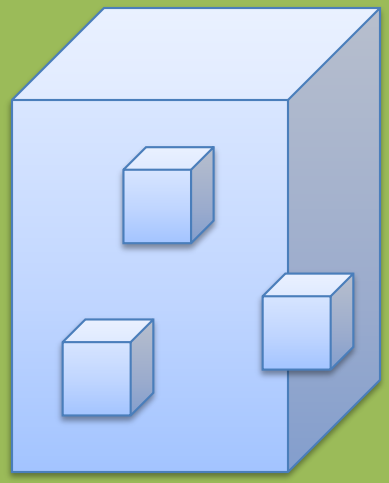
Globalization Advantage? Comparison for Rank 4



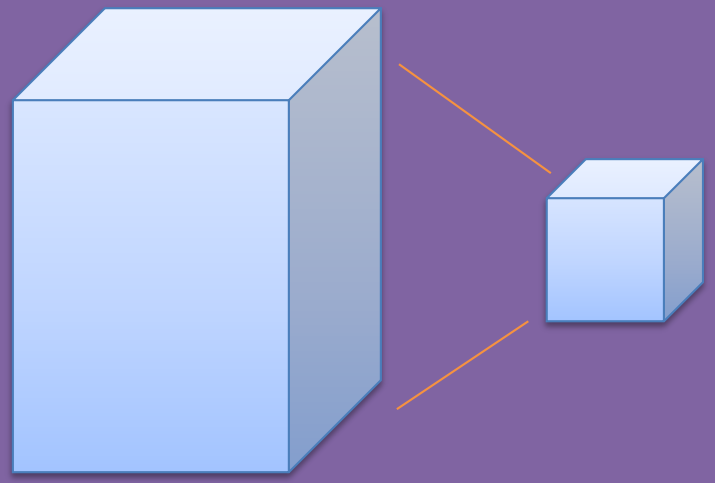


Other Randomized Methods

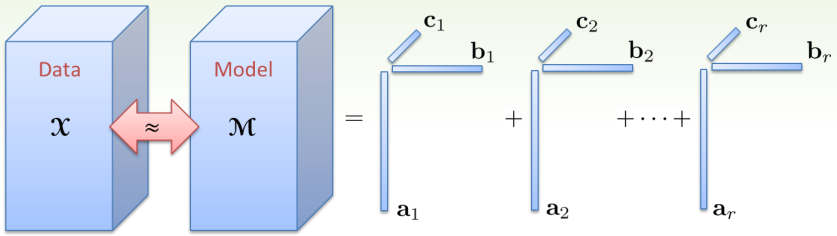
- Vervliet and De Lathauwer (2016) – Select a random d-way sub-block of data rather than random elements
  - Different sub-block at every iteration
  - Not contiguous as pictured



- Zhou, Cichocki, and Xie (2014) – Random projections (sketch) to smaller tensor
  - Project once and done



CP-OPT: Fitting CP Model via Direct Optimization



$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \underbrace{\|\mathcal{X} - \mathcal{M}\|^2}_{F(\mathbf{A},\mathbf{B},\mathbf{C})} \text{ s.t. } \underbrace{\mathcal{M} = [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]}_{m_{ijk} = \sum_{\ell=1}^{r} a_{i\ell}\, b_{j\ell}\, c_{k\ell}}$$

We have that $F(\mathbf{A},\mathbf{B},\mathbf{C}) = \sum_{ijk} f(x_{ijk}, m_{ijk})$ where $f(x,m) = \frac{1}{2}(x-m)^2$

Define a tensor $\mathcal{G}$ such that $g_{ijk} = \frac{\partial f}{\partial m}(x_{ijk}, m_{ijk}) = -(x_{ijk} - m_{ijk})$

$$\frac{\partial F}{\partial a_{i\ell}} = \sum_{jk} \frac{\partial f}{\partial m_{ijk}} \frac{\partial m_{ijk}}{\partial a_{i\ell}} = \sum_{jk} g_{ijk}\, b_{j\ell}\, c_{k\ell}$$

$$\frac{\partial F}{\partial b_{j\ell}} = \sum_{ik} \frac{\partial f}{\partial m_{ijk}} \frac{\partial m_{ijk}}{\partial b_{j\ell}} = \sum_{ij} g_{ijk}\, a_{i\ell}\, c_{k\ell}$$

$$\frac{\partial F}{\partial c_{k\ell}} = \sum_{ij} \frac{\partial f}{\partial m_{ijk}} \frac{\partial m_{ijk}}{\partial c_{k\ell}} = \sum_{ik} g_{ijk}\, a_{i\ell}\, b_{j\ell}$$

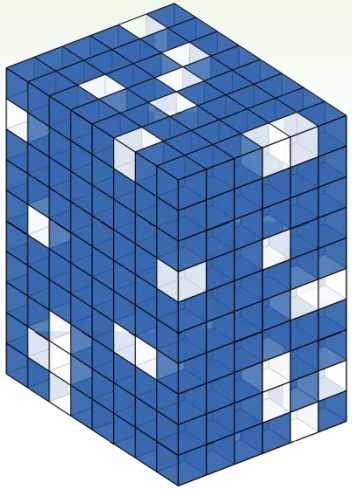
MTTKRP!

$$\frac{\partial F}{\partial \mathbf{A}} = \mathbf{G}_{(1)}(\mathbf{C} \odot \mathbf{B})$$

$$\frac{\partial F}{\partial \mathbf{B}} = \mathbf{G}_{(2)}(\mathbf{C} \odot \mathbf{A})$$

$$\frac{\partial F}{\partial \mathbf{C}} = \mathbf{G}_{(3)}(\mathbf{B} \odot \mathbf{A})$$

What if some data is missing?



Ω = *known entries (blue)*

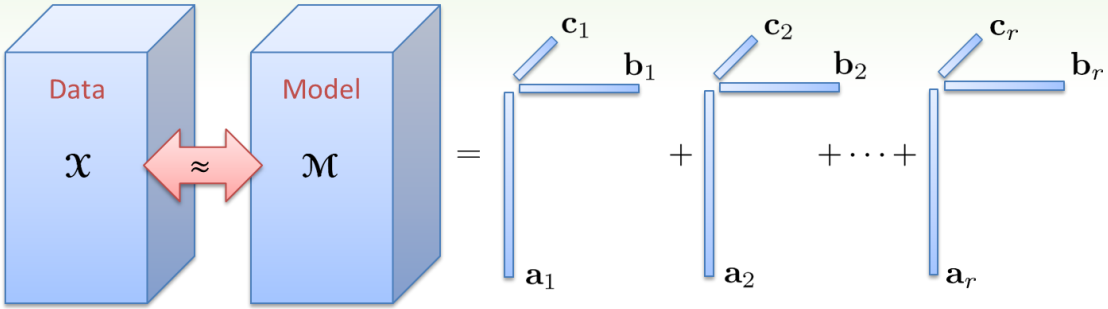
$$F(\mathbf{A},\mathbf{B},\mathbf{C}) = \sum_{ijk} f(x_{ijk}, m_{ijk})$$

$$F(\mathbf{A},\mathbf{B},\mathbf{C}) = \sum_{ijk\in\Omega} f(x_{ijk}, m_{ijk})$$

Redefine the tensor $\mathcal{G}$ such that $g_i = \begin{cases} \frac{\partial f}{\partial m}(x_i, m_i) & \text{if } i \in \Omega, \\ 0 & \text{if } i \notin \Omega. \end{cases}$

$$\frac{\partial F}{\partial \mathbf{A}} = \mathbf{G}_{(1)}(\mathbf{C} \odot \mathbf{B}) \qquad \frac{\partial F}{\partial \mathbf{B}} = \mathbf{G}_{(2)}(\mathbf{C} \odot \mathbf{A}) \qquad \frac{\partial F}{\partial \mathbf{C}} = \mathbf{G}_{(3)}(\mathbf{B} \odot \mathbf{A})$$

Generalized CP



$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \sum_{ijk\in\Omega} f(x_{ijk}, m_{ijk})$$

$$\text{s.t. } \mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

"Standard" CP uses:

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

"Exponential" CP uses :

$$f(x,m) = xm - \log m \quad (x \geq 0, m \geq 0)$$

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

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

"Boolean" CP uses:

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

Anderson-Bergman, Hong, and Kolda 2017

Fitting (Generalized) CP

Given data tensor $\mathcal{X}$ and initial guesses for $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

1: $F_0 \leftarrow \sum_{ijk \in \Omega} f(x_{ijk}, m_{ijk})$
2: **for** $\ell = 1, 2, \ldots$ **do**
3: Compute $\mathcal{G}$
4: $\Delta_{\mathbf{A}} \leftarrow \mathbf{G}_{(1)}(\mathbf{C} \odot \mathbf{B})$
5: $\Delta_{\mathbf{B}} \leftarrow \mathbf{G}_{(2)}(\mathbf{C} \odot \mathbf{A})$
6: $\Delta_{\mathbf{C}} \leftarrow \mathbf{G}_{(3)}(\mathbf{B} \odot \mathbf{A})$
7: Determine step length α
8: $\mathbf{A} \leftarrow \mathbf{A} - \alpha\Delta_{\mathbf{A}}$
9: $\mathbf{B} \leftarrow \mathbf{B} - \alpha\Delta_{\mathbf{B}}$
10: $\mathbf{C} \leftarrow \mathbf{C} - \alpha\Delta_{\mathbf{C}}$
11: $F_\ell \leftarrow \sum_{ijk \in \Omega} f(x_{ijk}, m_{ijk})$
12: Check convergence
13: **end for**

Compute Gradient

Compute optimization step and take it

Use either checks on function value or gradient

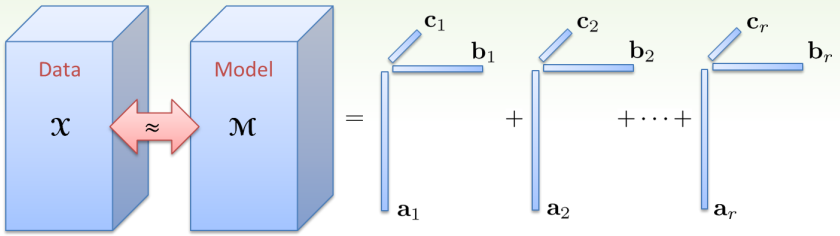
Acar, Dunlavy & Kolda, 2011

Some Observations

- Can use any optimization method
  - Showing steepest descent
  - Can use 1st and 2nd-order methods
  - Can add constraints
- Can easily add regularization

$$F(\mathbf{A},\mathbf{B},\mathbf{C}) = \sum_{ijk\in\Omega} f(x_{ijk}, m_{ijk}) + \rho_{\mathbf{A}}(\mathbf{A}) + \rho_{\mathbf{B}}(\mathbf{B}) + \rho_{\mathbf{C}}(\mathbf{C})$$

- G sparse ⇒ efficient gradient computation
  - Known data Ω "sparse" ⇒ G sparse
- Sparse data tensor (X) ⇏ Sparse gradient tensor (G)
  - Some choices for f yield special structure that can be exploited
  - But not generally the case
- Stochastic method yields sparse "known data"



Given data tensor $\mathcal{X}$ and initial guesses for $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

1: $F_0 \leftarrow \sum_{ijk\in\Omega} f(x_{ijk}, m_{ijk})$
2: **for** $\ell = 1, 2, \ldots$ **do**
3: Compute $\mathcal{G}$
4: $\Delta_{\mathbf{A}} \leftarrow \mathbf{G}_{(1)}(\mathbf{C} \odot \mathbf{B})$
5: $\Delta_{\mathbf{B}} \leftarrow \mathbf{G}_{(2)}(\mathbf{C} \odot \mathbf{A})$
6: $\Delta_{\mathbf{C}} \leftarrow \mathbf{G}_{(3)}(\mathbf{B} \odot \mathbf{A})$
7: Determine step length α
8: $\mathbf{A} \leftarrow \mathbf{A} - \alpha\Delta_{\mathbf{A}}$
9: $\mathbf{B} \leftarrow \mathbf{B} - \alpha\Delta_{\mathbf{B}}$
10: $\mathbf{C} \leftarrow \mathbf{C} - \alpha\Delta_{\mathbf{C}}$
11: $F_\ell \leftarrow \sum_{ijk\in\Omega} f(x_{ijk}, m_{ijk})$
12: Check convergence
13: **end for**

Fitting (Generalized) CP with SGD

Given data tensor $\mathcal{X}$ and initial guesses for $\mathbf{A}, \mathbf{B}, \mathbf{C}$.
Fix $\hat{\Omega} \subset \Omega$ with $|\hat{\Omega}| = s$.

1: $\hat{F}_0 \leftarrow \sum_{ijk \in \hat{\Omega}} f(x_{ijk}, m_{ijk})$
2: **for** $\ell = 1, 2, \ldots$ **do**
3: **for** $t = 1, \ldots, T$ **do**
4: $\bar{\Omega}_{\ell,t} \leftarrow p$ random entries in Ω
5: Compute $\mathcal{G}$ *only at entries in* $\bar{\Omega}_{\ell,t}$ (everywhere else is zero)
6: $\Delta_{\mathbf{A}} \leftarrow \bar{\mathbf{G}}_{(1)}(\mathbf{C} \odot \mathbf{B})$
7: $\Delta_{\mathbf{B}} \leftarrow \bar{\mathbf{G}}_{(2)}(\mathbf{C} \odot \mathbf{A})$
8: $\Delta_{\mathbf{C}} \leftarrow \bar{\mathbf{G}}_{(3)}(\mathbf{B} \odot \mathbf{A})$
9: $\alpha \leftarrow$ learning rate
10: $\mathbf{A} \leftarrow \mathbf{A} - \alpha\Delta_{\mathbf{A}}$
11: $\mathbf{B} \leftarrow \mathbf{B} - \alpha\Delta_{\mathbf{B}}$
12: $\mathbf{C} \leftarrow \mathbf{C} - \alpha\Delta_{\mathbf{C}}$
13: **end for**
14: $\hat{F}_\ell \leftarrow \sum_{ijk \in \hat{\Omega}} f(x_{ijk}, m_{ijk})$
15: Check convergence
16: **end for**

Epoch

Same sample every time

Different samples every time

Sparse MTTKRP, super cheap!

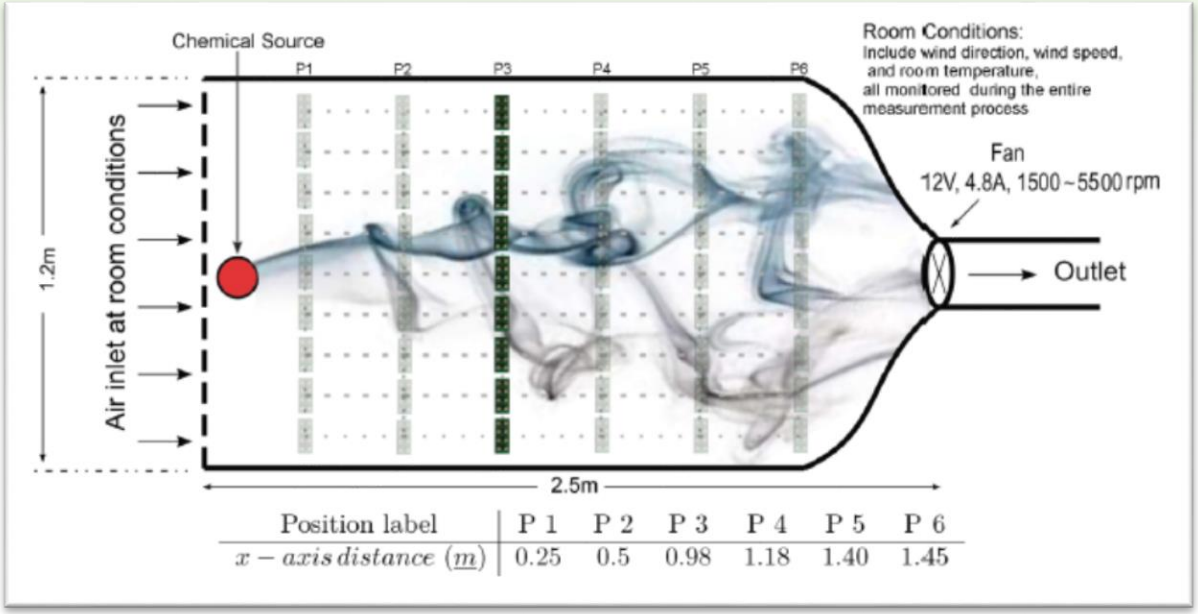
Compute Stochastic Gradient

Take tiny step (α is small)

Use *approximate* function value

Anderson-Bergman, Hong, Kolda (in progress)

Gas Dataset: 1.7 GB



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

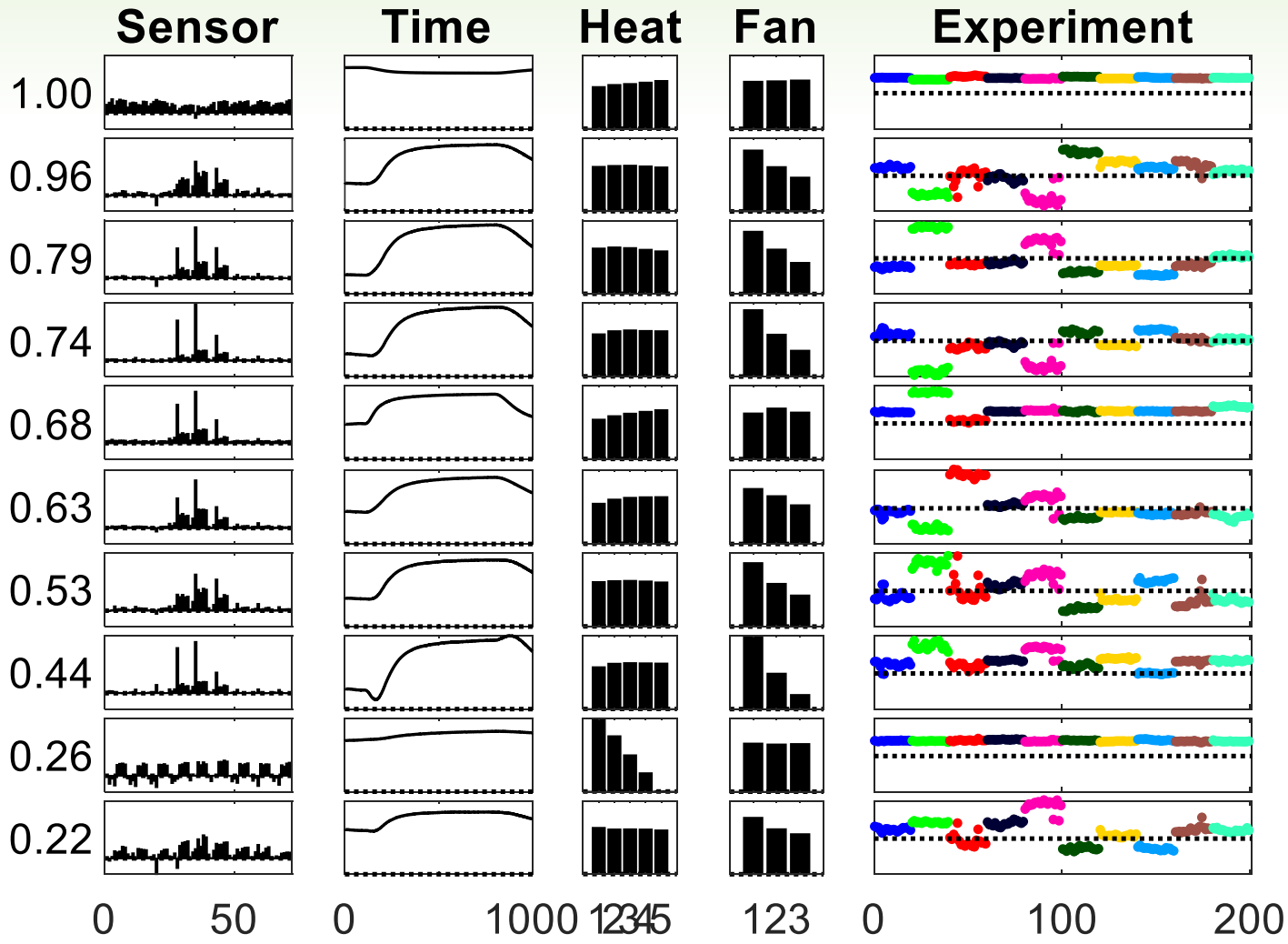
71 (sensors) x 1000 (timepoints) x 5 temps x 3 fan speeds x 200 exps

3.4% missing data

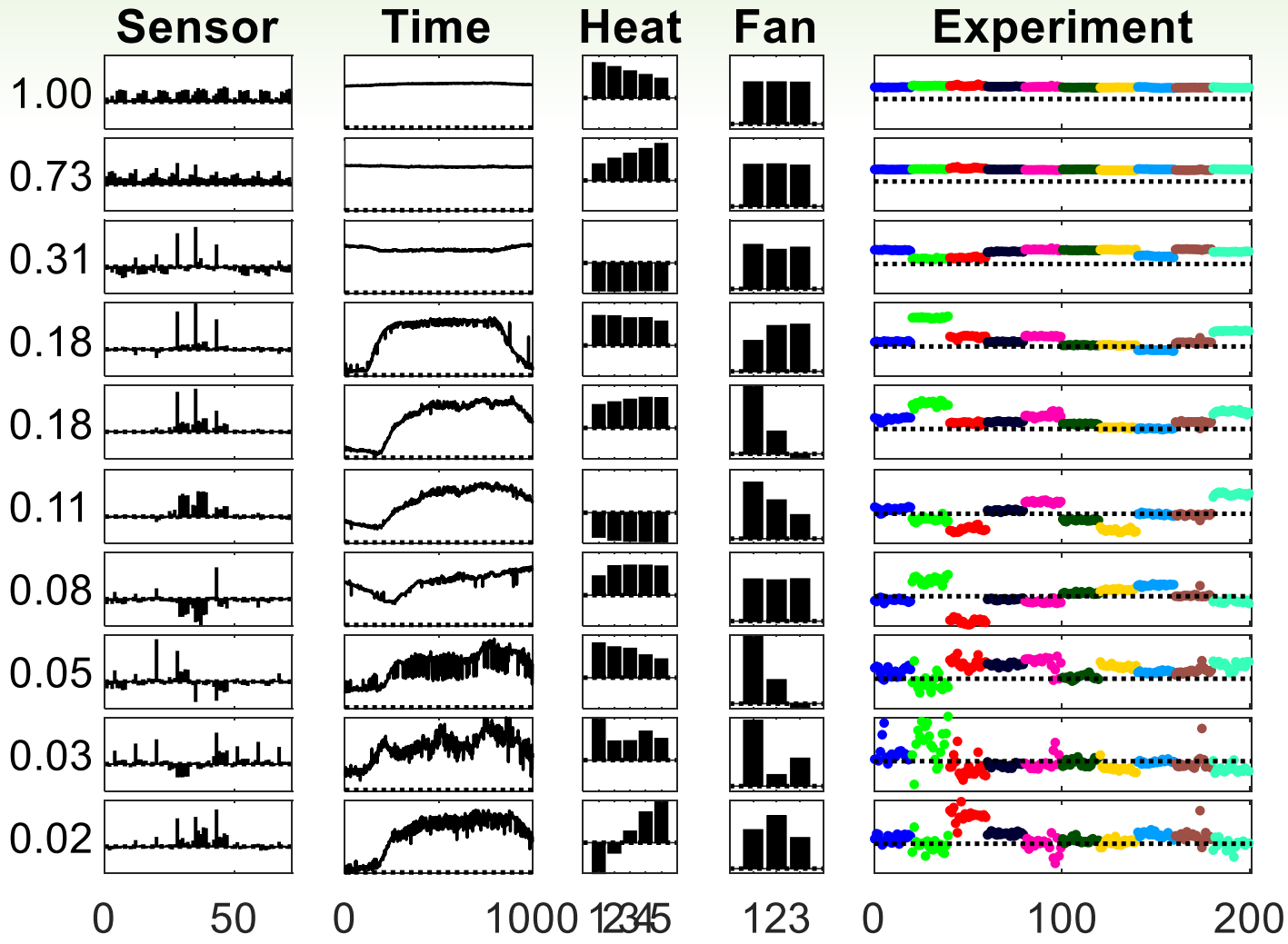
(10 gases x 20 replicates)

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

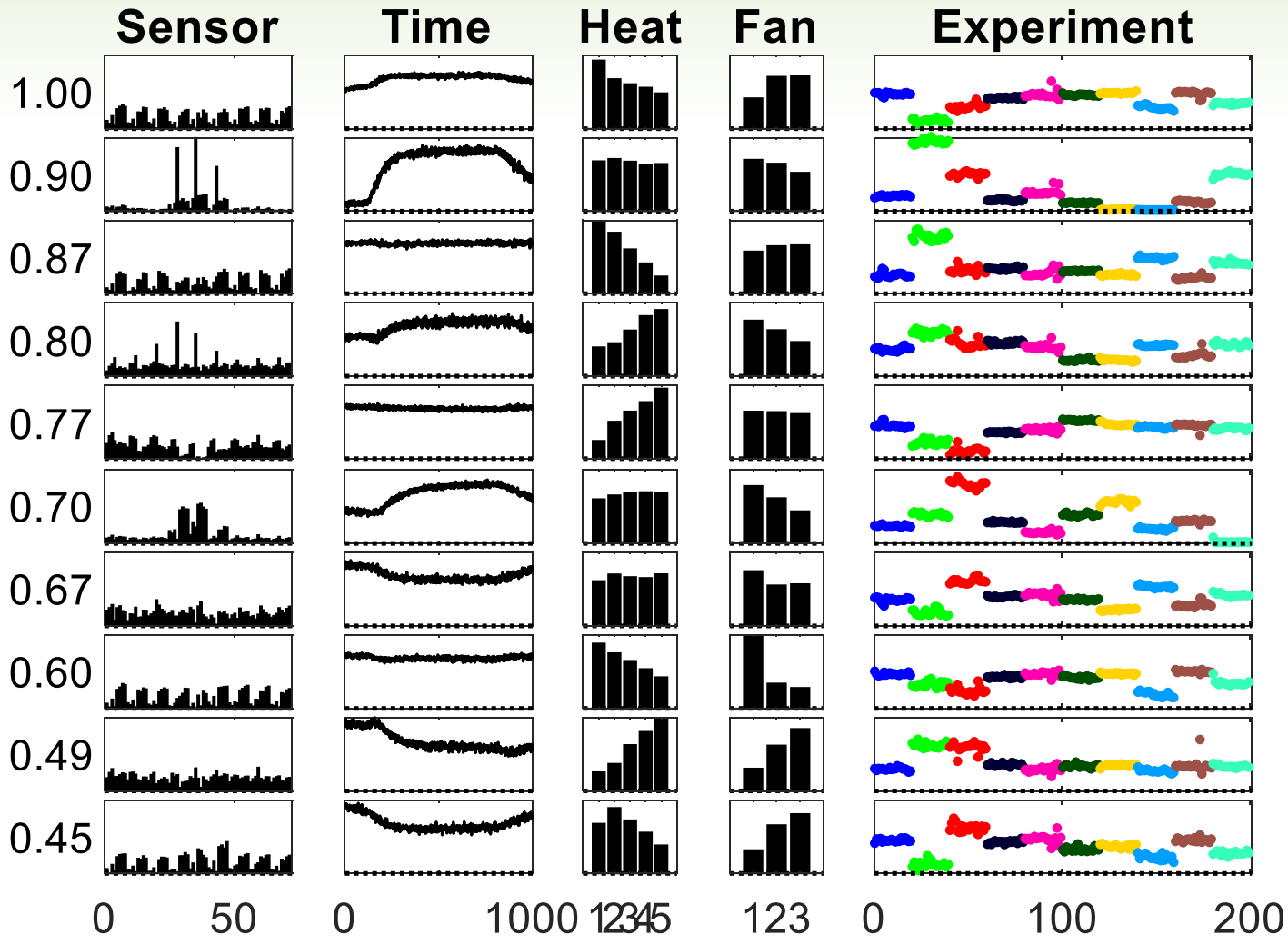
CP-ALS: 139s


Sensor
Time
Heat
Fan
Experiment
1.00
0.96
0.79
0.74
0.68
0.63
0.53
0.44
0.26
0.22
0
50
0
1000
1
2
3
4
5
1
2
3
0
100
200

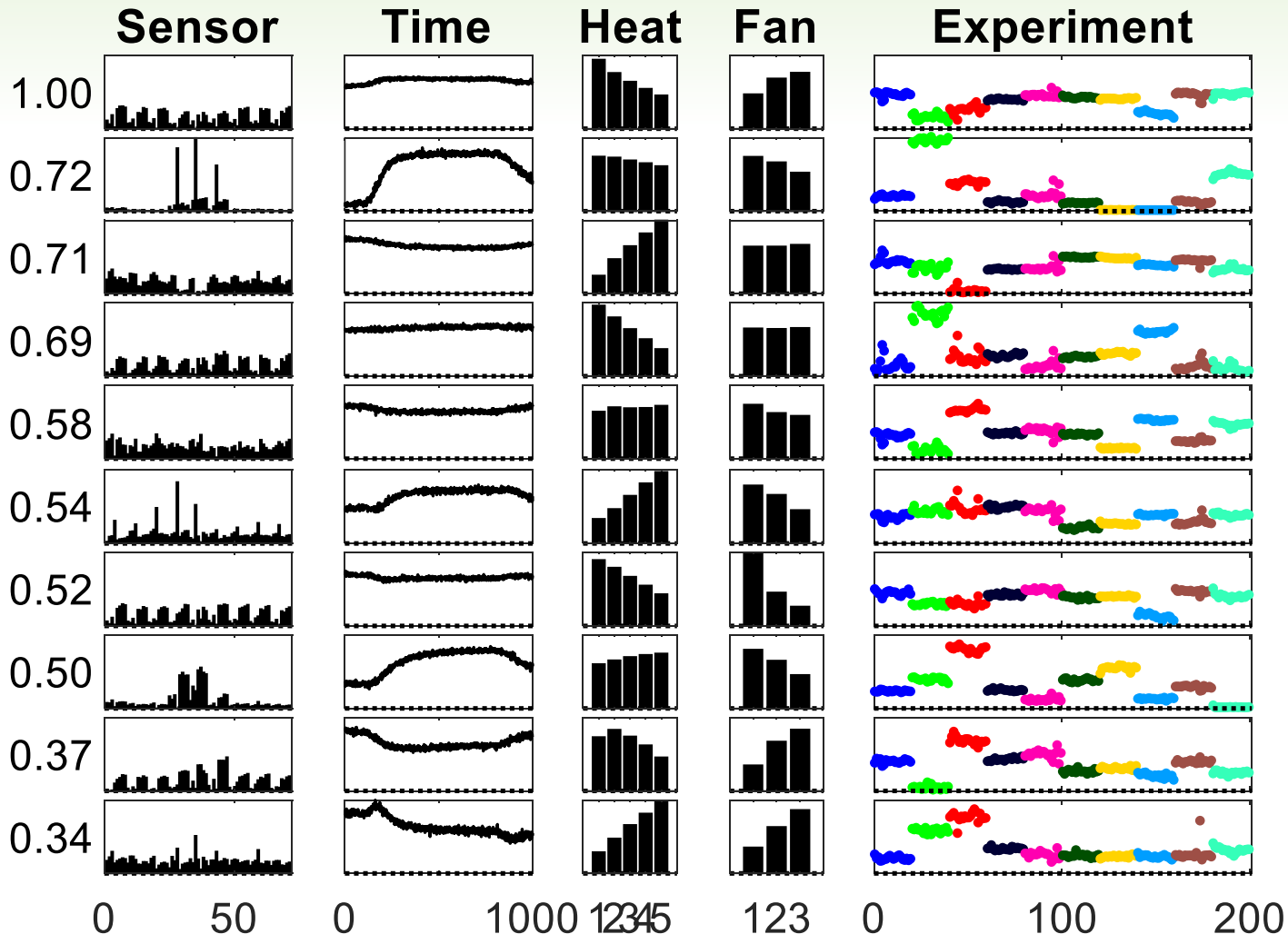
CP-ALS-RAND (no mixing): 18s



CP-SGD + nonneg: 102s

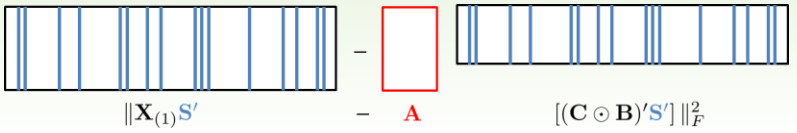


CP-SGD + nonneg + missing: 174s



Questions to Ponder This Week

- Matrix and Tensor Factorizations
  - Notion of "generalization error"?
- Contrasting and connections of the two approaches
- CP-ALS-RAND
  - # column samples
  - Sketching inside a larger algorithm?
- CP-SGD
  - # samples for gradient estimate?
  - Acceleration methods? (ADAM?)
  - Stratified sampling for sparse data?
- Both
  - # samples for function estimate?
  - # iterations per epoch?
  - # epochs w/o improvement before stopping?
  - Choice or rank (# components)
  - How to choose random samples



Given data tensor $\mathcal{X}$ and initial guesses for $\mathbf{A}, \mathbf{B}, \mathbf{C}$.
Fix $\hat{\Omega}$ with $|\hat{\Omega}| = p$.

1: $\hat{F}_0 \leftarrow \sum_{ijk\in\hat{\Omega}}(x_{ijk} - m_{ijk})^2$ — *Same sample every time*
2: **for** $\ell = 1, 2, \ldots$ **do**
3: **for** $t = 1, \ldots, T$ **do** — Epoch — *Different sampling matrices every time*
4: $\mathbf{A} \leftarrow \arg\min_{\mathbf{A}} \|\mathbf{X}_{(1)}\mathbf{S}^{\mathbf{A}}_{\ell,t} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})'\mathbf{S}^{\mathbf{A}}_{\ell,t}\|$
5: $\mathbf{B} \leftarrow \arg\min_{\mathbf{B}} \|\mathbf{X}_{(2)}\mathbf{S}^{\mathbf{B}}_{\ell,t} - \mathbf{B}(\mathbf{C}\odot\mathbf{A})'\mathbf{S}^{\mathbf{B}}_{\ell,t}\|$
6: $\mathbf{C} \leftarrow \arg\min_{\mathbf{C}} \|\mathbf{X}_{(3)}\mathbf{S}^{\mathbf{C}}_{\ell,t} - \mathbf{C}(\mathbf{B}\odot\mathbf{A})'\mathbf{S}^{\mathbf{C}}_{\ell,t}\|$
7: **end for**
8: $\hat{F}_\ell \leftarrow \sum_{ijk\in\hat{\Omega}}(x_{ijk} - m_{ijk})^2$ — Use *approximate* function value
9: Check convergence
10: **end for**

Randomized Least Squares

Given data tensor $\mathcal{X}$ and initial guesses for $\mathbf{A}, \mathbf{B}, \mathbf{C}$.
Fix $\hat{\Omega} \subset \Omega$ with $|\hat{\Omega}| = s$.

1: $\hat{F}_0 \leftarrow \sum_{ijk\in\hat{\Omega}} f(x_{ijk}, m_{ijk})$ — *Same sample every time*
2: **for** $\ell = 1, 2, \ldots$ **do**
3: **for** $t = 1, \ldots, T$ **do** — Epoch — *Different samples every time*
4: $\bar{\Omega}_{\ell,t} \leftarrow p$ random entries in Ω
5: Compute $\mathcal{G}$ *only at entries in* $\bar{\Omega}_{\ell,t}$ (everywhere else is zero)
6: $\Delta_{\mathbf{A}} \leftarrow \bar{\mathbf{G}}_{(1)}(\mathbf{C}\odot\mathbf{B})$
7: $\Delta_{\mathbf{B}} \leftarrow \bar{\mathbf{G}}_{(2)}(\mathbf{C}\odot\mathbf{A})$ — *Sparse MTTKRP, super cheap!*
8: $\Delta_{\mathbf{C}} \leftarrow \bar{\mathbf{G}}_{(3)}(\mathbf{B}\odot\mathbf{A})$
9: $\alpha \leftarrow$ learning rate
10: $\mathbf{A} \leftarrow \mathbf{A} - \alpha\Delta_{\mathbf{A}}$ — Take tiny step (α is small)
11: $\mathbf{B} \leftarrow \mathbf{B} - \alpha\Delta_{\mathbf{B}}$
12: $\mathbf{C} \leftarrow \mathbf{C} - \alpha\Delta_{\mathbf{C}}$
13: **end for**
14: $\hat{F}_\ell \leftarrow \sum_{ijk\in\hat{\Omega}} f(x_{ijk}, m_{ijk})$
15: Check convergence — Use *approximate* function value
16: **end for**

Compute Stochastic Gradient

References

- **Generalized CP:** Cliff Anderson-Bergman, D. Hong, T. G. Kolda, ***Generalized Canonical Polyadic Tensor Decomposition***, 2017 (coming soon)
- **Randomized Least Squares CP-ALS:** C. Battaglino, G. Ballard and T. G. Kolda, ***A Practical Randomized CP Tensor Decomposition***, January 2017, arXiv:1701.06600
- **Random Block CP-ALS:** N. Vervliet and L. D. Lathauwer, ***A Randomized Block Sampling Approach to Canonical Polyadic Decomposition of Large-Scale Tensors***, IEEE Journal of Selected Topics in Signal Processing, Vol. 10, No. 2, pp. 284-295, March 2016, doi:10.1109/JSTSP.2015.2503260
- **Sketching and CP-ALS:** G. Zhou, A. Cichocki and S. Xie, ***Decomposition of Big Tensors With Low Multilinear Rank***, December 2014, arXiv:1412.1885
- **Other related works…**
  - **Overall review:** T. G. Kolda and B. W. Bader, ***Tensor Decompositions and Applications***, SIAM Review, 2009, doi:10.1137/07070111X
  - **All-at-once Optimization Approach:** E. Acar, D. M. Dunlavy and T. G. Kolda, ***A Scalable Optimization Approach for Fitting Canonical Tensor Decompositions***, Journal of Chemometrics, 2011, doi:10.1002/cem.1335
  - **Missing data:** E. Acar, D. M. Dunlavy, T. G. Kolda, M. Mørup, ***Scalable Tensor Factorizations for Incomplete Data***, Chemometrics and Intelligent Laboratory Systems, 2011, doi:10.1016/j.chemolab.2010.08.004

Contact: Tammy Kolda, tgkolda@sandia.gov