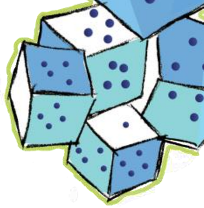
Illustration by Chris Brigman



Sandia National Laboratories

Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition

Tamara G. Kolda
Sandia National Labs

Brett Larsen
Stanford University

SIAM Annual Meeting 2020
Development in Machine Learning: Foundations and Applications
July 8, 2020

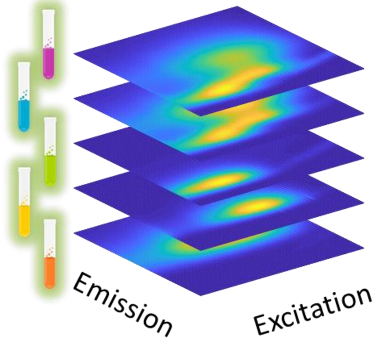
Supported by the DOE Office of Science Advanced Scientific Computing Research (ASCR) Applied Mathematics. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

U.S. DEPARTMENT OF ENERGY

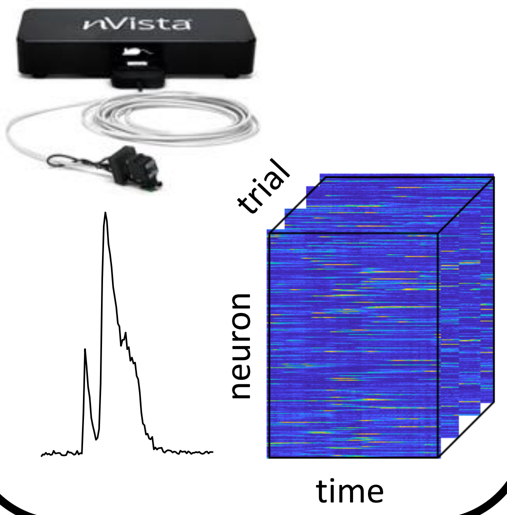
Tensors Come From Many Applications

- **Chemometrics:** Emission x Excitation x Samples (Fluorescence Spectroscopy)
- **Neuroscience:** Neuron x Time x Trial
- **Criminology:** Day x Hour x Location x Crime (Chicago Crime Reports)
- **Machine Learning:** Multivariate Gaussian Mixture Models Higher-Order Moments
- **Transportation:** Pickup x Dropoff x Time (Taxis)
- **Sports:** Player x Statistic x Season (Basketball)
- **Cyber-Traffic:** IP x IP x Port x Time
- **Social Network:** Person x Person x Time x Interaction-Type
- **Signal Processing:** Sensor x Frequency x Time
- **Trending Co-occurrence:** Term A x Term B x Time

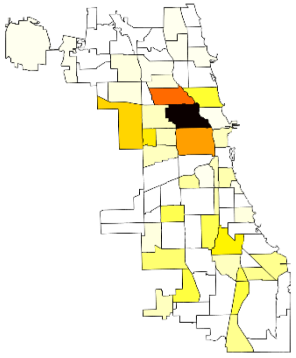
Chemometrics



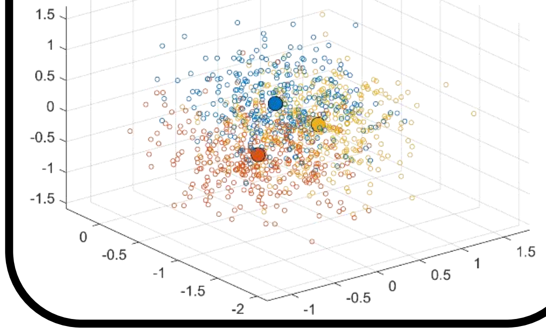
Neuroscience



Criminology



Machine Learning



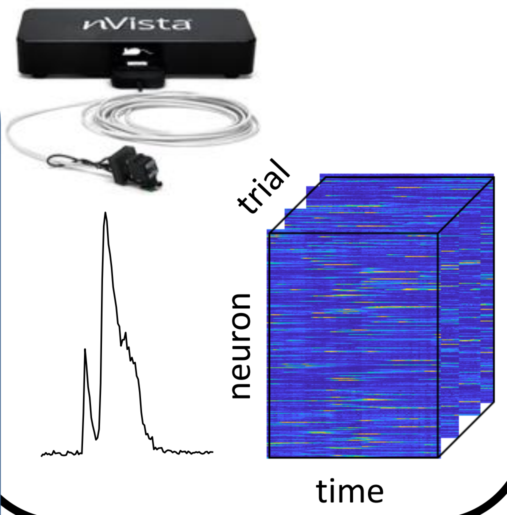
Tensors Come From Many Applications

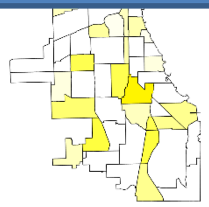
- **Chemometrics:** Emission x Excitation x Samples (Fluorescence Spectroscopy)
- **Neuroscience:** N
- **Criminology:** D (Chicago Crime
- **Machine Learn** Mixture Model
- **Transportation**
- **Sports:** Player
- **Cyber-Traffic:** I
- **Social Network** Interaction-Type
- **Signal Processing:** Sensor x Frequency x Time
- **Trending Co-occurrence:** Term A x Term B x Time

Tensor Decomposition Finds Patterns in Massive Data (Unsupervised Learning)

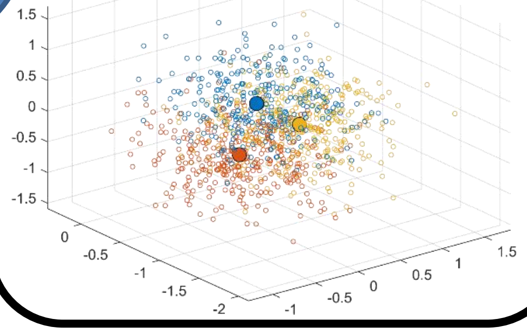
Chemometrics

Neuroscience

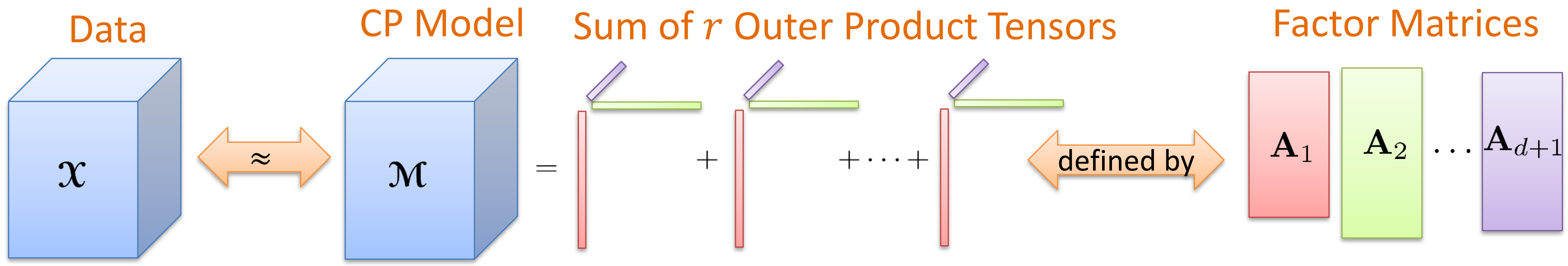




Machine Learning



Tensor Decomposition Identifies Factors



$$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d+1}}$$

$$\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{d+1}]\!] \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d+1}}$$

$$\mathbf{A}_k \in \mathbb{R}^{n_k \times r}$$

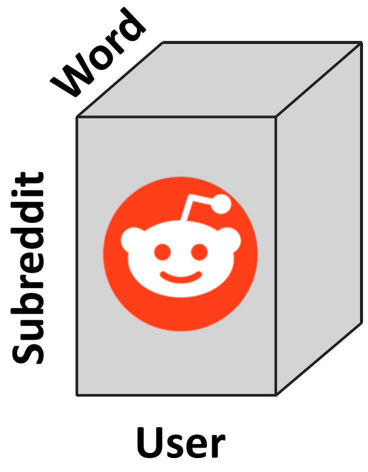
Model Rank

$$x_i = x(i_1, i_2, \ldots, i_{d+1})$$

$$m_i = m(i_1, i_2, \ldots, i_{d+1}) = \sum_{j=1}^{r} \prod_{k=1}^{d+1} a_k(i_k, j)$$

Example Sparse Multiway Data: Reddit

- Reddit is an American social news aggregator, web content rating, and discussion website
  - A “subreddit” is a discussion forum on a particular topic
- Tensor obtained from frost.io (http://frostt.io/tensors/reddit-2015/)
  - Built from reddit comments posted in the year 2015
  - Users and words with less than 5 entries have been removed



Reddit Tensor

8 million users
200 thousand subreddits
8 million words

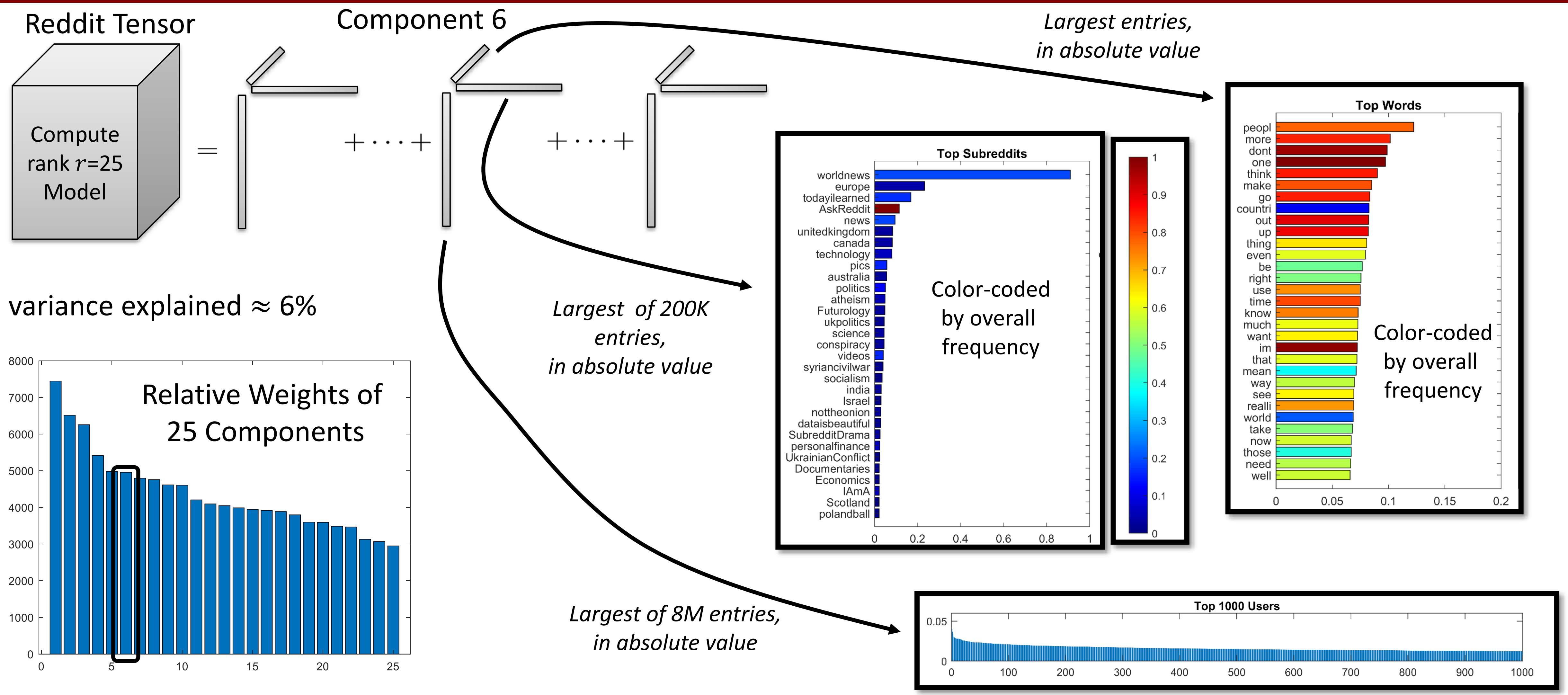
4.7 billion non-zeros ($10^{-8}\%$)
106 gigabytes

$x(i,j,k) = \log$ (1 + the number of times user i used word j in subreddit k)

Used a rank $\boldsymbol{r = 25}$ decompsition

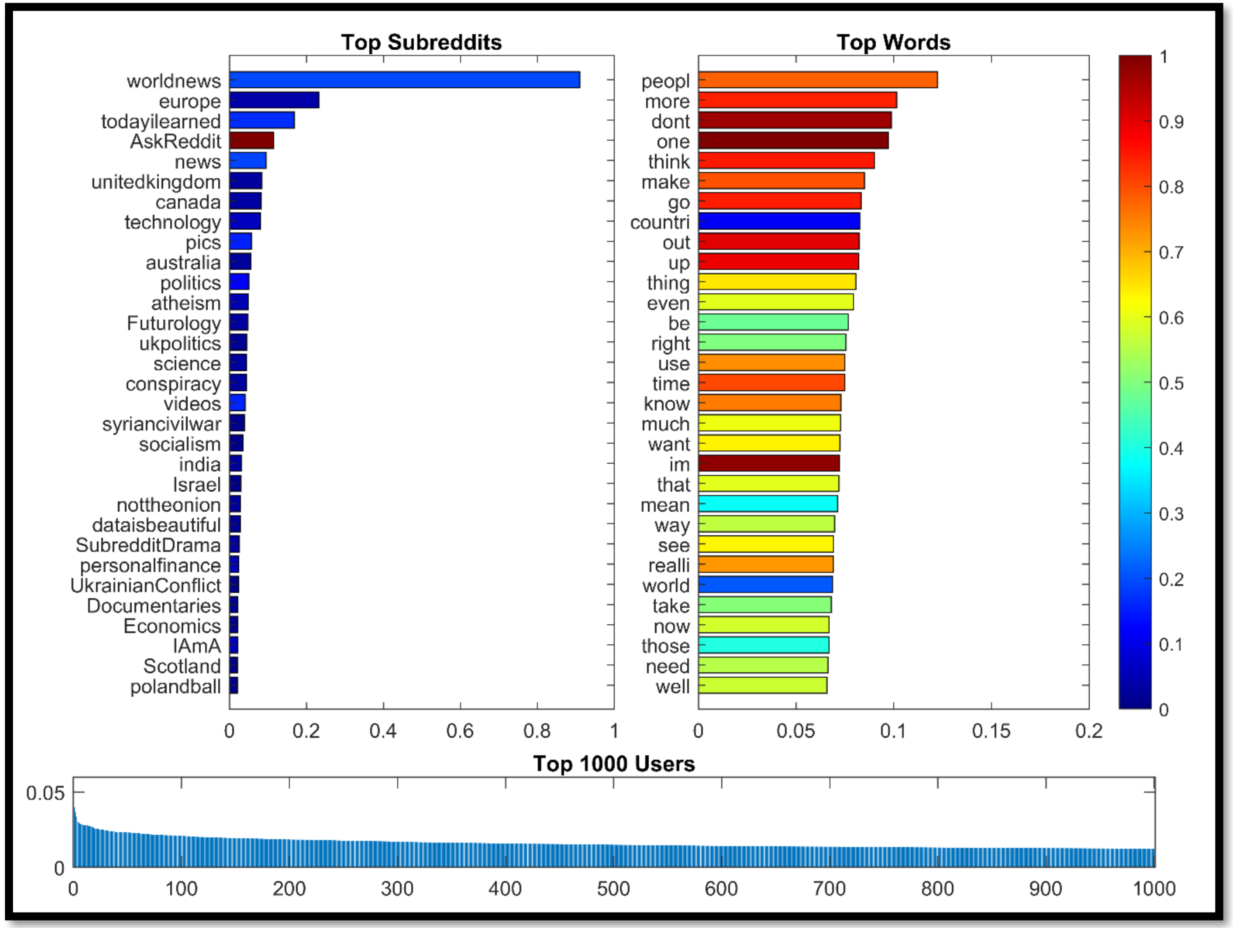
Smith et al (2017). “FROSTT: The Formidable Open Repository of Sparse Tensors and Tools”

Interpreting Reddit Components

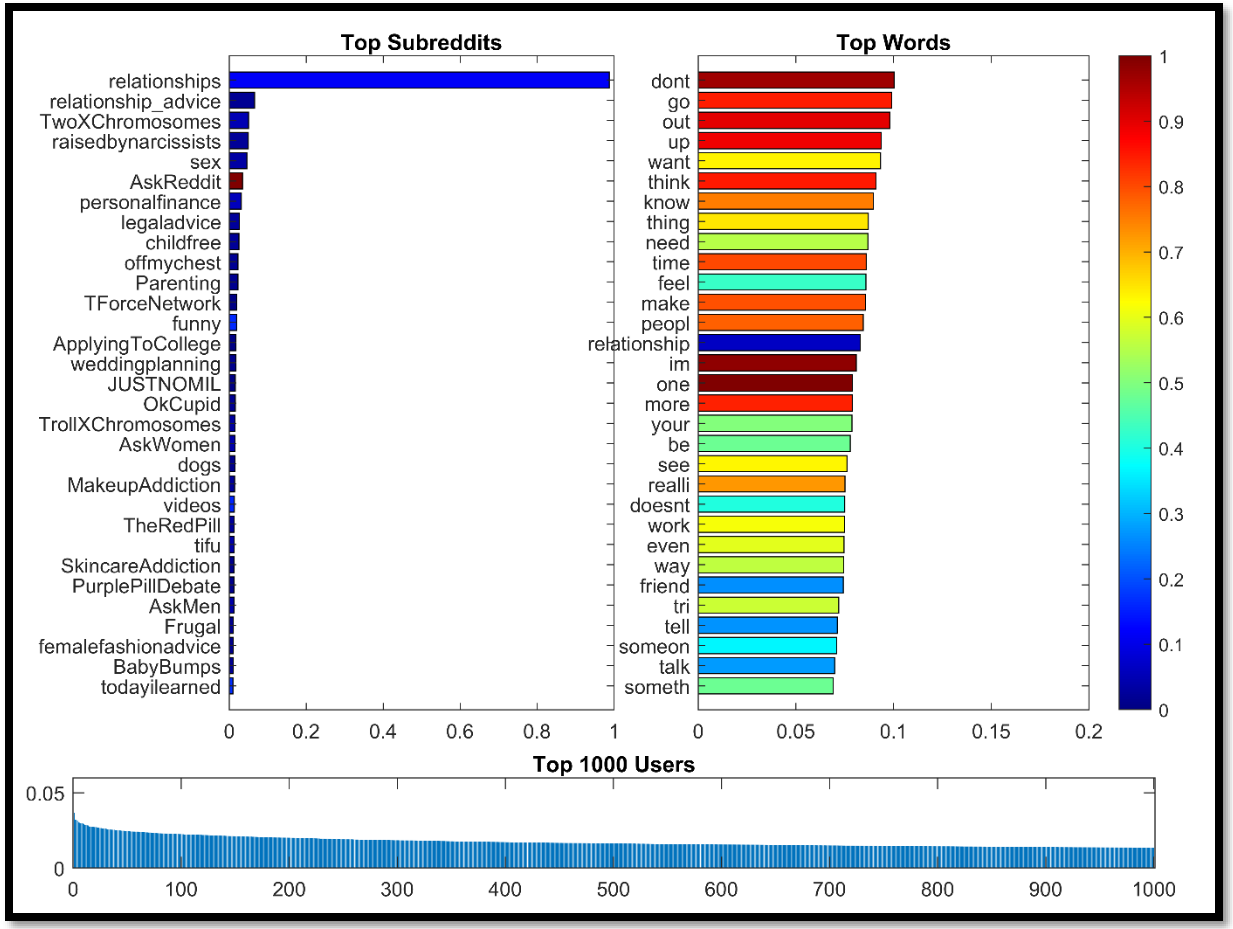


Example Reddit Components Include Rare Words Apropos to High-Scoring Reddits

Component #6: International News

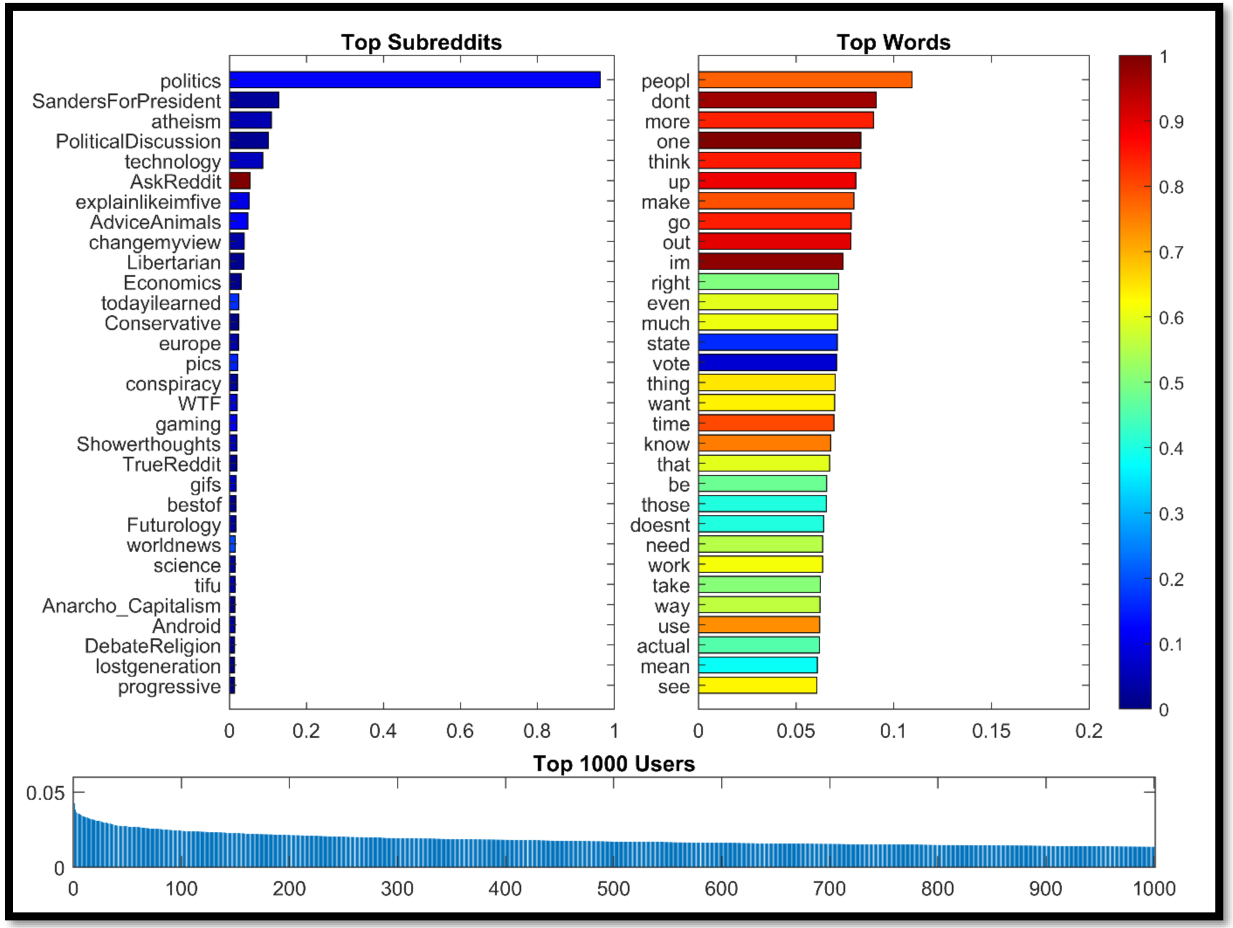


Component #8: Relationships

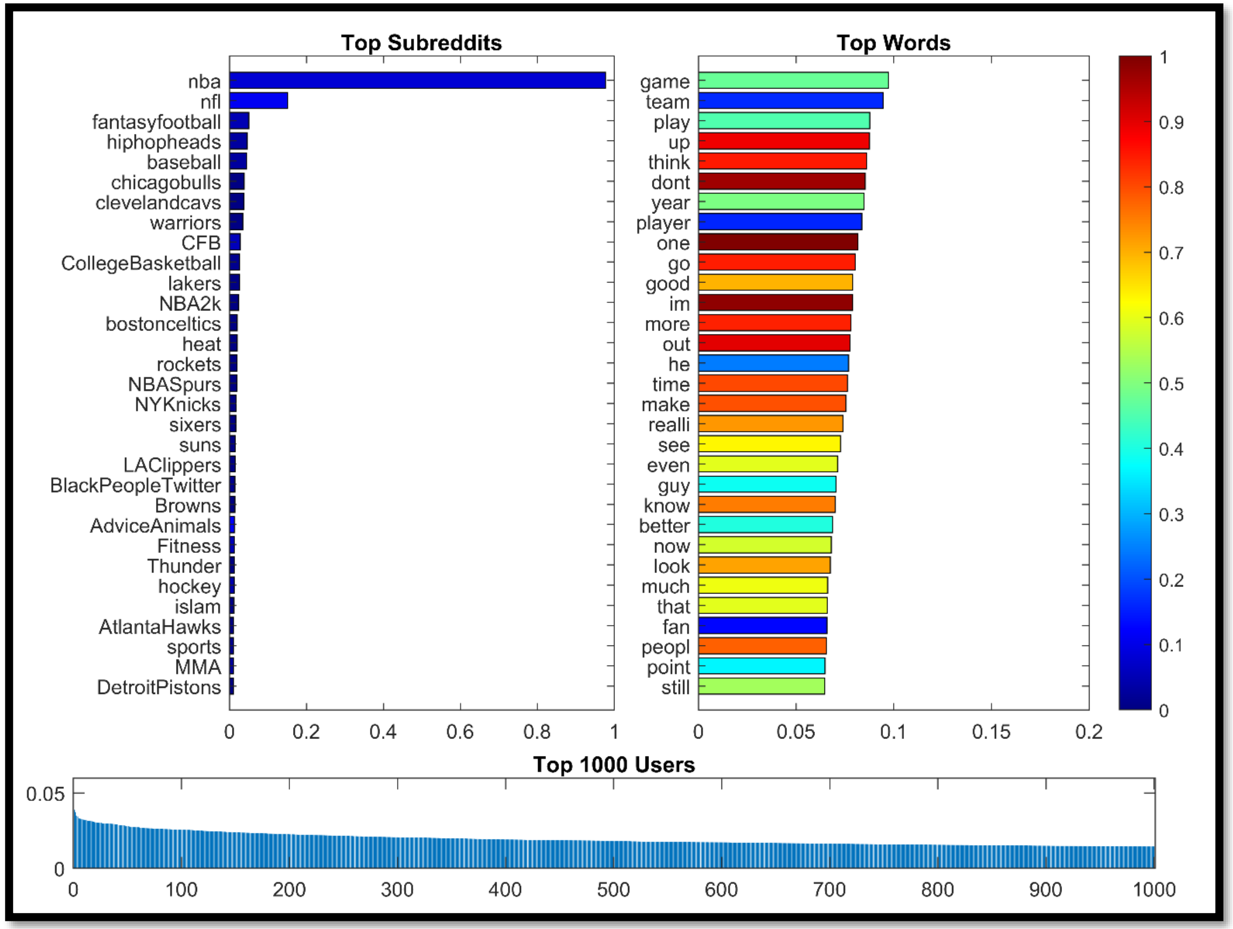


Example Reddit Components Include Rare Words Apropos to High-Scoring Reddits

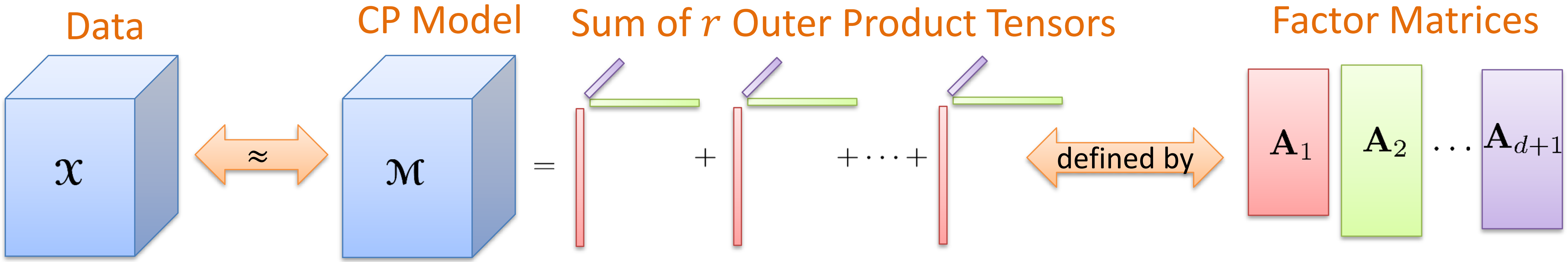
Component #9: U.S. Politics (2015)



Component #11: Sports



Tensor Decomposition Identifies Factors



$$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d+1}} \qquad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{d+1}]\!] \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d+1}} \qquad \mathbf{A}_k \in \mathbb{R}^{n_k \times r}$$

$$x_i = x(i_1, i_2, \ldots, i_{d+1}) \qquad m_i = m(i_1, i_2, \ldots, i_{d+1}) = \sum_{j=1}^{r} \prod_{k=1}^{d+1} a_k(i_k, j)$$

Model Rank

<u>Key Idea</u>: Alternate among the d factor matrices, fixing all but that one and solving. Each subproblem is linear least squares.

Prototypical CP Least Squares Problem has Khatri-Rao Product (KRP) Structure

$$\min_{\mathbf{B}} \|\mathbf{Z}\mathbf{B}^{\intercal} - \mathbf{X}^{\intercal}\|^2$$

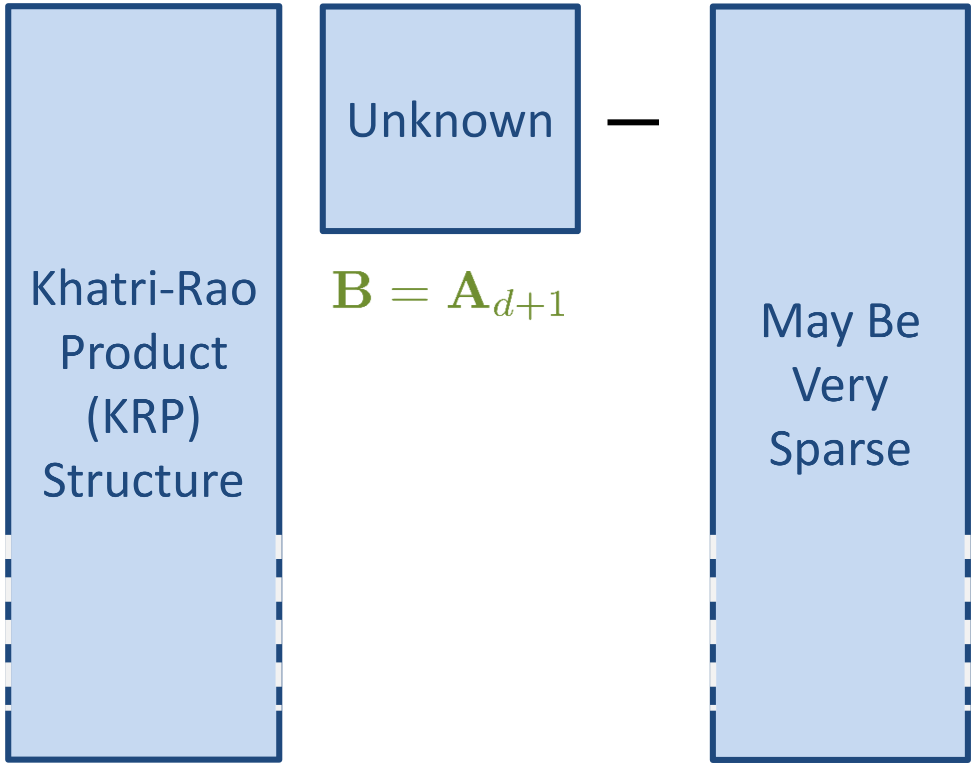
$N \gg r, n$

Linking back to mode-(d+1) least squares subproblem

$$N = \prod_{k=1}^{d} n_k$$

$$n = n_{d+1}$$

$\mathbf{Z} \in \mathbb{R}^{N \times r}$ $\mathbf{B}^{\intercal} \in \mathbb{R}^{r \times n}$ $\mathbf{X}^{\intercal} \in \mathbb{R}^{N \times n}$



$\mathbf{B} = \mathbf{A}_{d+1}$

$\mathbf{Z} = \mathbf{A}_d \odot \cdots \odot \mathbf{A}_1$

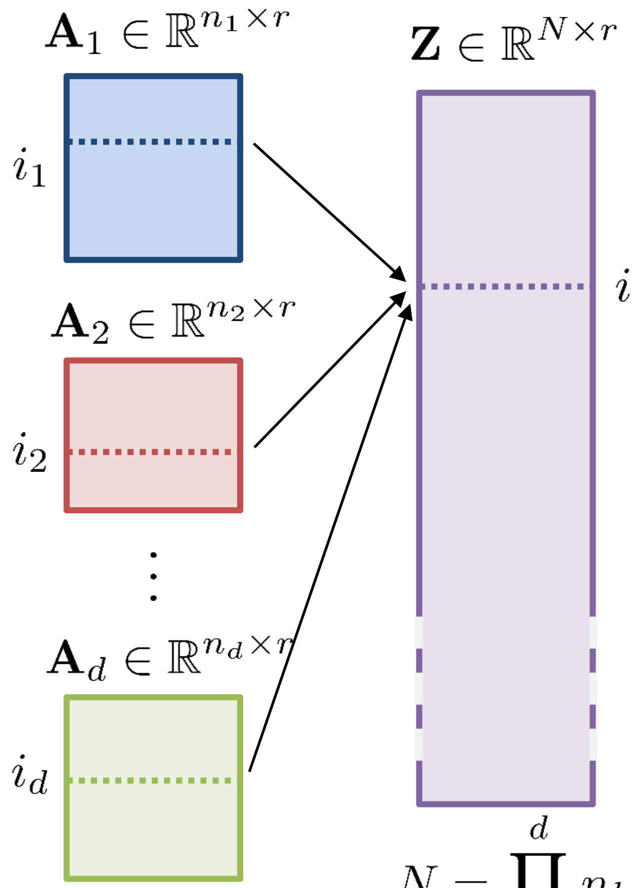
$\mathbf{X} = \mathbf{X}_{(d+1)}$

- KRP costs $O(Nr)$ to form
- System costs $O(Nnr^2)$ to solve
- KRP structure
  - Cost reduced to $O(Nnr)$
- KRP structure + data sparse
  - Cost reduced to $O(r\,\mathrm{nnz}(\mathbf{X}))$

Structure of Khatri-Rao Product (KRP): Hadamard Combinations of Rows of Inputs

KRP of d Matrices: $\mathbf{Z} = \mathbf{A}_d \odot \cdots \odot \mathbf{A}_1$

Number of columns is the same in all input matrices, but number of rows varies



$$N = \prod_{k=1}^{d} n_k$$

Each row of KRP is Hadamard product of specific rows in Factor Matrices:

$$\mathbf{Z}(i,:) = \mathbf{A}_1(i_1,:) * \cdots * \mathbf{A}_d(i_d,:)$$

where

$$\begin{aligned} i = \; & (n_{d-1} \cdots n_1)(i_d - 1) + (n_{d-2} \cdots n_1)(i_{d-1} - 1) \\ & + \cdots + n_1(i_2 - 1) + i_1 \quad \in [N] \end{aligned}$$

1-1 Correspondence between *linear index and multi index:*

$$i \in [N] \Leftrightarrow (i_1, \ldots, i_d) \in [n_1] \otimes \cdots \otimes [n_d]$$

Ingredient #1: Sample Subset of Rows in Overdetermined Least Squares System

$$\min_{\mathbf{B}} \|\mathbf{Z}\mathbf{B}^{\intercal} - \mathbf{X}^{\intercal}\|^2$$

$\mathbf{Z} \in \mathbb{R}^{N \times r}$ $\mathbf{B}^{\intercal} \in \mathbb{R}^{r \times n}$ $\mathbf{X}^{\intercal} \in \mathbb{R}^{N \times n}$

Khatri-Rao Product (KRP) Structure

Unknown

−

May Be Very Sparse

$$N \gg r, n$$

$$\min_{\mathbf{B}} \|\boldsymbol{\Omega}\mathbf{Z}\mathbf{B}^{\intercal} - \boldsymbol{\Omega}\mathbf{X}^{\intercal}\|^2$$

$\boldsymbol{\Omega}\mathbf{Z} \in \mathbb{R}^{s \times r}$ $\mathbf{B}^{\intercal} \in \mathbb{R}^{r \times n}$ $\boldsymbol{\Omega}\mathbf{X}^{\intercal} \in \mathbb{R}^{s \times n}$

Sampled KRP

Unknown

−

Sampled Data

Complexity reduced from $O(Nnr)$ to $O(snr^2)$

Key surveys:
M. W. Mahoney, ***Randomized Algorithms for Matrices and Data***, 2011;
D. P. Woodruff, ***Sketching as a Tool for Numerical Linear Algebra***, 2014

How sample so that solution of sampled problem yields something close to the optimal residual of the original problem?

Ingredient #2: Weight Sampled Rows by Probability of Selection to Eliminate Bias

Probability distribution on rows of linear system

$$\sum_{i=1}^{N} p_i = 1$$

Not specifying yet how these probabilities are selected

Pick a single random index ξ with probability p_ξ

Choose

$$\mathbf{\Omega} = \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{\sqrt{p_\xi}} & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{N \times 1}$$

ξth entry

Then (assuming all p_i positive) the sampled the sampled residual equals true residual in expectation:

$$\mathbb{E}\|\mathbf{\Omega}\mathbf{Z}\mathbf{B}^\intercal - \mathbf{\Omega}\mathbf{X}^\intercal\|^2 = \sum_{i=1}^{N} p_i \left(\left\| \frac{1}{\sqrt{p_i}} \mathbf{Z}(i,:)\mathbf{B}^\intercal - \frac{1}{\sqrt{p_i}} \mathbf{X}^\intercal(i,:) \right\|^2 \right)$$
$$= \|\mathbf{Z}\mathbf{B}^\intercal - \mathbf{X}^\intercal\|^2$$

Pick a s random indices ξ_j (with replacement) such that $P(\xi_j = i) = p_i$.

Choose $\mathbf{\Omega} \in \mathbb{R}^{s \times N}$ such that

Not specifying yet how s is determined

$$\omega(j,i) = \begin{cases} \frac{1}{\sqrt{s p_i}} & \text{if } \xi_j = i \\ 0 & \text{otherwise} \end{cases}$$

Each row has a single nonzero!

Then, as before, we have:

$$\mathbb{E}\|\mathbf{\Omega}\mathbf{Z}\mathbf{B}^\intercal - \mathbf{\Omega}\mathbf{X}^\intercal\|^2 = \|\mathbf{Z}\mathbf{B}^\intercal - \mathbf{X}^\intercal\|^2$$

Survey: D. P. Woodruff, ***Sketching as a Tool for Numerical Linear Algebra***, 2014

Ingredient #3: Use Factor Matrix Leverage Scores for Sampling Probabilities (Main Thm)

Sandia National Laboratories

Given linear system: $\|\mathbf{Z}\mathbf{B}^\intercal - \mathbf{X}^\intercal\|^2$ with $\mathbf{Z} = \mathbf{A}_d \odot \cdots \odot \mathbf{A}_1 \in \mathbb{R}^{N \times r}, \mathbf{X}^\intercal \in \mathbb{R}^{n \times N}$

Define sampling probabilities: $p_i = \frac{1}{r^d} \prod_{k=1}^{d} \ell_{i_k}(\mathbf{A}_k)$ for all $i \in [N]$

Leverage Scores

$\ell_{i_k}(\mathbf{A}_k) = \|\mathbf{Q}_k(i_k, :)\|_2$ where $\mathbf{Q}_k$ is orthonormal basis for column space of $\mathbf{A}_k$

And random sampling matrix: Pick a s random indices ξ_j such that $P(\xi_j = i) = p_i$ and define

$$\mathbf{\Omega} \in \mathbb{R}^{s \times N} \quad \text{with} \quad \omega(j,i) = \begin{cases} \frac{1}{\sqrt{s p_i}} & \text{if } \xi_j = i \\ 0 & \text{otherwise} \end{cases}$$

Solve sampled problem: $$\tilde{\mathbf{B}}_* \equiv \arg \min_{\mathbf{B} \in \mathbb{R}^{r \times n}} \|\mathbf{\Omega}\mathbf{Z}\mathbf{B}^\intercal - \mathbf{\Omega}\mathbf{X}\|_F^2$$

Get probabilistic error bound: With probability $1 - \delta$ for $\delta \in (0,1)$, we have

$$\|\mathbf{Z}\tilde{\mathbf{B}}_*^\intercal - \mathbf{X}^\intercal\|_F^2 \leq (1 + O(\epsilon))\|\mathbf{Z}\mathbf{B}_*^\intercal - \mathbf{X}^\intercal\|_F^2$$

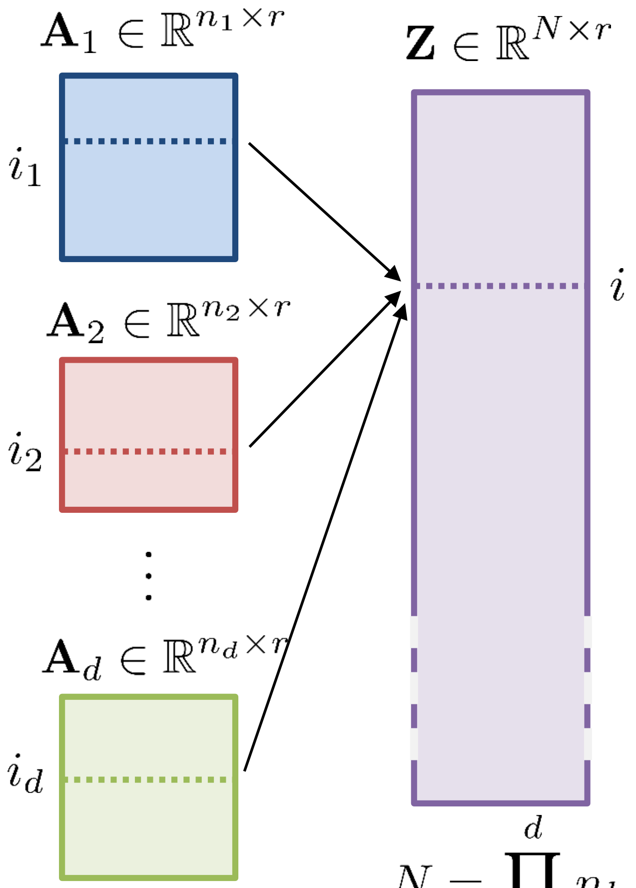
when number of samples satisfies: $s = O(r^d \log(n/\delta)/\epsilon^2)$

1-1 Correspondence between *linear index and multi index:*

$i \in [N] \Leftrightarrow (i_1, \ldots, i_d) \in [n_1] \otimes \cdots \otimes [n_d]$

Ingredient #4: Bound Leverage Scores
Ingredient #5: Efficient Sampling

KRP: $\mathbf{Z} = \mathbf{A}_d \odot \cdots \odot \mathbf{A}_1$



$$N = \prod_{k=1}^{d} n_k$$

Upper Bound on Leverage Score

Lemma (Cheng et al., NIPS 2016; Battaglino et al., SIMAX 2018):

$$\ell_i(\mathbf{Z}) \leq \prod_{k=1}^{d} \ell_{i_k}(\mathbf{A}_k)$$

Too expensive to calculate $O(Nr^2)$

Cheap to calculate individual leverage scores $O(r^2 \sum_k n_k)$

Recall probability of sampling row i

$$p_i \equiv \frac{1}{r^d} \prod_{k=1}^{d} \ell_{i_k}(\mathbf{A}_k)$$

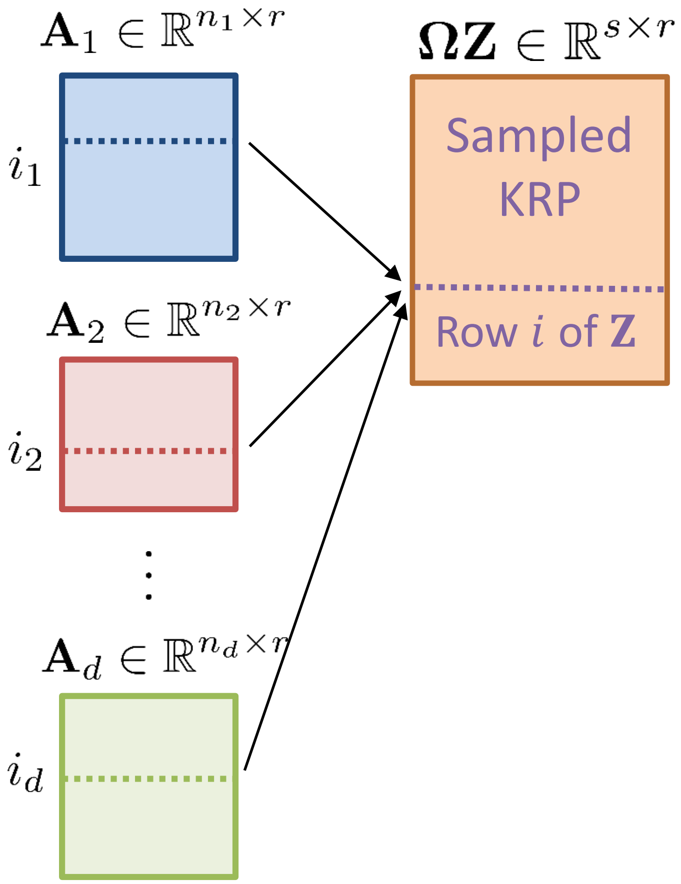
But still don't want to consider all N possible combinations corresponding to all rows of $\mathbf{Z}$!

1-1 Correspondence between *linear index* and *multi index:*

$$i \in [N] \Leftrightarrow (i_1, \ldots, i_d) \in [n_1] \otimes \cdots \otimes [n_d]$$

Ingredient #4: Bound Leverage Scores
Ingredient #5: Efficient Sampling

KRP: $\mathbf{Z} = \mathbf{A}_d \odot \cdots \odot \mathbf{A}_1$



Upper Bound on Leverage Score

Lemma (Cheng et al., NIPS 2016; Battaglino et al., SIMAX 2018):

$$\ell_i(\mathbf{Z}) \leq \prod_{k=1}^{d} \ell_{i_k}(\mathbf{A}_k)$$

Too expensive to calculate $O(Nr^2)$

Cheap to calculate individual leverage scores $O(r^2 \sum_k n_k)$

Recall probability of sampling row i

$$p_i \equiv \frac{1}{r^d} \prod_{k=1}^{d} \ell_{i_k}(\mathbf{A}_k)$$

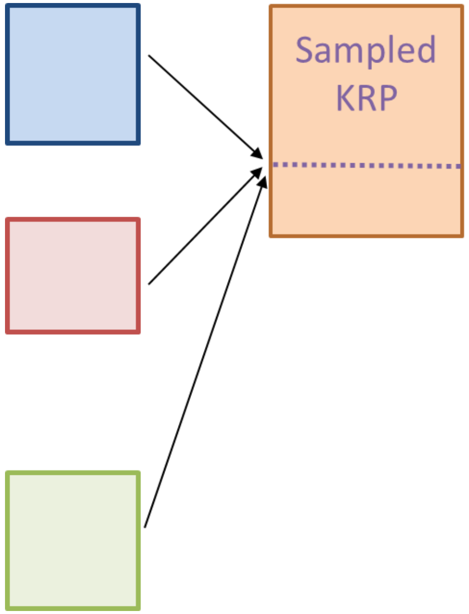
But still don't want to consider all N possible combinations corresponding to all rows of $\mathbf{Z}$!

1-1 Correspondence between *linear index* and *multi index:*

$$i \in [N] \Leftrightarrow (i_1, \dots, i_d) \in [n_1] \otimes \cdots \otimes [n_d]$$

Ingredient #6: Combine Repeated Rows

Problem: Concentrated sampling probabilities identify a few key rows *but* can lead to *many* repeats!

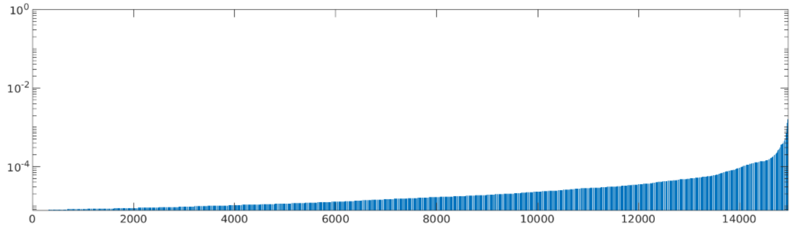


Least Squares Problems from Real-world Tensor Data Sets

Combining repeat rows ⇒ 2-20X speedup

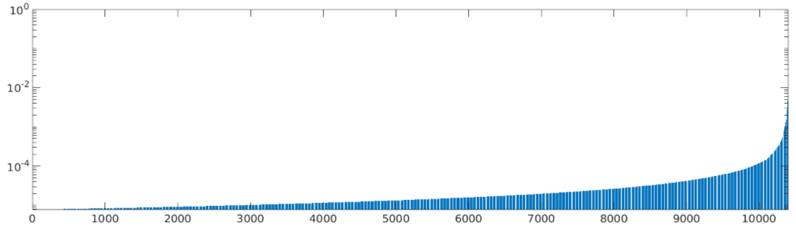
Example 1: $N = 3.2\text{e}12, s = 2^{17}, \tau = \frac{1}{s} = 8\text{e}{-6}$

$\mathcal{D} = \{i : p_i > \tau\}, |\mathcal{D}| \approx 15000, \sum_{i \in \mathcal{D}} p_i = 0.51$



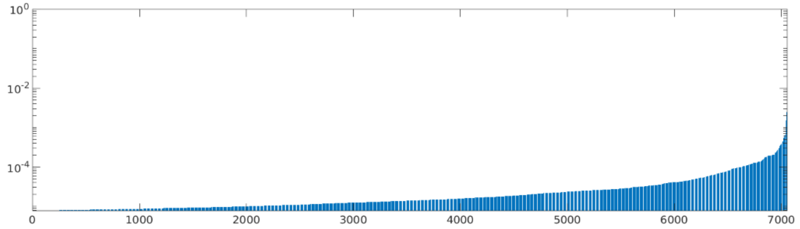
Example 2: $N = 8.7\text{e}12, s = 2^{17}, \tau = \frac{1}{s} = 8\text{e}{-6}$

$\mathcal{D} = \{i : p_i > \tau\}, |\mathcal{D}| \approx 10000, \sum_{i \in \mathcal{D}} p_i = 0.41$



Example 3: $N = 8.6\text{e}12, s = 2^{17}, \tau = \frac{1}{s} = 8\text{e}{-6}$

$\mathcal{D} = \{i : p_i > \tau\}, |\mathcal{D}| \approx 7000, \sum_{i \in \mathcal{D}} p_i = 0.25$



Ingredient #7: Hybrid Deterministic and Randomly-Sampled Rows

Deterministic Rows

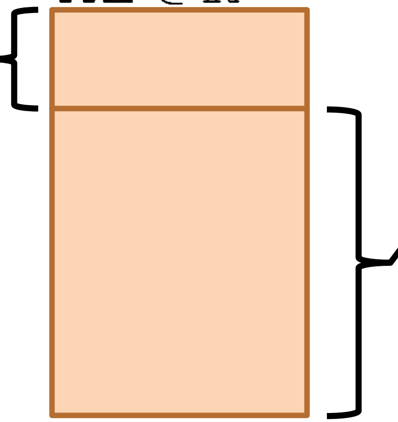
$$\mathcal{D}_\tau = \{\, i \in [N] \mid p_i \geq \tau \,\}$$

$$s_{\text{det}} = |\mathcal{D}_\tau|$$

$$p_{\text{det}} = \sum_{i \in \mathcal{D}_\tau} p_i$$

for $i \in \mathcal{D}_\tau$ **do**
  add row $\mathbf{A}_1(i_1,:) * \cdots * \mathbf{A}_d(i_d,:)$
end for

$\boldsymbol{\Omega}\mathbf{Z} \in \mathbb{R}^{s \times r}$



$$p_i \equiv \frac{1}{r^d} \prod_{k=1}^{d} \ell_{i_k}(\mathbf{A}_k)$$

Random Rows

$$s_{\text{rnd}} = s - s_{\text{det}}$$

for $j = 1 \ldots, s_{\text{rnd}}$ **do**
  repeat
    for $k = 1 \ldots, d$ **do**
      $i_k \leftarrow \texttt{multi}(\ell(\mathbf{A}_k)/r)$
    end for
  until $i \notin \mathcal{D}_\tau$
  $\omega \leftarrow \sqrt{(1 - p_{\text{det}})/(s_{\text{rnd}}\, p_i)}$
  add row $\omega\,(\mathbf{A}_1(i_1,:) * \cdots * \mathbf{A}_d(i_d,:))$
end for

1-1 Correspondence between *linear index and multi index:*

$$i \in [N] \Leftrightarrow (i_1, \ldots, i_d) \in [n_1] \otimes \cdots \otimes [n_d]$$

Ingredient #8: Find All High-Probability Rows without Computing All Probabilities

- Recall

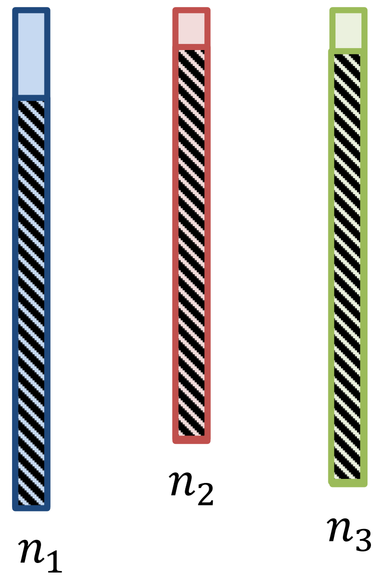
$$p_i \equiv \frac{1}{r^d} \prod_{k=1}^{d} \ell_{i_k}(\mathbf{A}_k)$$

- For given tolerance $\tau > 1/N$, define the set of deterministic rows to include

$$\mathcal{D}_\tau = \{\, i \in [N] \mid p_i \geq \tau \,\}$$

  - Compute *without* computing all p_i values
  - A few high leverage scores means all the others are necessarily low!
  - Use bounding procedure to eliminate most options
  - Compute products of at most a top few leverage scores in each mode

Sorted Leverages Scores (Descending)



1-1 Correspondence between *linear index and multi index:*

$$i \in [N] \Leftrightarrow (i_1, \ldots, i_d) \in [n_1] \otimes \cdots \otimes [n_d]$$

Ingredient #9: Efficiently Extract RHS from (Sparse) Unfolded Data Tensor

$$\min_{\mathbf{B}} \|\mathbf{Z}\mathbf{B}^\intercal - \mathbf{X}^\intercal\|^2$$

$\mathbf{Z} \in \mathbb{R}^{N \times r}$ — Khatri-Rao Product (KRP) Structure

$\mathbf{B}^\intercal \in \mathbb{R}^{r \times n}$ — Unknown

$\mathbf{X}^\intercal \in \mathbb{R}^{N \times n}$ — May Be Very Sparse

$$\min_{\mathbf{B}} \|\boldsymbol{\Omega}\mathbf{Z}\mathbf{B}^\intercal - \boldsymbol{\Omega}\mathbf{X}^\intercal\|^2$$

$\boldsymbol{\Omega}\mathbf{Z} \in \mathbb{R}^{s \times r}$ — Sampled KRP

$\mathbf{B}^\intercal \in \mathbb{R}^{r \times n}$ — Unknown

$\boldsymbol{\Omega}\mathbf{X}^\intercal \in \mathbb{R}^{s \times n}$ — Sampled Data

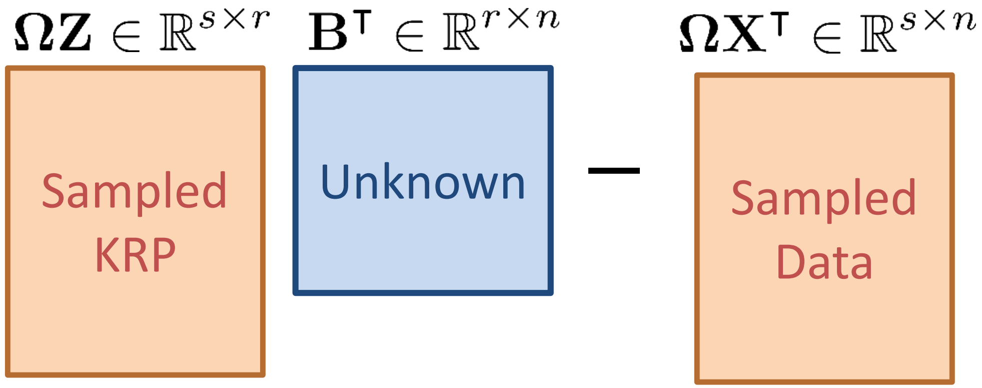
- Never form $\mathbf{X}^\mathrm{T}$ explicitly
- Precompute linear indices for every nonzero and every mode
- Results is sparse RHS

Similar in spirit to ideas for dense tensors in Battaglino et al., SIMAX 2018

Numerical Results

Solution Quality as Number of Samples Increase and Hybrid Improvements

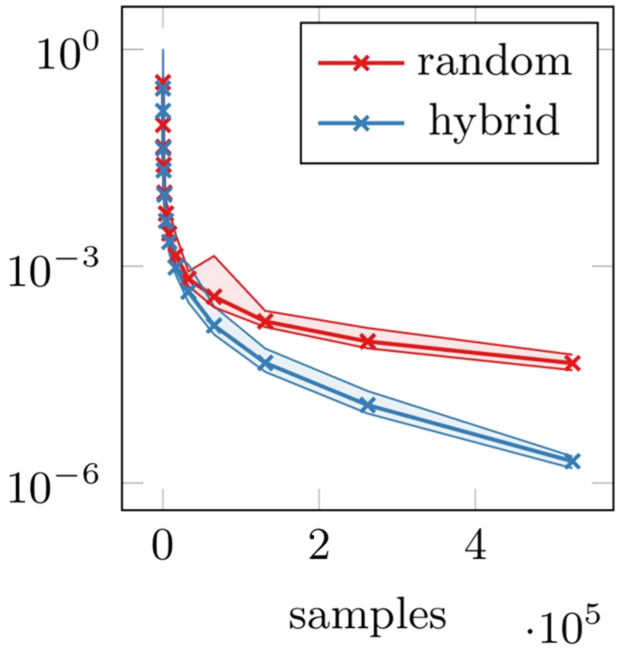
Single Least Squares Problem with N = 46M rows, r = 10 columns, n = 183 right-hand sides



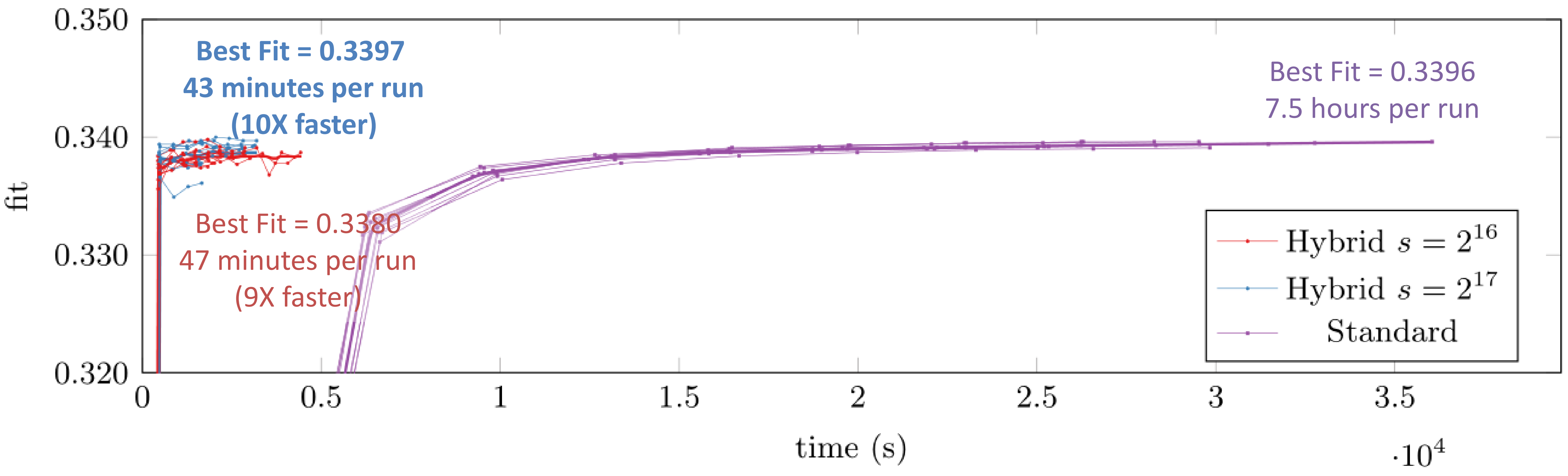
$$\tilde{\mathbf{B}}_* \equiv \arg\min_{\mathbf{B} \in \mathbb{R}^r} \|\Omega \mathbf{Z} \mathbf{B}^{\intercal} - \Omega \mathbf{X}^{\intercal}\|_2^2$$

$$\mathbf{B}_* \equiv \arg\min_{\mathbf{B} \in \mathbb{R}^r} \|\mathbf{Z} \mathbf{B}^{\intercal} - \mathbf{X}^{\intercal}\|_2^2$$

$$\frac{\left| \|\mathbf{Z}\mathbf{B}_*^{\intercal} - \mathbf{X}^{\intercal}\|_2 - \|\mathbf{Z}\tilde{\mathbf{B}}_*^{\intercal} - \mathbf{X}^{\intercal}\|_2 \right|}{\max\{1, \|\mathbf{Z}\mathbf{B}_*^{\intercal} - \mathbf{X}^{\intercal}\|_2\}}$$



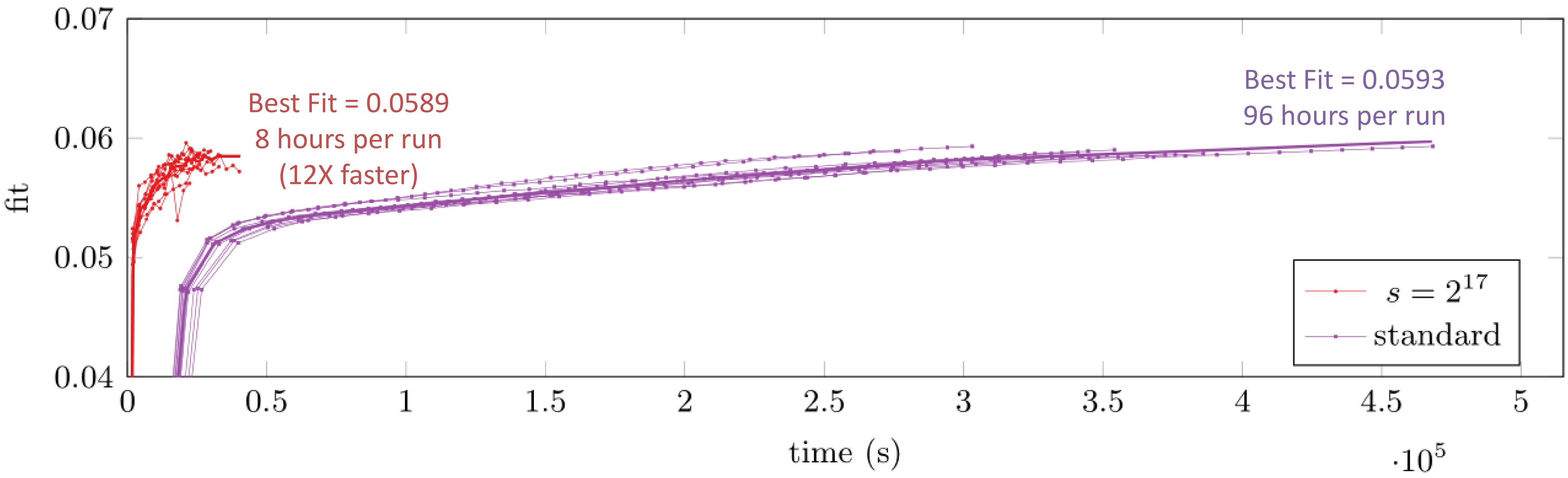
Over 9X Speed-up for Amazon Tensor with 1.7 Billion Nonzeros



Amazon Tensor: 4.8M x 1.8M x 1.8M Amazon Tensor with 1.7B nonzeros.
Rank r = 25 CP decomposition

Over 12X Speed-up for Reddit Tensor with 4.6 Billion Nonzeros (106 GB)

Sandia National Laboratories

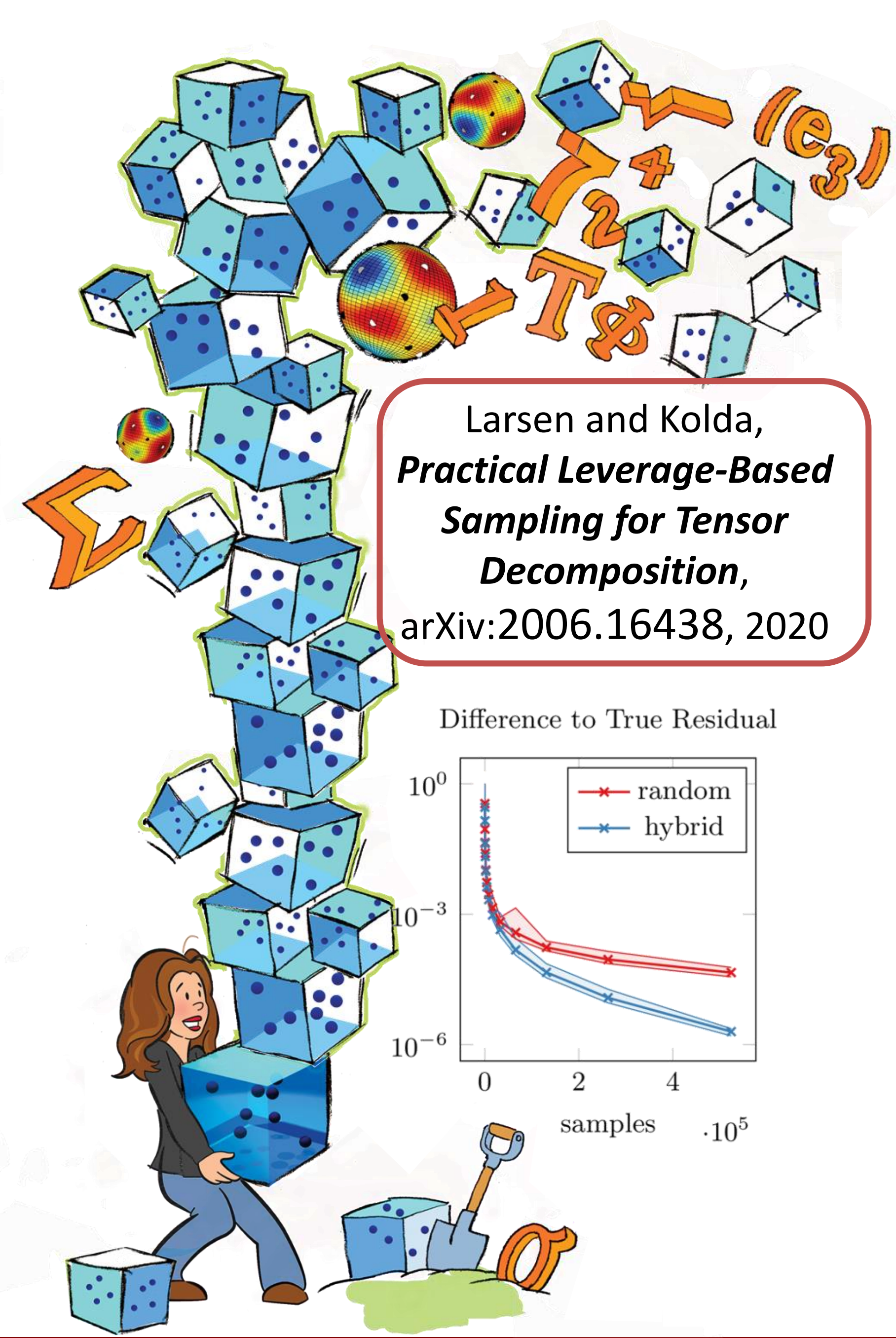


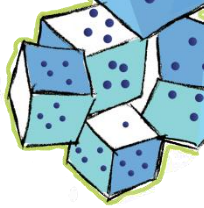
Amazon Tensor: 8.2M x 0.2M x 8.1M Reddit Tensor with 4.7B nonzeros.
Rank r = 25 CP decomposition

Hour-long version of talk on July 23rd
One World MINDS seminar

Conclusions & Future Work

Larsen and Kolda, ***Practical Leverage-Based Sampling for Tensor Decomposition***, arXiv:2006.16438, 2020





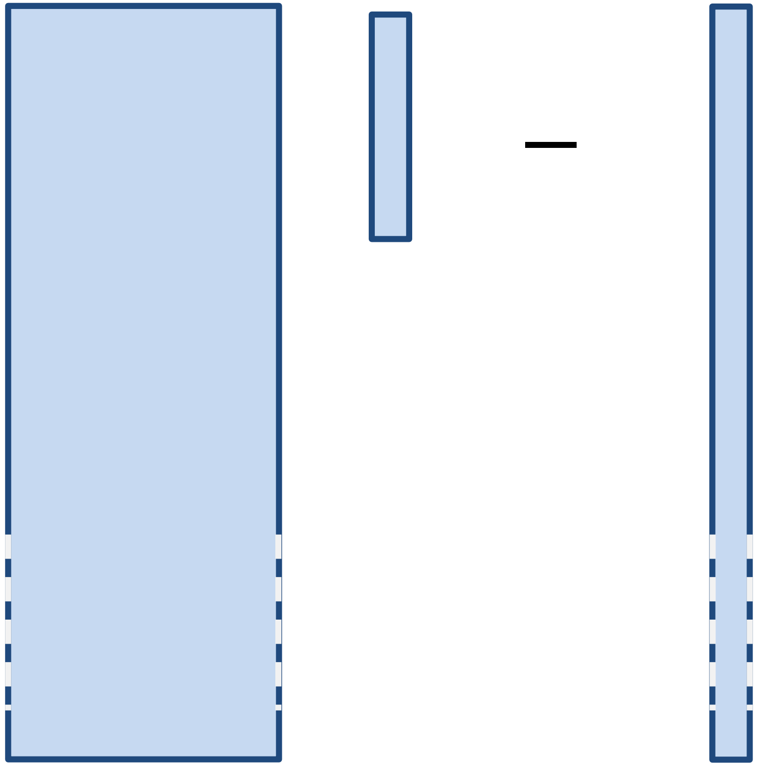
- How to make CP tensor decomposition faster for large-scale sparse tensors? Matrix sketching
- How to avoid repeated samples? Combine repeat rows or deterministically include high-probability rows
- How to efficiently sample? Sample independently from each factor matrix to build KRP
- How to extract data for RHS from data tensor? Pre-compute linear indices for tensor fibers
- Overall result: Order-of-magnitude speed-ups
- Many open problems: How to pick # samples (per mode even), deterministic threshold, robust stopping conditions, sampling based on data as well as KRP, parallelization of method, etc.

Contact Info: Brett bwlarsen@stanford.edu, Tammy tgkolda@sandia.gov

Backup: Uniform Sampling Okay for “Mixed” Dense Tensors (Inapplicable to Sparse)

Transform System: $\min_{\boldsymbol{\alpha} \in \mathbb{R}^r} \|\boldsymbol{\Phi}\mathbf{Z}\boldsymbol{\alpha} - \boldsymbol{\Phi}\boldsymbol{\nu}\|^2$

$\boldsymbol{\Phi}\mathbf{Z} \in \mathbb{R}^{N \times r}$ $\quad \boldsymbol{\alpha} \in \mathbb{R}^r$ $\quad \boldsymbol{\Phi}\boldsymbol{\nu} \in \mathbb{R}^N$



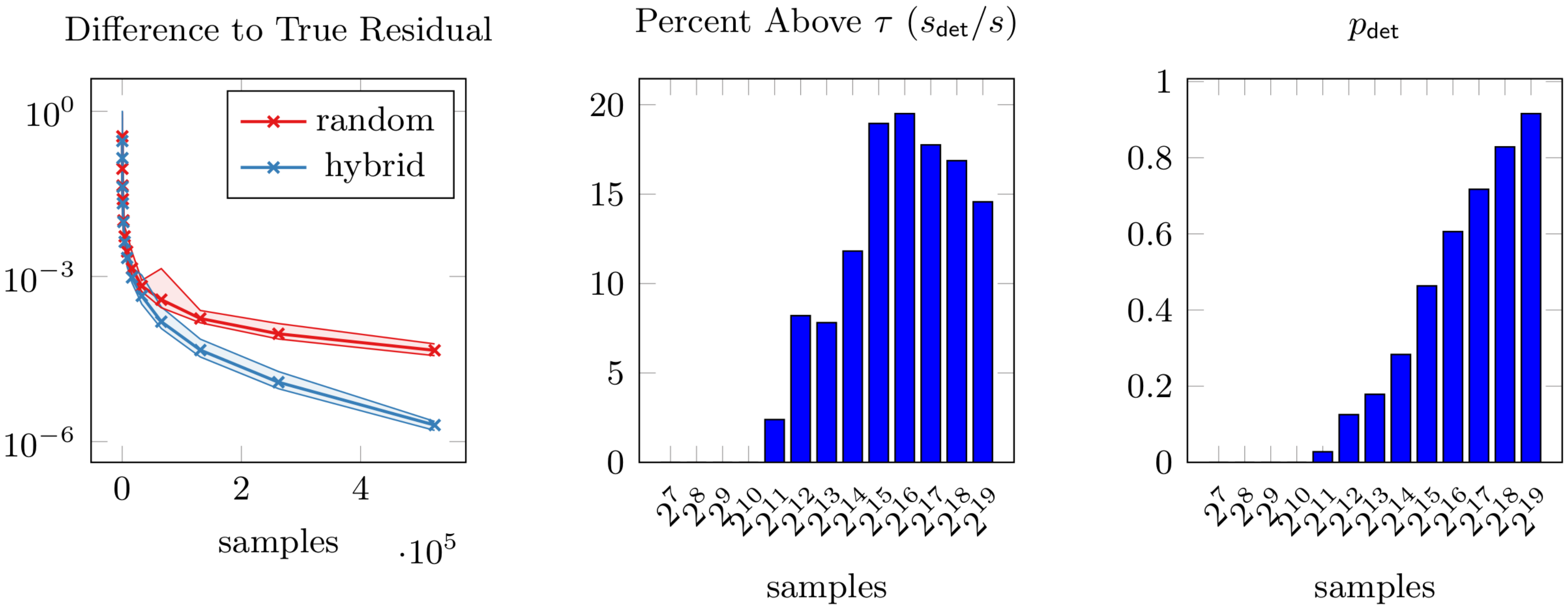
$N \gg r$

- Choose $\boldsymbol{\Phi}$ so that all leverage scores of $\boldsymbol{\Phi}\mathbf{Z}$ approximately equal, then uniform sampling yields $\beta \approx 1$
  - “Uniformize” the leverage scores per Mahoney
  - Fast Johnson-Lindenstrauss Transform (FJLT) uses random rows of matrix transformed by FFT and Rademacher diagonal
  - FJLT cost per iteration: $O(rN \log N)$
- Gaining Efficiency for KRP matrices
  - Transform individual factor matrices *before* forming $\mathbf{Z}$
  - Sample rows of $\mathbf{Z}$ implicitly
  - Kronecker Fast Johnson-Lindenstrauss Transform (KFJLT)
  - Special handling of right-hand side with preprocessing costs
  - KFJLT cost per iteration: $O(r \sum_k n_k \log n_k + sr^2)$
- References

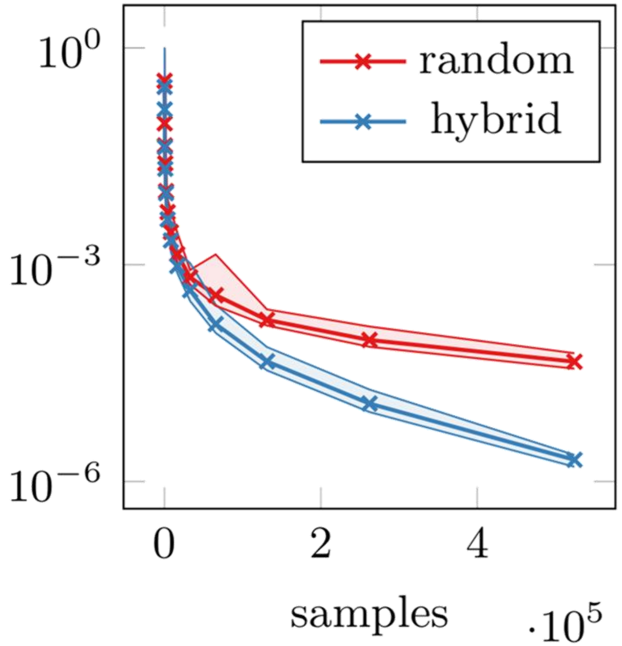
  - C. Battaglino, G. Ballard, T. G. Kolda. **A Practical Randomized CP Tensor Decomposition**. *SIAM Journal on Matrix Analysis and Applications*, Vol. 39, No. 2, pp. 876-901, 26 pages, 2018. https://doi.org/10.1137/17M1112303
  - R. Jin, T. G. Kolda, R. Ward. **Faster Johnson-Lindenstrauss Transforms via Kronecker Products,** 2019. http://arxiv.org/abs/1909.04801

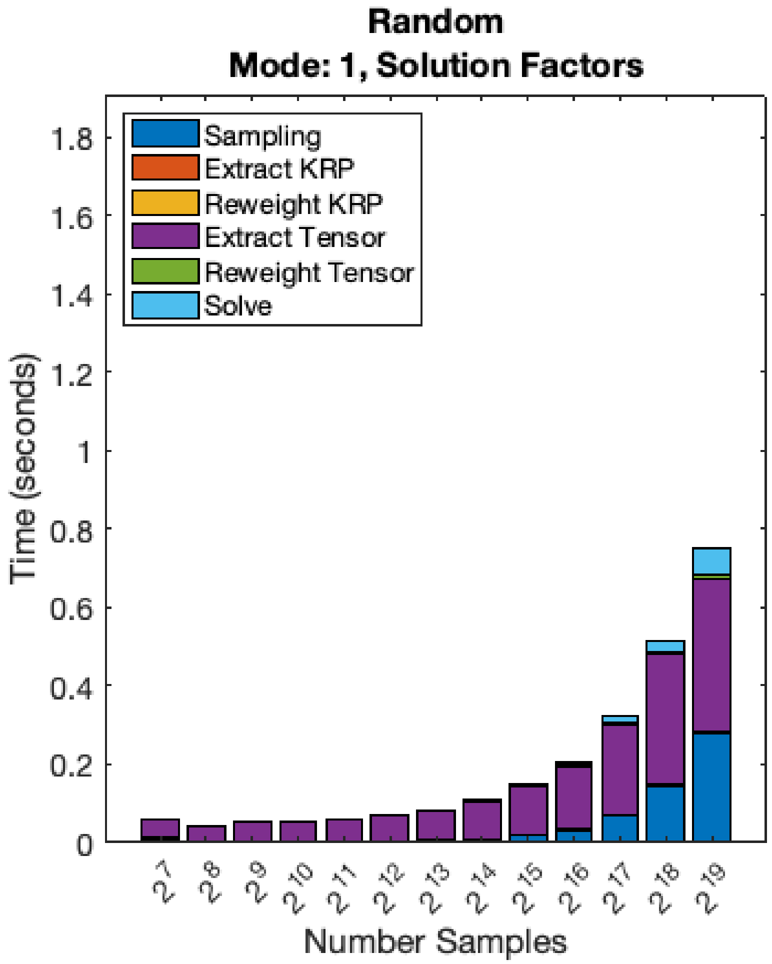
Deterministic Can Account for Substantial Portion of Probability

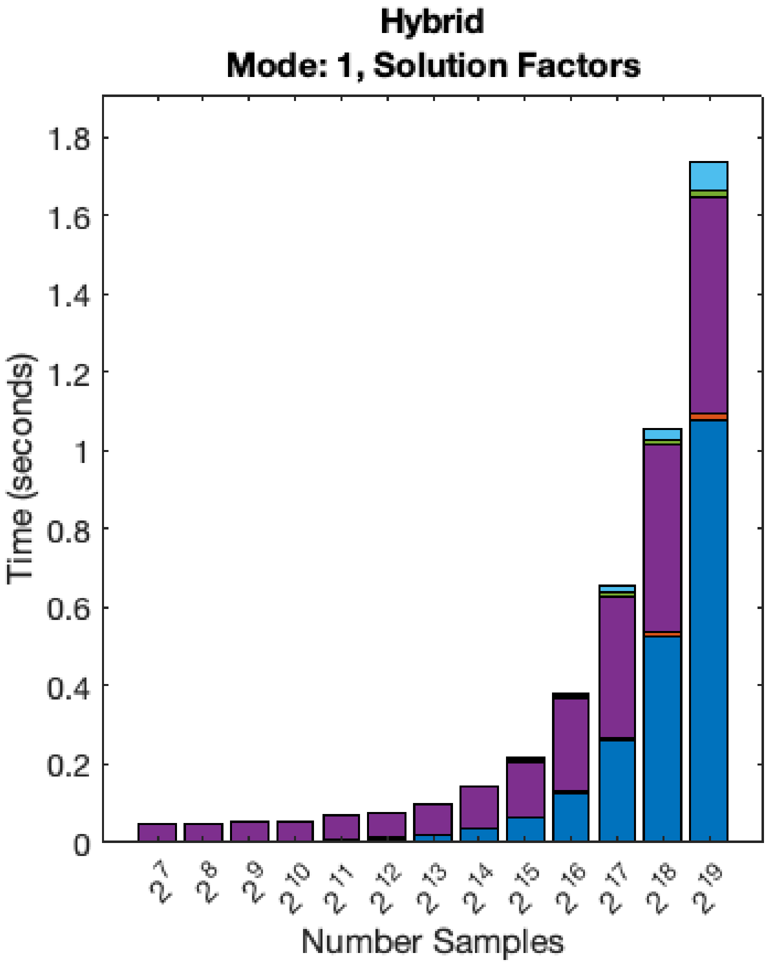
Single Least Squares Problem with N = 46M rows, r = 10 columns, n = 183 right-hand sides



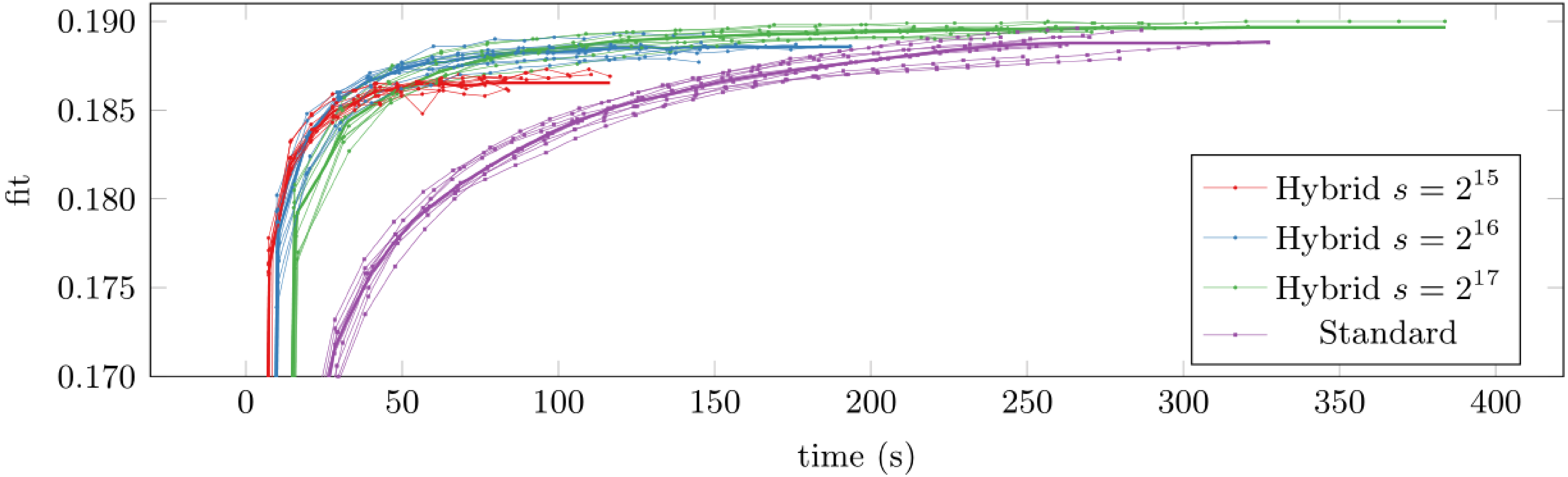
Some Trade-off Between Accuracy and Expense for Deterministic







CP-ARLS-LEV (Hybrid) Comparable to CP-ALS (Standard) on Small Uber Problem



Uber Tensor: 183 x 24 x 1140 x 1717 Uber Tensor with 3M nonzeros (0.038% dense).
Rank r = 25 CP decomposition