Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition

Tamara G. Kolda
Sandia National Labs, Livermore, CA
www.kolda.net

Joint work with
Brett Larsen
Stanford University

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
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
- **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

Tensor Decomposition Finds Patterns in Massive Data (Unsupervised Learning)
Tensor Decomposition Identifies Factors

\[ \mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d+1}} \]

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

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

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

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

Model Rank
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

### Formula

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

**Used a rank** $r = 25$ **decomposition**

Interpreting Reddit Components

Reddit Tensor

Compute rank $r=25$

Model

Component 6

Largest of 200K entries, in absolute value

Largest entries, in absolute value

Largest of 8M entries, in absolute value

variance explained $\approx 6\%$

Relative Weights of 25 Components

Model variance explained $\approx 6\%$

Compute rank $r=25$

Component 6

Largest of 200K entries, in absolute value

Largest entries, in absolute value

Largest of 8M entries, in absolute value
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 #11: Sports
Example Reddit Components Include Rare Words Apropos to High-Scoring Reddits

Component #15: Wrestling

Component #18: Soccer
Example Reddit Components Include Rare Words Apropos to High-Scoring Reddits

Component #19: Movies & TV

Component #18: Computer Card Game
Tensor Decomposition Identifies Factors

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

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

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

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

Key Idea: 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

\[ N \gg r, n \]

\[ Z \in \mathbb{R}^{N \times r} \]
\[ B^T \in \mathbb{R}^{r \times n} \]
\[ X^T \in \mathbb{R}^{N \times n} \]

\[ \min_B \|ZB^T - X^T\|^2 \]

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

\[ N = \prod_{k=1}^{d} n_k \]
\[ n = n_{d+1} \]

Khatri-Rao Product (KRP) Structure

\[ Z = A_d \otimes \cdots \otimes A_1 \]

Unknown

May Be Very Sparse

\[ B = A_{d+1} \]

\[ X = X_{(d+1)} \]

- 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 \( O(r \text{ nnz}(X)) \)
For Ease of Discussion: Simplify KRP Least Squares to Single Right-Hand Side

\[ \min_B \|ZB^T - X^T\|^2 \]

\[ Z \in \mathbb{R}^{N \times r} \]
\[ B^T \in \mathbb{R}^{r \times n} \]
\[ X^T \in \mathbb{R}^{N \times n} \]

Kharti-Rao Product (KRP) Structure

Unknown

May Be Very Sparse

\[ N \gg r, n \]

\[ \min_{\alpha \in \mathbb{R}^r} \|Z\alpha - \nu\|^2 \]

\[ Z \in \mathbb{R}^{N \times r} \]
\[ \alpha \in \mathbb{R}^r \]
\[ \nu \in \mathbb{R}^N \]

Kharti-Rao Product (KRP) Structure

\[ N \gg r \]
Structure of Khatri-Rao Product (KRP): Hadamard Combinations of Rows of Inputs

KRP of \( d \) Matrices: \( Z = A_d \odot \cdots \odot A_1 \)

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

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

\[
Z(i,:) = A_1(i_1,:) * \cdots * A_d(i_d,:)
\]

where

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

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

\[
\min_{\alpha \in \mathbb{R}^r} \|Z\alpha - \nu\|^2
\]

\[
Z \in \mathbb{R}^{N \times r} \quad \alpha \in \mathbb{R}^r \quad \nu \in \mathbb{R}^N
\]

Khatri-Rao Product (KRP) Structure

\[
N \gg r
\]

\[
\min_{\alpha \in \mathbb{R}^r} \|\Omega Z\alpha - \Omega \nu\|^2
\]

\[
\Omega Z \in \mathbb{R}^{s \times r} \quad \alpha \in \mathbb{R}^r \quad \Omega \nu \in \mathbb{R}^s
\]

Complexity reduced from \(O(Nr)\) to \(O(sr^2)\)

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

Key surveys:
M. W. Mahoney, *Randomized Algorithms for Matrices and Data*, 2011;
### 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** \( \xi \) with probability \( p_\xi \)

Choose

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

\( \xi \)th entry

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

\[
\mathbb{E} \| \Omega Z \alpha - \Omega \nu \|^2 = \sum_{i=1}^{N} p_i \left( \| \frac{1}{\sqrt{p_i}} Z(i,:) \alpha - \frac{1}{\sqrt{p_i}} \nu_i \|^2 \right) \\
= \| Z \alpha - \nu \|^2
\]

Not specifying yet how \( s \) is determined

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

Choose \( \Omega \in \mathbb{R}^{s \times N} \) such that

\[
\omega(j,i) = \begin{cases} 
\frac{1}{\sqrt{sp_i}} & \text{if } \xi_j = i \\
0 & \text{otherwise}
\end{cases}
\]

Each row has a single nonzero!

Then, as before, we have:

\[
\mathbb{E} \| \Omega Z \alpha - \Omega \nu \|^2 = \| Z \alpha - \nu \|^2
\]

Theory Review: Connecting Probabilities, Leverage Scores, and Number of Samples

Given linear system: $\|Z\alpha - \nu\|^2$ with $Z \in \mathbb{R}^{N \times r}$, $\nu \in \mathbb{R}^N$

And random sampling matrix:

- Pick a $s$ random indices $\xi_j$ such that $P(\xi_j = i) = p_i$ and define
- $\Omega \in \mathbb{R}^{s \times N}$ with $\omega(j, i) = \begin{cases} \frac{1}{\sqrt{sp_i}} & \text{if } \xi_j = i \\ 0 & \text{otherwise} \end{cases}$

Solve sampled problem:

- $\tilde{\alpha}_* \equiv \arg\min_{\alpha \in \mathbb{R}^r} \|\Omega Z\alpha - \Omega \nu\|^2$

Get probabilistic error bound:

- For error $\epsilon \in (0,1)$, confidence $1 - \delta \in (0,1)$, we have $P\left(\|Z\tilde{\alpha}_* - \nu\|^2 \leq (1 + O(\epsilon))\|Z\alpha_* - \nu\|^2\right) > 1 - \delta$

when number of samples satisfies:

- $s = O(\epsilon^{-2}\ln\left(\frac{r}{\delta}\right)r\beta^{-1})$

where $\beta$-term:

- $\beta = \min_{i\in[N]} \frac{r}{\ell_i(Z)} p_i \in (0,1)$

We get to pick!

Want $\beta$ as big as possible!

Ideally, $p_i = \ell_i(Z)/r$ for all $i \in [N]$, but ...


Ingredient #3: Leverage Scores Key to Limiting Samples (but too Expensive to Compute)

\[
\begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_r \\
0 \\
\end{bmatrix}
= \begin{bmatrix}
\nu_1 \\
\nu_2 \\
\vdots \\
\nu_N \\
\end{bmatrix}
\]

\[\ell_1(Z) = 1\]

\[Z \in \mathbb{R}^{N \times r}\]

**Leverage score:**

Let \(Q\) be any orthonormal basis of the column space of \(Z\).

Leverage score of row \(i\):

\[\ell_i(Z) = \|Q(i,:)\|_2^2 \in [0, 1]\]

**Coherence:**

\[\mu(Z) = \max_{i \in [N]} \ell_i(Z)\]

\[r/N \leq \mu(Z) \leq 1\]

**Rough Intuition:**

Key rows have high leverage score

What if we do uniform sampling?

\[p_i = \frac{1}{N} \text{ for all } i \in [N],\]

\[\beta = \min_{i \in [N]} \frac{r p_i}{\ell_i(Z)} = \min_{i \in [N]} \frac{r/N}{\ell_i(Z)}\]

\[s = O(\epsilon^{-2} \ln(r) r \beta^{-1})\]

Case 1: \(\mu(Z) = r/N\) (incoherent)

\[\Rightarrow \beta = 1 \Rightarrow s = O(\epsilon^{-2} \ln(r) r)\]

Case 2: \(\mu(Z) = 1\) (coherent)

\[\Rightarrow \beta = r/N \Rightarrow s = O(\epsilon^{-2} \ln(r) N)\]

In Case 2, prefer \(p_i = \ell_i(Z)/r\), but costs \(O(Nr^2)\) to compute leverage scores!
Aside: Uniform Sampling Okay for “Mixed” Dense Tensors (Inapplicable to Sparse)

Choose $\Phi$ so that all leverage scores of $\Phi Z$ 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 $Z$
- Sample rows of $Z$ implicitly
- Kronecker Fast Johnson-Lindenstrauss Transform (KFJLT)
- Special handling of right-hand side with preprocessing costs
- KFJLT cost per iteration: $O(r \sum_k n_k \log n_k + s r^2)$

References

Ingredient #4: Exploit KRP Structure to Bound Leverage Scores

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):

$$\mathcal{L}_i(\mathbf{Z}) \leq \prod_{k=1}^{d} \mathcal{L}_i(\mathbf{A}_k)$$

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

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

Tensor Least Squares Sketching with Leverage Scores

Thm: Using this sampling probability yields $(1+\epsilon)$ accuracy w.h.p. with number of rows

$$s = O(r^d \log(r/\delta)/\epsilon^2)$$

Set probability of sampling row $i$ to:

$$p_i = \frac{1}{r^d} \prod_{k=1}^{d} \mathcal{L}_i(\mathbf{A}_k)$$

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 #5: Randomly Sample Rows of KRP Using Implicit Leverage Score Bounds

KRP: $Z = A_d \circ \cdots \circ A_1$

$A_1 \in \mathbb{R}^{n_1 \times r}$

$A_2 \in \mathbb{R}^{n_2 \times r}$

$\vdots$

$A_d \in \mathbb{R}^{n_d \times r}$

$\Omega Z \in \mathbb{R}^{8 \times r}$

Sampled KRP

Row $i$ of $Z$

Probability of sampling row $i$:

$p_i \equiv \frac{1}{p_d} \prod_{k=1}^{d} \ell_{i_k}(A_k)$

- Recall our goal: Pick $s$ random indices $\xi_j$ such that $P(\xi_j = i) = p_i$
- For $j$-th sample for $j = 1, \ldots, s$:
  - Sample one row from each factor matrix such that $\text{Prob}(\text{row } i_k) = \ell_{i_k}(A_k)/r$
  - Set $\xi_j = i$, where $P(\xi_j = i) = p_i$
  - Compute Hadamard products of corresponding rows of factor matrices
  - Weight by $1/\sqrt{s p_i}$
- Never computes...
  - Matrix $Z$ nor its leverage scores
  - Weight matrix $\Omega$
- Computing factor matrix leverage scores costs only $O(r^2 \sum_k n_k)$
  - Versus $O(Nr^2)$ for computing leverage scores from $Z$

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 #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

Example 1: $N = 3.2e12, s = 2^{17}, \tau = \frac{1}{s} = 8e-6$

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

Example 2: $N = 8.7e12, s = 2^{17}, \tau = \frac{1}{s} = 8e-6$

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

Example 3: $N = 8.6e12, s = 2^{17}, \tau = \frac{1}{s} = 8e-6$

$D = \{i : p_i > \tau\}, |D| \approx 7000, \sum_{i \in D} p_i = 0.25$
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 \( \mathbf{A}_1(i_1,:) \times \cdots \times \mathbf{A}_d(i_d,:) \)
end for

Random Rows

\[ s_{\text{rnd}} = s - s_{\text{det}} \]

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

1-1 Correspondence between linear index and multi index:

\[ i \in [N] \Leftrightarrow (i_1, \ldots, i_d) \in [m_1] \times \cdots \times [m_d] \]
Ingredient #9: Find All High-Probability Rows without Computing All Probabilities

• Recall

\[ p_i = \frac{1}{n_d} \prod_{k=1}^{d} \ell_{i_k}(A_k) \]

• For given tolerance \( \tau > 1/N \), define the set of deterministic rows to include

\[ D_\tau = \{ i \in [N] \mid p_i \geq \tau \} \]

- 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)

\[ n_1 \quad n_2 \quad n_3 \]

1-1 Correspondence between linear index and multi index:

\[ i \in [N] \Leftrightarrow (i_1, \ldots, i_d) \in [n_1] \otimes \cdots \otimes [n_d] \]
Remember the Original Problem – Need to Sample the Right-Hand Side as Well

$$\min_B \|ZB^T - X^T\|^2$$

\[ Z \in \mathbb{R}^{N \times r} \quad B^T \in \mathbb{R}^{r \times n} \quad X^T \in \mathbb{R}^{N \times n} \]

Khatri-Rao Product (KRP) Structure

Unknown

May Be Very Sparse

$$\min_B \|\Omega ZB^T - \Omega X^T\|^2$$

\[ \Omega Z \in \mathbb{R}^{s \times r} \quad B^T \in \mathbb{R}^{r \times n} \quad \Omega X^T \in \mathbb{R}^{s \times n} \]

Sampled KRP

Unknown

Sampled Data

Complexity reduced from $O(Nrn)$ to $O(sr^2n)$
Ingredient #9: Efficiently Extract RHS from (Sparse) Unfolded Data Tensor

- Never form $X^T$ 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

\[ \Omega Z \in \mathbb{R}^{s \times r}, \quad B^T \in \mathbb{R}^{r \times n}, \quad \Omega X^T \in \mathbb{R}^{s \times n} \]

\[ \tilde{B}_* \equiv \arg \min_{B \in \mathbb{R}^r} \| \Omega Z B^T - \Omega X^T \|_2^2 \]

\[ B_* \equiv \arg \min_{B \in \mathbb{R}^r} \| Z B^T - X^T \|_2^2 \]

\[ \max \{ 1, \| Z B_*^T - X^T \|_2 \} \]

\[ \| Z B_*^T - X^T \|_2 - \| \tilde{B}_*^T - X^T \|_2 \]
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
Uber Tensor: 183 x 24 x 1140 x 1717 Uber Tensor with 3M nonzeros (0.038% dense).
Rank r = 25 CP decomposition
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)

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

Best Fit = 0.0589
8 hours per run
(12X faster)

Best Fit = 0.0593
96 hours per run
Conclusions & Future Work

- 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 bw Larsen@stanford.edu, Tammy tgkolda@sandia.gov