# CSE 392: Matrix and Tensor Algorithms for Data

Instructor: Shashanka Ubaru
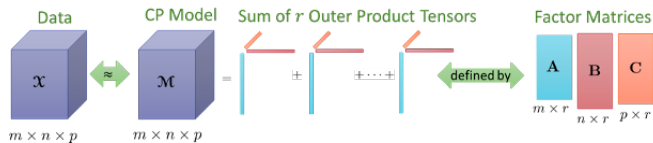
University of Texas, Austin
Spring 2024

# Lecture 17: Randomized CP - I

# Outline

# Alternating Least Squares (CP-ALS)



$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathcal{X} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|_F$$

**General Idea:** solve for ONE matrix, holding the others fixed.

- **CP-ALS:** Repeat until converged...
  - ▸ Solve for $\boldsymbol{A}$ (with $\boldsymbol{B}$ and $\boldsymbol{C}$ fixed)
  - ▸ Solve for $\boldsymbol{B}$ (with $\boldsymbol{A}$ and $\boldsymbol{C}$ fixed)
  - ▸ Solve for $\boldsymbol{C}$ (with $\boldsymbol{A}$ and $\boldsymbol{B}$ fixed)

## Special Structure of Least Squares Problem

$$\min_{\boldsymbol{A}} \|\boldsymbol{X}_{(1)} - \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^{\top}\|_F^2$$

$$\min_{\boldsymbol{A}} \|(\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{A}^{\top} - \boldsymbol{X}_{(1)}^{\top}\|_F^2$$

By normal equations:

$$(\boldsymbol{C} \odot \boldsymbol{B})^{\top}(\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{A}^{\top} = (\boldsymbol{C} \odot \boldsymbol{B})^{\top}\boldsymbol{X}_{(1)}^{\top}$$

$$(\boldsymbol{C}^{\top}\boldsymbol{C} * \boldsymbol{B}^{\top}\boldsymbol{B})\boldsymbol{A}^{\top} = (\boldsymbol{C} \odot \boldsymbol{B})^{\top}\boldsymbol{X}_{(1)}^{\top}$$

$$\boldsymbol{A}^{\top} = (\boldsymbol{C}^{\top}\boldsymbol{C} * \boldsymbol{B}^{\top}\boldsymbol{B})^{-1}(\boldsymbol{C} \odot \boldsymbol{B})^{\top}\boldsymbol{X}_{(1)}^{\top}$$

$$\boldsymbol{A} = \boldsymbol{X}_{(1)}(\boldsymbol{C} \odot \boldsymbol{B})(\boldsymbol{C}^{\top}\boldsymbol{C} * \boldsymbol{B}^{\top}\boldsymbol{B})^{-1}$$

# Special Structure of Least Squares Problem ($d$-way)

$$\min_{\boldsymbol{A}_k} \|\boldsymbol{X}_{(k)} - \boldsymbol{A}_k \underbrace{(\boldsymbol{A}_d \odot \cdots \odot \boldsymbol{A}_{k+1} \odot \boldsymbol{A}_{k-1} \odot \cdots \odot \boldsymbol{A}_1)}_{\boldsymbol{Z}_k}^{\top}\|_F^2$$

$$\min_{\boldsymbol{A}_k} \|\boldsymbol{Z}_k \boldsymbol{A}_k^{\top} - \boldsymbol{X}_{(k)}^{\top}|_F^2$$

$$\boldsymbol{Z}_k^{\top} \boldsymbol{Z}_k \boldsymbol{A}_k^{\top} = \boldsymbol{Z}_k^{\top} \boldsymbol{X}_{(k)}^{\top}$$

$$\underbrace{(\boldsymbol{A}_d^{\top} \boldsymbol{A}_d * \cdots * \boldsymbol{A}_{k+1}^{\top} \boldsymbol{A}_{k+1} * \boldsymbol{A}_{k-1}^{\top} \boldsymbol{A}_{k-1} \cdots * \boldsymbol{A}_1^{\top} \boldsymbol{A}_1)}_{\boldsymbol{V}_k} \boldsymbol{A}_k^{\top} = \boldsymbol{Z}_k^{\top} \boldsymbol{X}_{(k)}^{\top}$$

$$\boldsymbol{A}_k = \boldsymbol{X}_{(k)} \boldsymbol{Z}_k \boldsymbol{V}_k^{-1}$$

## CP-ALS Full Algorithm

**Inputs:** Tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, desired rank $r \in \mathbb{N}$.

1. Initialize $\boldsymbol{A}_k \in \mathbb{R}^{n_k \times r}$ for all $k \in [d]$
2. **repeat**
3.     **for** $k = 1, \ldots, d$ **do**
4.         $\boldsymbol{Z}_k \leftarrow \boldsymbol{A}_d \odot \cdots \odot \boldsymbol{A}_{k+1} \odot \boldsymbol{A}_{k-1} \odot \cdots \odot \boldsymbol{A}_1$
5.         $\boldsymbol{A}_k \leftarrow \arg\min_{\boldsymbol{B}} \|\boldsymbol{Z}_k \boldsymbol{B}^\top - \boldsymbol{X}_{(k)}^\top\|_F^2$
6.     **end**
7. **until** $\|\mathcal{X} - [\![\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_d]\!]\|_F^2$ ceases to decrease

## Can randomization help?
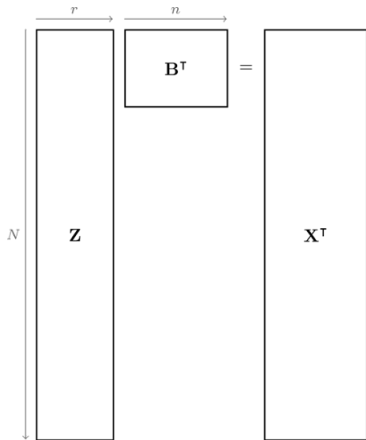


$$\min_{\boldsymbol{B}} \|\boldsymbol{Z}\boldsymbol{B}^\top - \boldsymbol{X}^\top\|_F^2$$

Conversion from tensor problem...

$$N = \prod_{\ell=1,\neq k}^{d} n_\ell, \quad n = n_k$$

$$\boldsymbol{Z} = \boldsymbol{A}_d \odot \cdots \odot \boldsymbol{A}_{k+1} \odot \boldsymbol{A}_{k-1} \odot \cdots \odot \boldsymbol{A}_1$$

$$\boldsymbol{X} = \boldsymbol{X}_{(k)}$$

$$\boldsymbol{B} = \boldsymbol{A}_k$$

## Recall: Sketch and solve

Use *Sketching*:

- Generate a sketching matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n}$.
- Compute sketches $\boldsymbol{SA}$ and $\boldsymbol{Sb}$.
- Solve:

$$\tilde{\boldsymbol{x}} = \min_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{SAx} - \boldsymbol{Sb}\|_2^2.$$

- Typically, $m = \text{poly}(d/\epsilon)$.
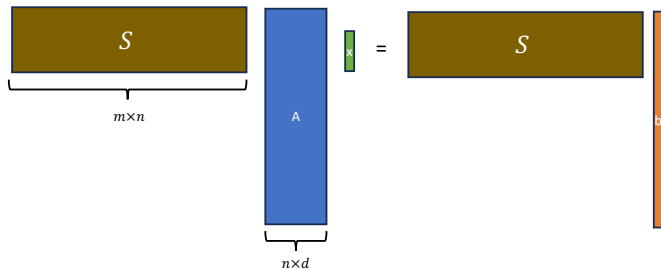
# Uniform sampling

$$\min_{\boldsymbol{B}} \|\boldsymbol{Z}\boldsymbol{B}^\top - \boldsymbol{X}^\top\|_F^2$$
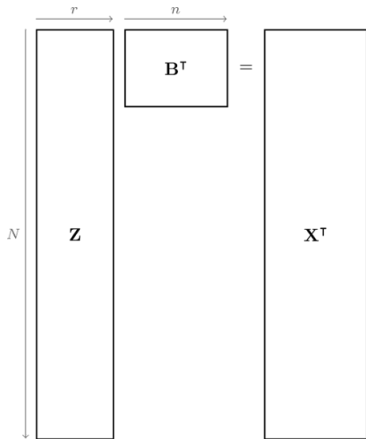


Constructing *sample* matrix $\boldsymbol{S}$ of size $s \times N$

- $s$ be the number of samples
- Each row of $\boldsymbol{S}$ is a random row of the $N \times N$ identity matrix, Scaled by $1/\sqrt{s}$.

$$\min_{\boldsymbol{B}} \|\boldsymbol{S}\boldsymbol{Z}\boldsymbol{B}^\top - \boldsymbol{S}\boldsymbol{X}^\top\|_F^2$$

# Uniform sampling

**Challenges:**

- Does uniform sampling "work"?
- $X^\top$ is expensive (in memory movement) to form
- $Z$ is expensive (in computations) to form
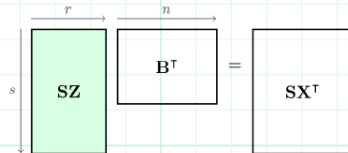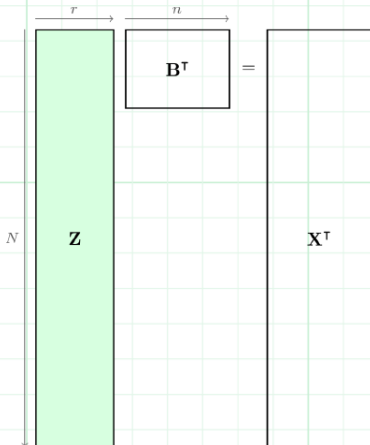- Checking convergence of overall CP ALS method

$$\underset{s}{\underset{\displaystyle\longrightarrow}{\overset{\displaystyle r}{\boxed{\mathbf{SZ}}}}}\ \overset{n}{\boxed{\mathbf{B^\top}}} = \boxed{\mathbf{SX^\top}}$$
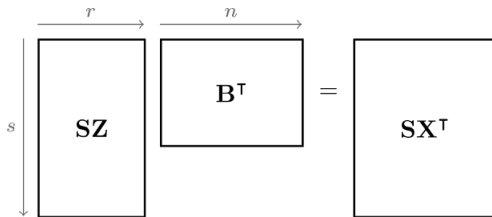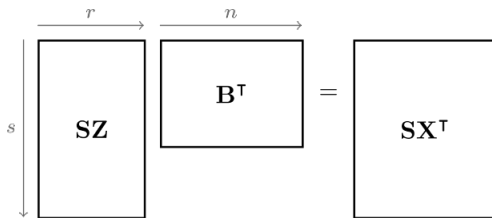
# Forming Sampled KRP



Factor Matrices
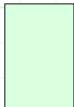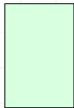
**Khatri-Rao Product**
$$\mathbf{Z} = \mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1$$
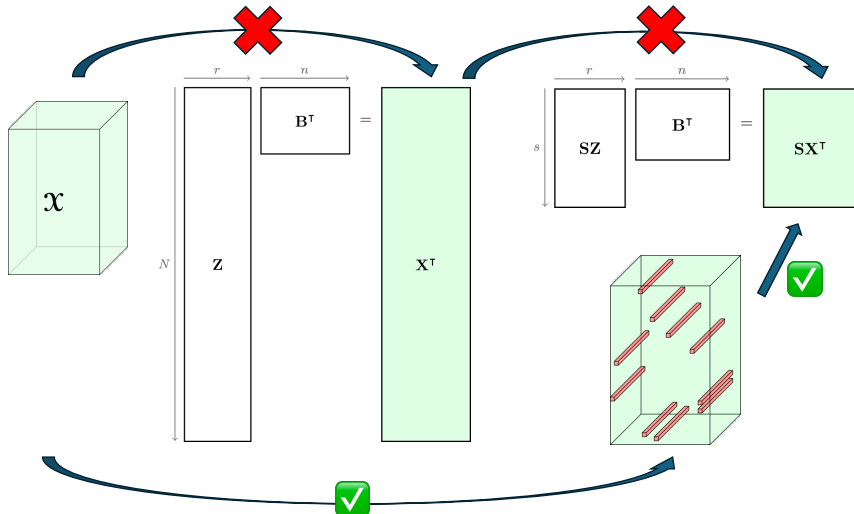
$\mathbf{B}^\mathsf{T}$ $=$ $\mathbf{X}^\mathsf{T}$

$\mathbf{Z}$

$\mathbf{SZ}$ $\mathbf{B}^\mathsf{T}$ $=$ $\mathbf{SX}^\mathsf{T}$

# Uniform sampling

**Challenges:**

- Does uniform sampling "work"?
- $X^\top$ is expensive (in memory movement) to form
- $Z$ is expensive (in computations) to form ✓
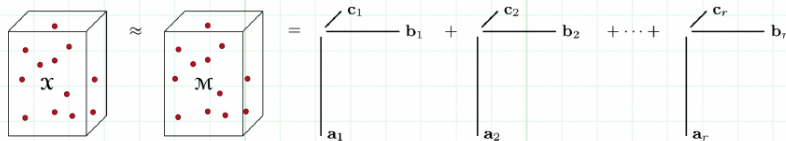- Checking convergence of overall CP ALS method

# Forming Sampled Right-hand Side

# Uniform sampling

**Challenges:**

- Does uniform sampling "work"?
- $X^\top$ is expensive (in memory movement) to form ✓
- $Z$ is expensive (in computations) to form ✓
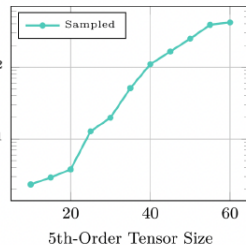- Checking convergence of overall CP ALS method

# Checking Convergence



$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathbf{X} - \mathbf{M}\|^2 \equiv \sum_{i,j,k} (x_{ijk} - m_{ijk})^2$$

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathbf{X} - \mathbf{M}\|^2 \approx \frac{\prod n_k}{|\Omega|} \sum_{(i,j,k) \in \Omega} (x_{ijk} - m_{ijk})^2$$

*Maybe don't even check every iteration...*

$$\frac{|f - \hat{f}|}{|f|} < 10^{-3}$$

**16000 samples < 1% of full data**

# Uniform sampling

**Challenges:**

- Does uniform sampling "work"?
- $X^\top$ is expensive (in memory movement) to form ✓
- $Z$ is expensive (in computations) to form ✓
- Checking convergence of overall CP ALS method ✓

## CP-ARLS Algorithm

**Inputs:** Tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, desired rank $r \in \mathbb{N}$, number of samples $s \in \mathbb{N}$.

1. Initialize $\boldsymbol{A}_k \in \mathbb{R}^{n_k \times r}$ for all $k \in [d]$
2. $\Omega \leftarrow$ sampled indices for function value estimation
3. **repeat**
4.      **for** $k = 1, \ldots, d$ **do**
5.          $\boldsymbol{S} \leftarrow$ random rows of $\boldsymbol{I}$ scaled by $1/\sqrt{s}$.
6.          $\hat{\boldsymbol{Z}} \leftarrow \mathrm{SKRP}(\boldsymbol{S}, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_{k-1}, \boldsymbol{A}_{k+1}, \ldots, \boldsymbol{A}_d)$
7.          $\hat{\boldsymbol{X}} \leftarrow \mathrm{STU}(\boldsymbol{S}, \mathcal{X}, k)$
8.          $\boldsymbol{A}_k \leftarrow \arg\min_{\boldsymbol{B}} \|\hat{\boldsymbol{Z}} \boldsymbol{B}^\top - \hat{\boldsymbol{X}}^\top\|_F^2$
9.      **end**
10. **until** $\mathbf{SFV}(\Omega, \mathcal{X}, \mathbf{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_d)$ ceases to decrease

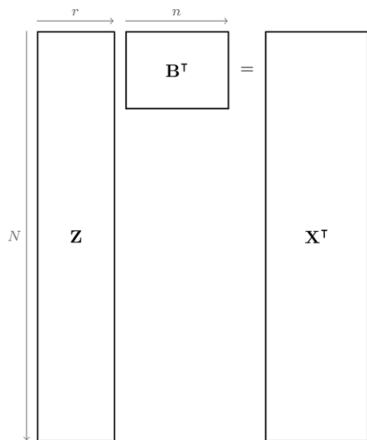Matlab Demo

# Sketching Problem with Plain Sampling

$$\min_{\boldsymbol{B}} \|\boldsymbol{Z}\boldsymbol{B}^\top - \boldsymbol{X}^\top\|_F^2$$



Constructing *sample* matrix $\boldsymbol{S}$ of size $s \times N$

- $s$ be the number of samples
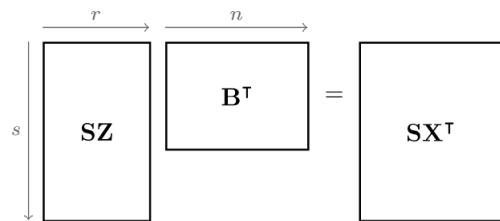- Each row of $\boldsymbol{S}$ is a random row of the $N \times N$ identity matrix, Scaled by $1/\sqrt{s}$.

$$\min_{\boldsymbol{B}} \|\boldsymbol{S}\boldsymbol{Z}\boldsymbol{B}^\top - \boldsymbol{S}\boldsymbol{X}^\top\|_F^2$$



Uniform sampling is only efficient if $\boldsymbol{Z}$ is incoherent.

# Recall: FJLT (SRHT/SRFT)
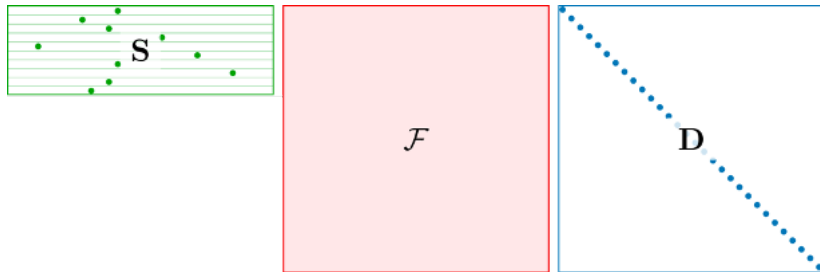


- $S$ is a sampling matrix
- $\mathcal{F}$ is an FFT (or Hadamard) matrix.
- $D$ is a diagonal matrix with $\pm 1$ (Radamacher) entries.

# Mixing using FJLTs

$$\min_{\boldsymbol{B}} \|\boldsymbol{Z}\boldsymbol{B}^\top - \boldsymbol{X}^\top\|_F^2$$
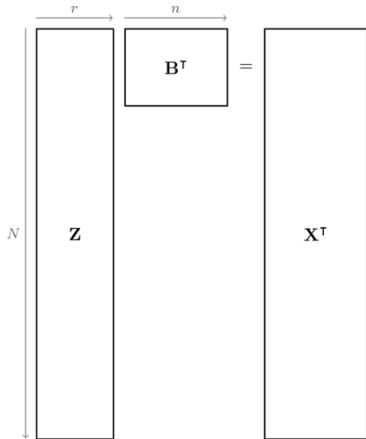


$$\min_{\boldsymbol{B}} \|\boldsymbol{S}\mathcal{F}\boldsymbol{D}\boldsymbol{Z}\boldsymbol{B}^\top - \boldsymbol{S}\mathcal{F}\boldsymbol{D}\boldsymbol{X}^\top\|_F^2$$

- $\boldsymbol{S}$ is $s \times N$ sampling matrix
- $\mathcal{F}$ is $N \times N$ FFT (or Hadamard) matrix.
- $\boldsymbol{D}$ is a $N \times N$ diagonal matrix with $\pm 1$ (Radamacher) entries.
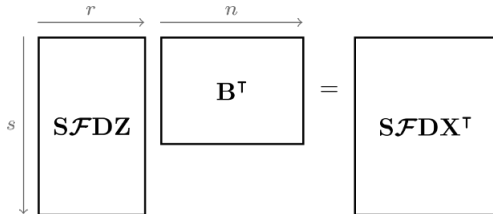
# Mixing using Mixing using Kronecker FJLTs

$$\min_{\boldsymbol{B}} \|\boldsymbol{Z}\boldsymbol{B}^{\top} - \boldsymbol{X}^{\top}\|_F^2$$
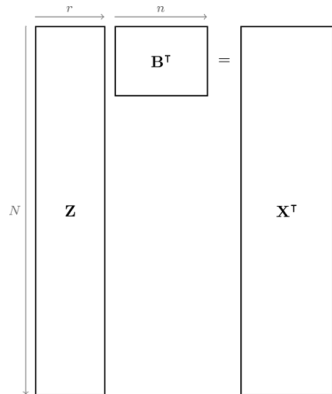


$$\min_{\boldsymbol{B}} \|\boldsymbol{S}\bar{\mathcal{F}}\bar{\boldsymbol{D}}\boldsymbol{Z}\boldsymbol{B}^{\top} - \boldsymbol{S}\bar{\mathcal{F}}\bar{\boldsymbol{D}}\boldsymbol{X}^{\top}\|_F^2$$

- $\boldsymbol{S}$ is $s \times N$ sampling matrix
- $\bar{\mathcal{F}} = \mathcal{F}_d \otimes \cdots \otimes \mathcal{F}_{k+1} \otimes \mathcal{F}_{k-1} \otimes \cdots \otimes \mathcal{F}_1$.
- $\bar{\boldsymbol{D}} = \boldsymbol{D}_d \otimes \cdots \otimes \boldsymbol{D}_{k+1} \otimes \boldsymbol{D}_{k-1} \otimes \cdots \otimes \boldsymbol{D}_1$.



$$\boldsymbol{Z} = \boldsymbol{A}_d \odot \cdots \odot \boldsymbol{A}_{k+1} \odot \boldsymbol{A}_{k-1} \odot \cdots \odot \boldsymbol{A}_1$$
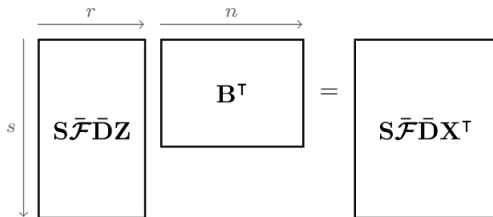
# Kronecker FJLTs (Simpler case)

$$\min_{B} \|ZB^\top - X^\top\|_F^2$$
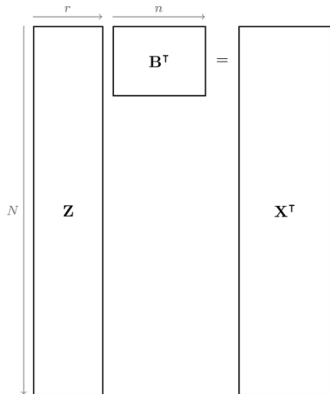


$$Z = A_2 \odot A_1$$

$$\min_{B} \|S\bar{\mathcal{F}}\bar{D}ZB^\top - S\bar{\mathcal{F}}\bar{D}X^\top\|_F^2$$

- $S$ is $s \times N$ sampling matrix
- $\bar{\mathcal{F}} = \mathcal{F}_2 \otimes \mathcal{F}_1$.
- $\bar{D} = D_2 \otimes D_1$.
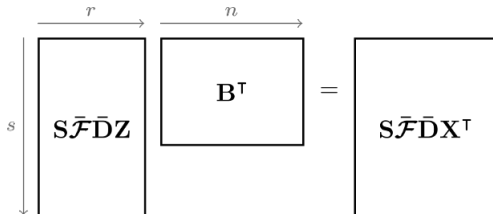
# Mixing KRP Efficiently Using Kronecker FJLT
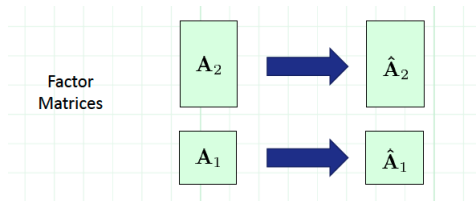


$$\boldsymbol{S}\bar{\mathcal{F}}\bar{\boldsymbol{D}}\boldsymbol{Z} = \boldsymbol{S}(\mathcal{F}_2 \otimes \mathcal{F}_1)(\boldsymbol{D}_2 \otimes \boldsymbol{D}_1)(\boldsymbol{A}_2 \odot \boldsymbol{A}_1)$$
$$= \boldsymbol{S}\left((\mathcal{F}_2\boldsymbol{D}_2) \otimes (\mathcal{F}_1\boldsymbol{D}_1)\right)(\boldsymbol{A}_2 \odot \boldsymbol{A}_1)$$
$$= \boldsymbol{S}\left((\mathcal{F}_2\boldsymbol{D}_2\boldsymbol{A}_2) \odot (\mathcal{F}_1\boldsymbol{D}_1\boldsymbol{A}_1)\right)$$
$$= \boldsymbol{S}(\hat{\boldsymbol{A}}_2 \odot \hat{\boldsymbol{A}}_1)$$

## Pre-Mixing Tensor

Need to compute sketched right hand side...

$$\boldsymbol{S}\bar{\mathcal{F}}\bar{\boldsymbol{D}}\boldsymbol{X}^\top = \boldsymbol{S}(\mathcal{F}_2 \otimes \mathcal{F}_1)(\boldsymbol{D}_2 \otimes \boldsymbol{D}_1)\boldsymbol{X}_{(3)}^\top$$

Pre-mixed tensor

$$\tilde{\mathcal{X}} = \mathcal{X} \times_1 \mathcal{F}_1 \boldsymbol{D}_1 \times_2 \mathcal{F}_2 \boldsymbol{D}_2 \times_3 \mathcal{F}_3 \boldsymbol{D}_3$$

$$\tilde{\boldsymbol{X}}_{(3)}^\top = (\mathcal{F}_2 \boldsymbol{D}_2 \otimes \mathcal{F}_1 \boldsymbol{D}_1)\boldsymbol{X}_{(3)}^\top (\mathcal{F}_3 \boldsymbol{D}_3)^\top$$

Sample before unmixing

$$\boldsymbol{S}\bar{\mathcal{F}}\bar{\boldsymbol{D}}\boldsymbol{X}^\top = (\boldsymbol{S}\tilde{\boldsymbol{X}}_{(3)}^\top)\boldsymbol{D}_3 \mathcal{F}_3^*$$

$$\min_{\boldsymbol{B}} \|\boldsymbol{S}\bar{\mathcal{F}}\bar{\boldsymbol{D}}\boldsymbol{Z}\boldsymbol{B}^\top - \boldsymbol{S}\bar{\mathcal{F}}\bar{\boldsymbol{D}}\boldsymbol{X}^\top\|_F^2$$

## CP-ARLS-Mix Algorithm

**Inputs:** Tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, desired rank $r \in \mathbb{N}$, number of samples $s \in \mathbb{N}$.

1. Initialize $\boldsymbol{A}_k \in \mathbb{R}^{n_k \times r}$ for all $k \in [d]$
2. Draw random diagonal $\boldsymbol{D}_k$ for all $k \in [d]$
3. Compute $\tilde{\boldsymbol{A}}_k = \mathcal{F}_k \boldsymbol{D}_k \boldsymbol{A}_k$ for all $k \in [d]$
4. Compute $\tilde{\mathcal{X}} = \mathcal{X} \times_1 \mathcal{F}_1 \boldsymbol{D}_1 \times_2 \mathcal{F}_2 \boldsymbol{D}_2 \cdots \times_d \mathcal{F}_d \boldsymbol{D}_d$
5. $\Omega \leftarrow$ sampled indices for function value estimation
6. **repeat**
7.     **for** $k = 1, \ldots, d$ **do**
8.         $\boldsymbol{S} \leftarrow$ random rows of $\boldsymbol{I}$ scaled by $1/\sqrt{s}$.
9.         $\hat{\boldsymbol{Z}} \leftarrow \text{SKRP}(\boldsymbol{S}, \tilde{\boldsymbol{A}}_1, \ldots, \tilde{\boldsymbol{A}}_{k-1}, \tilde{\boldsymbol{A}}_{k+1}, \ldots, \tilde{\boldsymbol{A}}_d)$
10.         $\hat{\boldsymbol{X}} \leftarrow \mathcal{F}_k^* \boldsymbol{D}_k \left( \text{STU}(\boldsymbol{S}, \tilde{\mathcal{X}}, k) \right)$
11.         $\boldsymbol{A}_k \leftarrow \arg\min_{\boldsymbol{B}} \|\hat{\boldsymbol{Z}} \boldsymbol{B}^\top - \hat{\boldsymbol{X}}^\top\|_F^2$
12.         $\tilde{\boldsymbol{A}}_k \leftarrow \mathcal{F}_k \boldsymbol{D}_k \boldsymbol{A}_k$
13.     **end**
14. **until** $\text{SFV}(\Omega, \mathcal{X}, \boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_d)$ ceases to decrease

# Randomized-CP

**Further Reading:**

- D. Cheng, R. Peng, I. Perros , and Y. Liu. *SPALS: Fast Alternating Least Squares via Implicit Leverage Scores Sampling* , NeurIPS'16

- C. Battaglino , G. Ballard, and T. G. Kolda. *A Practical Randomized CP Tensor Decomposition* , SIAM J. Matrix Analysis and Applications, 2018

- R. Jin, T. G. Kolda, and R. Ward. *Faster Johnson Lindenstrauss Transforms via Kronecker Products*, Information and Inference, 2020

- O. A. Malik, and S. Becker. *Guarantees for the Kronecker Fast Johnson Lindenstrauss Transform Using a Coherence and Sampling Argument*, Linear Algebra and its Applications, 2020

- M. A. Iwen , D. Needell, E. Rebrova , and A. Zare . *Lower Memory Oblivious (Tensor) Subspace Embeddings with Fewer Random Bits: Modewise Methods for Least Squares* , SIAM J. Matrix Analysis and Applications, 2021

Matlab Demo