2 Randomized CP- I

2.1 Alternating Least Squares (CP-ALS) Review

Finding the rank of tensor X is an NP hard problem and thus an approximation algorithm will be necessary. One potential algorithm is the Alternating Least Squares algorithm which optimizes across all three variables, A, B, and C, one at a time. The general idea is to fix B and C and solve for A, then fix A and C and solve for B, and fix A and B and solve for C, and repeat until convergence. The equation to solve for A is shown below:

$$\min_A \|X_{(1)} - A(C \odot B)^T\|_F^2 =$$
$$\min_A \|(C \odot B)A^T - X_{(1)}^T\|_F^2$$

From normal equations:

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

This can be written for a general d-way tensor as the following:

$$\min_{A_k} \|X_{(k)} - A_{A_k}(A_d \odot ... \odot A_{k+1} \odot A_{k-1} \odot ... \odot A_1)^T\|_F^2 =$$
$$\min_{A_k} \|Z_kA^T - X_{(k)}^T\|_F^2$$

where $Z_k = A_d \odot ... \odot A_{k+1} \odot A_{k-1} \odot ... \odot A_1$

$$Z_k^TZ_kA_k^T = Z_k^TX_{(k)}^T$$

$$(A_d^TA_d * ... * A_{k+1}^TA_{k+1} * A_{k-1}^TA_{k-1} * ... * A_1^TA_1)A_k^T = Z_k^TX_{(k)}^T$$

$$A_k = X_{(k)}Z_kV_k^{-1}$$

where $V_k = A_d^TA_d * ... * A_{k+1}^TA_{k+1} * A_{k-1}^TA_{k-1} * ... * A_1^TA_1$

Finally for a d-way tensor the generalized alternating least squares algorithm is given as follows for a desired rank r, also followed by a MATLAB example:

1. Initialize $A_k \in \mathbb{R}^{n_k \times r}$ for all $k \in [d]$
2. repeat
3. for k = 1,...,d do
4. $Z_k \leftarrow A_d \odot ... \odot A_{k+1} \odot A_{k-1} \odot ... \odot A_1$
5. $A_k \leftarrow \arg\min_B \|Z_k B^T - X_{(k)}^T\|_F^2$
6. end
7. until $\|X - [[A_1, A_2, ..., A_d]]\|_F^2$ under threshold

```
% Tensor decomposition

% Example for CP-ALS using tensor-toolbox

% Tensor Toolbox : https://www.tensortoolbox.org/

% The Excitation Emission Matrix (EEM) tensor data
% has been curated from a series of
% Fluorescence Spectroscopy experiments

% Dataset from https://gitlab.com/tensors/tensor_data_eem

addpath(genpath(pwd))
%% Load data

load eem18

% 18 samples x 251 emissions x 21 excitations.

%% Visualization

viz_slices(X,mixtures,1,'X-slices')

%% CP decomposition

rng('default')
M = cp_als(X,5);
viz_eem_cp(M,mixtures,[],'eem_model');

%% Error

err = norm(X - tensor(M))
```

Figure 3: MATLAB example of CP-ALS

2.2 Adding Randomization

Similar to matrices, it is possible to add randomized sampling to tensors to aid in the optimization process by reducing the dimensionality of the dataset. Recall the sketch and solve method for matrices where a sketching matrix S $\in \mathbb{R}^{m \times n}$ is used to solve the following minimization: $\tilde{x} = \min_{x \in \mathbb{R}^d} \|SAx - Sb\|_2^2$ which approximates the solution x. Similar methods can be applied to tensors.

For following methods the tensor problem is defined as:

$\min_B \|ZB^T - X^T\|_F^2$ where $Z = A_d \odot ... \odot A_{k+1} \odot A_{k-1} \odot ... \odot A_1$, and is of size r x N, where $N = \prod_{l=1,\neq k}^{d} n_l$

$X = X_(k)$, the matricization of X along the kth dimension

$B = A_k$, the factorization matrix for the kth dimension.

2.3 Uniform Sampling

A simple method of adding randomization is by uniform sampling where sample matrix S of size s x N is constructed, where s is the number of samples, and each row of S is a random row of N x N identity matrix, scaled by $\frac{1}{\sqrt{s}}$. This matrix S can then be applied to Z and X^T to reduce the dimensionality from N to s and solve the following minimization problem: $\min_B \|SZB^T - SX^T\|_F^2$.

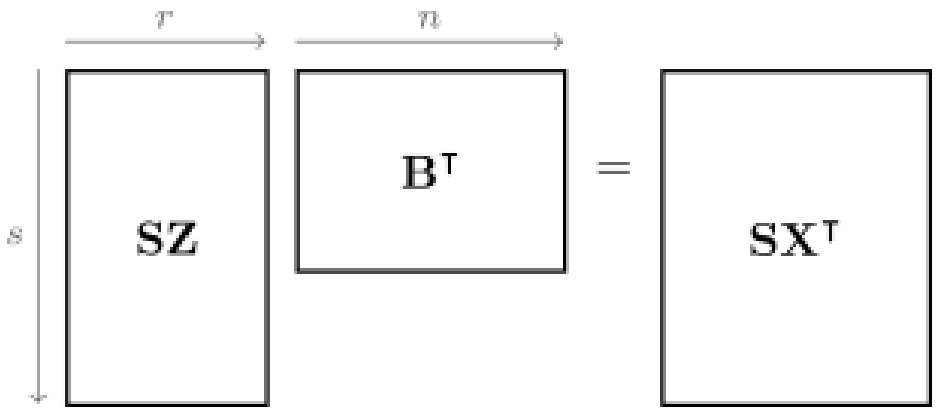


Figure 4: Randomized Reduced Tensor Problem

Uniform sampling is shown to be an effective optimization for the tensor problem since it helps improve the computational and memory efficiency of the CP-ALS algorithm while still converging to a satisfactory approximation. More specifically, it helps reduce the number of computations needed to form Z by down sampling to SZ with a smaller size, s x r. Secondly, it reduces the memory size of storing X^T, also by down sampling to SX^T. The following figures depict how the reduced matrices improve the efficiency:

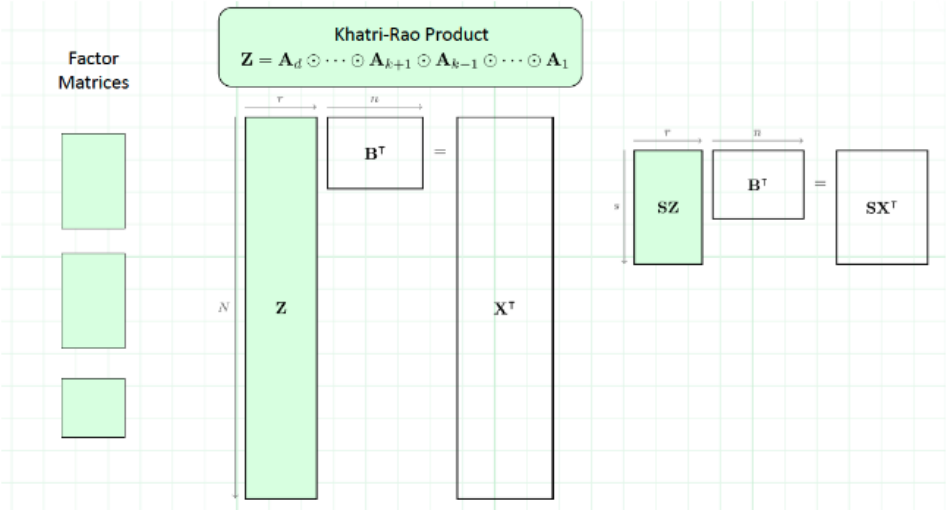


Figure 5: Reduced KRP

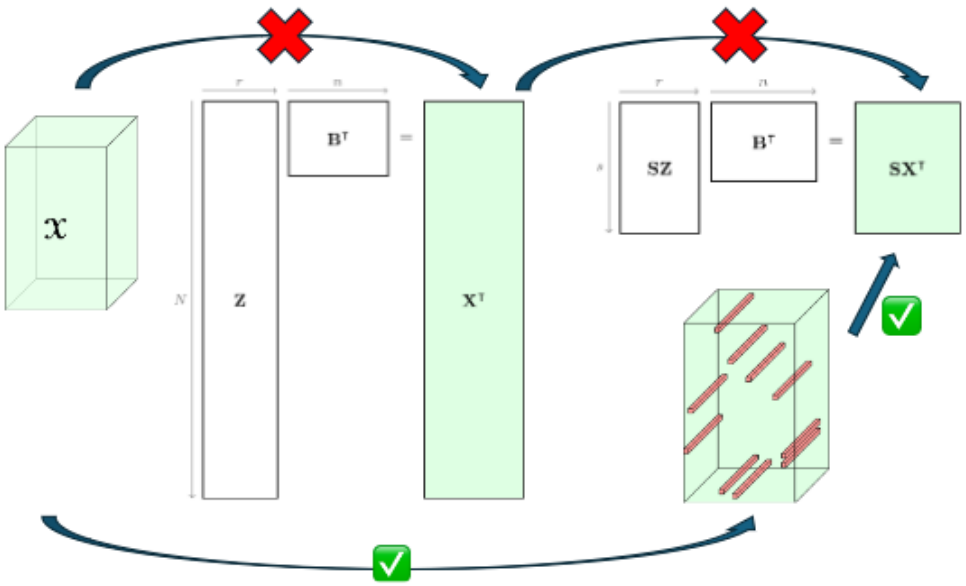
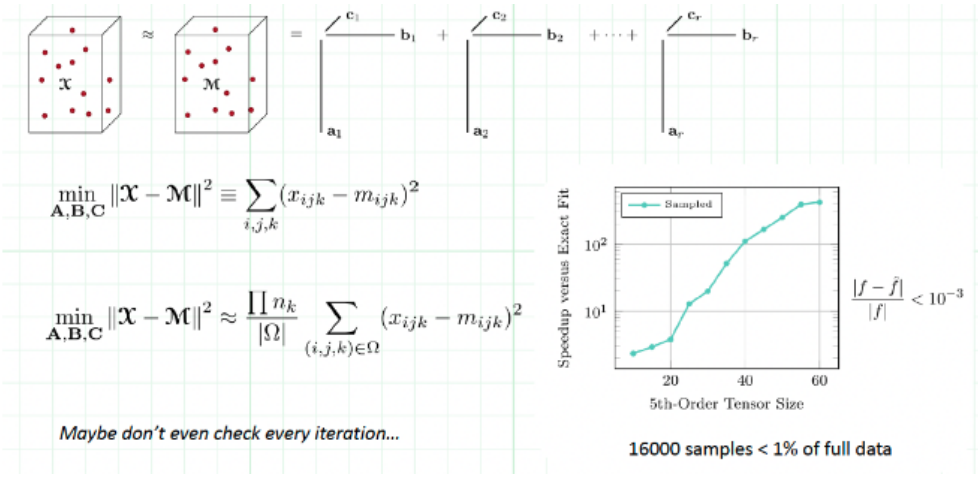


Figure 6: Reduced X^T

Finally it is proven that the sampling algorithm still provides a satisfactory approximation even when sampling less than 1% of the entire data and the speedup benefits improve as the tensor size increases. The following figure depicts these finding:



Thus, the Alternating Least Squared algorithm with randomization, CP-ARLS, is given as follows for a desired rank r and number of samples s, also followed by a MATLAB example:

1. Initialize $A_k \in \mathbb{R}^{n_k \times r}$ for all $k \in [d]$
2. $\Omega \leftarrow$ the sampled indices for function value estimation
3. repeat
4. for k = 1,...,d do
5. $S \leftarrow$ random rows of I scaled by $\frac{1}{\sqrt{s}}$
6. $\hat{Z} \leftarrow SKRP(S, A_1, ..., A_{k-1}, A_{k+1}, ..., A_d)$
7. $\hat{X} \leftarrow STU(S, X, k)$
8. $\hat{A}_k \leftarrow \arg\min_B \|\hat{Z}B^T - \hat{X}^T\|_F^2$
9. end
10. until $SFV(\Omega, X, A_1, A_2, ..., A_d)$ ceases to decrease

```
% Tensor decomposition

% Example for CP-ARLS using tensor-toolbox

% Tensor Toolbox : https://www.tensortoolbox.org/

% The Excitation Emission Matrix (EEM) tensor data
% has been curated from a series of
% Fluorescence Spectroscopy experiments

% Dataset from https://gitlab.com/tensors/tensor_data_eem
clc
close all
addpath(genpath(pwd))
%% Load data

load eem18

% 18 samples x 251 emissions x 21 excitations.

%% Visualization

viz_slices(X,mixtures,1,'X-slices')

%% CP-ALS decomposition

rng('default')
tic;
M1 = cp_als(X,3);
toc;
viz_eem_cp(M1,mixtures,[],'eem_model');

%% CP-ARLS decomposition

rng('default')
tic;
M2 = cp_arls(X,3,'mix',true);
toc;
viz_eem_cp(M2,mixtures,[],'eem_model');

%% Errors

err1 = norm(X - tensor(M1))/norm(X)

err2 = norm(X - tensor(M2))/norm(X)
```

Figure 7: MATLAB example of CP-ARLS

2.4 Sampling Alternatives

While the randomized uniform sampling method does converge and improves efficiency of the CP-ALS algorithm it is only the most efficient when Z is incoherent. In other cases mixing the sampling can lead to better optimizations.

One such mixing method involves using FJLTs. The randomized tensor problem with JFLTS is defined as follows:

S is a s x N sampling matrix

F is an N x N FFT, or Hadamard, matrix

D is a N x N diagonal matrix with ± 1 Radamacher entries

The above matrices can be applied to Z and X^T to reduce the dimensionality from N to s and solve the following minimization problem: $\min_B \|SFDZB^T - SFDX^T\|_F^2$.

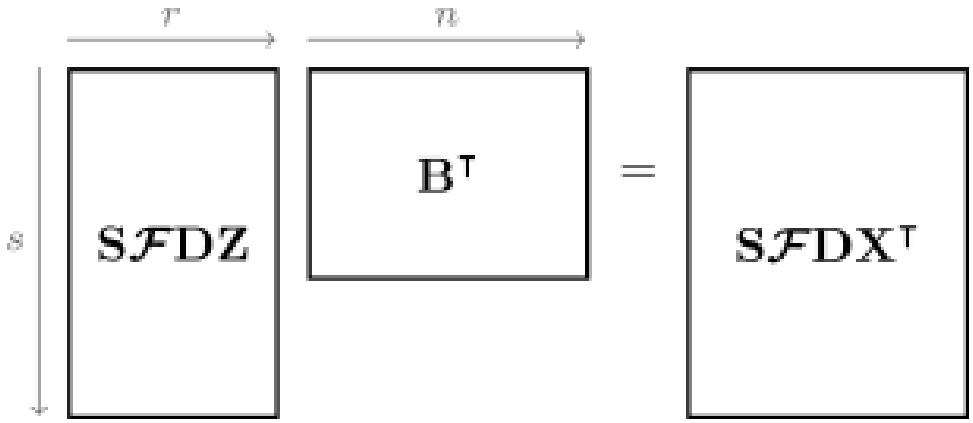


Figure 8: Randomized Reduced Tensor Problem with FJLT mixing

Another alternative method involves applying Kronecker products to FJLTS. The randomized tensor problem with Kronecker JFLTS is defines as follows:

S is a s x N sampling matrix

$\bar{F} = F_d \otimes ... \otimes F_{k+1} \otimes F_{k-1} \otimes ... \otimes F_1$

$\bar{D} = D_d \otimes ... \otimes D_{k+1} \otimes D_{k-1} \otimes ... \otimes D_1$

The above matrices can be applied to Z and X^T to reduce the dimensionality from N to s and solve the following minimization problem: $\min_B \|S\bar{F}\bar{D}ZB^T - S\bar{F}\bar{D}X^T\|_F^2$.

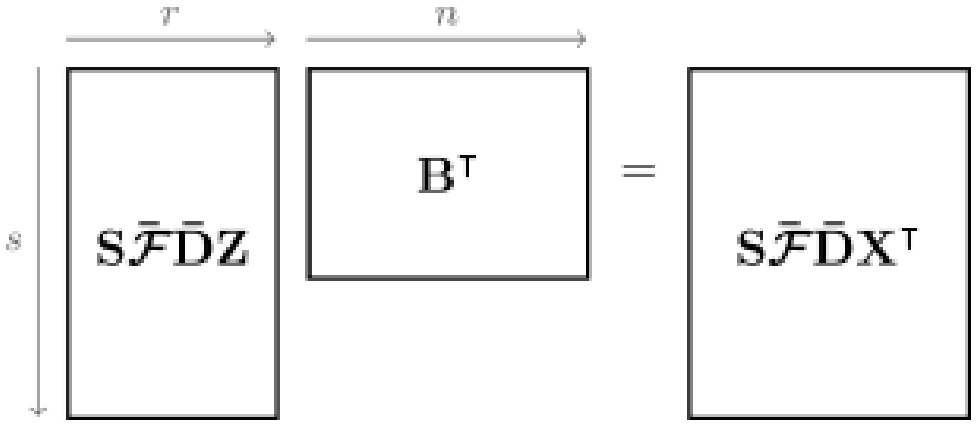


Figure 9: Randomized Reduced Tensor Problem with Kronecker FJLT mixing

Finally, the Alternating Least Squared algorithm with randomization and mixing, CP-ARLS-Mix, is given as follows for a desired rank r and number of samples s:

1. Initialize $A_k \in \mathbb{R}^{n_k \times r}$ for all $k \in [d]$
2. Draw random diagonal D_k for all $k \in [d]$
3. Compute $\hat{A}_k = F_k D_k A_k$ for all $k \in [d]$
4. Compute $\hat{X} = X \times F_1 D_1 \times F_2 D_2 ... \times F_d D_d$
5. $\Omega \leftarrow$ the sampled indices for function value estimation
6. repeat
7. for k = 1,...,d do
8. $S \leftarrow$ random rows of I scaled by $\frac{1}{\sqrt{s}}$
9. $\hat{Z} \leftarrow SKRP(S, A_1, ..., A_{k-1}, A_{k+1}, ..., A_d)$
10. $\hat{X} \leftarrow F_k^* D_k(STU(S, X, k))$
11. $A_k \leftarrow \arg\min_B \|\hat{Z}B^T - \hat{X}^T\|_F^2$
12. $\hat{A}_k \leftarrow F_k D_k A_k$

13. end

14. until $SFV(\Omega, X, A_1, A_2, ..., A_d)$ ceases to decrease