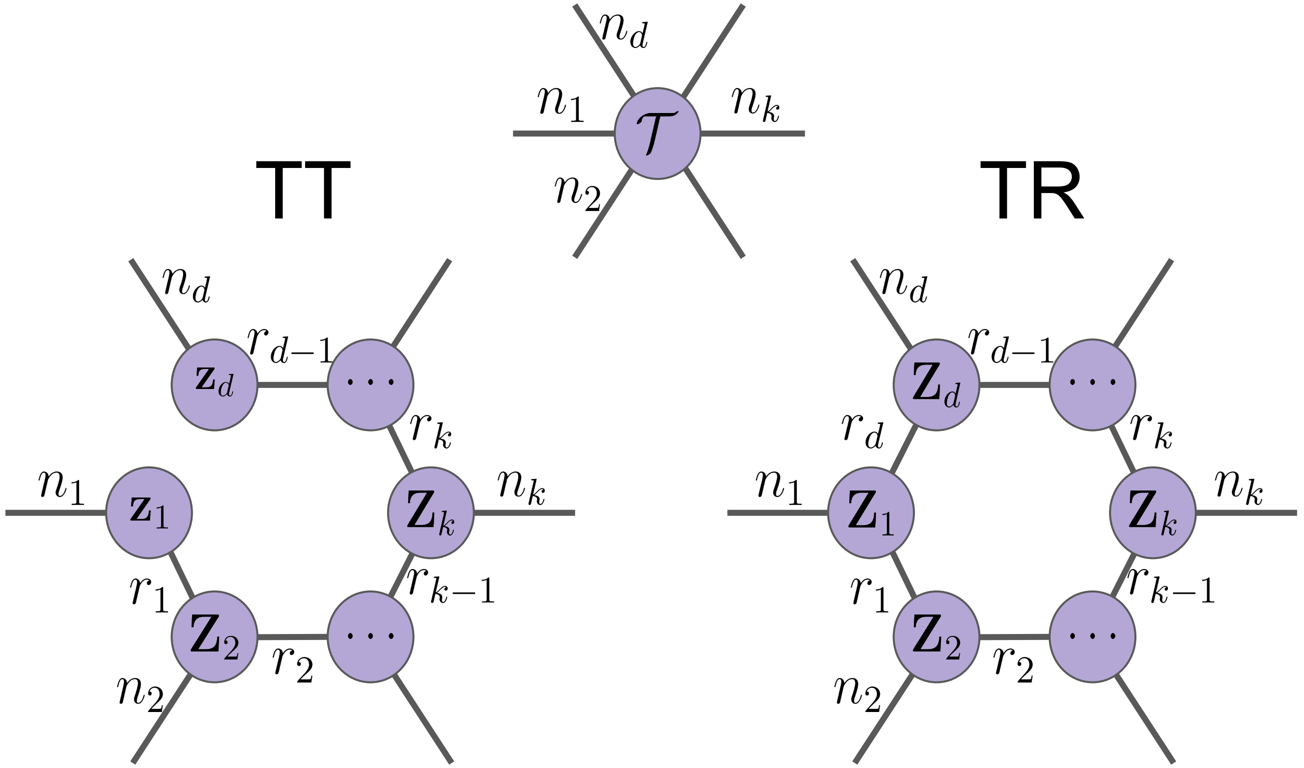
Accelerating TR-ALS for Tensor Completion

Maxim Kurkin, Igor Udovichenko
Daniil Merkulov, **Nazariy Tupitsa**

Introduction

Tensor Train [1] and Tensor Ring [2] formats can be used to compress tensor of $O(n^d)$ size to $O(dnr^2)$ elements.

Tensor Train and Tensor Ring decompositions



Tensor Train and Tensor Ring decompositions

Consider a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

Tensor Ring representation in index format can be written as trace of chain of three-dimensional tensors $\mathbf{Z}_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$:

$$\mathcal{T}(i_1, i_2, \ldots, i_d) = \mathrm{Tr}\{\mathbf{Z}_1(i_1)\mathbf{Z}_2(i_2) \cdots \mathbf{Z}_d(i_d)\}$$

Tensor Train is similar chain of multiplications, but with first and last factors being a matrix:

$$\mathcal{T}(i_1, i_2, \ldots, i_d) = \mathbf{z}_1^T(i_1)\mathbf{Z}_2(i_2) \cdots \mathbf{Z}_{d-1}(i_{d-1})\mathbf{z}_d(i_d)$$

Problem

The goal is to find the TR factors that minimize the Frobenius norm between the given tensor and the recovered tensor:

$$\sum_{i_1,\ldots,i_d} \left(\mathcal{T}(i_1, i_2, \ldots, i_d) - \operatorname{Tr}\{\mathbf{Z}_1(i_1)\mathbf{Z}_2(i_2)\cdots\mathbf{Z}_d(i_d)\}\right)^2 \to \min_{\mathbf{Z}_1,\ldots,\mathbf{Z}_d}$$

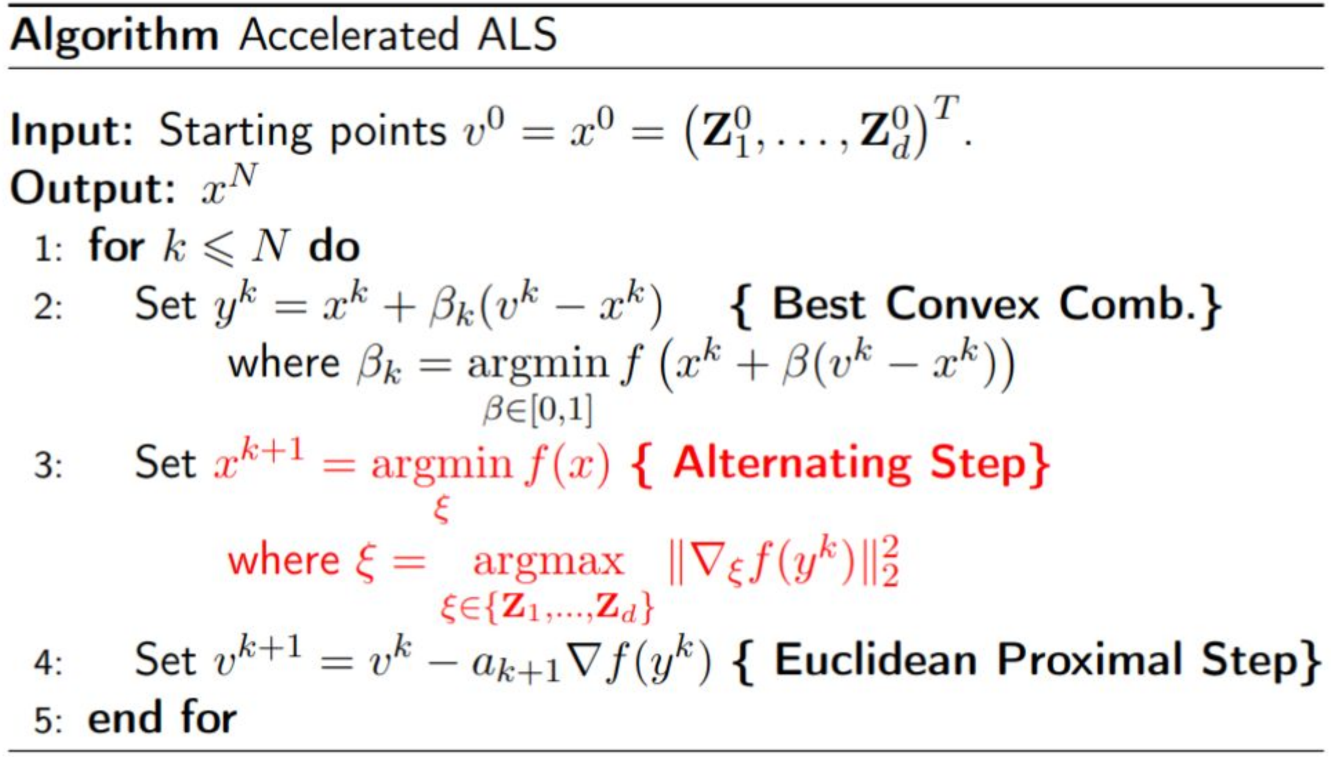
ALS algorithm

Due to complexity of construction of full-rank exact decompositions, approximate truncated rank methods like ALS or SVD are used.

Alternating Least Squares algorithm finds minima of least squares problem w.r.t. to only one factor while others stay fixed:

$$\sum_{i_1,\ldots,i_d} \left(\mathcal{T}(i_1, i_2, \ldots, i_d) - \mathrm{Tr}\{\mathbf{Z}_1(i_1)\mathbf{Z}_2(i_2)\cdots\mathbf{Z}_d(i_d)\}\right)^2 \to \min_{\mathbf{Z}_i}$$

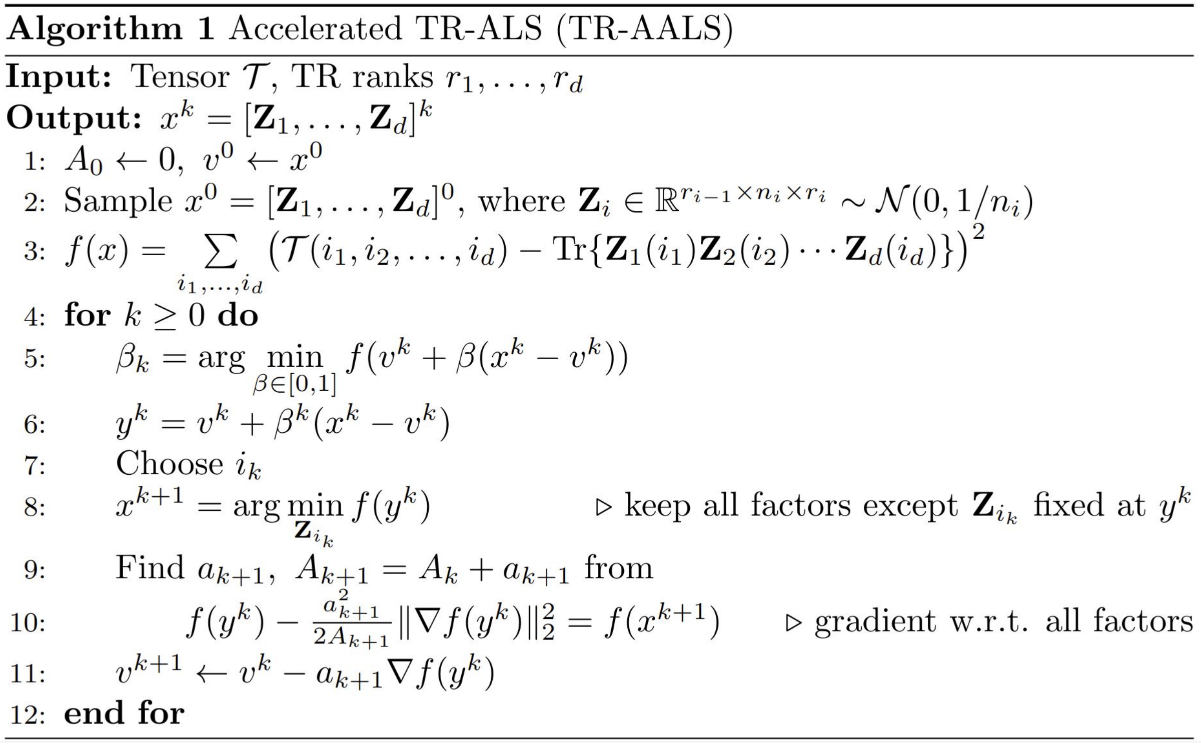
Accelerated ALS algorithm



Algorithm Accelerated ALS

```
Input: Starting points $v^0 = x^0 = (\mathbf{Z}_1^0, \ldots, \mathbf{Z}_d^0)^T$.
Output: $x^N$
1: for $k \leqslant N$ do
2:     Set $y^k = x^k + \beta_k(v^k - x^k)$    { Best Convex Comb.}
           where $\beta_k = \underset{\beta \in [0,1]}{\operatorname{argmin}} f\left(x^k + \beta(v^k - x^k)\right)$
3:     Set $x^{k+1} = \underset{\xi}{\operatorname{argmin}} f(x)$ { Alternating Step}
           where $\xi = \underset{\xi \in \{\mathbf{Z}_1, \ldots, \mathbf{Z}_d\}}{\operatorname{argmax}} \|\nabla_\xi f(y^k)\|_2^2$
4:     Set $v^{k+1} = v^k - a_{k+1}\nabla f(y^k)$ { Euclidean Proximal Step}
5: end for
```

Accelerated ALS algorithm

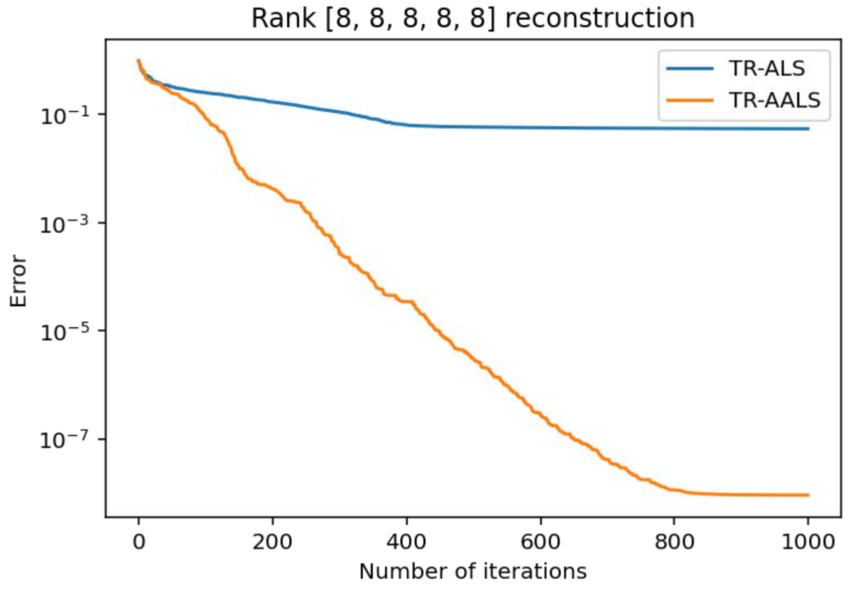


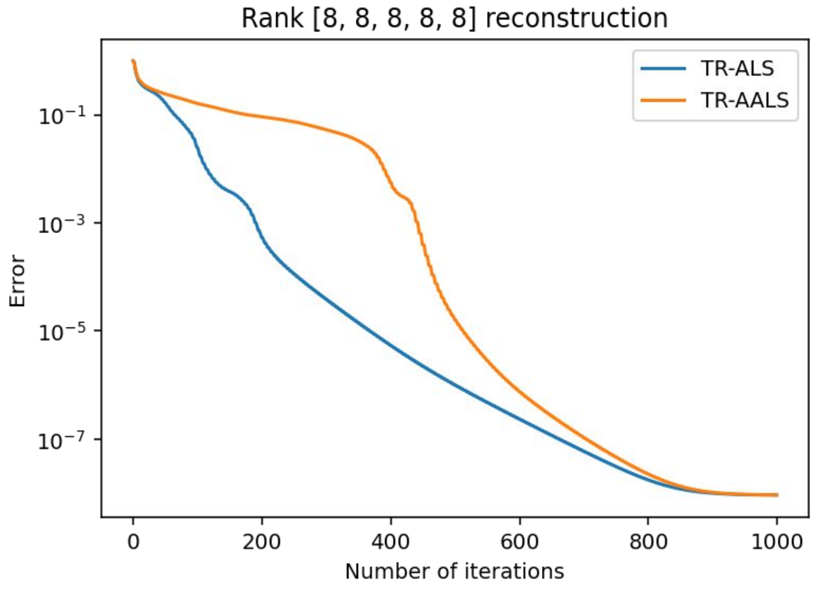
Algorithm 1 Accelerated TR-ALS (TR-AALS)

```
Input: Tensor $\mathcal{T}$, TR ranks $r_1, \dots, r_d$
Output: $x^k = [\mathbf{Z}_1, \dots, \mathbf{Z}_d]^k$
 1: $A_0 \leftarrow 0,\ v^0 \leftarrow x^0$
 2: Sample $x^0 = [\mathbf{Z}_1, \dots, \mathbf{Z}_d]^0$, where $\mathbf{Z}_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i} \sim \mathcal{N}(0, 1/n_i)$
 3: $f(x) = \sum_{i_1,\dots,i_d} \left(\mathcal{T}(i_1, i_2, \dots, i_d) - \mathrm{Tr}\{\mathbf{Z}_1(i_1)\mathbf{Z}_2(i_2)\cdots\mathbf{Z}_d(i_d)\}\right)^2$
 4: for $k \geq 0$ do
 5:     $\beta_k = \arg\min_{\beta \in [0,1]} f(v^k + \beta(x^k - v^k))$
 6:     $y^k = v^k + \beta^k(x^k - v^k)$
 7:     Choose $i_k$
 8:     $x^{k+1} = \arg\min_{\mathbf{Z}_{i_k}} f(y^k)$          ▷ keep all factors except $\mathbf{Z}_{i_k}$ fixed at $y^k$
 9:     Find $a_{k+1},\ A_{k+1} = A_k + a_{k+1}$ from
10:         $f(y^k) - \frac{a_{k+1}^2}{2A_{k+1}} \|\nabla f(y^k)\|_2^2 = f(x^{k+1})$          ▷ gradient w.r.t. all factors
11:     $v^{k+1} \leftarrow v^k - a_{k+1} \nabla f(y^k)$
12: end for
```

Results

- TR-AALS is robust to random choice of factors
- TR-AALS converges not worse than TR-ALS





Conclusion

- TT-ALS converges in one run
- TR-AALS usually converges faster

Thanks for your attention!