Tensor Train: description and applications

Martina Iannacito

Dipartimento di Matematica
Alma Mater Studiorum - Università di Bologna

Seminar for the Matrix and Tensor methods for Data Science course

December 12, 2024

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Table of Contents

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$, the possible decompositions are

Tucker decomposition

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}_1 \cdots \times_d \mathbf{U}_d \qquad \text{where}$$

- $\mathbf{U}_k$ are orthogonal matrices of size $(n_k \times r_k)$ s.t.

$$\mathbf{X}_{(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\mathsf{T} \qquad k = 1, \ldots, d$$

- $\mathcal{S}$ is a tensor of size $(r_1 \times \cdots \times r_d)$ s.t.

$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^\mathsf{T} \cdots \times_d \mathbf{U}_d^\mathsf{T};$$

- the $\mathbf{j} = (j_1, \ldots, j_d)$ element of $\mathcal{X}$ is

$$\mathcal{X}(\mathbf{j}) = \sum_{\mathbf{h}} \mathcal{S}(\mathbf{h}) \mathbf{U}_1(j_1, h_1) \cdots \mathbf{U}_d(j_d, h_d);$$

where $\mathbf{h} = (h_1, \ldots, h_d)$ for $h_k = 1, \ldots, r_k$ and $k = 1, \ldots, d$.

Preliminaries – I

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$, the possible decompositions are

Tucker decomposition

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}_1 \cdots \times_d \mathbf{U}_d \qquad \text{where}$$

- $\mathbf{U}_k$ are orthogonal matrices of size $(n_k \times r_k)$ s.t.

$$\mathbf{X}_{(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top \qquad k = 1, \ldots, d$$

- $\mathcal{S}$ is a tensor of size $(r_1 \times \cdots \times r_d)$ s.t.

$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^\top \cdots \times_d \mathbf{U}_d^\top;$$

- the $\mathbf{j} = (j_1, \ldots, j_d)$ element of $\mathcal{X}$ is

$$\mathcal{X}(\mathbf{j}) = \sum_{\mathbf{h}} \mathcal{S}(\mathbf{h}) \mathbf{U}_1(j_1, h_1) \cdots \mathbf{U}_d(j_d, h_d);$$

where $\mathbf{h} = (h_1, \ldots, h_d)$ for $h_k = 1, \ldots, r_k$ and $k = 1, \ldots, d$.

Preliminaries – I

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$, the possible decompositions are

Tucker decomposition

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}_1 \cdots \times_d \mathbf{U}_d \qquad \text{where}$$

- $\mathbf{U}_k$ are orthogonal matrices of size $(n_k \times r_k)$ s.t.

$$\mathbf{X}_{(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top \qquad k = 1, \ldots, d$$

- $\mathcal{S}$ is a tensor of size $(r_1 \times \cdots \times r_d)$ s.t.

$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^\top \cdots \times_d \mathbf{U}_d^\top;$$

- the $\mathbf{j} = (j_1, \ldots, j_d)$ element of $\mathcal{X}$ is

$$\mathcal{X}(\mathbf{j}) = \sum_{\mathbf{h}} \mathcal{S}(\mathbf{h}) \mathbf{U}_1(j_1, h_1) \cdots \mathbf{U}_d(j_d, h_d);$$

where $\mathbf{h} = (h_1, \ldots, h_d)$ for $h_k = 1, \ldots, r_k$ and $k = 1, \ldots, d$.

Preliminaries – I

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$, the possible decompositions are

Tucker decomposition

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}_1 \cdots \times_d \mathbf{U}_d \qquad \text{where}$$

- $\mathbf{U}_k$ are orthogonal matrices of size $(n_k \times r_k)$ s.t.
$$\mathbf{X}_{(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top \qquad k = 1, \ldots, d$$
- $\mathcal{S}$ is a tensor of size $(r_1 \times \cdots \times r_d)$ s.t.
$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^\top \cdots \times_d \mathbf{U}_d^\top;$$
- the $\mathbf{j} = (j_1, \ldots, j_d)$ element of $\mathcal{X}$ is
$$\mathcal{X}(\mathbf{j}) = \sum_{\mathbf{h}} \mathcal{S}(\mathbf{h}) \mathbf{U}_1(j_1, h_1) \cdots \mathbf{U}_d(j_d, h_d);$$
where $\mathbf{h} = (h_1, \ldots, h_d)$ for $h_k = 1, \ldots, r_k$ and $k = 1, \ldots, d$.

Preliminaries – I

Remark

Existing and widely used algorithms to compute the decomposition and approximation at given ML-rank are HOSVD, T-HOSVD and ST-HOSVD, HOOI.
See [Kolda et al. 2009; Vannieuwenhoven et al. 2012; De Lathauwer, De Moor, et al. 2000a; De Lathauwer, De Moor, et al. 2000b].

Preliminaries – II

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$, the possible decompositions are

Canonical Polyadic Decomposition

Assuming $\mathcal{X}$ has rank R, its CPD is

$$\mathcal{X} = \sum_{k=1}^{R} \mathbf{x}_k^{(1)} \otimes \cdots \otimes \mathbf{x}_k^{(d)}$$

where $\mathbf{x}_h^{(k)}$ is a vector of length n_k. The $\mathbf{j} = (j_1, \ldots, j_d)$ element of $\mathcal{X}$ is

$$\mathcal{X}(\mathbf{j}) = \sum_{k=1}^{R} \mathbf{x}_k^{(1)}(j_1) \cdots \mathbf{x}_k^{(d)}(j_d);$$

Remark

Existing and widely used algorithms to compute the decomposition and approximation at given a target rank are CPD-ALS and CPD-GEVD.
See [Kolda et al. 2009; De Lathauwer 2006].

Preliminaries – II

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$, the possible decompositions are

Canonical Polyadic Decomposition

Assuming $\mathcal{X}$ has rank R, its CPD is

$$\mathcal{X} = \sum_{k=1}^{R} \mathbf{x}_k^{(1)} \otimes \cdots \otimes \mathbf{x}_k^{(d)}$$

where $\mathbf{x}_h^{(k)}$ is a vector of length n_k. The $\mathbf{j} = (j_1, \ldots, j_d)$ element of $\mathcal{X}$ is

$$\mathcal{X}(\mathbf{j}) = \sum_{k=1}^{R} \mathbf{x}_k^{(1)}(j_1) \cdots \mathbf{x}_k^{(d)}(j_d);$$

Remark

Existing and widely used algorithms to compute the decomposition and approximation at given a target rank are CPD-ALS and CPD-GEVD.
See [Kolda et al. 2009; De Lathauwer 2006].

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$, the possible decompositions are

Canonical Polyadic Decomposition

Assuming $\mathcal{X}$ has rank R, its CPD is

$$\mathcal{X} = \sum_{k=1}^{R} \mathbf{x}_k^{(1)} \otimes \cdots \otimes \mathbf{x}_k^{(d)}$$

where $\mathbf{x}_h^{(k)}$ is a vector of length n_k. The $\mathbf{j} = (j_1, \ldots, j_d)$ element of $\mathcal{X}$ is

$$\mathcal{X}(\mathbf{j}) = \sum_{k=1}^{R} \mathbf{x}_k^{(1)}(j_1) \cdots \mathbf{x}_k^{(d)}(j_d);$$

Remark

Existing and widely used algorithms to compute the decomposition and approximation at given a target rank are CPD-ALS and CPD-GEVD.
See [Kolda et al. 2009; De Lathauwer 2006].

Pro and cons

- Tucker decomposition
  - + Always possible to compute it;
  - – Not unique ⇒ not a unique interpretation of data
  - – Storage cost depends **exponentially** on the order d, i.e. $\mathcal{O}(dnr + r^d)$
- CPD decomposition
  - + Unique* ⇒ unique interpretation of data
  - + Storage cost depends **linearly** on the order d, i.e. $\mathcal{O}(dnR)$
  - – NP-hard in general to find the true decomposition

assuming $n = \max\{n_k\}$ and $r = \max\{r_k\}$

Question

Does a mid-way decomposition technique exist?

Pro and cons

- Tucker decomposition
  - + Always possible to compute it;
  - − Not unique $\Rightarrow$ not a unique interpretation of data
  - − Storage cost depends **exponentially** on the order d, i.e. $\mathcal{O}(dnr + r^d)$
- CPD decomposition
  - + Unique* $\Rightarrow$ unique interpretation of data
  - + Storage cost depends **linearly** on the order d, i.e. $\mathcal{O}(dnR)$
  - − NP-hard in general to find the true decomposition

assuming $n = \max\{n_k\}$ and $r = \max\{r_k\}$

Question

Does a mid-way decomposition technique exist?

Pro and cons

- Tucker decomposition
  - \+ Always possible to compute it;
  - − Not unique ⇒ not a unique interpretation of data
  - − Storage cost depends **exponentially** on the order d, i.e. $\mathcal{O}(dnr + r^d)$
- CPD decomposition
  - \+ Unique* ⇒ unique interpretation of data
  - \+ Storage cost depends **linearly** on the order d, i.e. $\mathcal{O}(dnR)$
  - − NP-hard in general to find the true decomposition

assuming $n = \max\{n_k\}$ and $r = \max\{r_k\}$

Question

Does a mid-way decomposition technique exist?

Pro and cons

- Tucker decomposition
  - \+ Always possible to compute it;
  - − Not unique $\Rightarrow$ not a unique interpretation of data
  - − Storage cost depends **exponentially** on the order d, i.e. $\mathcal{O}(dnr + r^d)$
- CPD decomposition
  - \+ Unique* $\Rightarrow$ unique interpretation of data
  - \+ Storage cost depends **linearly** on the order d, i.e. $\mathcal{O}(dnR)$
  - − NP-hard in general to find the true decomposition

assuming $n = \max\{n_k\}$ and $r = \max\{r_k\}$

Question

Does a mid-way decomposition technique exist?

Yes!

Tensor-Train and Hierarchical Tucker

Table of Contents

Tensor-Train – I

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$,

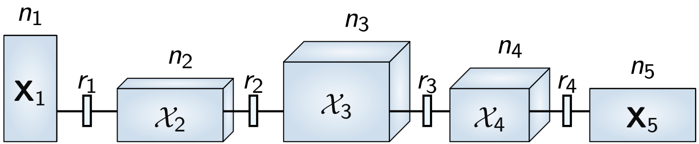
Definition [Oseledets 2011]

The **TT-decomposition** at **TT-rank** is $(r_1, \ldots, r_d)$ is given by

- $d-2$ tensors of order 3 and size $(r_{k-1} \times n_k \times r_k)$, $\mathcal{X}_k$ for $k = 2, \ldots, d-1$
- 2 matrices of size $(n_1 \times r_1)$ and $(r_{d-1} \times n_d)$, $\mathbf{X}_1$ and $\mathbf{X}_d$.

Remark

With an *abuse of notation*, we consider $\mathbf{X}_1$ and $\mathbf{X}_d$ as **tensors** of order 3 and size $(r_0 \times n_1 \times r_1)$ and $(r_{d-1} \times n_d \times r_d)$ with $r_0 = r_d = 1$. They are denoted by $\mathcal{X}_1$ and $\mathcal{X}_d$.



Tensor-Train – I

Given an $(n_1 \times \cdots \times n_d)$ tensor $\mathcal{X}$,

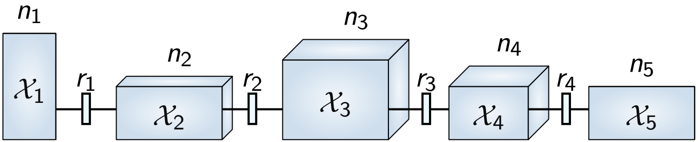
Definition [Oseledets 2011]

The **TT-decomposition** at **TT-rank** is $(r_1, \ldots, r_d)$ is given by

- $d-2$ tensors of order 3 and size $(r_{k-1} \times n_k \times r_k)$, $\mathcal{X}_k$ for $k = 2, \ldots, d-1$
- 2 matrices of size $(n_1 \times r_1)$ and $(r_{d-1} \times n_d)$, $\mathbf{X}_1$ and $\mathbf{X}_d$.

Remark

With an *abuse of notation*, we consider $\mathbf{X}_1$ and $\mathbf{X}_d$ as **tensors** of order 3 and size $(r_0 \times n_1 \times r_1)$ and $(r_{d-1} \times n_d \times r_d)$ with $r_0 = r_d = 1$. They are denoted by $\mathcal{X}_1$ and $\mathcal{X}_d$.



Tensor-Train – II

Definition

The tensors $\mathcal{X}_k$ are the **TT-cores** of $\mathcal{X}$.
The j_kth slice w.r.t. mode-2 of the kth core, $\mathcal{X}_k(:, j_k, :)$, is an $(r_{k-1} \times r_k)$ matrix denoted by $\mathbf{X}_k(j_k)$. The TT-representation of a tensor is called **TT-vector**.

Given $\mathbf{j} = (j_1, \ldots, j_d)$, the $\mathbf{j}$th element of $\mathcal{X}$ is

$$\mathcal{X}(\mathbf{j}) = \sum_{k=1}^{d} \sum_{h_k=1}^{r_k} \mathcal{X}_1(1, j_1, h_1)\mathcal{X}_2(h_1, j_2, h_2) \cdots \mathcal{X}_d(h_{d-1}, j_d, 1)$$

$$= \mathbf{X}_1(j_1)\mathbf{X}_2(j_2) \cdots \mathbf{X}_d(j_d).$$

Compactly, $\mathcal{X} = \mathcal{X}_1\mathcal{X}_2 \cdots \mathcal{X}_d$, with the **contraction** (i.e., sum over an index) as underlying operation.

Tensor-Train – II

Definition

The tensors $\mathcal{X}_k$ are the **TT-cores** of $\mathcal{X}$.
The j_kth slice w.r.t. mode-2 of the kth core, $\mathcal{X}_k(:, j_k, :)$, is an $(r_{k-1} \times r_k)$ matrix denoted by $\mathbf{X}_k(j_k)$. The TT-representation of a tensor is called **TT-vector**.

Given $\mathbf{j} = (j_1, \ldots, j_d)$, the $\mathbf{j}$th element of $\mathcal{X}$ is

$$\begin{aligned}\mathcal{X}(\mathbf{j}) &= \sum_{k=1}^{d} \sum_{h_k=1}^{r_k} \mathcal{X}_1(1, j_1, h_1)\mathcal{X}_2(h_1, j_2, h_2) \cdots \mathcal{X}_d(h_{d-1}, j_d, 1) \\ &= \mathbf{X}_1(j_1)\mathbf{X}_2(j_2) \cdots \mathbf{X}_d(j_d).\end{aligned}$$

Compactly, $\mathcal{X} = \mathcal{X}_1\mathcal{X}_2 \cdots \mathcal{X}_d$, with the **contraction** (i.e., sum over an index) as underlying operation.

Tensor-Train – II

Definition

The tensors $\mathcal{X}_k$ are the **TT-cores** of $\mathcal{X}$.
The j_kth slice w.r.t. mode-2 of the kth core, $\mathcal{X}_k(:, j_k, :)$, is an $(r_{k-1} \times r_k)$ matrix denoted by $\mathbf{X}_k(j_k)$. The TT-representation of a tensor is called **TT-vector**.

Given $\mathbf{j} = (j_1, \ldots, j_d)$, the $\mathbf{j}$th element of $\mathcal{X}$ is

$$\mathcal{X}(\mathbf{j}) = \sum_{k=1}^{d} \sum_{h_k=1}^{r_k} \mathcal{X}_1(1, j_1, h_1)\mathcal{X}_2(h_1, j_2, h_2) \cdots \mathcal{X}_d(h_{d-1}, j_d, 1)$$

$$= \mathbf{X}_1(j_1)\mathbf{X}_2(j_2) \cdots \mathbf{X}_d(j_d).$$

Compactly, $\mathcal{X} = \mathcal{X}_1\mathcal{X}_2 \cdots \mathcal{X}_d$, with the **contraction** (i.e., sum over an index) as underlying operation.

Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{m_1 \times \cdots \times m_d}$ be a multi-linear operator. By fixing a basis for both vector space, $\boldsymbol{\mathcal{A}}$ can be associated with a tensor $\mathcal{A}$ of order $2d$ and size $((m_1 \times n_1) \times \cdots \times (m_d \times n_d))$.

Definition [Oseledets 2011]

The **TT-decomposition** at **TT-rank** is $(r_1, \ldots, r_d)$ of $\mathcal{A}$ is given by

- $d-2$ tensors of order-4 and size $(r_{k-1} \times m_k \times n_k \times r_3)$, $\mathcal{A}_k$ for $k = 2, \ldots, d-1$
- 2 tensors of order-3 and size $(m_1 \times n_1 \times r_1)$ and $(r_{d-1} \times m_d \times n_d)$, $\mathcal{A}_1$ and $\mathcal{A}_d$.

Remark

$\mathcal{A}_1$ and $\mathcal{A}_d$ can be seen as order-4 **tensors** of size $(r_0 \times m_1 \times n_1 \times r_1)$ and $(r_{d-1} \times m_d \times n_d \times r_d)$ with $r_0 = r_d = 1$.

TT and operators – I

Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{m_1 \times \cdots \times m_d}$ be a multi-linear operator. By fixing a basis for both vector space, $\boldsymbol{\mathcal{A}}$ can be associated with a tensor $\mathcal{A}$ of order $2d$ and size $((m_1 \times n_1) \times \cdots \times (m_d \times n_d))$.

Definition [Oseledets 2011]

The **TT-decomposition** at **TT-rank** is $(r_1, \dots, r_d)$ of $\mathcal{A}$ is given by

- $d-2$ tensors of order-4 and size $(r_{k-1} \times m_k \times n_k \times r_3)$, $\mathcal{A}_k$ for $k = 2, \dots, d-1$
- 2 tensors of order-3 and size $(m_1 \times n_1 \times r_1)$ and $(r_{d-1} \times m_d \times n_d)$, $\mathcal{A}_1$ and $\mathcal{A}_d$.

Remark

$\mathcal{A}_1$ and $\mathcal{A}_d$ can be seen as order-4 **tensors** of size $(r_0 \times m_1 \times n_1 \times r_1)$ and $(r_{d-1} \times m_d \times n_d \times r_d)$ with $r_0 = r_d = 1$.

Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{m_1 \times \cdots \times m_d}$ be a multi-linear operator. By fixing a basis for both vector space, $\boldsymbol{\mathcal{A}}$ can be associated with a tensor $\mathcal{A}$ of order $2d$ and size $((m_1 \times n_1) \times \cdots \times (m_d \times n_d))$.

Definition [Oseledets 2011]

The **TT-decomposition** at **TT-rank** is $(r_1, \ldots, r_d)$ of $\mathcal{A}$ is given by

- $d-2$ tensors of order-4 and size $(r_{k-1} \times m_k \times n_k \times r_3)$, $\mathcal{A}_k$ for $k = 2, \ldots, d-1$
- 2 tensors of order-3 and size $(m_1 \times n_1 \times r_1)$ and $(r_{d-1} \times m_d \times n_d)$, $\mathcal{A}_1$ and $\mathcal{A}_d$.

Remark

$\mathcal{A}_1$ and $\mathcal{A}_d$ can be seen as order-4 **tensors** of size $(r_0 \times m_1 \times n_1 \times r_1)$ and $(r_{d-1} \times m_d \times n_d \times r_d)$ with $r_0 = r_d = 1$.

TT and operators – II

Definition

The tensors $\mathcal{A}_k$ are the **TT-cores** of $\mathcal{A}$.
The (i_k, j_k)th slice w.r.t. modes-(2, 3) of the kth core, $\mathcal{A}_k(:, i_k, j_k, :)$, is an $(r_{k-1} \times r_k)$ matrix denoted by $\mathbf{A}_k(i_k, j_k)$. The TT-representation of a multilinear operator is called **TT-matrix**

Given $\mathbf{i} = (i_1, \dots, i_d) \in \mathbb{N}^{m_1 \times \cdots m_d}$ and $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{N}^{n_1 \times \cdots n_d}$, the $(\mathbf{i}, \mathbf{j})$th element of $\mathcal{A}$ is

$$\mathcal{A}(\mathbf{i}, \mathbf{j}) = \sum_{k=1}^{d} \sum_{h_k=1}^{r_k} \mathcal{A}_1(1, i_1, j_1, h_1)\mathcal{A}_2(h_1, i_2, j_2, h_2) \cdots \mathcal{A}_d(h_{d-1}, i_d, j_d, 1)$$

$$= \mathbf{A}_1(i_1, j_1)\mathbf{A}_2(i_2, j_2) \cdots \mathbf{A}_d(i_d, j_d).$$

Compactly, $\mathcal{A} = \mathcal{A}_1\mathcal{A}_2 \cdots \mathcal{A}_d$, with the contraction as underlying operation.

TT and operators – II

Definition

The tensors $\mathcal{A}_k$ are the **TT-cores** of $\mathcal{A}$.
The (i_k, j_k)th slice w.r.t. modes-(2, 3) of the kth core, $\mathcal{A}_k(:, i_k, j_k, :)$, is an $(r_{k-1} \times r_k)$ matrix denoted by $\mathbf{A}_k(i_k, j_k)$. The TT-representation of a multilinear operator is called **TT-matrix**

Given $\mathbf{i} = (i_1, \dots, i_d) \in \mathbb{N}^{m_1 \times \cdots m_d}$ and $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{N}^{n_1 \times \cdots n_d}$, the $(\mathbf{i}, \mathbf{j})$th element of $\mathcal{A}$ is

$$
\begin{aligned}
\mathcal{A}(\mathbf{i}, \mathbf{j}) &= \sum_{k=1}^{d} \sum_{h_k=1}^{r_k} \mathcal{A}_1(1, i_1, j_1, h_1)\mathcal{A}_2(h_1, i_2, j_2, h_2) \cdots \mathcal{A}_d(h_{d-1}, i_d, j_d, 1) \\
&= \mathbf{A}_1(i_1, j_1)\mathbf{A}_2(i_2, j_2) \cdots \mathbf{A}_d(i_d, j_d).
\end{aligned}
$$

Compactly, $\mathcal{A} = \mathcal{A}_1\mathcal{A}_2 \cdots \mathcal{A}_d$, with the contraction as underlying operation.

Definition

The tensors $\mathcal{A}_k$ are the **TT-cores** of $\mathcal{A}$.
The (i_k, j_k)th slice w.r.t. modes-(2, 3) of the kth core, $\mathcal{A}_k(:, i_k, j_k, :)$, is an $(r_{k-1} \times r_k)$ matrix denoted by $\mathbf{A}_k(i_k, j_k)$. The TT-representation of a multilinear operator is called **TT-matrix**

Given $\mathbf{i} = (i_1, \dots, i_d) \in \mathbb{N}^{m_1 \times \cdots m_d}$ and $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{N}^{n_1 \times \cdots n_d}$, the $(\mathbf{i}, \mathbf{j})$th element of $\mathcal{A}$ is

$$\mathcal{A}(\mathbf{i}, \mathbf{j}) = \sum_{k=1}^{d} \sum_{h_k=1}^{r_k} \mathcal{A}_1(1, i_1, j_1, h_1)\mathcal{A}_2(h_1, i_2, j_2, h_2) \cdots \mathcal{A}_d(h_{d-1}, i_d, j_d, 1)$$

$$= \mathbf{A}_1(i_1, j_1)\mathbf{A}_2(i_2, j_2) \cdots \mathbf{A}_d(i_d, j_d).$$

Compactly, $\mathcal{A} = \mathcal{A}_1\mathcal{A}_2 \cdots \mathcal{A}_d$, with the contraction as underlying operation.

Matrix Product States [Östlund et al. 1995; Klümper et al. 1992; Fannes et al. 1992; Vidal 2003]

The TT-format originated from the **quantum physics** field, where it is called **Matrix Product States** (MPS).

The quantum physicists rely on a visual notation, **Tensor Network**, to describe MPS/TT objects (and not only).

In the **tensor network**, there is

- a node with the *label* of the object
- one (or more) **output edge** with the *dimension* of the space to which it belong

Remark

We just draw the vector $\mathbf{x}$ of length n

Matrix Product States [Östlund et al. 1995; Klümper et al. 1992; Fannes et al. 1992; Vidal 2003]

The TT-format originated from the **quantum physics** field, where it is called **Matrix Product States** (MPS).

The quantum physicists rely on a visual notation, **Tensor Network**, to describe MPS/TT objects (and not only).

In the **tensor network**, there is

- a node with the *label* of the object
- one (or more) **output edge** with the *dimension* of the space to which it belong

Remark

We just draw the vector $\mathbf{x}$ of length n

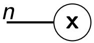
Matrix Product States [Östlund et al. 1995; Klümper et al. 1992; Fannes et al. 1992; Vidal 2003]

The TT-format originated from the **quantum physics** field, where it is called **Matrix Product States** (MPS).

The quantum physicists rely on a visual notation, **Tensor Network**, to describe MPS/TT objects (and not only).

In the **tensor network**, there is

- a node with the *label* of the object
- one (or more) **output edge** with the *dimension* of the space to which it belong



Remark

We just draw the vector $\mathbf{x}$ of length n

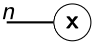
Matrix Product States [Östlund et al. 1995; Klümper et al. 1992; Fannes et al. 1992; Vidal 2003]

The TT-format originated from the **quantum physics** field, where it is called **Matrix Product States** (MPS).

The quantum physicists rely on a visual notation, **Tensor Network**, to describe MPS/TT objects (and not only).

In the **tensor network**, there is

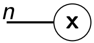
- a node with the *label* of the object
- one (or more) **output edge** with the *dimension* of the space to which it belong



Remark

We just draw the vector $\mathbf{x}$ of length n

- the length n vector $\mathbf{x}$ is depicted as



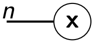
- the $(m \times n)$ matrix $\mathbf{A}$ is depicted as

- the $(m \times n)$ orthogonal matrix $\mathbf{Q}$ is depicted as

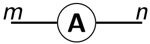
- the $(m \times n \times p)$ tensor $\mathcal{X}$ is depicted as

Example of TN – I

- the length n vector $\mathbf{x}$ is depicted as



- the $(m \times n)$ matrix $\mathbf{A}$ is depicted as

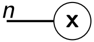


- the $(m \times n)$ orthogonal matrix $\mathbf{Q}$ is depicted as

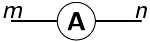
- the $(m \times n \times p)$ tensor $\mathcal{X}$ is depicted as

Example of TN – I

- the length n vector $\mathbf{x}$ is depicted as



- the $(m \times n)$ matrix $\mathbf{A}$ is depicted as

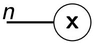


- the $(m \times n)$ orthogonal matrix $\mathbf{Q}$ is depicted as

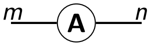
- the $(m \times n \times p)$ tensor $\mathcal{X}$ is depicted as

Example of TN – I

- the length n vector $\mathbf{x}$ is depicted as

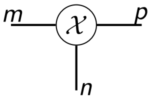


- the $(m \times n)$ matrix $\mathbf{A}$ is depicted as



- the $(m \times n)$ orthogonal matrix $\mathbf{Q}$ is depicted as

- the $(m \times n \times p)$ tensor $\mathcal{X}$ is depicted as



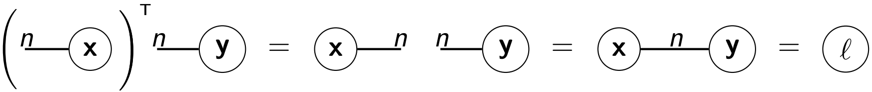
Contraction with TN

TNs allow the representation of standard linear alegbra operations and decomposition.

The inner product between two length n vectors, $\mathbf{x}$ and $\mathbf{y}$

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{\mathsf{T}}\mathbf{y} = \sum_{k=1}^{n} \mathbf{x}(k)\mathbf{y}(k) = \ell$$

is depicted joining by an inner edge the node of $\mathbf{x}$ and $\mathbf{y}$, i.e.



Remark

A contraction is depicted by an **inner edge** joining two nodes. Example of contractions: the matrix-vector product, the matrix-tensor product, the contraction between tensors...

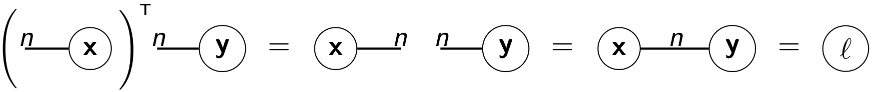
Contraction with TN

TNs allow the representation of standard linear alegbra operations and decomposition.

The inner product between two length n vectors, $\mathbf{x}$ and $\mathbf{y}$

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{\mathsf{T}}\mathbf{y} = \sum_{k=1}^{n} \mathbf{x}(k)\mathbf{y}(k) = \ell$$

is depicted joining by an inner edge the node of $\mathbf{x}$ and $\mathbf{y}$, i.e.

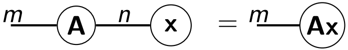


Remark

A contraction is depicted by an **inner edge** joining two nodes. Example of contractions: the matrix-vector product, the matrix-tensor product, the contraction between tensors...

Examples of TN contraction – I

- the matrix-vector product **Ax** is depicted as

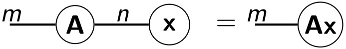


- the QR-decomposition of **A** is depicted as

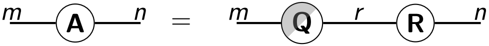
- the SVD decomposition of **A** is depicted as

Examples of TN contraction – I

- the matrix-vector product **Ax** is depicted as



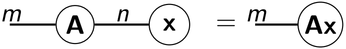
- the QR-decomposition of **A** is depicted as



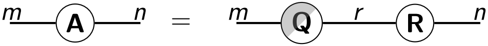
- the SVD decomposition of **A** is depicted as

Examples of TN contraction – I

- the matrix-vector product **Ax** is depicted as



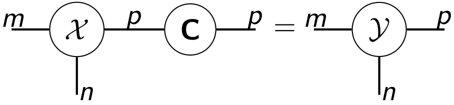
- the QR-decomposition of **A** is depicted as



- the SVD decomposition of **A** is depicted as

Examples of TN contraction – II

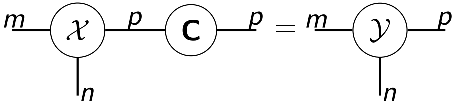
- the matrix-tensor product $\mathcal{X} \times_3 \mathbf{C}$, resulting in $\mathcal{Y}$, is depicted as



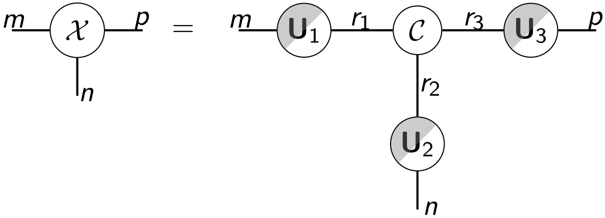
- the Tucker decomposition of $\mathcal{X}$ is depicted as

Examples of TN contraction – II

- the matrix-tensor product $\mathcal{X} \times_3 \mathbf{C}$, resulting in $\mathcal{Y}$, is depicted as



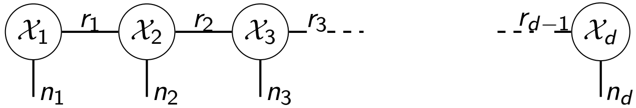
- the Tucker decomposition of $\mathcal{X}$ is depicted as



TT and TN

The TN-representation of

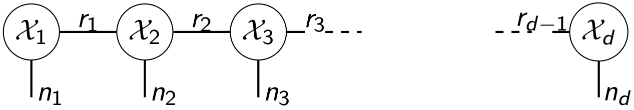
- TT-vector $\mathcal{X}$ is



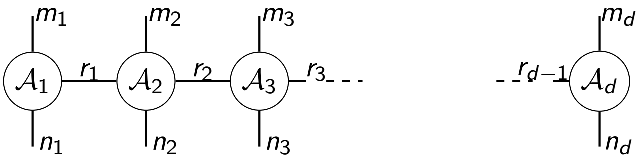
- TT-matrix $\mathcal{A}$ is

The TN-representation of

- TT-vector $\mathcal{X}$ is



- TT-matrix $\mathcal{A}$ is



Given two TT-vectors, $\mathcal{X}$ and $\mathcal{Y}$, of the same size, it can be proven that

Proposition [Oseledets 2011]

The sum TT-vector, $\mathcal{Z}(\mathbf{j}) = \mathcal{X}(\mathbf{j}) + \mathcal{Y}(\mathbf{j})$, is such that

$$\mathbf{Z}_1(j_1) = \begin{bmatrix} \mathbf{X}_1(j_1) & \mathbf{Y}_1(j_1) \end{bmatrix} \quad \text{and} \quad \mathbf{Z}_d(j_d) = \begin{bmatrix} \mathbf{X}_d(j_d) \\ \mathbf{Y}_d(j_d) \end{bmatrix}$$

$$\mathbf{Z}_k(j_k) = \begin{bmatrix} \mathbf{X}_k(j_k) & \\ & \mathbf{Y}_k(j_k) \end{bmatrix}$$

where $\mathbf{j} = (j_1, \ldots, j_d)$, $j_k = 1, \ldots, n_k$ and $k = 1, \ldots, d$.

Let $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{N}^{n_1 \times \cdots \times n_d}$, the $\mathbf{j}$th element of $\mathcal{Z}$ is

$$\mathcal{Z}(\mathbf{j}) = \mathbf{Z}_1(j_1)\mathbf{Z}_2(j_2)\cdots\mathbf{Z}_{d-1}(j_{d-1})\mathbf{Z}_d(j_d)$$

$$= \begin{bmatrix} \mathbf{X}_1(j_1) & \mathbf{Y}_1(j_1) \end{bmatrix} \begin{bmatrix} \mathbf{X}_2(j_2) & \\ & \mathbf{Y}_2(j_2) \end{bmatrix} \cdots \begin{bmatrix} \mathbf{X}_{d-1}(j_{d-1}) & \\ & \mathbf{Y}_{d-1}(j_{d-1}) \end{bmatrix} \begin{bmatrix} \mathbf{X}_d(j_d) \\ \mathbf{Y}_d(j_d) \end{bmatrix}.$$

By directly multiplying the blocks, we obtain

$$\mathcal{Z}(\mathbf{j}) = \mathbf{X}_1(j_1)\mathbf{X}_2(j_2)\cdots\mathbf{X}_d(j_d) + \mathbf{Y}_1(j_1)\mathbf{Y}_2(j_2)\cdots\mathbf{Y}_d(j_d),$$

that is the thesis.

Remark

The TT-rank of $\mathcal{Z}$ is the sum of the TT-ranks of $\mathcal{X}$ and $\mathcal{Y}$. The linear combination of two TT-elements can be obtained only by memory manipulation.

Proof – I

Let $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{N}^{n_1 \times \cdots \times n_d}$, the $\mathbf{j}$th element of $\mathcal{Z}$ is

$$\mathcal{Z}(\mathbf{j}) = \mathbf{Z}_1(j_1)\mathbf{Z}_2(j_2)\cdots\mathbf{Z}_{d-1}(j_{d-1})\mathbf{Z}_d(j_d)$$
$$= \begin{bmatrix} \mathbf{X}_1(j_1) & \mathbf{Y}_1(j_1) \end{bmatrix} \begin{bmatrix} \mathbf{X}_2(j_2) & \\ & \mathbf{Y}_2(j_2) \end{bmatrix} \cdots \begin{bmatrix} \mathbf{X}_{d-1}(j_{d-1}) & \\ & \mathbf{Y}_{d-1}(j_{d-1}) \end{bmatrix} \begin{bmatrix} \mathbf{X}_d(j_d) \\ \mathbf{Y}_d(j_d) \end{bmatrix}.$$

By directly multiplying the blocks, we obtain

$$\mathcal{Z}(\mathbf{j}) = \mathbf{X}_1(j_1)\mathbf{X}_2(j_2)\cdots\mathbf{X}_d(j_d) + \mathbf{Y}_1(j_1)\mathbf{Y}_2(j_2)\cdots\mathbf{Y}_d(j_d),$$

that is the thesis.

Remark

The TT-rank of $\mathcal{Z}$ is the sum of the TT-ranks of $\mathcal{X}$ and $\mathcal{Y}$. The linear combination of two TT-elements can be obtained only by memory manipulation.

Proof – I

Let $\mathbf{j} = (j_1, \ldots, j_d) \in \mathbb{N}^{n_1 \times \cdots \times n_d}$, the $\mathbf{j}$th element of $\mathcal{Z}$ is

$$\begin{aligned}\mathcal{Z}(\mathbf{j}) &= \mathbf{Z}_1(j_1)\mathbf{Z}_2(j_2)\cdots\mathbf{Z}_{d-1}(j_{d-1})\mathbf{Z}_d(j_d)\\ &= \begin{bmatrix}\mathbf{X}_1(j_1) & \mathbf{Y}_1(j_1)\end{bmatrix}\begin{bmatrix}\mathbf{X}_2(j_2) & \\ & \mathbf{Y}_2(j_2)\end{bmatrix}\cdots\begin{bmatrix}\mathbf{X}_{d-1}(j_{d-1}) & \\ & \mathbf{Y}_{d-1}(j_{d-1})\end{bmatrix}\begin{bmatrix}\mathbf{X}_d(j_d)\\ \mathbf{Y}_d(j_d)\end{bmatrix}.\end{aligned}$$

By directly multiplying the blocks, we obtain

$$\mathcal{Z}(\mathbf{j}) = \mathbf{X}_1(j_1)\mathbf{X}_2(j_2)\cdots\mathbf{X}_d(j_d) + \mathbf{Y}_1(j_1)\mathbf{Y}_2(j_2)\cdots\mathbf{Y}_d(j_d),$$

that is the thesis.

Remark

The TT-rank of $\mathcal{Z}$ is the sum of the TT-ranks of $\mathcal{X}$ and $\mathcal{Y}$. The linear combination of two TT-elements can be obtained only by memory manipulation.

Proof – I

Let $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{N}^{n_1 \times \cdots \times n_d}$, the $\mathbf{j}$th element of $\mathcal{Z}$ is

$$\begin{aligned}\mathcal{Z}(\mathbf{j}) &= \mathbf{Z}_1(j_1)\mathbf{Z}_2(j_2)\cdots\mathbf{Z}_{d-1}(j_{d-1})\mathbf{Z}_d(j_d)\\ &= \begin{bmatrix}\mathbf{X}_1(j_1) & \mathbf{Y}_1(j_1)\end{bmatrix}\begin{bmatrix}\mathbf{X}_2(j_2) & \\ & \mathbf{Y}_2(j_2)\end{bmatrix}\cdots\begin{bmatrix}\mathbf{X}_{d-1}(j_{d-1}) & \\ & \mathbf{Y}_{d-1}(j_{d-1})\end{bmatrix}\begin{bmatrix}\mathbf{X}_d(j_d)\\ \mathbf{Y}_d(j_d)\end{bmatrix}.\end{aligned}$$

By directly multiplying the blocks, we obtain

$$\mathcal{Z}(\mathbf{j}) = \mathbf{X}_1(j_1)\mathbf{X}_2(j_2)\cdots\mathbf{X}_d(j_d) + \mathbf{Y}_1(j_1)\mathbf{Y}_2(j_2)\cdots\mathbf{Y}_d(j_d),$$

that is the thesis.

Remark

The TT-rank of $\mathcal{Z}$ is the sum of the TT-ranks of $\mathcal{X}$ and $\mathcal{Y}$. The linear combination of two TT-elements can be obtained only by memory manipulation.

TT-arithmetic – II

Given a TT-vectors, $\mathcal{X}$, and a rank-1 tensor, $\mathcal{Y}$, of the same size, it can be proven that

Lemma [Oseledets 2011]

Their inner product is such that

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \mathbf{z}_1^\top (\mathbf{Z}_2 \cdots \mathbf{Z}_{d-1} \mathbf{z}_d),$$

where

$$\mathbf{z}_1 = \sum_{j_1=1}^{n_1} \mathbf{y}_1(j_1) \mathbf{X}_1(j_1), \qquad \mathbf{z}_d = \sum_{j_d=1}^{n_d} \mathbf{y}_d(j_d) \mathbf{X}_d(j_d), \qquad \text{and} \qquad \mathbf{Z}_k = \sum_{j_k=1}^{n_k} \mathbf{y}_k(j_k) \mathbf{X}_k(j_k)$$

for $k = 2, \ldots, d-1$.

Proof – II

The inner product between two tensors is $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{X}(\mathbf{j})\mathcal{Y}(\mathbf{j})$. As $\mathcal{Y}$ is a rank-1 tensor, it gets

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{X}(\mathbf{j})(\mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_d)(\mathbf{j}) = \sum_{k=1}^{d} \sum_{j_k=1}^{n_k} \mathbf{X}_1(j_1) \cdots \mathbf{X}_d(j_d)\mathbf{y}_1(j_1) \cdots \mathbf{y}_d(j_d)$$

$$= \left(\sum_{j_1=1}^{n_1} \mathbf{y}_1(j_1)\mathbf{X}_1(j_1) \right) \cdots \left(\sum_{j_d=1}^{n_d} \mathbf{y}_d(j_d)\mathbf{X}_d(j_d) \right).$$

We recognize the expressions of $\mathbf{z}_h$ and $\mathbf{Z}_k$ for $h = 1, d$ and $k = 2, \ldots, d-1$, that is the thesis, i. e.,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \langle \mathcal{X}, \mathcal{Y} \rangle = \mathbf{z}_1^T(\mathbf{Z}_2 \cdots \mathbf{Z}_{d-1}\mathbf{z}_d).$$

Proof – II

The inner product between two tensors is $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{X}(\mathbf{j})\mathcal{Y}(\mathbf{j})$. As $\mathcal{Y}$ is a rank-1 tensor, it gets

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{X}(\mathbf{j})(\mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_d)(\mathbf{j}) = \sum_{k=1}^{d} \sum_{j_k=1}^{n_k} \mathbf{X}_1(j_1) \cdots \mathbf{X}_d(j_d)\mathbf{y}_1(j_1) \cdots \mathbf{y}_d(j_d)$$

$$= \Big(\sum_{j_1=1}^{n_1} \mathbf{y}_1(j_1)\mathbf{X}_1(j_1) \Big) \cdots \Big(\sum_{j_d=1}^{n_d} \mathbf{y}_d(j_d)\mathbf{X}_d(j_d) \Big).$$

We recognize the expressions of $\mathbf{z}_h$ and $\mathbf{Z}_k$ for $h = 1, d$ and $k = 2, \ldots, d-1$, that is the thesis, i. e.,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \langle \mathcal{X}, \mathcal{Y} \rangle = \mathbf{z}_1^{\top}(\mathbf{Z}_2 \cdots \mathbf{Z}_{d-1}\mathbf{z}_d).$$

Proof – II

The inner product between two tensors is $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{X}(\mathbf{j})\mathcal{Y}(\mathbf{j})$. As $\mathcal{Y}$ is a rank-1 tensor, it gets

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{X}(\mathbf{j})(\mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_d)(\mathbf{j}) = \sum_{k=1}^{d} \sum_{j_k=1}^{n_k} \mathbf{X}_1(j_1) \cdots \mathbf{X}_d(j_d) \mathbf{y}_1(j_1) \cdots \mathbf{y}_d(j_d)$$
$$= \left(\sum_{j_1=1}^{n_1} \mathbf{y}_1(j_1)\mathbf{X}_1(j_1) \right) \cdots \left(\sum_{j_d=1}^{n_d} \mathbf{y}_d(j_d)\mathbf{X}_d(j_d) \right).$$

We recognize the expressions of $\mathbf{z}_h$ and $\mathbf{Z}_k$ for $h = 1, d$ and $k = 2, \ldots, d-1$, that is the thesis, i. e.,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \langle \mathcal{X}, \mathcal{Y} \rangle = \mathbf{z}_1^{\top}(\mathbf{Z}_2 \cdots \mathbf{Z}_{d-1}\mathbf{z}_d).$$

Proof – II

The inner product between two tensors is $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{X}(\mathbf{j})\mathcal{Y}(\mathbf{j})$. As $\mathcal{Y}$ is a rank-1 tensor, it gets

$$\begin{aligned}\langle \mathcal{X}, \mathcal{Y} \rangle &= \sum_{\mathbf{j}} \mathcal{X}(\mathbf{j})(\mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_d)(\mathbf{j}) = \sum_{k=1}^{d} \sum_{j_k=1}^{n_k} \mathbf{X}_1(j_1) \cdots \mathbf{X}_d(j_d) \mathbf{y}_1(j_1) \cdots \mathbf{y}_d(j_d) \\ &= \left(\sum_{j_1=1}^{n_1} \mathbf{y}_1(j_1) \mathbf{X}_1(j_1) \right) \cdots \left(\sum_{j_d=1}^{n_d} \mathbf{y}_d(j_d) \mathbf{X}_d(j_d) \right).\end{aligned}$$

We recognize the expressions of $\mathbf{z}_h$ and $\mathbf{Z}_k$ for $h = 1, d$ and $k = 2, \ldots, d-1$, that is the thesis, i. e.,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \langle \mathcal{X}, \mathcal{Y} \rangle = \mathbf{z}_1^{\top}(\mathbf{Z}_2 \cdots \mathbf{Z}_{d-1} \mathbf{z}_d).$$

TT-arithmetic – III

Given two TT-vectors, $\mathcal{X}$ and $\mathcal{Y}$, of the same size, it can be proven that

Proposition [Oseledets 2011]

define $\mathcal{Z}(\mathbf{j}) = \mathcal{X}(\mathbf{j})\mathcal{Y}(\mathbf{j})$. Then

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{Z}(\mathbf{j}) = \sum_{\mathbf{j}} \mathbf{Z}_1(j_1) \cdots \mathbf{Z}_d(j_d),$$

where

$$\mathbf{Z}_k(j_k) = \mathbf{X}_k(j_k) \otimes_{\mathrm{K}} \mathbf{Y}_k(j_k) \qquad \text{for} \qquad k = 1, \ldots, d.$$

Remark

The TT-cores of $\mathcal{Z}$ are equal to the product of the TT-ranks of $\mathcal{X}$ and $\mathcal{Y}$. The TT-cores of $\mathcal{Z}$ can be computed by *nd* Kronecker products.

TT-arithmetic – III

Given two TT-vectors, $\mathcal{X}$ and $\mathcal{Y}$, of the same size, it can be proven that

Proposition [Oseledets 2011]

define $\mathcal{Z}(\mathbf{j}) = \mathcal{X}(\mathbf{j})\mathcal{Y}(\mathbf{j})$. Then

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{j}} \mathcal{Z}(\mathbf{j}) = \sum_{\mathbf{j}} \mathbf{Z}_1(j_1) \cdots \mathbf{Z}_d(j_d),$$

where

$$\mathbf{Z}_k(j_k) = \mathbf{X}_k(j_k) \otimes_{\mathrm{K}} \mathbf{Y}_k(j_k) \qquad \text{for} \qquad k = 1, \dots, d.$$

Remark

The TT-cores of $\mathcal{Z}$ are equal to the product of the TT-ranks of $\mathcal{X}$ and $\mathcal{Y}$. The TT-cores of $\mathcal{Z}$ can be computed by *nd* Kronecker products.

Proof – III

Let $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{N}^{n_1 \times \cdots \times n_d}$, the $\mathbf{j}$th element of $\mathcal{Z}$ is

$$\mathcal{Z}(\mathbf{j}) = \mathcal{X}(\mathbf{j})\mathcal{Y}(\mathbf{j}) = \Big(\mathbf{X}_1(j_1)\cdots\mathbf{X}_d(j_d)\Big)\Big(\mathbf{Y}_1(j_1)\cdots\mathbf{Y}_d(j_d)\Big).$$

Notice now that $\mathbf{X}_d(j_d)$ has size $(r_{d_1} \times 1)$ and $\mathbf{Y}_1(j_1)$ has size $(1 \times s_{d_1})$. Thus, we can make explicit in the previous equation the Kronecker product, writing

$$\mathcal{Z}(\mathbf{j}) = \Big(\mathbf{X}_1(j_1)\cdots\mathbf{X}_d(j_d)\Big) \otimes_K \Big(\mathbf{Y}_1(j_1)\cdots\mathbf{Y}_d(j_d)\Big).$$

Recalling the mixed-product property of the Kronecker product, we can rewrite the previous equation as

$$\mathcal{Z}(\mathbf{j}) = \Big(\mathbf{X}_1(j_1) \otimes_{\mathrm{K}} \mathbf{Y}_1(j_1)\Big)\cdots\Big(\mathbf{X}_d(j_d) \otimes_{\mathrm{K}} \mathbf{Y}_d(j_d)\Big).$$

that is the thesis.

Let $\mathbf{j} = (j_1, \ldots, j_d) \in \mathbb{N}^{n_1 \times \cdots \times n_d}$, the **j**th element of $\mathcal{Z}$ is

$$\mathcal{Z}(\mathbf{j}) = \mathcal{X}(\mathbf{j})\mathcal{Y}(\mathbf{j}) = \Big(\mathbf{X}_1(j_1)\cdots\mathbf{X}_d(j_d)\Big)\Big(\mathbf{Y}_1(j_1)\cdots\mathbf{Y}_d(j_d)\Big).$$

Notice now that $\mathbf{X}_d(j_d)$ has size $(r_{d_1} \times 1)$ and $\mathbf{Y}_1(j_1)$ has size $(1 \times s_{d_1})$. Thus, we can make explicit in the previous equation the Kronecker product, writing

$$\mathcal{Z}(\mathbf{j}) = \Big(\mathbf{X}_1(j_1)\cdots\mathbf{X}_d(j_d)\Big) \otimes_K \Big(\mathbf{Y}_1(j_1)\cdots\mathbf{Y}_d(j_d)\Big).$$

Recalling the mixed-product property of the Kronecker product, we can rewrite the previous equation as

$$\mathcal{Z}(\mathbf{j}) = \Big(\mathbf{X}_1(j_1) \otimes_{\mathrm{K}} \mathbf{Y}_1(j_1)\Big)\cdots\Big(\mathbf{X}_d(j_d) \otimes_{\mathrm{K}} \mathbf{Y}_d(j_d)\Big).$$

that is the thesis.

TT-arithmetic – IV

Given a TT-vector, $\mathcal{X}$, and a TT-matrix, $\mathcal{A}$, of compatible size, it can be proven that

Proposition [Oseledets 2011]

The contraction TT-vector, $\mathcal{Y}(\mathbf{i}) = \mathcal{A}(\mathbf{i},\mathbf{j})\mathcal{X}(\mathbf{j})$ is such that

$$\mathbf{Y}_k(i_k) = \sum_{j_k=1}^{n_k} \mathbf{A}_k(i_k, j_k) \otimes_{\mathrm{K}} \mathbf{X}_k(j_k)$$

where $\mathbf{j} = (j_1, \ldots, j_d)$ and $\mathbf{i} = (i_1, \ldots, i_d)$, $j_k = 1, \ldots, n_k$ and $i_k = 1, \ldots, m_k$, $k = 1, \ldots, d$.

Remark

The TT-rank of $\mathcal{Y}$ is the product of the TT-ranks of $\mathcal{A}$ and $\mathcal{X}$. The contraction between a TT-matrix and a TT-vector can be obtained by dnm Kronecker products.

TT-arithmetic – IV

Given a TT-vector, $\mathcal{X}$, and a TT-matrix, $\mathcal{A}$, of compatible size, it can be proven that

Proposition [Oseledets 2011]

The contraction TT-vector, $\mathcal{Y}(\mathbf{i}) = \mathcal{A}(\mathbf{i}, \mathbf{j})\mathcal{X}(\mathbf{j})$ is such that

$$\mathbf{Y}_k(i_k) = \sum_{j_k=1}^{n_k} \mathbf{A}_k(i_k, j_k) \otimes_{\mathrm{K}} \mathbf{X}_k(j_k)$$

where $\mathbf{j} = (j_1, \ldots, j_d)$ and $\mathbf{i} = (i_1, \ldots, i_d)$, $j_k = 1, \ldots, n_k$ and $i_k = 1, \ldots, m_k$, $k = 1, \ldots, d$.

Remark

The TT-rank of $\mathcal{Y}$ is the product of the TT-ranks of $\mathcal{A}$ and $\mathcal{X}$. The contraction between a TT-matrix and a TT-vector can be obtained by *dnm* Kronecker products.

Proof – IV

The **i**th element of $\mathcal{Y}$ is

$$\mathcal{Y}(\mathbf{i}) = \sum_{\mathbf{j}} \mathcal{A}(\mathbf{i},\mathbf{j})\mathcal{X}(\mathbf{j}) = \sum_{k=1}^{d} \sum_{h_k=1}^{n_k} \Big(\mathbf{A}_1(i_1, j_1) \cdots \mathbf{A}_d(i_d, j_d)\Big)\Big(\mathbf{X}_1(j_1) \cdots \mathbf{X}_d(j_d)\Big)$$

Notice now that $\mathbf{A}_d(i_d, j_d)$ has size $(r_{d_1} \times 1)$ and $\mathbf{X}_1(j_1)$ has size $(1 \times s_{d_1})$. Thus, we can make explicit in the previous equation the Kronecker product, writing

$$\mathcal{Y}(\mathbf{i}) = \sum_{k=1}^{d} \sum_{h_k=1}^{n_k} \Big(\mathbf{A}_1(i_1, j_1) \cdots \mathbf{A}_d(i_d, j_d)\Big) \otimes_{\mathrm{K}} \Big(\mathbf{X}_1(j_1) \cdots \mathbf{X}_d(j_d)\Big).$$

Recalling the mixed-product property of the Kronecker product and the sum linearity, we can rewrite the previous equation as $\mathcal{Y}(\mathbf{i}) = \mathbf{Y}_1(i_1) \cdots \mathbf{Y}_d(i_d)$ where

$$\mathbf{Y}_k(i_k) = \sum_{j_k=1}^{n_k} \mathbf{A}_k(i_k, j_k) \otimes_{\mathrm{K}} \mathbf{X}(j_k)$$

that is the thesis.

Proof – IV

The **i**th element of $\mathcal{Y}$ is

$$\mathcal{Y}(\mathbf{i}) = \sum_{\mathbf{j}} \mathcal{A}(\mathbf{i},\mathbf{j})\mathcal{X}(\mathbf{j}) = \sum_{k=1}^{d} \sum_{h_k=1}^{n_k} \Big(\mathbf{A}_1(i_1, j_1) \cdots \mathbf{A}_d(i_d, j_d)\Big)\Big(\mathbf{X}_1(j_1) \cdots \mathbf{X}_d(j_d)\Big)$$

Notice now that $\mathbf{A}_d(i_d, j_d)$ has size $(r_{d_1} \times 1)$ and $\mathbf{X}_1(j_1)$ has size $(1 \times s_{d_1})$. Thus, we can make explicit in the previous equation the Kronecker product, writing

$$\mathcal{Y}(\mathbf{i}) = \sum_{k=1}^{d} \sum_{h_k=1}^{n_k} \Big(\mathbf{A}_1(i_1, j_1) \cdots \mathbf{A}_d(i_d, j_d)\Big) \otimes_{\mathrm{K}} \Big(\mathbf{X}_1(j_1) \cdots \mathbf{X}_d(j_d)\Big).$$

Recalling the mixed-product property of the Kronecker product and the sum linearity, we can rewrite the previous equation as $\mathcal{Y}(\mathbf{i}) = \mathbf{Y}_1(i_1) \cdots \mathbf{Y}_d(i_d)$ where

$$\mathbf{Y}_k(i_k) = \sum_{j_k=1}^{n_k} \mathbf{A}_k(i_k, j_k) \otimes_{\mathrm{K}} \mathbf{X}(j_k)$$

that is the thesis.

TT pro and cons

+ storage cost linear w.r.t. the tensor order, i.e., $\mathcal{O}(dnr^2)$;
+ arithmetic *ad hoc* to perform linear algebra operations;
+ always possible to compute with stable operations the TT-decomposition;
- linear algebra operations increase (significantly) the rank

Further info

- suggested reading: Oseledets 2011; Gelß 2017; Orús 2014;
- computational packages: TT-Toolbox, TTpy, torchTT, t3f...

TT pro and cons

- \+ storage cost linear w.r.t. the tensor order, i.e., $\mathcal{O}(dnr^2)$;
- \+ arithmetic *ad hoc* to perform linear algebra operations;
- \+ always possible to compute with stable operations the TT-decomposition;
- − linear algebra operations increase (significantly) the rank

Further info

- suggested reading: Oseledets 2011; Gelß 2017; Orús 2014;
- computational packages: TT-Toolbox, TTpy, torchTT, t3f...

TT pro and cons

+ storage cost linear w.r.t. the tensor order, i.e., $\mathcal{O}(dnr^2)$;
+ arithmetic *ad hoc* to perform linear algebra operations;
+ always possible to compute with stable operations the TT-decomposition;
– linear algebra operations increase (significantly) the rank

Further info

- suggested reading: Oseledets 2011; Gelß 2017; Orús 2014;
- computational packages: TT-Toolbox, TTpy, torchTT, t3f...

TT pro and cons

+ storage cost linear w.r.t. the tensor order, i.e., $\mathcal{O}(dnr^2)$;
+ arithmetic *ad hoc* to perform linear algebra operations;
+ always possible to compute with stable operations the TT-decomposition;

− linear algebra operations increase (significantly) the rank

Further info

- suggested reading: Oseledets 2011; Gelß 2017; Orús 2014;
- computational packages: TT-Toolbox, TTpy, torchTT, t3f...

TT pro and cons

- $+$ storage cost linear w.r.t. the tensor order, i.e., $\mathcal{O}(dnr^2)$;
- $+$ arithmetic *ad hoc* to perform linear algebra operations;
- $+$ always possible to compute with stable operations the TT-decomposition;
- $-$ linear algebra operations increase (significantly) the rank

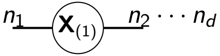
Further info

- suggested reading: Oseledets 2011; Gelß 2017; Orús 2014;
- computational packages: TT-Toolbox, TTpy, torchTT, t3f...

TT-decomposition – I

Let $\mathcal{X}$ be an order-d tensor of size $(n_1 \times \cdots \times n_d)$, and $N = \prod_{k=1}^{d} n_k$.

1. $\mathbf{X}_{(1)} \leftarrow \texttt{reshape}(\mathcal{X}, (n_1 \times m_1))$,
   $m_1 = N/n_1$



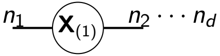
2. $[\mathbf{U}_1, \mathbf{\Sigma}_1, \mathbf{V}_1^{\mathsf{T}}] \leftarrow \texttt{SVD}(\mathbf{X}_{(1)})$;

3. $\mathcal{X}_1 \leftarrow \mathbf{U}_1$, 1st TT-core of size $(n_1 \times r_1)$

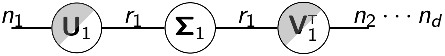
4. $\mathbf{E}_2 \leftarrow \texttt{reshape}(\mathbf{\Sigma}_1 \mathbf{V}_1^{\mathsf{T}}, (n_2 r_1 \times m_2))$,
   $m_2 = m_1/n_2$

Let $\mathcal{X}$ be an order-d tensor of size $(n_1 \times \cdots \times n_d)$, and $N = \prod_{k=1}^{d} n_k$.

1. $\mathbf{X}_{(1)} \leftarrow \texttt{reshape}(\mathcal{X}, (n_1 \times m_1))$,
   $m_1 = N/n_1$



2. $[\mathbf{U}_1, \mathbf{\Sigma}_1, \mathbf{V}_1^{\mathsf{T}}] \leftarrow \texttt{SVD}(\mathbf{X}_{(1)})$;



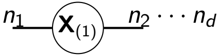
3. $\mathcal{X}_1 \leftarrow \mathbf{U}_1$, 1st TT-core of size $(n_1 \times r_1)$

4. $\mathbf{E}_2 \leftarrow \texttt{reshape}(\mathbf{\Sigma}_1 \mathbf{V}_1^{\mathsf{T}}, (n_2 r_1 \times m_2))$,
   $m_2 = m_1/n_2$

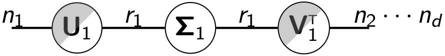
TT-decomposition – I

Let $\mathcal{X}$ be an order-d tensor of size $(n_1 \times \cdots \times n_d)$, and $N = \prod_{k=1}^{d} n_k$.

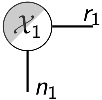
1. $\mathbf{X}_{(1)} \leftarrow \texttt{reshape}(\mathcal{X}, (n_1 \times m_1))$, $m_1 = N/n_1$



2. $[\mathbf{U}_1, \mathbf{\Sigma}_1, \mathbf{V}_1^{\intercal}] \leftarrow \texttt{SVD}(\mathbf{X}_{(1)})$;



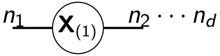
3. $\mathcal{X}_1 \leftarrow \mathbf{U}_1$, 1st TT-core of size $(n_1 \times r_1)$



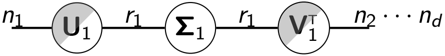
4. $\mathbf{E}_2 \leftarrow \texttt{reshape}(\mathbf{\Sigma}_1 \mathbf{V}_1^{\intercal}, (n_2 r_1 \times m_2))$, $m_2 = m_1/n_2$

Let $\mathcal{X}$ be an order-d tensor of size $(n_1 \times \cdots \times n_d)$, and $N = \prod_{k=1}^{d} n_k$.

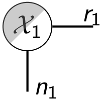
1. $\mathbf{X}_{(1)} \leftarrow \texttt{reshape}(\mathcal{X}, (n_1 \times m_1))$,
   $m_1 = N/n_1$



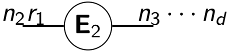
2. $[\mathbf{U}_1, \boldsymbol{\Sigma}_1, \mathbf{V}_1^{\mathsf{T}}] \leftarrow \texttt{SVD}(\mathbf{X}_{(1)})$;



3. $\mathcal{X}_1 \leftarrow \mathbf{U}_1$, 1st TT-core of size $(n_1 \times r_1)$

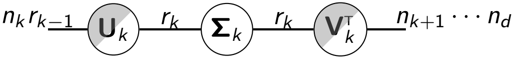


4. $\mathbf{E}_2 \leftarrow \texttt{reshape}(\boldsymbol{\Sigma}_1 \mathbf{V}_1^{\mathsf{T}}, (n_2 r_1 \times m_2))$,
   $m_2 = m_1/n_2$



At the kth step

1. $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^\top] \leftarrow \texttt{SVD}(\mathbf{E}_k)$;



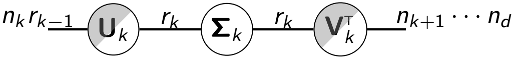
2. $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{U}_k, (r_{k-1} \times n_k \times r_k))$, kth TT-core

3. $\mathbf{E}_{k+1} \leftarrow \texttt{reshape}(\mathbf{\Sigma}_k \mathbf{V}_k^\top, (n_{k+1} r_k \times m_{k+1}))$, $m_{k+1} = m_k / n_k$

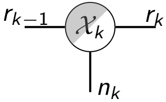
TT-decomposition – II

At the kth step

1. $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^\top] \leftarrow \texttt{SVD}(\mathbf{E}_k)$;



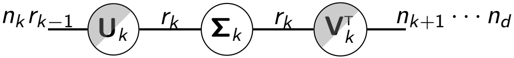
2. $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{U}_k, (r_{k-1} \times n_k \times r_k))$, kth TT-core



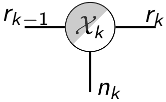
3. $\mathbf{E}_{k+1} \leftarrow \texttt{reshape}(\mathbf{\Sigma}_k \mathbf{V}_k^\top, (n_{k+1} r_k \times m_{k+1}))$,
$m_{k+1} = m_k / n_k$

At the kth step

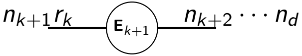
1. $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^\top] \leftarrow \texttt{SVD}(\mathbf{E}_k)$;



2. $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{U}_k, (r_{k-1} \times n_k \times r_k))$, kth TT-core

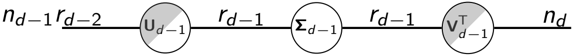


3. $\mathbf{E}_{k+1} \leftarrow \texttt{reshape}(\mathbf{\Sigma}_k \mathbf{V}_k^\top, (n_{k+1} r_k \times m_{k+1}))$,
$m_{k+1} = m_k / n_k$



At the $(d-1)$th step

1. $[\mathbf{U}_{d-1}, \mathbf{\Sigma}_{d-1}, \mathbf{V}_{d-1}^{\mathsf{T}}] \leftarrow \texttt{SVD}(\mathbf{E}_{d-1})$;



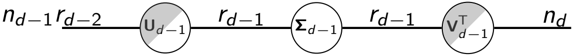
2. $\mathcal{X}_{d-1} \leftarrow \texttt{reshape}(\mathbf{U}_{d-1}, (r_{d-2} \times n_{d-1} \times r_{d-1}))$, $(d-1)$th TT-core

3. $\mathcal{X}_d \leftarrow \mathbf{\Sigma}_{d-1}\mathbf{V}_{d-1}^{\mathsf{T}}$, dth TT-core

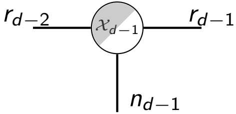
TT-decomposition – III

At the $(d-1)$th step

1. $[\mathbf{U}_{d-1}, \boldsymbol{\Sigma}_{d-1}, \mathbf{V}_{d-1}^{\mathsf{T}}] \leftarrow \texttt{SVD}(\mathbf{E}_{d-1})$;



2. $\mathcal{X}_{d-1} \leftarrow \texttt{reshape}(\mathbf{U}_{d-1}, (r_{d-2} \times n_{d-1} \times r_{d-1}))$,
   $(d-1)$th TT-core

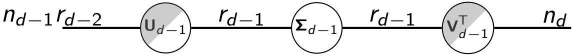


3. $\mathcal{X}_d \leftarrow \boldsymbol{\Sigma}_{d-1}\mathbf{V}_{d-1}^{\mathsf{T}}$, dth TT-core

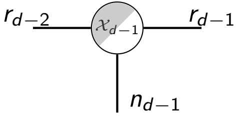
TT-decomposition – III

At the $(d-1)$th step

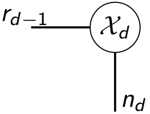
1. $[\mathbf{U}_{d-1}, \mathbf{\Sigma}_{d-1}, \mathbf{V}^{\mathsf{T}}_{d-1}] \leftarrow \texttt{SVD}(\mathbf{E}_{d-1})$;



2. $\mathcal{X}_{d-1} \leftarrow \texttt{reshape}(\mathbf{U}_{d-1}, (r_{d-2} \times n_{d-1} \times r_{d-1}))$, $(d-1)$th TT-core



3. $\mathcal{X}_d \leftarrow \mathbf{\Sigma}_{d-1}\mathbf{V}^{\mathsf{T}}_{d-1}$, dth TT-core



Algorithm 1: TT-SVD [Oseledets 2011]

Input: an $(n_1 \times \cdots \times n_d)$ dense tensor $\mathcal{X}$

Output: $\{\mathcal{X}_k\}$ TT-cores of $\mathcal{X}$

$m_1 = n_2 \cdots n_d$, and $r_0 = 1$;

$\mathbf{E} \leftarrow \texttt{reshape}(\mathcal{X}, (r_0 n_1 \times m_1))$;

for $k = 1, \ldots, d-1$ **do**

  $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^\top] \leftarrow \texttt{SVD}(\mathbf{E})$; ▷ $r_k = \text{rank}(\mathbf{E})$

  $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{U}, (r_{k-1} \times n_k \times r_k))$;

  $m_{k+1} \leftarrow m_k / n_k$;

  $\mathbf{E} \leftarrow \texttt{reshape}(\mathbf{\Sigma}_k \mathbf{V}_k^\top, (r_k n_{k+1} \times m_{k+1}))$;

$\mathcal{X}_d \leftarrow \texttt{reshape}(\mathbf{E}, (r_{d-1} \times n_d \times r_d))$ ▷ $r_d = 1$

TT-SVD quality

Theorem [Oseledets 2011]

Given a tensor, $\mathcal{X}$, the best approximation of $\mathcal{X}$ in the Frobenius norm with TT-ranks bounded by r_k always exists, denoted by $\mathcal{X}^*$.
If $\mathcal{Y}$ denotes the TT-decomposition of $\mathcal{X}$ computed by the TT-SVD algorithm, $\mathcal{Y}$ is quasi-optimal, i.e.,

$$||\mathcal{X} - \mathcal{Y}|| \leq \sqrt{d-1}||\mathcal{X} - \mathcal{X}^*||$$

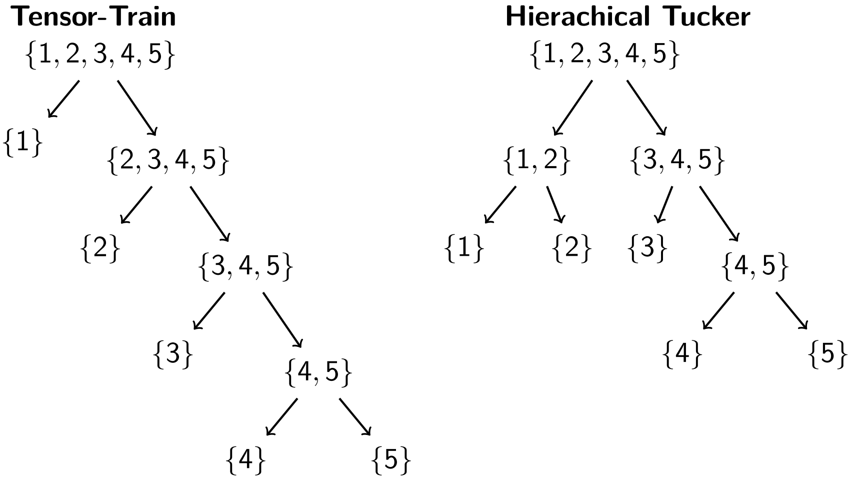
TT-SVD and graph

Consider $\mathcal{X}$ an $(n_1 \times n_2 \times n_3 \times n_4 \times n_5)$ tensor

$\{1,2,3,4,5\}$
↙ ↘
$\{1\}$ $\{2,3,4,5\}$
↙ ↘
$\{2\}$ $\{3,4,5\}$
↙ ↘
$\{3\}$ $\{4,5\}$
↙ ↘
$\{4\}$ $\{5\}$

$(n_1 \times n_2 n_3 n_4 n_5)$
↙ ↘
$(n_1 \times r_1)$ $(r_1 \times n_2 n_3 n_4 n_5)$
$(r_1 n_2 \times n_3 n_4 n_5)$
↙ ↘
$(r_1 n_2 \times r_2)$ $(r_2 \times n_3 n_4 n_5)$
$(r_2 n_3 \times n_4 n_5)$
↙ ↘
$(r_2 n_3 \times r_3)$ $(r_3 \times n_4 n_5)$
$(r_3 n_4 \times n_5)$
↙ ↘
$(r_3 n_4 \times r_4)$ $(r_4 \times n_5)$

A different tree? Hierarchical Tucker [Hackbusch 2019; Grasedyck 2010]



Let $\mathcal{X}$ be a TT-vector of size $(n_1 \times \cdots \times n_d)$ and TT-rank $(r_1, \ldots, r_d)$.
Aim: finding a TT-vector $\mathcal{Y}$ such that $||\mathcal{X} - \mathcal{Y}|| \leq \varepsilon ||\mathcal{X}||$.
Step I: $\delta = ||\mathcal{X}|| \varepsilon / \sqrt{d-1}$;
Step II: orthogonalize from right to left the TT-cores of $\mathcal{X}$, for $k = d, \ldots, 2$

1. $\mathbf{G}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (r_{k-1} \times n_k r_k))$
2. $[\mathbf{Q}_k, \mathbf{R}_k] \leftarrow \texttt{QR}(\mathbf{G}_k^\top)$;
3. $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{Q}_k, (r_{k-1} \times n_k \times r_k))$
4. $\mathcal{X}_{k-1} \leftarrow \mathcal{X}_{k-1} \times_3 \mathbf{R}_k$

TT-rounding – I

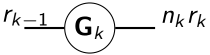
Let $\mathcal{X}$ be a TT-vector of size $(n_1 \times \cdots \times n_d)$ and TT-rank $(r_1, \ldots, r_d)$.
Aim: finding a TT-vector $\mathcal{Y}$ such that $||\mathcal{X} - \mathcal{Y}|| \leq \varepsilon \, ||\mathcal{X}||$.
Step I: $\delta = ||\mathcal{X}|| \, \varepsilon / \sqrt{d-1}$;
Step II: orthogonalize from right to left the TT-cores of $\mathcal{X}$, for $k = d, \ldots, 2$

1. $\mathbf{G}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (r_{k-1} \times n_k r_k))$
2. $[\mathbf{Q}_k, \mathbf{R}_k] \leftarrow \texttt{QR}(\mathbf{G}_k^\top)$;
3. $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{Q}_k, (r_{k-1} \times n_k \times r_k))$
4. $\mathcal{X}_{k-1} \leftarrow \mathcal{X}_{k-1} \times_3 \mathbf{R}_k$

TT-rounding – I

Let $\mathcal{X}$ be a TT-vector of size $(n_1 \times \cdots \times n_d)$ and TT-rank $(r_1, \ldots, r_d)$.
Aim: finding a TT-vector $\mathcal{Y}$ such that $||\mathcal{X} - \mathcal{Y}|| \leq \varepsilon \, ||\mathcal{X}||$.
Step I: $\delta = ||\mathcal{X}|| \, \varepsilon / \sqrt{d-1}$;
Step II: orthogonalize from right to left the TT-cores of $\mathcal{X}$, for $k = d, \ldots, 2$

1. $\mathbf{G}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (r_{k-1} \times n_k r_k))$



2. $[\mathbf{Q}_k, \mathbf{R}_k] \leftarrow \texttt{QR}(\mathbf{G}_k^T)$;

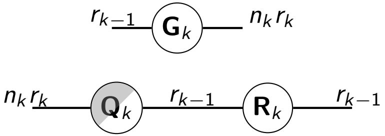
3. $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{Q}_k, (r_{k-1} \times n_k \times r_k))$

4. $\mathcal{X}_{k-1} \leftarrow \mathcal{X}_{k-1} \times_3 \mathbf{R}_k$

TT-rounding – I

Let $\mathcal{X}$ be a TT-vector of size $(n_1 \times \cdots \times n_d)$ and TT-rank $(r_1, \ldots, r_d)$.
Aim: finding a TT-vector $\mathcal{Y}$ such that $||\mathcal{X} - \mathcal{Y}|| \leq \varepsilon \, ||\mathcal{X}||$.
Step I: $\delta = ||\mathcal{X}|| \, \varepsilon / \sqrt{d-1}$;
Step II: orthogonalize from right to left the TT-cores of $\mathcal{X}$, for $k = d, \ldots, 2$

1. $\mathbf{G}_k \leftarrow \mathtt{reshape}(\mathcal{X}_k, (r_{k-1} \times n_k r_k))$
2. $[\mathbf{Q}_k, \mathbf{R}_k] \leftarrow \mathtt{QR}(\mathbf{G}_k^\top)$;

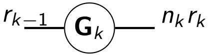


3. $\mathcal{X}_k \leftarrow \mathtt{reshape}(\mathbf{Q}_k, (r_{k-1} \times n_k \times r_k))$
4. $\mathcal{X}_{k-1} \leftarrow \mathcal{X}_{k-1} \times_3 \mathbf{R}_k$

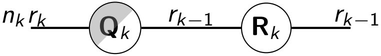
TT-rounding – I

Let $\mathcal{X}$ be a TT-vector of size $(n_1 \times \cdots \times n_d)$ and TT-rank $(r_1, \ldots, r_d)$.
Aim: finding a TT-vector $\mathcal{Y}$ such that $||\mathcal{X} - \mathcal{Y}|| \leq \varepsilon \, ||\mathcal{X}||$.
Step I: $\delta = ||\mathcal{X}|| \, \varepsilon / \sqrt{d-1}$;
Step II: orthogonalize from right to left the TT-cores of $\mathcal{X}$, for $k = d, \ldots, 2$

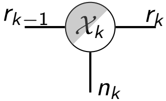
1. $\mathbf{G}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (r_{k-1} \times n_k r_k))$



2. $[\mathbf{Q}_k, \mathbf{R}_k] \leftarrow \texttt{QR}(\mathbf{G}_k^T)$;



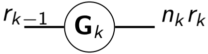
3. $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{Q}_k, (r_{k-1} \times n_k \times r_k))$



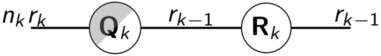
4. $\mathcal{X}_{k-1} \leftarrow \mathcal{X}_{k-1} \times_3 \mathbf{R}_k$

Let $\mathcal{X}$ be a TT-vector of size $(n_1 \times \cdots \times n_d)$ and TT-rank $(r_1, \ldots, r_d)$.
Aim: finding a TT-vector $\mathcal{Y}$ such that $||\mathcal{X} - \mathcal{Y}|| \leq \varepsilon\, ||\mathcal{X}||$.
Step I: $\delta = ||\mathcal{X}||\, \varepsilon / \sqrt{d-1}$;
Step II: orthogonalize from right to left the TT-cores of $\mathcal{X}$, for $k = d, \ldots, 2$

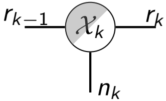
1. $\mathbf{G}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (r_{k-1} \times n_k r_k))$



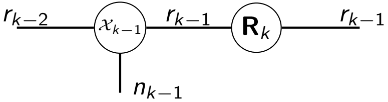
2. $[\mathbf{Q}_k, \mathbf{R}_k] \leftarrow \texttt{QR}(\mathbf{G}_k^T)$;



3. $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{Q}_k, (r_{k-1} \times n_k \times r_k))$



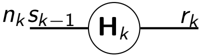
4. $\mathcal{X}_{k-1} \leftarrow \mathcal{X}_{k-1} \times_3 \mathbf{R}_k$



TT-rounding – II

Step III: truncate from left to right the TT-cores of $\mathcal{X}$, for $k = 1, \ldots, d-1$ and $s_0 = 1$

1. $\mathbf{H}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (s_{k-1} n_k \times r_k))$



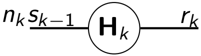
2. $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^T] \leftarrow \texttt{SVD}(\mathbf{H}_k, \delta)$;

3. $\mathcal{Y}_k \leftarrow \texttt{reshape}(\mathbf{U}_k, (s_{k-1} \times n_k \times s_k))$

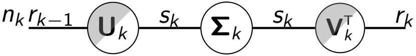
4. $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_{k+1} \times_1 (\mathbf{\Sigma}_k \mathbf{V}_k^T)^T$

Step III: truncate from left to right the TT-cores of $\mathcal{X}$, for $k = 1, \ldots, d-1$ and $s_0 = 1$

1. $\mathbf{H}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (s_{k-1} n_k \times r_k))$



2. $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^\mathsf{T}] \leftarrow \texttt{SVD}(\mathbf{H}_k, \delta)$;



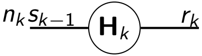
3. $\mathcal{Y}_k \leftarrow \texttt{reshape}(\mathbf{U}_k, (s_{k-1} \times n_k \times s_k))$

4. $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_{k+1} \times_1 (\mathbf{\Sigma}_k \mathbf{V}_k^\mathsf{T})^\mathsf{T}$

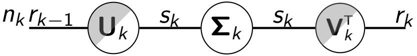
TT-rounding – II

Step III: truncate from left to right the TT-cores of $\mathcal{X}$, for $k = 1, \ldots, d-1$ and $s_0 = 1$

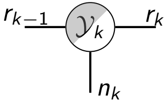
1. $\mathbf{H}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (s_{k-1}n_k \times r_k))$



2. $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^\top] \leftarrow \texttt{SVD}(\mathbf{H}_k, \delta)$;



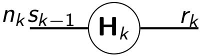
3. $\mathcal{Y}_k \leftarrow \texttt{reshape}(\mathbf{U}_k, (s_{k-1} \times n_k \times s_k))$



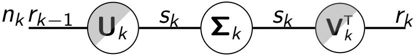
4. $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_{k+1} \times_1 (\mathbf{\Sigma}_k \mathbf{V}_k^\top)^\top$

Step III: truncate from left to right the TT-cores of $\mathcal{X}$, for $k = 1, \ldots, d-1$ and $s_0 = 1$

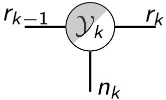
1. $\mathbf{H}_k \leftarrow \texttt{reshape}(\mathcal{X}_k, (s_{k-1} n_k \times r_k))$



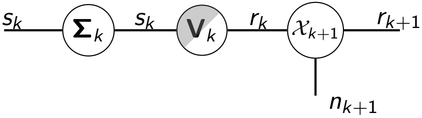
2. $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^\top] \leftarrow \texttt{SVD}(\mathbf{H}_k, \delta)$;



3. $\mathcal{Y}_k \leftarrow \texttt{reshape}(\mathbf{U}_k, (s_{k-1} \times n_k \times s_k))$



4. $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_{k+1} \times_1 (\mathbf{\Sigma}_k \mathbf{V}_k^\top)^\top$



Algorithm – I

Algorithm 2: TT-rounding [Oseledets 2011]

Input: $\{\mathcal{X}_k\}$ TT-cores of $\mathcal{X}$, $\varepsilon \in (0,1)$

Output: $\{\mathcal{Y}_k\}$ TT-cores of $\mathcal{Y}$ s.t. $||\mathcal{X} - \mathcal{Y}|| \leq \varepsilon \, ||\mathcal{X}||$

▷ *Step I: preparation*

$\delta = ||\mathcal{X}|| \, \varepsilon / \sqrt{d-1}$;

$\mathbf{G} \leftarrow \texttt{reshape}(\mathcal{X}_d, (r_{d-1} \times n_d r_d))$;

▷ *Step II: orthogonalization*

for $k = d, \dots, 2$ **do**

  $[\mathbf{Q}_k, \mathbf{R}_k] \leftarrow \texttt{QR}(\mathbf{G}^\top)$;

  $\mathcal{X}_k \leftarrow \texttt{reshape}(\mathbf{Q}_k, (r_{k-1} \times n_k \times r_k))$;

  $\mathbf{G} \leftarrow \texttt{reshape}(\mathcal{X}_{k-1} \times_3 \mathbf{R}_k, (r_{k-2} \times n_{k-1} r_{k-1}))$;

$\mathcal{X}_1 \leftarrow \texttt{reshape}(\mathbf{G}, (r_0 \times n_1 \times r_1))$;

Algorithm – II

▷ *Step III: truncation*

$s_0 = 1$;

$\mathbf{H} \leftarrow \texttt{reshape}(\mathcal{X}_1, (s_0 n_1 \times r_1))$;

for $k = 1, \dots, d-1$ **do**

  $[\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k^\top] \leftarrow \texttt{SVD}(\mathbf{H}, \delta)$; ▷ $s_k = \text{rank}_\delta(\mathbf{H})$

  $\mathcal{Y}_k \leftarrow \texttt{reshape}(\mathbf{U}_k, (s_{k-1} \times n_k \times s_k))$;

  $\mathbf{H} \leftarrow \texttt{reshape}(\mathcal{X}_{k+1} \times_1 (\mathbf{\Sigma}_k \mathbf{V}_k^\top)^\top, (s_k n_{k+1} \times r_{k+1}))$;

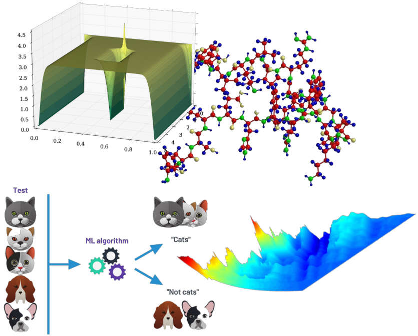
$\mathcal{Y}_d \leftarrow \texttt{reshape}(\mathbf{H}, (s_{d-1} \times n_d \times s_d))$ ▷ $s_d = 1$

Table of Contents

Application fields

Numerical simulations are necessary in

- Stochastic equations
- Uncertainty quantification problems
- Quantum and vibration chemistry
- Optimization
- Machine learning



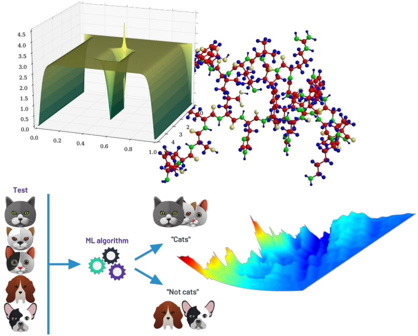
Frequently, they involve solving

Least-squares	Eigenpairs	Linear systems
$\min_{x \in \mathcal{S}} \|\|b - Ax\|\|$	$Ax = \lambda x$	$Ax = b$

Application fields

Numerical simulations are necessary in

- Stochastic equations
- Uncertainty quantification problems
- Quantum and vibration chemistry
- Optimization
- Machine learning



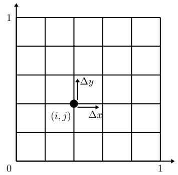
Frequently, they involve solving

Least-squares	Eigenpairs	**Linear systems**
$\min_{x \in \mathcal{S}} \|\|b - Ax\|\|$	$Ax = \lambda x$	$Ax = b$

Context

The problem

$$\begin{cases} \mathcal{L}(u) & = f \quad \text{in} \quad \Omega \\ u & = f_0 \quad \text{in} \quad \partial\Omega \end{cases} \quad \text{for} \quad \Omega \subseteq \mathbb{R}^{n_1 \times \cdots \times n_d}.$$



$$\boldsymbol{\mathcal{A}}(\mathcal{X}) = \mathcal{B}$$

where $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a multilinear operator and $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ a tensor.
For large scale-simulations we have to take into account

- memory costs $\mathcal{O}(n^d)$
- computational model
- numerical method

Multilinear Solvers

Given a multilinear operator $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \times \cdots \times n_d}$ and a tensor $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, consider the multilinear system

$$\boldsymbol{\mathcal{A}}(\mathcal{X}) = \mathcal{B}$$

Optimization methods

- Density Matrix Renormalisation Group (DMRG) [White 1992];
- Alternating Minimal Energy (AMEn) [Dolgov and Savostyanov 2014];
- ...

based on (M)ALS.

Numerical linear algebra methods

- Conjugate Gradient (CG) [Tobler 2012];
- Generalized minimal residual method (GMRES) [Dolgov 2013];
- ...

based on iterative schemes.

1-site DMRG: Optimization setting

Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \times \cdots \times n_d}$ with $\mathcal{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ its associated tensor, such that $\mathcal{A}(\mathbf{j}, \mathbf{i}) = \mathcal{A}(\mathbf{i}, \mathbf{j})$.
The solution $\mathcal{X}^*$ of the the multilinear system $\mathcal{A}\mathcal{X} = \mathcal{B}$ is the minimizer of the functional

$$\mathcal{J}(\mathcal{X}) = \frac{1}{2}\langle \mathcal{X}, \mathcal{A}\mathcal{X} \rangle - \langle \mathcal{X}, \mathcal{B} \rangle$$

Notation

To ease the presentation, we use the following notation:

- $\mathbf{r} = (r_0, r_1, \ldots, r_d)$
- $\mathbb{R}^{\mathbf{m} \times \mathbf{n}} = \mathbb{R}^{(m_1 \times n_1) \times \cdots \times (m_d \times n_d)}$;
- $\mathcal{M}_{TT}(\mathbf{n}, \mathbf{r})$ denotes the set* of the TT-vectors of size $\mathbf{n}$ and TT-rank $\mathbf{r}$;
- $\mathcal{M}_{TT}(\mathbf{m} \times \mathbf{n}, \mathbf{r})$ denotes the set* of the TT-matrices of size $\mathbf{m} \times \mathbf{n}$ and TT-rank $\mathbf{r}$.

Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \times \cdots \times n_d}$ with $\mathcal{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ its associated tensor, such that $\mathcal{A}(\mathbf{j}, \mathbf{i}) = \mathcal{A}(\mathbf{i}, \mathbf{j})$.
The solution $\mathcal{X}^*$ of the the multilinear system $\mathcal{A}\mathcal{X} = \mathcal{B}$ is the minimizer of the functional

$$\mathcal{J}(\mathcal{X}) = \frac{1}{2}\langle \mathcal{X}, \mathcal{A}\mathcal{X} \rangle - \langle \mathcal{X}, \mathcal{B} \rangle$$

Notation

To ease the presentation, we use the following notation:

- $\mathbf{r} = (r_0, r_1, \ldots, r_d)$
- $\mathbb{R}^{\mathbf{m} \times \mathbf{n}} = \mathbb{R}^{(m_1 \times n_1) \times \cdots \times (m_d \times n_d)}$;
- $\mathcal{M}_{TT}(\mathbf{n}, \mathbf{r})$ denotes the set* of the TT-vectors of size $\mathbf{n}$ and TT-rank $\mathbf{r}$;
- $\mathcal{M}_{TT}(\mathbf{m} \times \mathbf{n}, \mathbf{r})$ denotes the set* of the TT-matrices of size $\mathbf{m} \times \mathbf{n}$ and TT-rank $\mathbf{r}$.

1-site DMRG: from d modes to 1 mode

For every $k = 1, \dots, d$, define the retraction operator

$$\mathcal{Q}_k : \mathcal{M}_{TT}(\mathbf{n}, \mathbf{r}) \mapsto \mathcal{M}_{TT}(\mathbf{n} \times \mathbf{m}_k, \mathbf{r})$$

such that

- $\mathbf{m}_k = (1, \dots, m_k, \dots, 1)$ where $m_k = r_{k-1} \cdot n_k \cdot r_k$;
- the j-th TT core of $\mathcal{Q}_k(\mathcal{X})$ is equal to the j-th TT core of $\mathcal{X}$, i.e.

$$\Big(\mathcal{Q}_k(\mathcal{X})\Big)_j = \mathcal{X}_j \qquad \text{for} \qquad j \neq k$$

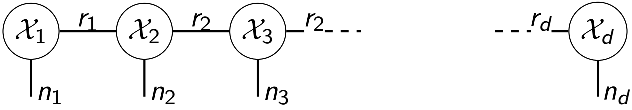
- the k-th TT core of $\mathcal{Q}_k(\mathcal{X})$ is the identity operator, i.e.

$$\Big(\mathcal{Q}_k(\mathcal{X})\Big)_k(\mathbf{i}, \mathbf{j}) = \mathbb{I}.$$

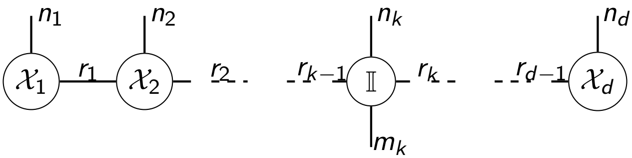
Retraction as TN

The TN-representation of the retraction operator

Given the TT-vector $\mathcal{X}$



the TT-matrix $\mathcal{Q}_k(\mathcal{X})$ is



1-site DMRG: core idea

To approximate the minimizer $\mathcal{X}^*$, let $\mathcal{X}^{(0)} \in \mathcal{M}_{TT}(\mathbf{n}, \mathbf{r})$ be an initial guess, and for every $k = 1, \ldots, d$ we construct the retraction operator $\mathcal{Q}_k(\mathcal{X})$ and we compute

$$\left(\mathcal{J} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})\right)\mathbf{x} = \frac{1}{2}\langle \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}, \boldsymbol{\mathcal{A}} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}\rangle - \langle \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}, \mathcal{B}\rangle$$

where $\mathcal{X}^{(k-1)}$ has all the first $(k-1)$ TT-cores updated by the 1-side DMRG and the remaining ones equal to the TT-cores of the initial guess $\mathcal{X}^{(0)}$.

The minimizer of $\mathcal{J} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})$ is $\mathbf{x}_k \in \mathbb{R}^{m_k}$.

Then, $\mathbf{x}_k$ is reshaped as an $(r_{k-1} \times n_k \times r_k)$ tensor $\mathcal{X}_k$, the kth TT-core of an approximation of the minimizer $\mathcal{X}^*$.

1-site DMRG: core idea

To approximate the minimizer $\mathcal{X}^*$, let $\mathcal{X}^{(0)} \in \mathcal{M}_{TT}(\mathbf{n}, \mathbf{r})$ be an initial guess, and for every $k = 1, \ldots, d$ we construct the retraction operator $\mathcal{Q}_k(\mathcal{X})$ and we compute

$$\Big(\mathcal{J} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})\Big)\mathbf{x} = \frac{1}{2}\langle \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}, \mathcal{A} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}\rangle - \langle \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}, \mathcal{B}\rangle$$

where $\mathcal{X}^{(k-1)}$ has all the first $(k-1)$ TT-cores updated by the 1-side DMRG and the remaining ones equal to the TT-cores of the initial guess $\mathcal{X}^{(0)}$.

The minimizer of $\mathcal{J} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})$ is $\mathbf{x}_k \in \mathbb{R}^{m_k}$.

Then, $\mathbf{x}_k$ is reshaped as an $(r_{k-1} \times n_k \times r_k)$ tensor $\mathcal{X}_k$, the kth TT-core of an approximation of the minimizer $\mathcal{X}^*$.

To approximate the minimizer $\mathcal{X}^*$, let $\mathcal{X}^{(0)} \in \mathcal{M}_{TT}(\mathbf{n}, \mathbf{r})$ be an initial guess, and for every $k = 1, \dots, d$ we construct the retraction operator $\mathcal{Q}_k(\mathcal{X})$ and we compute

$$\left(\mathcal{J} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})\right)\mathbf{x} = \frac{1}{2}\langle \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}, \mathcal{A} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}\rangle - \langle \mathcal{Q}_k(\mathcal{X}^{(k-1)})\mathbf{x}, \mathcal{B}\rangle$$

where $\mathcal{X}^{(k-1)}$ has all the first $(k-1)$ TT-cores updated by the 1-side DMRG and the remaining ones equal to the TT-cores of the initial guess $\mathcal{X}^{(0)}$.

The minimizer of $\mathcal{J} \circ \mathcal{Q}_k(\mathcal{X}^{(k-1)})$ is $\mathbf{x}_k \in \mathbb{R}^{m_k}$.

Then, $\mathbf{x}_k$ is reshaped as an $(r_{k-1} \times n_k \times r_k)$ tensor $\mathcal{X}_k$, the kth TT-core of an approximation of the minimizer $\mathcal{X}^*$.

Remark that

- the retraction of the multilinear operator as the contraction of all the left and right indexes of $\mathcal{A}$ by $\mathcal{Q}_k(\mathcal{X})$, i.e.,

$$\mathbf{A}_k = \mathcal{Q}_k(\mathcal{X})^\intercal \mathcal{A} \mathcal{Q}_k(\mathcal{X}) \qquad \text{is} \qquad (m_k \times m_k) \qquad \text{matrix}$$

- the retraction of the right hand-side as the contraction of all the indexes of $\mathcal{B}$ by $\mathcal{Q}_k(\mathcal{X})$, i.e.,

$$\mathbf{b}_k = \mathcal{Q}_k(\mathcal{X})^\intercal \mathcal{B} \qquad \text{is a length} \qquad m_k \qquad \text{vector}$$

By solving

$$\mathbf{A}_k \mathbf{x} = \mathbf{b}_k$$

we find a vector $\mathbf{x}_k$ of length m_k, and update the k-th TT-core of the tensor we are looking for as

$$\mathcal{X}_k = \mathtt{reshape}(\mathbf{x}_k, (r_{k-1} \times n_k \times r_k)).$$

1-site DMRG: details

Remark that

- the retraction of the multilinear operator as the contraction of all the left and right indexes of $\mathcal{A}$ by $\mathcal{Q}_k(\mathcal{X})$, i.e.,

$$\mathbf{A}_k = \mathcal{Q}_k(\mathcal{X})^\intercal \mathcal{A} \mathcal{Q}_k(\mathcal{X}) \qquad \text{is} \qquad (m_k \times m_k) \qquad \text{matrix}$$

- the retraction of the right hand-side as the contraction of all the indexes of $\mathcal{B}$ by $\mathcal{Q}_k(\mathcal{X})$, i.e.,

$$\mathbf{b}_k = \mathcal{Q}_k(\mathcal{X})^\intercal \mathcal{B} \qquad \text{is a length} \qquad m_k \qquad \text{vector}$$

By solving

$$\mathbf{A}_k \mathbf{x} = \mathbf{b}_k$$

we find a vector $\mathbf{x}_k$ of length m_k, and update the k-th TT-core of the tensor we are looking for as

$$\mathcal{X}_k = \texttt{reshape}(\mathbf{x}_k, (r_{k-1} \times n_k \times r_k)).$$

1-site DMRG: details

Remark that

- the retraction of the multilinear operator as the contraction of all the left and right indexes of $\mathcal{A}$ by $\mathcal{Q}_k(\mathcal{X})$, i.e.,

$$\mathbf{A}_k = \mathcal{Q}_k(\mathcal{X})^\top \mathcal{A} \mathcal{Q}_k(\mathcal{X}) \qquad \text{is} \qquad (m_k \times m_k) \qquad \text{matrix}$$

- the retraction of the right hand-side as the contraction of all the indexes of $\mathcal{B}$ by $\mathcal{Q}_k(\mathcal{X})$, i.e.,

$$\mathbf{b}_k = \mathcal{Q}_k(\mathcal{X})^\top \mathcal{B} \qquad \text{is a length} \qquad m_k \qquad \text{vector}$$

By solving

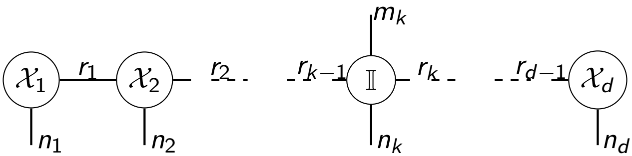
$$\mathbf{A}_k \mathbf{x} = \mathbf{b}_k$$

we find a vector $\mathbf{x}_k$ of length m_k, and update the k-th TT-core of the tensor we are looking for as

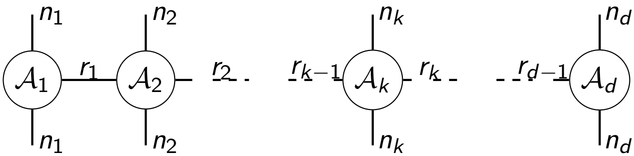
$$\mathcal{X}_k = \texttt{reshape}(\mathbf{x}_k, (r_{k-1} \times n_k \times r_k)).$$

TN - core idea – I

The TN-representation of the TT-matrix $\mathcal{Q}_k(\mathcal{X})^\top$ is

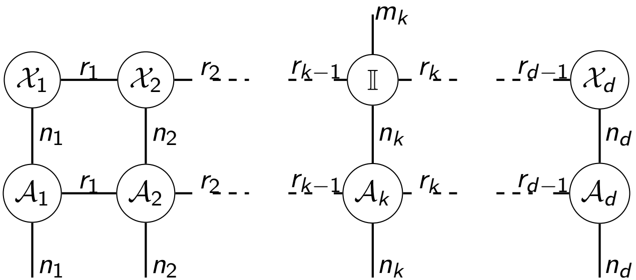


and the TT-matrix $\mathcal{A}$ is



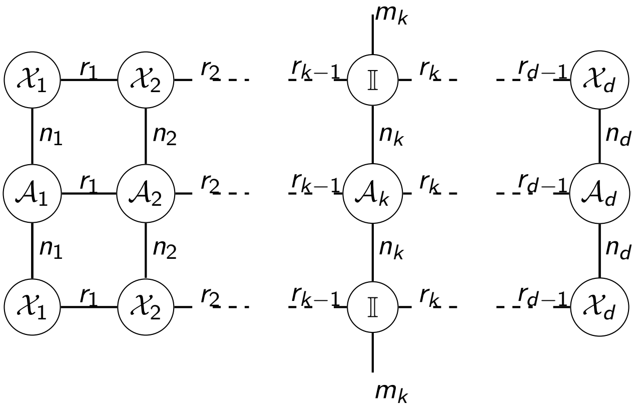
TN - core idea – I

Thus, the TN-representation of the TT-matrix $\mathcal{Q}_k(\mathcal{X})^\top \mathcal{A}$ is

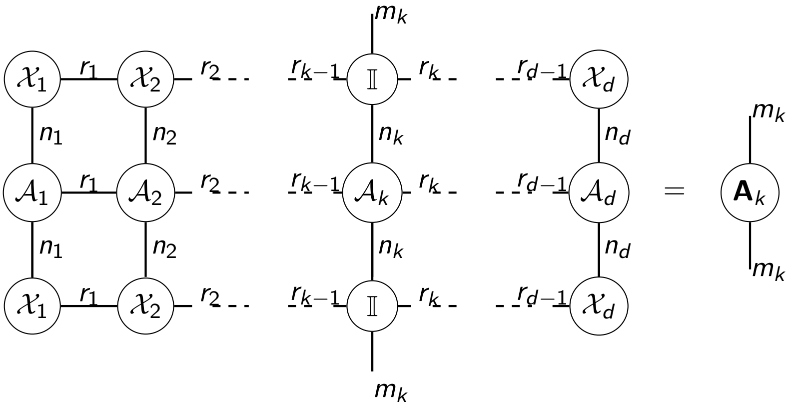


TN - core idea – I

Finally, the TN-representation of the TT-matrix $\mathcal{Q}_k(\mathcal{X})^\top \mathcal{A} \mathcal{Q}_k(\mathcal{X})$ is

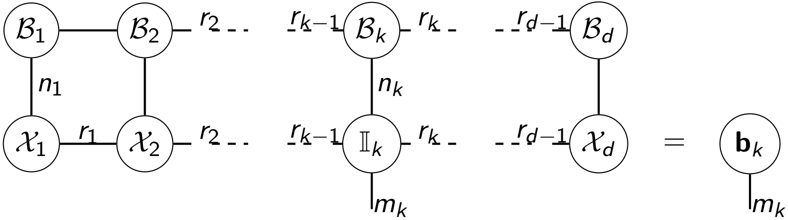


Finally, the TN-representation of the TT-matrix $\mathcal{Q}_k(\mathcal{X})^\top \mathcal{A} \mathcal{Q}_k(\mathcal{X})$ is



$\mathbf{A}_k$ is an $(m_k \times m_k)$ matrix!

Similarly, the TN-representation of the TT-vector $\mathcal{Q}_k(\mathcal{X})^\top \mathcal{B}$ is



$\mathbf{b}_k$ is an $(m_k \times 1)$ vector!

Optimization methods – comments

1-site DMRG relies on ALS scheme. **ALS** represents the basis for the other optimization methods, which

- converts a global minimization problem into a local one;
- may merge two adjacent TT-cores and separates them by SVD;
- may vary the TT-ranks;
- may introduce expansion of the TT-cores.

Optimization methods

+ suitable for more complex problems;
± developed independently in several research groups from different fields;
− implementations are not well organized and sometimes hard to employ;

Optimization methods – comments

1-site DMRG relies on ALS scheme. **ALS** represents the basis for the other optimization methods, which

- converts a global minimization problem into a local one;
- may merge two adjacent TT-cores and separates them by SVD;
- may vary the TT-ranks;
- may introduce expansion of the TT-cores.

Optimization methods

+ suitable for more complex problems;
± developed independently in several research groups from different fields;
− implementations are not well organized and sometimes hard to employ;

Numerical linear algebra – comments

Iterative schemes are employed, replacing

- matrices by TT-matrices;
- vectors by TT-vectors;
- TT-rounding steps are introduced.

The TT-rounding steps make the algorithm inexact, (hopefully) linking the solution accuracy to the rounding precision.

+ usually straightforward to implement;
+ many theoretical results available for the matrix case can be adapted to the TT-case;
– suitable for more simpler problems;
– TT-rounding is an expensive operation, its use has to be tuned;
– TT-rounding affects the quality of the iterative solution.

Numerical linear algebra – comments

Iterative schemes are employed, replacing

- matrices by TT-matrices;
- vectors by TT-vectors;
- TT-rounding steps are introduced.

The TT-rounding steps make the algorithm inexact, (hopefully) linking the solution accuracy to the rounding precision.

+ usually straightforward to implement;
+ many theoretical results available for the matrix case can be adapted to the TT-case;
– suitable for more simpler problems;
– TT-rounding is an expensive operation, its use has to be tuned;
– TT-rounding affects the quality of the iterative solution.

Study case: MGS

Producing an orthogonal basis is a common task of iterative methods. Given a set of linearly independent vectors $\mathcal{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_m\}$, the aim is producing a set $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_m\}$ such that

$$\langle \mathbf{q}_i, \mathbf{q}_j \rangle = \delta_{ij}.$$

The Modified Gram-Schmidt (MGS) is a classical choice thanks to its algorithmic clean structure and its good performance*.

Algorithm 3: $\mathcal{Q}, \mathbf{R} = \texttt{MGS}(\mathcal{A})$

Input: $\mathcal{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_m\}$

Output: $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_m\}, \mathbf{R}$

```
for i = 1, ..., m do
    p = a_i
    for j = 1, ..., i - 1 do
        R(i, j) = <p, q_j>
        p = p - R(i, j) q_j
    R(i, i) = ||p||
    q_i = p / R(i, i)
```

Producing an orthogonal basis is a common task of iterative methods. Given a set of linearly independent vectors $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$, the aim is producing a set $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$ such that

$$\langle \mathbf{q}_i, \mathbf{q}_j \rangle = \delta_{ij}.$$

The Modified Gram-Schmidt (MGS) is a classical choice thanks to its algorithmic clean structure and its good performance*.

Algorithm 4: $\mathcal{Q}, \mathbf{R} = \texttt{MGS}(\mathcal{A})$

Input: $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$
Output: $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_m\}, \mathbf{R}$

```
for i = 1, ..., m do
    p = a_i
    for j = 1, ..., i − 1 do
        R(i, j) = ⟨p, q_j⟩
        p = p − R(i, j) q_j
    R(i, i) = ||p||
    q_i = p / R(i, i)
```

Orthogonalization schemes

A key property of an orthonalization scheme is producing a basis with a good orthogonality regardless of the possible collinearities the input set.
The loss of orthogonality measures the quality in terms of orthogonality of the computed basis.

Definition

Let $\mathbf{Q}_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$ be the orthogonal basis produced by an orthogonalization kernel, then the **Loss Of Orthogonality** (LOO) is

$$||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k||.$$

Several theoretical results link the LOO with $\kappa(\mathbf{A}_k)$, i.e. the measure of the linearly dependency of the input vectors $\mathbf{A}_k = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$.

Remark

For MGS, it was proven in [Björck 1967] that $||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k|| \sim \mathcal{O}(u\kappa(\mathbf{A}_k))$ where u is the machine working precision.

Orthogonalization schemes

A key property of an orthonalization scheme is producing a basis with a good orthogonality regardless of the possible collinearities the input set.
The loss of orthogonality measures the quality in terms of orthogonality of the computed basis.

Definition

Let $\mathbf{Q}_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$ be the orthogonal basis produced by an orthogonalization kernel, then the **Loss Of Orthogonality** (LOO) is

$$||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k||.$$

Several theoretical results link the LOO with $\kappa(\mathbf{A}_k)$, i.e. the measure of the linearly dependency of the input vectors $\mathbf{A}_k = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$.

Remark

For MGS, it was proven in [Björck 1967] that $||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k|| \sim \mathcal{O}(u\kappa(\mathbf{A}_k))$ where u is the machine working precision.

TT-MGS

The MGS algorithm is straightforwardly formulated in TT-format.

The two fundamental modifications:

- rounding precision $\delta \in (0, 1)$ as input;
- TT-rounding step after the inner for loop.

These modifications are needed because the linear combinations of line 5 sequentially increases the TT-ranks!

Algorithm 5: $\mathcal{Q}, \mathbf{R} = \texttt{TT-MGS}(\mathcal{A}, \delta)$

Input: $\mathcal{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_m\}$, $\delta \in \mathbb{R}_+$

Output: $\mathcal{Q} = \{\mathcal{Q}_1, \dots, \mathcal{Q}_m\}$, $\mathbf{R}$

1 **for** $i = 1, \dots, m$ **do**

2 $\quad \mathcal{P} = \mathcal{A}_i$

3 $\quad$ **for** $j = 1, \dots, i-1$ **do**

4 $\quad\quad \mathbf{R}(i,j) = \langle \mathcal{P}, \mathcal{Q}_j \rangle$

5 $\quad\quad \mathcal{P} = \mathcal{P} - \mathbf{R}(i,j)\mathcal{Q}_j$

6 $\quad \mathcal{P} = \texttt{TT-rounding}(\mathcal{P}, \delta)$

7 $\quad \mathbf{R}(i,i) = ||\mathcal{P}||$

8 $\quad \mathcal{Q}_i = \mathcal{P}/\mathbf{R}(i,i)$

TT-MGS

The MGS algorithm is straightforwardly formulated in TT-format.
The two fundamental modifications:

- rounding precision $\delta \in (0, 1)$ as input;
- TT-rounding step after the inner for loop.

These modifications are needed because the linear combinations of line 5 sequentially increases the TT-ranks!

Algorithm 6: $\mathcal{Q}, \mathbf{R} = \texttt{TT-MGS}(\mathcal{A}, \delta)$

Input: $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$, $\delta \in \mathbb{R}_+$
Output: $\mathcal{Q} = \{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$, $\mathbf{R}$

```
1 for i = 1, ..., m do
2     P = A_i
3     for j = 1, ..., i − 1 do
4         R(i, j) = ⟨P, Q_j⟩
5         P = P − R(i, j) Q_j
6     P = TT-rounding(P, δ)
7     R(i, i) = ||P||
8     Q_i = P / R(i, i)
```

1 **for** $i = 1, \ldots, m$ **do**
2 $\quad \mathcal{P} = \mathcal{A}_i$
3 $\quad$ **for** $j = 1, \ldots, i-1$ **do**
4 $\quad\quad \mathbf{R}(i,j) = \langle \mathcal{P}, \mathcal{Q}_j \rangle$
5 $\quad\quad \mathcal{P} = \mathcal{P} - \mathbf{R}(i,j)\mathcal{Q}_j$
6 $\quad \mathcal{P} = \texttt{TT-rounding}(\mathcal{P}, \delta)$
7 $\quad \mathbf{R}(i,i) = ||\mathcal{P}||$
8 $\quad \mathcal{Q}_i = \mathcal{P}/\mathbf{R}(i,i)$

TT-MGS

The MGS algorithm is straightforwardly formulated in TT-format.
The two fundamental modifications:

- rounding precision $\delta \in (0,1)$ as input;
- TT-rounding step after the inner for loop.

These modifications are needed because the linear combinations of line 5 sequentially increases the TT-ranks!

Algorithm 7: $\mathcal{Q}, \mathbf{R} = \texttt{TT-MGS}(\mathcal{A}, \delta)$

Input: $\mathcal{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_m\}$, $\delta \in \mathbb{R}_+$
Output: $\mathcal{Q} = \{\mathcal{Q}_1, \dots, \mathcal{Q}_m\}$, $\mathbf{R}$

```
1 for i = 1, ..., m do
2     P = A_i
3     for j = 1, ..., i − 1 do
4         R(i, j) = ⟨P, Q_j⟩
5         P = P − R(i, j) Q_j
6     P = TT-rounding(P, δ)
7     R(i, i) = ||P||
8     Q_i = P / R(i, i)
```

TT-MGS: numerical evaluation [Coulaud et al. 2022]

We produce a sequence of 20 TT-vectors of size $(15 \times 15 \times 15)$ which get more and more collinear, as

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\boldsymbol{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1)$$
$$\mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||}\mathcal{X}_{k+1}$$

Figure: The LOO of **MGS** vs $\kappa(\mathbf{A}_k)$.

We compute $[\mathcal{Q}, \sim] = \texttt{TT-MGS}(\mathcal{A}, \delta = 10^{-5})$. Let $\mathbf{Q}_k^\top \mathbf{Q}_k(i,j) = \langle \mathcal{Q}_i, \mathcal{Q}_j \rangle$ and $\mathbf{A}_k(:,j) = \texttt{reshape}(\mathcal{A}_j, n^3)$ for $i, j = 1, \dots, k$ and $k = 1, \dots, 20$.

Experimentally

$$||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k|| \sim \mathcal{O}(\delta \kappa(\mathbf{A}_k))$$

TT-MGS: numerical evaluation [Coulaud et al. 2022]

We produce a sequence of 20 TT-vectors of size $(15 \times 15 \times 15)$ which get more and more collinear, as

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\boldsymbol{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1)$$

$$\mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||} \mathcal{X}_{k+1}$$

We compute $[\mathcal{Q}, \sim] = \texttt{TT-MGS}(\mathcal{A}, \delta = 10^{-5})$.
Let $\mathbf{Q}_k^\top \mathbf{Q}_k(i,j) = \langle \mathcal{Q}_i, \mathcal{Q}_j \rangle$ and $\mathbf{A}_k(:,j) = \texttt{reshape}(\mathcal{A}_j, n^3)$ for $i,j = 1,\ldots,k$ and $k = 1,\ldots,20$.
Experimentally

$$||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k|| \sim \mathcal{O}(\delta \kappa(\mathbf{A}_k))$$

Figure: The LOO of **MGS** vs $\kappa(\mathbf{A}_k)$.

We produce a sequence of 20 TT-vectors of size $(15 \times 15 \times 15)$ which get more and more collinear, as

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\boldsymbol{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1)$$
$$\mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||} \mathcal{X}_{k+1}$$

We compute $[\mathcal{Q}, \sim] = \texttt{TT-MGS}(\mathcal{A}, \delta = 10^{-5})$. Let $\mathbf{Q}_k^\top \mathbf{Q}_k(i,j) = \langle \mathcal{Q}_i, \mathcal{Q}_j \rangle$ and $\mathbf{A}_k(:,j) = \texttt{reshape}(\mathcal{A}_j, n^3)$ for $i,j = 1, \dots, k$ and $k = 1, \dots, 20$.

Experimentally

$$||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k|| \sim \mathcal{O}(\delta \kappa(\mathbf{A}_k))$$

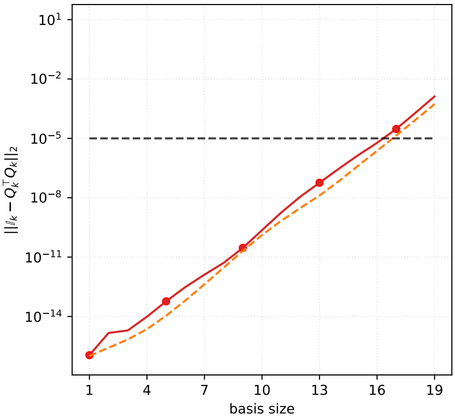


Figure: The LOO of **MGS** vs $\kappa(\mathbf{A}_k)$.

TT-MGS: numerical evaluation [Coulaud et al. 2022]

We produce a sequence of 20 TT-vectors of size $(15 \times 15 \times 15)$ which get more and more collinear, as

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\boldsymbol{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1)$$
$$\mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||} \mathcal{X}_{k+1}$$

We compute $[\mathcal{Q}, \sim] = \texttt{TT-MGS}(\mathcal{A}, \delta = 10^{-5})$.
Let $\mathbf{Q}_k^\top \mathbf{Q}_k(i,j) = \langle \mathcal{Q}_i, \mathcal{Q}_j \rangle$ and $\mathbf{A}_k(:,j) = \texttt{reshape}(\mathcal{A}_j, n^3)$ for $i,j = 1,\dots,k$ and $k = 1,\dots,20$.
Experimentally

$$||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k|| \sim \mathcal{O}(\delta \kappa(\mathbf{A}_k))$$

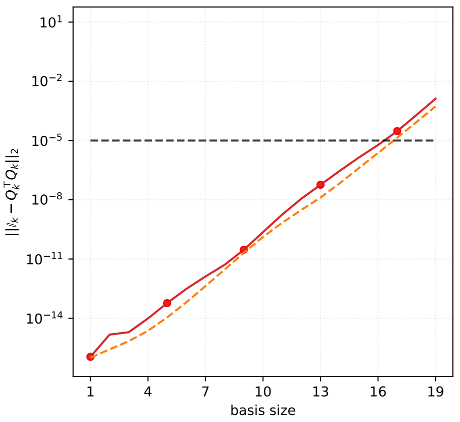


Figure: The LOO of **MGS** vs $\kappa(\mathbf{A}_k)$.

Table of Contents

Conclusion

- description of the TT format:
  - advantages (break of the *curse of dimensionality*);
  - disadvantages (memory growth with tensors operations);
- TN formalism:
  - introduced in the quantum physics community;
  - convenient to describe high order tensors interactions;
- TT application: solution of multilinear systems:
  - 1-site DMRG: optimization-based, convenient for complex problems;
  - TT-MGS: NLA-based, suitable for simpler problems.

References I

[1] T. G. Kolda and B. W. Bader. "Tensor Decompositions and Applications". In: *SIAM Review* 51.3 (Aug. 2009), pp. 455–500. DOI: 10.1137/07070111x.

[2] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen. "A New Truncation Strategy for the Higher-Order Singular Value Decomposition". In: *SIAM Journal on Scientific Computing* 34.2 (2012), A1027–A1052. DOI: 10.1137/110836067.

[3] L. De Lathauwer, B. De Moor, and J. Vandewalle. "A Multilinear Singular Value Decomposition". In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1253–1278. DOI: 10.1137/S0895479896305696.

[4] L. De Lathauwer, B. De Moor, and J. Vandewalle. "On the Best Rank-1 and Rank-$(R_1, R_2, \dots, R_N)$ approximation of high order tensors". In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1324–1342. DOI: 10.1137/S0895479898346995.

References II

[5] L. De Lathauwer. "A Link between the Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization". In: *SIAM Journal on Matrix Analysis and Applications* 28.3 (2006), pp. 642–666. DOI: 10.1137/040608830.

[6] I. V. Oseledets. "Tensor-Train Decomposition". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317. DOI: 10.1137/090752286.

[7] S. Östlund and S. Rommer. "Thermodynamic Limit of Density Matrix Renormalization". In: *Phys. Rev. Lett.* 75 (19 Nov. 1995), pp. 3537–3540. DOI: 10.1103/PhysRevLett.75.3537.

[8] A. Klümper, A. Schadschneider, and J. Zittartz. "Groundstate properties of a generalized VBS-model". In: *Zeitschrift für Physik B Condensed Matter* 87.3 (Oct. 1992), pp. 281–287. DOI: 10.1007/BF01309281.

[9] M. Fannes, B. Nachtergaele, and R. F. Werner. “Finitely correlated states on quantum spin chains”. In: *Communications in Mathematical Physics* 144.3 (Mar. 1992), pp. 443–490. DOI: 10.1007/BF02099178.

[10] G. Vidal. “Efficient Classical Simulation of Slightly Entangled Quantum Computations”. In: *Phys. Rev. Lett.* 91 (14 Oct. 2003), p. 147902. DOI: 10.1103/PhysRevLett.91.147902.

[11] P. Gelß. “The Tensor-Train Format and Its Applications”. PhD thesis. 2017. DOI: 10.17169/refubium-7566.

[12] R. Orús. “A practical introduction to tensor networks: Matrix product states and projected entangled pair states”. In: *Annals of Physics* 349 (2014), pp. 117–158. DOI: https://doi.org/10.1016/j.aop.2014.06.013.

References IV

[13] W. Hackbusch. "Hierarchical Tensor Representation". In: *Tensor Spaces and Numerical Tensor Calculus*. Cham: Springer International Publishing, 2019, pp. 387–451. DOI: 10.1007/978-3-030-35554-8_11.

[14] L. Grasedyck. "Hierarchical Singular Value Decomposition of Tensors". In: *SIAM Journal on Matrix Analysis and Applications* 31.4 (2010), pp. 2029–2054. DOI: 10.1137/090764189.

[15] S. R. White. "Density matrix formulation for quantum renormalization groups". In: *Phys. Rev. Lett.* 69 (19 Nov. 1992), pp. 2863–2866. DOI: 10.1103/PhysRevLett.69.2863.

[16] S. V. Dolgov and D. V. Savostyanov. "Alternating Minimal Energy Methods for Linear Systems in Higher Dimensions". In: *SIAM Journal on Scientific Computing* 36.5 (2014), A2248–A2271. DOI: 10.1137/140953289.

References V

[17] C. Tobler. "Low-rank tensor methods for linear systems and eigenvalue problems". en. PhD thesis. ETH Zurich, 2012. DOI: 10.3929/ETHZ-A-007587832.

[18] S. V. Dolgov. "TT-GMRES: solution to a linear system in the structured tensor format". In: *Russian Journal of Numerical Analysis and Mathematical Modelling* 28.2 (2013), pp. 149–172. DOI: 10.1515/rnam-2013-0009.

[19] Å. Björck. "Solving linear least squares problems by Gram-Schmidt orthogonalization". In: *BIT Numerical Mathematics* 7.1 (Mar. 1967), pp. 1–21. DOI: 10.1007/BF01934122.

[20] O. Coulaud, L. Giraud, and M. Iannacito. *On some orthogonalization schemes in Tensor Train format*. Research Report RR-9491. Inria center at the University of Bordeaux, Nov. 2022.