Approximately Optimal Core Shapes for Tensor Decompositions

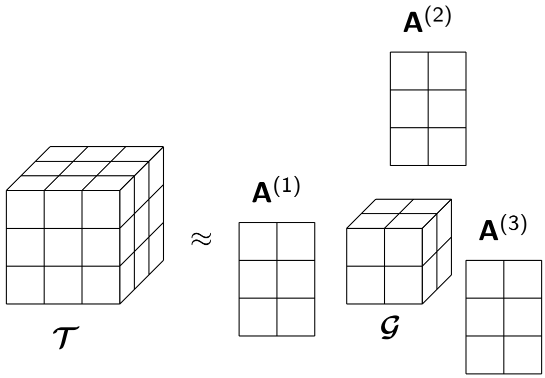
Mehrdad Ghadiri[1] Matthew Fahrbach[2]
Gang Fu[2] Vahab Mirrokni[2]

[1]Georgia Institute of Technology

[2]Google Research

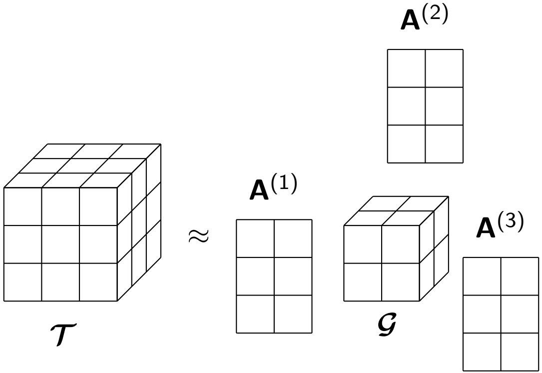
ICML 2023

Tucker Decomposition



Tucker decomposition writes a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ of order N as product of N **factor matrices**, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ for $n \in [N]$, and a **core tensor** $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$.

Tucker Decomposition



Tucker decomposition writes a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ of order N as product of N **factor matrices**, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ for $n \in [N]$, and a **core tensor** $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$.

$$t_{i_1 i_2 \dots i_N} \approx \widehat{t}_{i_1 i_2 \dots i_N} \stackrel{\text{def}}{=} \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} a^{(1)}_{i_1 r_1} \cdots a^{(N)}_{i_N r_N}$$

Problem: Core Shape Selection

Tucker decomposition writes a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ of order N as product of N **factor matrices**, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ for $n \in [N]$, and a **core tensor** $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$.

$$t_{i_1 i_2 \dots i_N} \approx \widehat{t}_{i_1 i_2 \dots i_N} \stackrel{\text{def}}{=} \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} a_{i_1 r_1}^{(1)} \cdots a_{i_N r_N}^{(N)}$$

Problem: Core Shape Selection

Tucker decomposition writes a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ of order N as product of N **factor matrices**, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ for $n \in [N]$, and a **core tensor** $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$.

$$t_{i_1 i_2 \ldots i_N} \approx \widehat{t}_{i_1 i_2 \ldots i_N} \stackrel{\text{def}}{=} \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \ldots r_N} a^{(1)}_{i_1 r_1} \cdots a^{(N)}_{i_N r_N}$$

Tucker decomposition goal: $\min_{\mathcal{G}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}} \|\mathcal{T} - \hat{\mathcal{T}}\|_{\mathrm{F}}^2$

Problem: Core Shape Selection

Tucker decomposition writes a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ of order N as product of N **factor matrices**, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ for $n \in [N]$, and a **core tensor** $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$.

$$t_{i_1 i_2 \ldots i_N} \approx \widehat{t}_{i_1 i_2 \ldots i_N} \stackrel{\text{def}}{=} \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \ldots r_N} a^{(1)}_{i_1 r_1} \cdots a^{(N)}_{i_N r_N}$$

Tucker decomposition goal: $\min_{\mathcal{G}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}} \|\mathcal{T} - \hat{\mathcal{T}}\|_{\mathrm{F}}^2$

$$\|\mathcal{T} - \hat{\mathcal{T}}\|_{\mathrm{F}}^2 = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} (t_{i_1 i_2 \ldots i_N} - \widehat{t}_{i_1 i_2 \ldots i_N})^2$$

Problem: Core Shape Selection

Tucker decomposition writes a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ of order N as product of N **factor matrices**, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ for $n \in [N]$, and a **core tensor** $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$.

$$t_{i_1 i_2 \ldots i_N} \approx \widehat{t}_{i_1 i_2 \ldots i_N} \stackrel{\text{def}}{=} \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \ldots r_N} a^{(1)}_{i_1 r_1} \cdots a^{(N)}_{i_N r_N}$$

Tucker decomposition goal: $\min_{\mathcal{G}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}} \|\mathcal{T} - \hat{\mathcal{T}}\|_{\mathrm{F}}^2$

$$\|\mathcal{T} - \hat{\mathcal{T}}\|_{\mathrm{F}}^2 = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} (t_{i_1 i_2 \ldots i_N} - \widehat{t}_{i_1 i_2 \ldots i_N})^2$$

Problem Statement (Informal)

How to select $R_1, \ldots, R_n$? i.e., the shape of the tensor $\mathcal{G}$.

Importance of Shape of Core Tensor

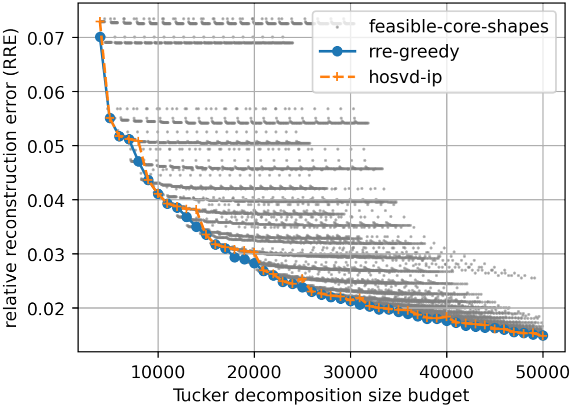


Figure: Pareto frontier of core shapes $\mathbf{r} \in [20]^3$ for hyperspectral tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{1024 \times 1344 \times 33}$. RRE is $L(\boldsymbol{\mathcal{X}}, \mathbf{r}) / \|\boldsymbol{\mathcal{X}}\|_F^2$. RRE-greedy adds to dimensions of $\mathbf{r}$ greedily. HOSVD-IP is our approach that uses integer programming and a surrogate packing problem on higher-order singular values.

Formal Statement of Problem

For $\mathbf{r} = (R_1, \ldots, R_N)$

Formal Statement of Problem

For $\mathbf{r} = (R_1, \ldots, R_N)$ with $R_1 \in [I_1], \ldots, R_N \in [I_N]$,

Formal Statement of Problem

For $\mathbf{r} = (R_1, \ldots, R_N)$ with $R_1 \in [I_1], \ldots, R_N \in [I_N]$, we define

$$L(\mathcal{T}, \mathbf{r}) \stackrel{\text{def}}{=} \min_{\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}, \mathbf{A}^{(1)} \in \mathbb{R}^{I_1 \times R_1}, \ldots, \mathbf{A}^{(N)} \in \mathbb{R}^{I_N \times R_N}} \|\mathcal{T} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)}\|_{\mathrm{F}}^2$$

Formal Statement of Problem

For $\mathbf{r} = (R_1, \ldots, R_N)$ with $R_1 \in [I_1], \ldots, R_N \in [I_N]$, we define

$$L(\mathcal{T}, \mathbf{r}) \stackrel{\text{def}}{=} \min_{\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}, \mathbf{A}^{(1)} \in \mathbb{R}^{I_1 \times R_1}, \ldots, \mathbf{A}^{(N)} \in \mathbb{R}^{I_N \times R_N}} \|\mathcal{T} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)}\|_{\mathrm{F}}^2$$

Problem Statement (formal)

Given a tensor $\mathcal{T}$ and a budget $B > 0$,

$$\begin{aligned} \min \quad & L(\mathcal{T}, \mathbf{r}) \\ \text{s.t.} \quad & \sum_{n \in [N]} I_n R_n + \prod_{n \in [N]} R_n \leq B \end{aligned}$$

Challenges

- Computing $L(\mathcal{T}, \mathbf{r})$ for a given $\mathbf{r}$ is an NP-hard problem.

Challenges

- Computing $L(\mathcal{T}, \mathbf{r})$ for a given $\mathbf{r}$ is an NP-hard problem.
  - We alleviate this issue by using a proxy function that approximates $L(\mathcal{T}, \mathbf{r})$ within a factor of N.

Challenges

- Computing $L(\mathcal{T}, \mathbf{r})$ for a given $\mathbf{r}$ is an NP-hard problem.
  - We alleviate this issue by using a proxy function that approximates $L(\mathcal{T}, \mathbf{r})$ within a factor of N.

$$\widetilde{L}(\mathcal{T}, \mathbf{r}) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2$$

Challenges

- Computing $L(\mathcal{T}, \mathbf{r})$ for a given $\mathbf{r}$ is an NP-hard problem.
    - We alleviate this issue by using a proxy function that approximates $L(\mathcal{T}, \mathbf{r})$ within a factor of N.

$$\widetilde{L}(\mathcal{T}, \mathbf{r}) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2$$

- The constraint $\sum_{n\in[N]} I_n R_n + \prod_{n\in[N]} R_n \leq B$ is non-linear and non-convex.

Challenges

- Computing $L(\mathcal{T}, \mathbf{r})$ for a given $\mathbf{r}$ is an NP-hard problem.
  - We alleviate this issue by using a proxy function that approximates $L(\mathcal{T}, \mathbf{r})$ within a factor of N.

$$\widetilde{L}(\mathcal{T}, \mathbf{r}) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2$$

- The constraint $\sum_{n\in[N]} I_n R_n + \prod_{n\in[N]} R_n \leq B$ is non-linear and non-convex.
  - We use different splits of the budget between the two terms.

Challenges

- Computing $L(\mathcal{T}, \mathbf{r})$ for a given $\mathbf{r}$ is an NP-hard problem.
  - We alleviate this issue by using a proxy function that approximates $L(\mathcal{T}, \mathbf{r})$ within a factor of N.

$$\widetilde{L}(\mathcal{T}, \mathbf{r}) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2$$

- The constraint $\sum_{n\in[N]} I_n R_n + \prod_{n\in[N]} R_n \leq B$ is non-linear and non-convex.
  - We use different splits of the budget between the two terms.

$$\sum_{n\in[N]} I_n R_n \leq B_1$$

$$\prod_{n\in[N]} R_n \leq B - B_1$$

Integer Linear Programming Formulation

$$
\begin{aligned}
\text{maximize} \quad & \sum_{n=1}^{N} \sum_{i_n=1}^{I_n} p_{i_n}^{(n)} x_{i_n}^{(n)} \\
\text{subject to} \quad & \sum_{n=1}^{N} \sum_{i_n=1}^{I_n} \log(i_n) x_{i_n}^{(n)} \leq \log(B_1) \\
& \sum_{n=1}^{N} \sum_{i_n=1}^{I_n} I_n i_n x_{i_n}^{(n)} \leq B - B_1 \\
& \sum_{i_n=1}^{I_n} x_{i_n}^{(n)} = 1 \qquad \forall n \in [N] \\
& x_{i_n}^{(n)} \in \{0, 1\} \qquad \forall n \in [N], i_n \in [I_n]
\end{aligned}
$$

Our Results

1. We show even solving the optimization problem for the proxy function is NP-hard.

Our Results

1. We show even solving the optimization problem for the proxy function is NP-hard.
2. We present a polynomial-time approximation scheme (PTAS) for optimizing the proxy function

Our Results

1. We show even solving the optimization problem for the proxy function is NP-hard.
2. We present a polynomial-time approximation scheme (PTAS) for optimizing the proxy function, i.e., for any fixed $\epsilon > 0$, there is a polynomial time algorithm that finds a $(1 + \epsilon)$-approximation.

Our Results

1. We show even solving the optimization problem for the proxy function is NP-hard.
2. We present a polynomial-time approximation scheme (PTAS) for optimizing the proxy function, i.e., for any fixed $\epsilon > 0$, there is a polynomial time algorithm that finds a $(1 + \epsilon)$-approximation.
3. We give an $(1 + \epsilon) \cdot N$ approximation algorithm for finding the optimal core shape for Tucker decomposition.

Experiments