Overcomplete Tensor Decomposition via Koszul-Young Flattenings

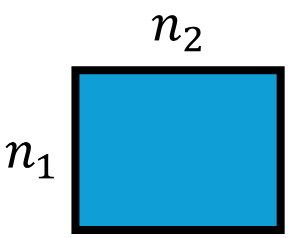
Alex Wein

UC Davis

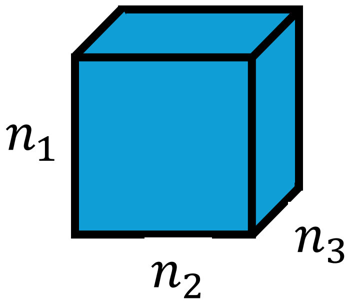
Joint work with Pravesh Kothari (Princeton) and Ankur Moitra (MIT)

arXiv:2411.14344

Tensors



- Order-2 tensor: matrix $M \in \mathbb{R}^{n_1 \times n_2}$ $M = (M_{ij})$
- Order-3 tensor $T \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ $T = (T_{ijk})$
  - Equivalent: $T \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \mathbb{R}^{n_3}$



- $u \in \mathbb{R}^{n_1}$, $v \in \mathbb{R}^{n_2}$, $w \in \mathbb{R}^{n_3}$
- Rank-1 order-2 tensor (matrix) $M = uv^\top$ $M_{ij} = u_i v_j$
- Rank-1 order-3 tensor $T = u \otimes v \otimes w$ $T_{ijk} = u_i v_j w_k$
- $\mathrm{rank}(T)$ = smallest r s.t. T is the sum of r rank-1 tensors

"CP rank"

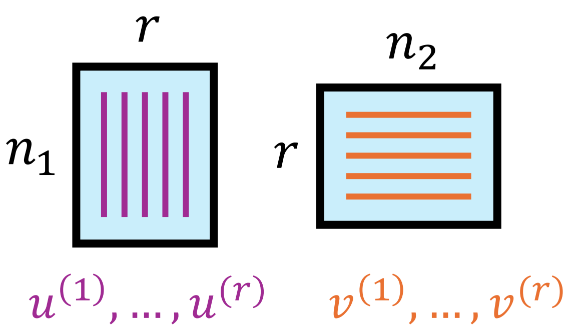
"Simple tensor"

Matrix Decomposition

Polynomial-time algorithm

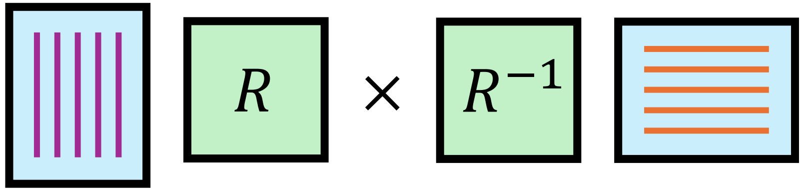
- Given $M \in \mathbb{R}^{n_1 \times n_2}$, can we efficiently compute…
  - … $\mathrm{r} \coloneqq \mathrm{rank}(M)$? Yes (ignoring numerical precision…)
  - … a decomposition of M into r rank-1 terms? Yes
- Using SVD…

$M =$



$$= \sum_{i=1}^{r} u^{(i)} v^{(i)\top}$$

- … but the answer is not unique: $M =$



Tensor Decomposition

- Given $T \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, can we efficiently compute…
  - … $\mathrm{r} \coloneqq \mathrm{rank}(T)$? NP-hard [Håstad '90; Hillar, Lim '09]
  - … a decomposition of T into r rank-1 terms?
- Can still hope to decompose "most" rank-r tensors (for small r)

$$T = \sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}$$

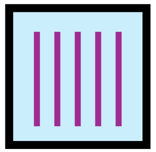
with $u^{(i)}, v^{(i)}, w^{(i)}$ chosen *generically*

Probability 1

- Tensor decompositions (of order ≥ 3) are often unique!
  - Inherent ambiguity: $u \otimes v \otimes w = 2u \otimes \frac{1}{2} v \otimes w$

Formal Meaning of "Generic"

- A predicate $P(x)$ depending on formal variables $x = (x_1, \dots, x_m)$ is "true generically" if there exists a not-identically-zero polynomial $g(x)$ such that: $g(x) \neq 0 \;\Rightarrow\; P(x)$
  - Implies that $P(x)$ holds for "almost all" x
- Ex: Generically chosen vectors $\underbrace{u^{(1)}, \dots, u^{(n)}}_{x} \in \mathbb{R}^n$ are linearly independent
- Pf: $\underbrace{\text{Lin. indep.}}_{P(x)} \Leftrightarrow \underbrace{\det M}_{g(x)} \neq 0$

$M =$ 

Motivation

- Why care about tensor decomposition?
- 2 answers…
  1) Fundamental problem, needs no justification ☺
  2) Applications in statistics / data science
     - Phylogenetic reconstruction, topic modeling, community detection, learning Gaussian mixtures, independent component analysis, dictionary learning, …
     - Either used for analyzing tensor-valued data (higher-order PCA), or method of moments

↳ Imagine samples drawn from a distribution $y \sim D$ where $y \in \mathbb{R}^n$

First moment: $\mathbb{E}[y] = \vec{0}$ (assume centered)

Second moment: $\mathbb{E}[yy^\top]$ ($n \times n$ covariance matrix)

Third moment: $\mathbb{E}[y \otimes y \otimes y]$ ($n \times n \times n$ tensor)

Prior Work

- Goal: given an $n \times n \times n$ tensor $T = \sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}$ with generic components, want an algorithm to provably recover the rank-1 terms in polynomial time (as $n \to \infty$)
  - Decomposition is unique provided $r \leq cn^2$ [Bocci, Chiantini, Ottaviani '13]
  - Classical (~1970) "Jennrich's algorithm": $r \leq n$ ("undercomplete")
  - [Chen, Rademacher '20] $r = n + O(1)$
  - [Koiran '24] $r \leq \frac{4}{3}n$
  - Our result [Kothari, Moitra, W. '24] $r \leq (2-\epsilon)n$
    - Runtime $n^{C(\epsilon)}$
    - More generally, for $n_1 \times n_2 \times n_3$ tensor with $n_3 \geq n_2 \geq n_1$ and $n_1 \to \infty$ and $n_2 \asymp n_3$: $r \leq (1-\epsilon)(n_2+n_3)$
  - For *random* components, can reach $r \approx n^{3/2}$ [Ma, Shi, Steurer '16]

"overcomplete": $r > n$

Some proof ideas…

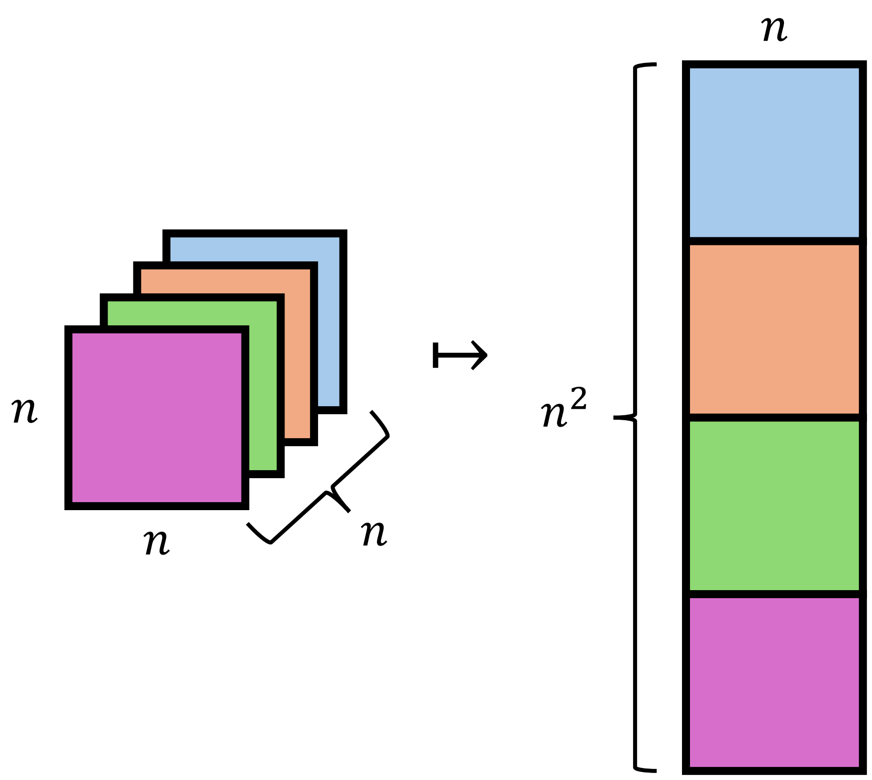
Rank Detection

- For simplicity, consider an easier task…
- Goal: given an $n \times n \times n$ tensor $T = \sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}$ with generic components, want an algorithm to provably ~~recover the rank-1 terms~~ compute r in polynomial time (as $n \to \infty$)
  - [Persu '18] $r \leq \frac{3}{2}n$
  - Our result [Kothari, Moitra, W. '24] $r \leq (2-\epsilon)n$
- Approach: construct a map ("flattening") $T \mapsto M(T)$

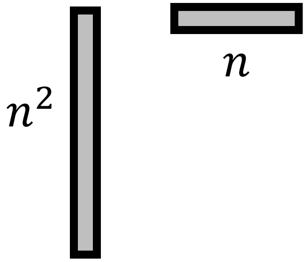
  T: $n \times n \times n$ $\qquad$ $M(T)$: $N_1 \times N_2$

- Hope: $\mathrm{rank}(T)$ can be deduced from $\mathrm{rank}(M(T))$

"Trivial" Flattening



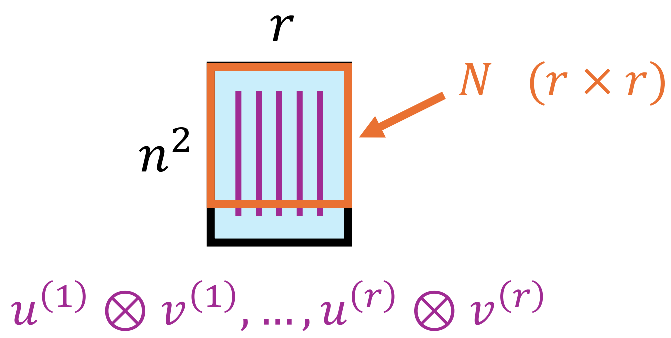
- Flatten $n \times n \times n$ tensor to $n^2 \times n$ matrix:
- Rank-1 $T \mapsto$ Rank-1 M
  - $u \otimes v \otimes w \mapsto (u \otimes w)v^\top$



- Rank-r $T \mapsto$ Rank-? M
  - Answer: Rank-r … provided $r \leq n$ and components are generic
- Solves rank detection for $r \leq n$ (we'll beat this by a factor of 2)

Proof: Rank Detection by Trivial Flattening

- $f\colon u \otimes v \otimes w \mapsto (u \otimes w)v^{\top}$
- Goal: $\operatorname{rank} f\left(\sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}\right) = r$ (if $r \leq n$ and generic)
- $f\left(\sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}\right) = \sum_{i=1}^{r} (u^{(i)} \otimes v^{(i)})\,(w^{(i)})^{\top}$
- Sufficient: $\{u^{(i)} \otimes v^{(i)}\}$ and $\{w^{(i)}\}$ each linearly independent
- $N(u, v, w)$ symbolic matrix
- Suff: $\det N \neq 0$ for generic u, v, w
- Suff: $\det N(u, v, w) \not\equiv 0$ as a polynomial in u, v, w
- Suff: $\exists u, v, w : \det N(u, v, w) = 0$

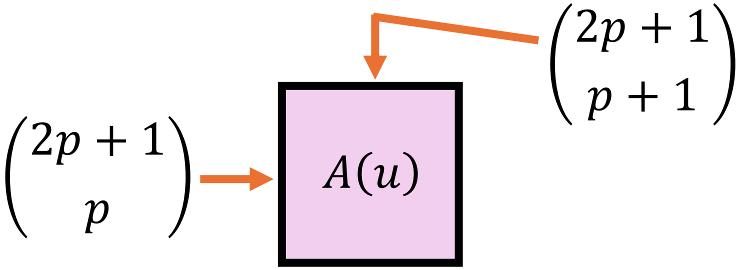


From Rank Detection to Decomposition

- $f\colon u \otimes v \otimes w \mapsto (u \otimes w)v^\top$
- $M = f\left(\sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}\right)$
- Know: $\operatorname{rank} M = r$ (if $r \leq n$ and generic)
- Goal: recover $u^{(i)}, v^{(i)}, w^{(i)}$
- $\operatorname{colspan} M = ?$
- Answer: $\operatorname{span}\{u^{(i)} \otimes v^{(i)}\}$
- Search for simple tensors (rank-1 matrices) in this linear subspace
- [Johnston, Lovitz, Vijayaraghavan '22] Can find the r rank-1 $m \times n$ matrices that span a given subspace, provided $r \leq \frac{1}{4}(m-1)(n-1)$
- Forthcoming (w. Jeshu Dastidar & Tait Weicht) improvement $\frac{1}{4}$ to $\frac{1}{2}$

Factor-2 Improvement: A Better Flattening

- Inspired by "Koszul-Young flattening" [Landsberg, Ottaviani '13]
- Parameter $p \geq 1$ (integer; large constant)
- Linear map $T \mapsto M(T)$
- $u \otimes v \otimes w \mapsto A(u) \otimes (vw^\top)$



E.g. $p = 1$: $A(u) = $

$$\begin{array}{c|ccc} & \{1,2\} & \{1,3\} & \{2,3\} \\ \hline \{1\} & -u_2 & -u_3 & 0 \\ \{2\} & u_1 & 0 & -u_3 \\ \{3\} & 0 & u_1 & u_2 \end{array}$$

- $\mathrm{rank}(M(T)) = \binom{2p}{p} \mathrm{rank}(T)$ when $r \leq (2 - \epsilon)n$

$\binom{2p}{p}$ → $\mathrm{rank}(A)$

Looking Forward…

- Is $\mathrm{r} \approx 2n$ the limit (for efficient algorithms)?
- Recall: for *random* tensors, can do $\mathrm{r} \approx n^{3/2}$
- For order-4 tensors, no gap between random and generic ($\mathrm{r} \approx n^2$)
- Lower bounds, building on [Efremenko, Garg, Oliveira, Wigderson '17]
  - Linear flattenings of the form $u \otimes v \otimes w \mapsto A(u) \otimes (vw^\top)$ cannot surpass $2n$
  - General linear flattenings cannot surpass $6n$
  - Degree-d polynomial flattenings cannot surpass $C_d n$
- ???

Thanks!