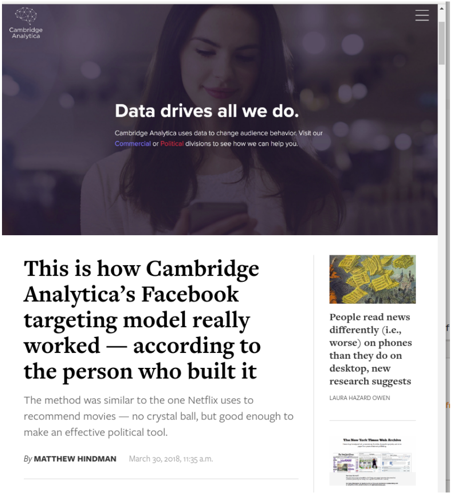
Low-Rank Matrix and Tensor Approximation

Daniel Kressner

Institute of Mathematics
daniel.kressner@epfl.ch
http://anchp.epfl.ch

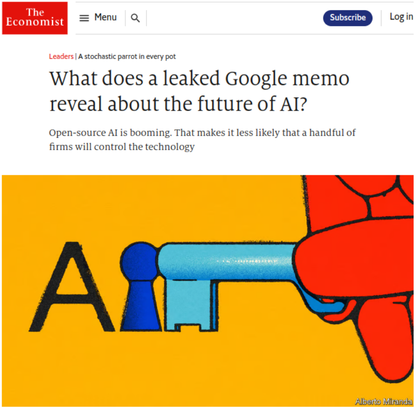
Computational Mathematics for Data Science
DTU 2023

From http://www.niemanlab.org



... his [Aleksandr Kogan's] message went on to confirm that his approach was indeed similar to SVD or other matrix factorization methods, like in the Netflix Prize competition, and the Kosinki-Stillwell-Graepel Facebook model. Dimensionality reduction of Facebook data was the core of his model.

Leaked Internal Google Document, May 2023



But the uncomfortable truth is, we aren't positioned to win this arms race and neither is OpenAI. While we've been squabbling, a third faction has been quietly eating our lunch... Open-source models are faster, more customizable, more private, and pound-for-pound more capable. They are doing things with $100 and 13B params that we struggle with at $10M and 540B. And they are doing so in weeks, not months.

...

In both cases, low-cost public involvement was enabled by a vastly cheaper mechanism for fine tuning called low rank adaptation, or LoRA [arXiv:2106.09685] ...

Rest of this tutorial

1. Foundations

   Low-rank matrix approximation algorithms
2. Deterministic Sampling
3. Stochastic Sampling
4. Tensors
5. Alternating Optimization
6. Riemannian Optimization

1. Foundations

- Matrix rank
- SVD
- Best low-rank approximation
- Low-rank and subspace approximation
- When (not) to expect good low-rank approximations
- Stability considerations

References: [Golub/Van Loan'2013][1], [Horn/Johnson'2013][2]

[1] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, 2013.

[2] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 2013.

Rank and basic properties

Let $A \in \mathbb{R}^{m \times n}$. Then

$$\operatorname{rank}(A) := \dim(\operatorname{range}(A)).$$

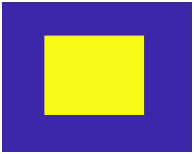
Rank and basic properties

Let $A \in \mathbb{R}^{m \times n}$. Then

$$\mathrm{rank}(A) := \dim(\mathrm{range}(A)).$$

Quiz

1. What is the rank of this matrix?



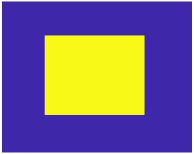
Rank and basic properties

Let $A \in \mathbb{R}^{m \times n}$. Then

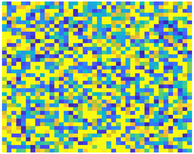
$$\text{rank}(A) := \dim(\text{range}(A)).$$

Quiz

1. What is the rank of this matrix?



2. What is the rank of `randn(40)`?



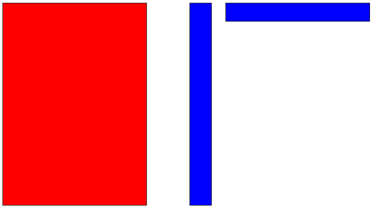
Rank and matrix factorizations

Lemma. A matrix $A \in \mathbb{R}^{m\times n}$ of rank r admits a factorization of the form

$$A = BC^T, \quad B \in \mathbb{R}^{m\times r}, \quad C \in \mathbb{R}^{n\times r}.$$

We say that A has low rank if $\operatorname{rank}(A) \ll m, n$.

Illustration of low-rank factorization:



	A	BC^T
#entries	mn	$mr + nr$

- Generically (and in most applications), A has full rank, that is, $\operatorname{rank}(A) = \min\{m, n\}$.
- Aim instead at approximating A by a low-rank matrix.

The singular value decomposition

Theorem (SVD). Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. Then there are orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ such that

$$A = U\Sigma V^T, \quad \text{with} \quad \Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & 0 & \end{bmatrix} \in \mathbb{R}^{m\times n}$$

and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

- $\sigma_1, \ldots, \sigma_n$ are called singular values
- $u_1, \ldots, u_n$ are called *left* singular vectors
- $v_1, \ldots, v_n$ are called *right* singular vectors
- $Av_i = \sigma_i u_i$, $A^T u_i = \sigma_i v_i$ for $i = 1, \ldots, n$.
- Singular values are always uniquely defined by A.
- Singular values are *never* unique. If $\sigma_1 > \sigma_2 > \cdots \sigma_n > 0$ then unique up to $u_i \leftarrow \pm u_i$, $v_i \leftarrow \pm v_i$.

The singular value decomposition

Theorem (SVD). Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. Then there are orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ such that

$$A = U\Sigma V^T, \quad \text{with} \quad \Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & 0 & \end{bmatrix} \in \mathbb{R}^{m\times n}$$

and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

Quiz: Which properties of A can be extracted from the SVD?

The singular value decomposition

Theorem (SVD). Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. Then there are orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ such that

$$A = U\Sigma V^T, \quad \text{with} \quad \Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & 0 & \end{bmatrix} \in \mathbb{R}^{m\times n}$$

and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

Quiz: Which properties of A can be extracted from the SVD?

$r = \text{rank}(A) =$ number of nonzero singular values of A,

$\mathrm{kernel}(A) = \mathrm{span}\{v_{r+1}, \ldots, v_n\}$, $\text{range}(A) = \mathrm{span}\{u_1, \ldots, u_r\}$

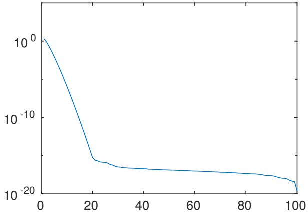
$\|A\|_2 = \sigma_1$, $\|A^\dagger\|_2 = 1/\sigma_r$, $\|A\|_F^2 = \sigma_1^2 + \cdots + \sigma_n^2$

$\sigma_1^2, \ldots, \sigma_n^2$ eigenvalues of AA^T and A^TA.

SVD: Computational aspects

- Standard implementations (LAPACK, Matlab's `svd`, ...) require $\mathcal{O}(mn^2)$ operations to compute (economy size) SVD of $m \times n$ matrix A.
- Beware of roundoff error when interpreting singular value plots.

  Example: `semilogy(svd(hilb(100)))`



  - Kink is caused by roundoff error and does not reflect true behavior of singular values.
  - Exact singular values are known to decay exponentially.[3]
  - *Sometimes* more accuracy possible.[4].

[3]Beckermann, B. The condition number of real Vandermonde, Krylov and positive definite Hankel matrices. Numer. Math. 85 (2000), no. 4, 553–577.

[4]Drmač, Z.; Veselić, K. New fast and accurate Jacobi SVD algorithm. I. SIAM J. Matrix Anal. Appl. 29 (2007), no. 4, 1322–1342

Best low-rank approximation

For $k < n$, partition SVD as

$$U\Sigma V^T = \begin{bmatrix} U_k & * \end{bmatrix} \begin{bmatrix} \Sigma_k & 0 \\ 0 & * \end{bmatrix} \begin{bmatrix} V_k & * \end{bmatrix}^T, \quad \Sigma_k = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}$$

Rank-k truncation:

$$A \approx \mathcal{T}_k(A) := U_k \Sigma_k V_k^T.$$

has rank at most k. By unitary invariance of $\|\cdot\| \in \{\|\cdot\|_2, \|\cdot\|_F\}$:

$$\|\mathcal{T}_k(A) - A\| = \big\|\mathrm{diag}(0,\dots,0,\sigma_{k+1},\dots,\sigma_n)\big\|.$$

In particular:

$$\|A - \mathcal{T}_k(A)\|_2 = \sigma_{k+1}, \qquad \|A - \mathcal{T}_k(A)\|_F = \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_n^2}.$$

Nearly equal iff singular values decay quickly.

Best low-rank approximation

Theorem (Schmidt-Mirsky). Let $A \in \mathbb{R}^{m \times n}$. Then

$$\|A - \mathcal{T}_k(A)\| = \min\left\{\|A - B\| : \ B \in \mathbb{R}^{m\times n} \text{ has rank at most } k\right\}$$

holds for any unitarily invariant norm $\|\cdot\|$.

Proof: See Section 7.4.9 in [Horn/Johnson'2013] for general case.

Proof for $\|\cdot\|_F$: Let $\sigma(A), \sigma(B)$ denote the vectors of singular values of A and B and use the matrix inner product $\langle A, B\rangle = \text{trace}(B^T A)$. Then von Neumann's trace inequality states that

$$|\langle A, B\rangle| \leq \langle \sigma(A), \sigma(B)\rangle$$

Hence,

$$\begin{aligned} \|A - B\|_F^2 &= \langle A - B, A - B\rangle = \|A\|_F^2 - 2\langle A, B\rangle + \|B\|_F^2 \\ &\geq \|\sigma(A)\|_2^2 - 2\langle \sigma(A), \sigma(B)\rangle + \|\sigma(B)\|_2^2 \\ &= \sum_{i=1}^{n} (\sigma_i(A) - \sigma_i(B))^2 \geq \|A - \mathcal{T}_k(A)\|_F^2. \end{aligned}$$

Best low-rank approximation

Theorem (Schmidt-Mirsky). Let $A \in \mathbb{R}^{m\times n}$. Then

$$\|A - \mathcal{T}_k(A)\| = \min\left\{\|A - B\| : \ B \in \mathbb{R}^{m\times n} \text{ has rank at most } k\right\}$$

holds for any unitarily invariant norm $\|\cdot\|$.

Quiz. Is the best rank-k approximation unique if $\sigma_k > \sigma_{k+1}$?

Best low-rank approximation

Theorem (Schmidt-Mirsky). Let $A \in \mathbb{R}^{m \times n}$. Then

$$\|A - \mathcal{T}_k(A)\| = \min\left\{\|A - B\| : \; B \in \mathbb{R}^{m \times n} \text{ has rank at most } k\right\}$$

holds for any unitarily invariant norm $\|\cdot\|$.

Quiz. Is the best rank-k approximation unique if $\sigma_k > \sigma_{k+1}$?

- If $\sigma_k > \sigma_{k+1}$ best rank-k approximation unique wrt $\|\cdot\|_F$.
- Wrt $\|\cdot\|_2$ only unique if $\sigma_{k+1} = 0$. For example, $\mathrm{diag}(2, 1, \epsilon)$ with $0 < \epsilon < 1$ has infinitely many best rank-two approximations:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 2-\epsilon/2 & 0 & 0 \\ 0 & 1-\epsilon/2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 2-\epsilon/3 & 0 & 0 \\ 0 & 1-\epsilon/3 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ldots.$$

- If $\sigma_k = \sigma_{k+1}$ best rank-k approximation never unique. I_3 has several best rank-two approximations:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Some uses of low-rank approximation

- Data compression.
- Fast solvers for linear systems: Kernel matrices, integral operators, under the hood of sparse direct solvers (MUMPS, PaStiX), . . .
- Fast solvers for dynamical systems: Dynamical low-rank method.
- Low-rank compression / training of neural nets.
- Defeating quantum supremacy claims by Google/IBM. Science'2022:

NEWS | PHYSICS

Ordinary computers can beat Google's quantum computer after all

Superfast algorithm put crimp in 2019 claim that Google's machine had achieved "quantum supremacy"

2 AUG 2022 · 5:05 PM ET · BY ADRIAN CHO

Approximating the range of a matrix

Aim at finding a matrix $Q \in \mathbb{R}^{m \times k}$ with orthonormal columns such that

$$\mathrm{range}(Q) \approx \mathrm{range}(A).$$

QQ^T is orthogonal projector onto range(Q) $\rightsquigarrow$ Aim at solving

$$\min \left\{ \|A - QQ^T A\| : Q^T Q = I_k \right\}$$

for $\|\cdot\| \in \{\|\cdot\|_2, \|\cdot\|_F\}$. Because $\mathrm{rank}(QQ^T A) \le k$,

$$\|A - QQ^T A\| \ge \|A - \mathcal{T}_k(A)\|.$$

Setting $Q = U_k$ one obtains

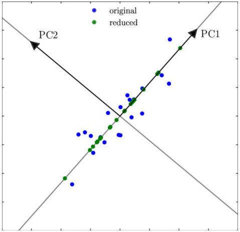
$$U_k U_k^T A = U_k U_k^T U \Sigma V^T = U_k \Sigma_k V_k^T = \mathcal{T}_k(A).$$

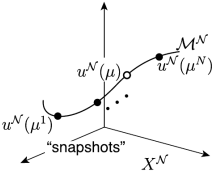
$\rightsquigarrow$ $Q = U_k$ is optimal.

Low-rank approximation and range approximation
are essentially the same tasks!

Two popular uses of range approximation

Principal component analysis (PCA): Dominant left singular vectors of data matrix $X = [x_1, \ldots, x_n]$ (with mean subtracted) provide directions of maximum variance, 2nd maximum variance, etc.





Proper orthogonal decomposition (POD), reduced basis methods: Collect snapshots of time-dependent and/or parameter-dependent equations and perform model reduction by projection to dominant left singular vectors U_k of snapshot matrix.

When to expect good low-rank approximations

Smoothness.

Example 1: Snapshot matrix with snapshots depending smoothly on time/parameter

$$\begin{aligned} A &= \begin{bmatrix} u(t_1) & u(t_2) & \cdots & u(t_n) \end{bmatrix} \\ &\approx \underbrace{\begin{bmatrix} p_1 & p_2 & \cdots & p_k \end{bmatrix}}_{\text{low-dim. polynomial basis}} \times \underbrace{\begin{bmatrix} \ell_1(t_1) & \ell_1(t_2) & \cdots & \ell_1(t_n) \\ \ell_2(t_1) & \ell_2(t_2) & \cdots & \ell_2(t_n) \\ \vdots & \vdots & & \vdots \\ \ell_2(t_1) & \ell_2(t_2) & \cdots & \ell_2(t_n) \end{bmatrix}}_{\text{Vandermonde-like matrix}} \end{aligned}$$

where $u(t) \approx p(t) = p_1\ell_1(t) + \cdots + p_n\ell_n(t)$ (polynomial approximation of degree k in basis of Lagrange polynomials).

If $u : [-1, 1] \to \mathbb{R}^n$ admits analytic extension to Bernstein ellipse $\mathcal{E}_\rho$ (focii ± 1 and sum of half axes equal to $\rho > 1$) then polynomial approximation implies

$$\sigma_k(A) \lesssim \max_{z \in \mathcal{E}_\rho} \|u(z)\|_2 \cdot \rho^{-k}.$$

Exponential decay of singular values!

When to expect good low-rank approximations

Smoothness.

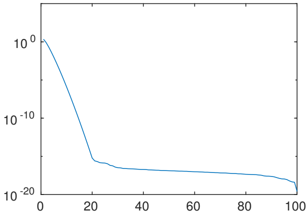
Example 2: Kernel matrix for smooth (low-dimensional) kernel:

$$K = \begin{bmatrix} \kappa(x_1, x_1) & \cdots & \kappa(x_1, x_n) \\ \vdots & & \vdots \\ \kappa(x_n, x_1) & \cdots & \kappa(x_n, x_n) \end{bmatrix}, \quad \kappa : \Omega \times \Omega \to \mathbb{R}.$$

Hilbert matrix:

$$K = \left[\frac{1}{i + j - 1} \right]_{i,j=1}^{n}$$

Kernel $\kappa(x, y) = 1/(x + y - 1)$.



Exponential singular value decay established through Taylor expansion [Börm'2010] or exponential sum approximation [Braess/Hackbusch'2005]:

$$\frac{1}{x + y} \approx \sum_{i=1}^{k} \gamma_i \exp(\beta_i (x + y)) = \sum_{i=1}^{k} \gamma_i \exp(\beta_i x) \cdot \exp(\beta_i y).$$

When to expect good low-rank approximations

Algebraic structure.

If X satisfies low-rank Sylvester matrix equation:

$$AX + XB = \text{low rank}$$

and spectra of A, B are disjoint then singular values of X (usually) decay exponentially[5].

- Basis of fast solvers for matrix equations.
- Captures many structured matrices: Vandermonde, Cauchy, Pick, ... matrices, canoncial Krylov bases, ...

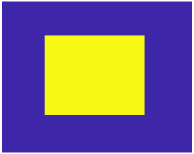
[5] Bernhard Beckermann and Alex Townsend. "On the singular values of matrices with displacement structure". In: *SIAM J. Matrix Anal. Appl.* 38.4 (2017), pp. 1227–1248.

When *not* to expect good low-rank approximations

In most over situations:

- Kernel matrices with singular/non-smooth kernels
- Snapshot matrices for time-dependent / parametrized solutions featuring a slowly decaying Kolmogoroff N-width.
- Images
- White noise
- ...

$\exists$ Exceptions to these rules:



Also: Low-rank methods are often used even when there is no notable singular value decay in, e.g., statistical inference.

When *not* to expect good low-rank approximations

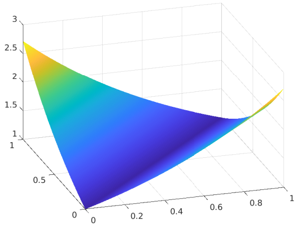
Consider kernel matrix

$$K = \begin{bmatrix} \kappa(x_1, x_1) & \cdots & \kappa(x_1, x_n) \\ \vdots & & \vdots \\ \kappa(x_n, x_1) & \cdots & \kappa(x_n, x_n) \end{bmatrix}, \quad \kappa : D \times D \to \mathbb{R}.$$

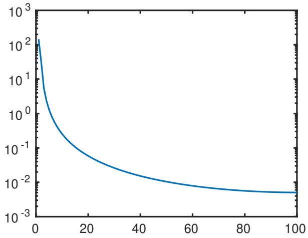
for 1D-kernel κ with diagonal singularity/non-smoothness. Example:

$$\kappa(x, y) = \exp(-|x - y|), \quad x, y \in [0, 1]$$

Function



Singular values

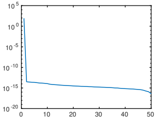


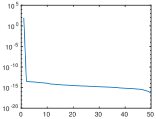
But not everything is lost..

Block partition K. Level 1:

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} =$$

K_{11}

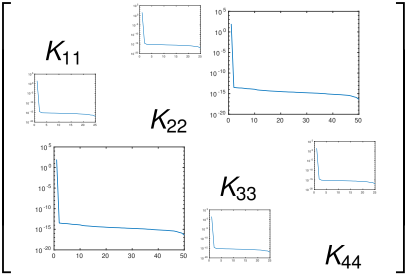




K_{22}

But not everything is lost..

Block partition K. Level 2:

$K =$ 

etc. $\rightsquigarrow$ HODLR. More general constructions [Hackbusch'2015]:

- $\mathcal{H}$-matrices $=$ general recursive block partition.
- HSS/$\mathcal{H}^2$-matrices impose additional nestedness conditions on the low-rank factors on different levels of the recursion.

Exciting news: Recovery of such matrices from mat-vec products[6].

[6] D. Halikias and A. Townsend. *Structured matrix recovery from matrix-vector products*. arXiv:2212.09841. 2022, J. Levitt and P. G. Martinsson. *Linear-complexity black-box randomized compression of rank-structured matrices*. arXiv:2205.02990. 2022.

Stability considerations

What happens to SVD if A is perturbed by noise (roundoff error, ...)?

Weyl's inequality:

$$|\sigma_i(A+E) - \sigma_i(A)| \le \|E\|_2.$$

Singular values are perfectly well conditioned.

Singular vectors tend to be less stable! Example:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1+\varepsilon \end{bmatrix}, \qquad E = \begin{bmatrix} 0 & \varepsilon \\ \varepsilon & -\varepsilon \end{bmatrix}.$$

- A has right singular vectors $\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.
- $A+E$ has right singular vectors $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Wedin'1972: Error in U_k, $V_k \lesssim \varepsilon/[\sigma_k(A) - \sigma_{k+1}(A)]$.

Bad news for stability of low-rank approximation?

Stability of low-rank approximation

Lemma. Let $A \in \mathbb{R}^{m\times n}$ have rank $\leq k$. Then

$$\|\mathcal{T}_k(A+E) - A\| \leq C\|E\|$$

holds with $C = 2$ for any unitarily invariant norm $\|\cdot\|$. For the Frobenius norm, the constant can be improved to $C = (1+\sqrt{5})/2$.

Proof. Schmidt-Mirsky gives $\|\mathcal{T}_k(A+E) - (A+E)\| \leq \|E\|$. Triangle inequality implies

$$\|\mathcal{T}_k(A+E) - (A+E) + (A+E) - A\| \leq 2\|E\|.$$

Second part is result by Hackbusch[7]. $\square$

Implication for general matrix A:

$$\begin{aligned}\|\mathcal{T}_k(A+E) - \mathcal{T}_k(A)\| &= \big\|\mathcal{T}_k\big(\mathcal{T}_k(A) + (A - \mathcal{T}_k(A)) + E\big) - \mathcal{T}_k(A)\big\| \\ &\leq C\|(A - \mathcal{T}_k(A)) + E\| \leq C(\|A - \mathcal{T}_k(A)\| + \|E\|).\end{aligned}$$

Perturbations on the level of truncation error pose no danger.

[7] Hackbusch, W. New estimates for the recursive low-rank truncation of block-structured matrices. Numer. Math. 132 (2016), no. 2, 303–328

Low-rank matrix approximation algorithms

Landscape of algorithms

Landscape of algorithms

Choice of algorithm for performing low-rank approximation of A depends critically on how A is accessed:

1. Small matrices: If $m, n = O(10^2)$, don't think twice, apply svd.
2. Mat-vecs: A is accessed through matrix-vector products $v \mapsto Av$. massive dense matrices, sparse matrices, implicit representation (e.g., through matrix functions, Schur complements, ...).

   Randomized SVD and friends (e.g., block Lanczos)
   Talk by Yuji Nakatsukasa
3. Entry-by-entry: Individual entries $A(i, j)$ can be directly computed but it is too expensive to compute/hold the whole matrix. kernel matrices, distances matrices, discretizations of nonlocal equations (integral eqns, fractional diff eqns), ...

   Sampling-based techniques.
4. Semi-analytical techniques: Polynomial approximation, exponential sum approximation, Random Fourier features.
5. Implicit: A satisfies linear system/eigenvalue problem/opt problem/...

   Alternating optimization, Riemannian optimization, ...

2. Deterministic sampling

Sampling based approximation

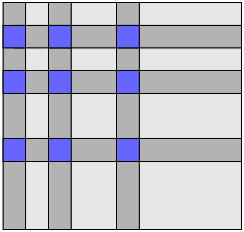
Aim: Obtain rank-r approximation of $m \times n$ matrix A from selected entries of A.
Two different situations:

- Unstructured sampling: Let $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$. Solve

$$\min \|A - BC^T\|_\Omega, \qquad \|M\|_\Omega^2 = \sum_{(i,j) \in \Omega} m_{ij}^2.$$

  Matrix completion problem solved by general optimization techniques (ALS, Riemannian optimization, convex relaxation).

- Column/row sampling:



  Focus of this part.

Row selection from orthonormal basis

Task. Given orthonormal basis $U \in \mathbb{R}^{n \times r}$ find a "good" $r \times r$ submatrix of U.

Classical problem already considered by Knuth.[8]

Quantification of "good": Smallest singular value not too small.

Some notation:

- Given an $m \times n$ matrix A and index sets

$$\begin{aligned} I &= \{i_1, \dots, i_k\}, \quad 1 \le i_1 < i_2 < \cdots i_k \le m, \\ J &= \{j_1, \dots, j_\ell\}, \quad 1 \le j_1 < j_2 < \cdots j_\ell \le n, \end{aligned}$$

  we let

$$A(I,J) = \begin{pmatrix} a_{i_1,j_1} & \cdots & a_{i_1,j_n} \\ \vdots & & \vdots \\ a_{i_m,j_1} & \cdots & a_{i_m,j_n} \end{pmatrix} \in \mathbb{R}^{k \times \ell}.$$

  The full index set is denoted by $:$, e.g., $A(I,:)$.
- $|\det A|$ denotes the volume of a *square* matrix A.

[8] Knuth, Donald E. Semioptimal bases for linear dependencies. Linear and Multilinear Algebra 17 (1985), no. 1, 1–4.

Row selection from orthonormal basis

Lemma (Maximal volume yields good submatrix)

Let index set I, $\#I = r$, be chosen such that $|det(U(I,:))|$ is maximized among all $r \times r$ submatrices. Then

$$\frac{1}{\sigma_{\min}(U(I,:))} \leq \sqrt{r(n-r)+1}$$

Proof.[9] W.l.o.g. $I = \{1, \ldots, r\}$. Consider

$$\tilde{U} = UU(I,:)^{-1} = \begin{pmatrix} I_r \\ B \end{pmatrix}.$$

Because of $\det \tilde{U}(J,:) = \det U(J,:)/\det U(I,:)$ for any J, submatrix $\#J = r$, $\tilde{U}(I,:)$ has maximal volume among all $r \times r$ submatrices of $\tilde{U}$.

[9] Following Lemma 2.1 in [Goreinov, S. A.; Tyrtyshnikov, E. E.; Zamarashkin, N. L. A theory of pseudoskeleton approximations. Linear Algebra Appl. 261 (1997), 1–21].

Maximality of $\tilde{U}(I,:)$ implies $\max|b_{ij}| \le 1$. Argument: If there was b_{ij} with $|b_{ij}| > 1$ then interchanging rows $r+i$ and j of $\tilde{U}$ would increase volume of $\tilde{U}(I,:)$.

We have

$$\|B\|_2 \le \|B\|_F \le \sqrt{(n-r)r}\max|b_{ij}| \le \sqrt{(n-r)r}.$$

This yields the result:

$$\|U(I,:)^{-1}\|_2 = \|UU(I,:)^{-1}\|_2 = \sqrt{1+\|B\|_2^2} \le \sqrt{1+(n-r)r}.$$

Greedy row selection from orthonormal basis

Finding submatrix of maximal volume is NP hard.[10]

Greedy algorithm (column-by-column):[11]

- First step is easy: Choose i such that $|u_{i1}|$ is maximal.
- Now, assume that $k < r$ steps have been performed and the first k columns have been processed. Task: Choose optimal index in column $k + 1$.

There is a one-to-one connection between greedy row selection and Gaussian elimination with column pivoting!

[10] Civril, A., Magdon-Ismail, M.: On selecting a maximum volume sub-matrix of a matrix and related problems. Theoret. Comput. Sci. 410(47-49), 4801–4811 (2009)

[11] Reinvented multiple times in the literature.

Greedy row selection from orthonormal basis

Simplified form of Gaussian elimination with column pivoting:

Input: $n \times r$ matrix U
Output: "Good" index set $I \subset \{1, \ldots, n\}$, $\#I = r$.

Set $I = \emptyset$.
for $k = 1, \ldots, r$ **do**
  Choose $i^* = \text{argmax}_{i=1,\ldots,n} |u_{ik}|$.
  Set $I \leftarrow I \cup i^*$.
  $U \leftarrow U - \frac{1}{u_{i^*,k}} U(:,k)U(i^*,:)$
end for

Theorem

For the index set returned by greedy algorithm applied to orthnormal $U \in \mathbb{R}^{n \times r}$, it holds that

$$\|U(I,:)^{-1}\|_2 \leq \sqrt{nr}2^{r-1}.$$

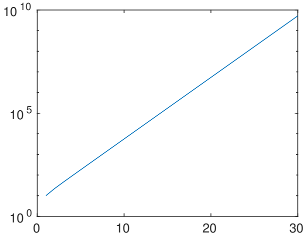
Performance of greedy algorithm in practice often quite good, although this bound is sharp.

Counter example for greedy

Let U be Q-factor of economy sized QR factorization of $n \times r$ matrix

$$A = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ \vdots & \ddots & \ddots & \\ -1 & \cdots & -1 & 1 \\ -1 & \cdots & -1 & -1 \\ \vdots & & \vdots & \vdots \\ -1 & \cdots & -1 & -1 \end{pmatrix}$$

Variation of famous example by Wilkinson. Greedy performs no pivoting, at least in exact arithmetic.



$\|U(I,:)^{-1}\|_2$ vs. r for $n = 100$ returned by greedy.

Improvements over greedy

Improve upon maxvol-based greedy (in a deterministic framework) via Knuth's iterative exchange of rows. Given index set I, $\#I = r$, and $\mu \geq 1$, $\mu \approx 1$, form

$$\tilde{U} = UU(I,:)^{-1}.$$

Search for largest element

$$(i^*, j^*) = \text{argmax}|\tilde{u}_{ij}|.$$

If

$$|\tilde{u}_{i^*j^*}| \leq \mu, \tag{1}$$

terminate algorithm. Otherwise, set $I \leftarrow I \backslash \{j^*\} \cup \{i^*\}$ and repeat.

Alternative: Apply existing methods for rank-revealing QR to U^T [Golub/Van Loan'2013].

Vector approximation

Goal: Want to approximate vector f in subspace $\mathrm{range}(U)$. For $I = \{i_1, \ldots, i_k\}$ define selection operator:

$$\mathbb{S}_I = \begin{bmatrix} e_{i_1} & e_{i_2} & \cdots & e_{i_k} \end{bmatrix}.$$

Minimal error attained by orthogonal projection UU^T. When replaced by *oblique* projection

$$U(\mathbb{S}_I^T U)^{-1}\mathbb{S}_I^T f$$

increase of error bounded by result of lemma.

Lemma

$$\|f - U(\mathbb{S}_I^T U)^{-1}\mathbb{S}_I^T f\|_2 \leq \|(\mathbb{S}_I^T U)^{-1}\|_2 \cdot \|f - UU^T f\|_2.$$

Proof. Let $\Pi = U(\mathbb{S}_I^T U)^{-1}\mathbb{S}_I^T$. Then

$$\|(I - \Pi)f\|_2 = \|(I - \Pi)(f - UU^T f)\|_2 \leq \|I - \Pi\|_2 \|f - UU^T f\|_2.$$

The proof is completed by noting (and using the exercises),

$$\|I - \Pi\|_2 = \|\Pi\|_2 \leq \|(\mathbb{S}_I^T U)^{-1}\mathbb{S}_I^T\|_2 = \|(\mathbb{S}_I^T U)^{-1}\|_2.$$

Connection to interpolation

We have

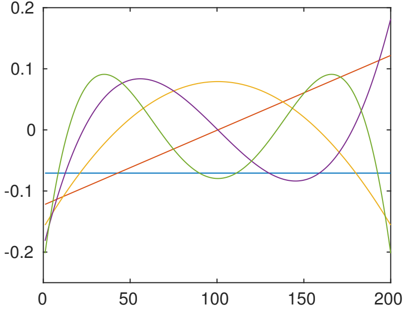
$$\mathbb{S}_I^T (I - U(\mathbb{S}_I^T U)^{-1} \mathbb{S}_I^T) = 0$$

and hence

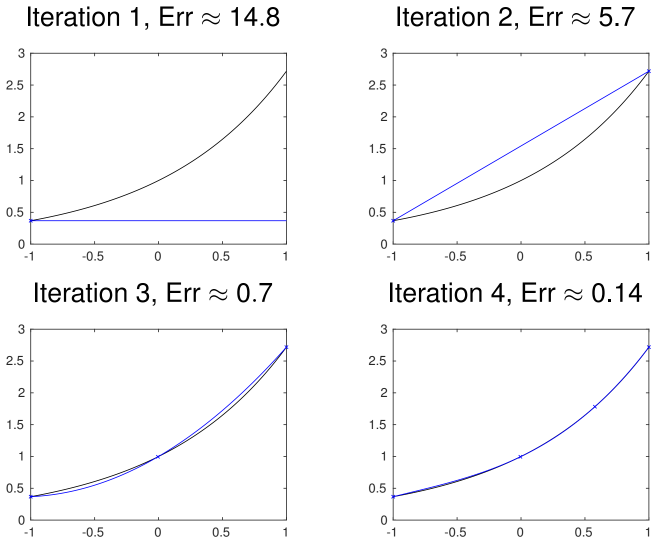
$$\|\mathbb{S}_I^T (f - U(\mathbb{S}_I^T U)^{-1} \mathbb{S}_I^T f)\|_2 = 0.$$

Interpretation: f is "interpolated" exactly at selected indices.

Example: Let f contain discretization of $\exp(x)$ on $[-1, 1]$ let U contain orthonormal basis of discretized monomials $\{1, x, x^2, \ldots\}$.

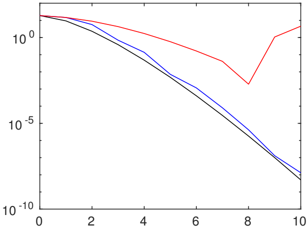


Connection to interpolation



Connection to interpolation

Comparison between best approximation, greedy approximation, approximation obtained by simply selecting first r indices.



Terminology:

- Continuous setting: EIM (Empirical Interpolation method),
  [M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An “empirical interpolation” method: Application to efficient reduced-basis discretization of partial differential equations, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672].
- Discrete setting: DEIM (Discrete EIM),
  [S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. SIAM Journal on Scientific Computing, 32(5), 2737–2764, 2010].

POD+DEIM

Consider LARGE ODE of the form

$$\dot{x}(t) = Ax(t) + F(x(t)).$$

A is $n \times n$ matrix. Idea of POD[12]:

1. Simulate ODE for one or more initial conditions and collect trajectories at discrete time points into snapshot matrix:

$$X = \begin{pmatrix} x(t_1) & \cdots & x(t_m) \end{pmatrix}.$$

2. Compute ONB $V \in \mathbb{R}^{n \times r}$, $r \ll n$, of dominant left subspace of X (e.g., by SVD).
3. Assume approximation $x \approx UU^T x = Uy$ and project dynamical system onto $\mathtt{range}(U)$:

$$\dot{y}(t) = U^T A U y(t) + U^T F(Uy(t)).$$

[12]See [S. Volkwein. Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling. Lecture Notes, 2013] for a comprehensive introduction.

POD+DEIM

Problem: $U^T F(Uy(t))$ still involves (large) dimension of original system.

Using DEIM:

$$U^T F(Uy(t)) \approx (\mathbb{S}_I^T U)^{-1} \mathbb{S}_I^T F(Uy(t)).$$

$$\dot{y}(t) = U^T AUy(t) + (\mathbb{S}_I^T U)^{-1} \mathbb{S}_I^T F(Uy(t)).$$

$\rightsquigarrow$ Only need to evaluate $\#I = r$ instead of n entries of function F.

Particularly efficient for

$$F(x) = \begin{pmatrix} f_1(x_1) \\ \vdots \\ f_n(x_n) \end{pmatrix} \quad \Rightarrow \quad \mathbb{S}_I^T F(Uy(t)) = \begin{pmatrix} f_{i_1}(x_{i_1}) \\ \vdots \\ f_{i_r}(x_{i_r}) \end{pmatrix}$$

Example from [Chaturantabut/Sorensen'2010]: Discretized FitzHugh-Nagumo equations involve $F(x) = x \odot (x - 0.1) \odot (1 - x)$.

The CUR decomposition: Existence results

$$A = CUR,$$

where C contains columns of A, R contains rows of A, U is chosen "wisely".

Theorem (Goreinov/Tyrtyshnikov/Zamarshkin'1997). Let $\varepsilon := \sigma_{k+1}(A)$. Then there exist row indices $I \subset \{1, \dots, m\}$ and column indices $J \subset \{1, \dots, n\}$ and a matrix $S \in \mathbb{R}^{k \times k}$ such that

$$\|A - A(:, J)SA(I, :)\|_2 \leq \varepsilon(1 + 2\sqrt{k}(\sqrt{m} + \sqrt{n})).$$

Proof. Let U_k, V_k contain k dominant left/right singular vectors of A. Choose I, J by selecting rows from U_k, V_k. According to max volume lemma, the square matrices $\hat{U} = U_k(I, :)$, $\hat{V} = V_k(J, :)$ satisfy

$$\|\hat{U}^{-1}\|_2 \leq \sqrt{k(m-k)+1}, \quad \|\hat{V}^{-1}\|_2 \leq \sqrt{k(n-k)+1}.$$

$+$ complicated choice of S.

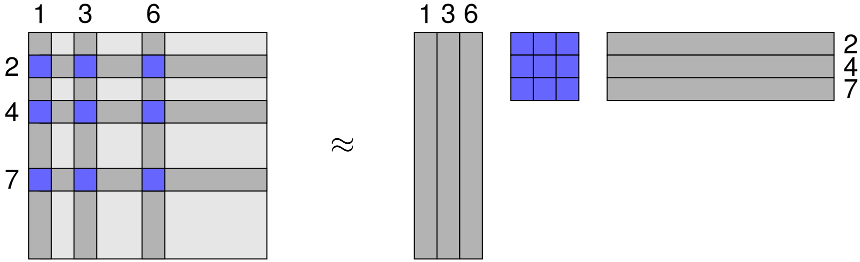
The CUR decomposition: Existence results

Choice of $S = (A(I,J))^{-1}$ in CUR $\rightsquigarrow$ Remainder term

$$R := A - A(:,J)(A(I,J))^{-1}A(I,:)$$

has zero rows at I and zero columns at J.

Cross approximation:



Adaptive Cross Approximation (ACA)

A more direct attempt to find a good cross..

Theorem (Goreinov/Tyrtyshnikov'2001). Suppose that

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where $A_{11} \in \mathbb{R}^{r \times r}$ has maximal volume among all $r \times r$ submatrices of A. Then

$$\|A_{22} - A_{21}A_{11}^{-1}A_{12}\|_C \leq (r+1)\sigma_{r+1}(A),$$

where $\|M\|_C := \max_{ij} |m_{ij}|$

As we already know, finding A_{11} is NP hard [Çivril/Magdon-Ismail'2013].

Adaptive Cross Approximation (ACA)

ACA with full pivoting [Bebendorf/Tyrtyshnikov'2000]

Set $R_0 := A$, $I := \{\}$, $J := \{\}$, $k := 0$
repeat
  $k := k + 1$
  $(i_k, j_k) := \arg\max_{i,j} |R_{k-1}(i,j)|$
  $I \leftarrow I \cup \{i_k\}$, $J \leftarrow J \cup \{j_k\}$
  $\delta_k := R_{k-1}(i_k, j_k)$
  $u_k := R_{k-1}(:, j_k)$, $v_k := R_{k-1}(i_k, :)^T / \delta_k$
  $R_k := R_{k-1} - u_k v_k^T$
until $\|R_k\|_F \leq \varepsilon \|A\|_F$

- This is greedy for maxvol.
- Still too expensive for general matrices.

Adaptive Cross Approximation (ACA)

ACA with partial pivoting

Set $R_0 := A$, $I := \{\}$, $J := \{\}$, $k := 1$, $i^* := 1$
repeat
    $j^* := \arg\max_j |R_{k-1}(i^*, j)|$
    $\delta_k := R_{k-1}(i^*, j^*)$
    if $\delta_k = 0$ **then**
        if $\# I = \min\{m, n\} - 1$ **then**
            Stop
        end if
    else
        $u_k := R_{k-1}(:, j^*)$, $v_k := R_{k-1}(i^*, :)^T / \delta_k$
        $R_k := R_{k-1} - u_k v_k^T$
        $k := k + 1$
    end if
    $I \leftarrow I \cup \{i^*\}$, $J \leftarrow J \cup \{j^*\}$
    $i^* := \arg\max_{i \notin I} |u_k(i)|$
until stopping criterion is satisfied

Adaptive Cross Approximation (ACA)

ACA with partial pivoting. Remarks:

- R_k is never formed explicitly. Entries of R_k are computed from

$$R_k(i,j) = A(i,j) - \sum_{\ell=1}^{k} u_\ell(i) v_\ell(j).$$

- Ideal stopping criterion $\|u_k\|_2 \|v_k\|_2 \le \varepsilon \|A\|_F$ elusive.
  Replace $\|A\|_F$ by $\|A_k\|_F$, recursively computed via

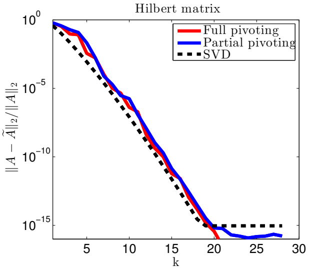
$$\|A_k\|_F^2 = \|A_{k-1}\|_F^2 + 2 \sum_{j=1}^{k-1} u_k^T u_j v_j^T v_k + \|u_k\|_2^2 \|v_k\|_2^2.$$

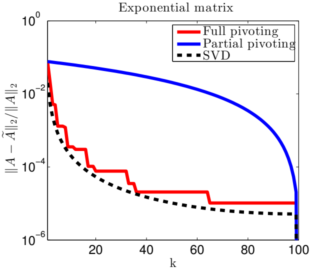
Adaptive Cross Approximation (ACA)

Two 100×100 matrices:

(a) The Hilbert matrix A defined by $A(i,j) = 1/(i+j-1)$.

(b) The matrix A defined by $A(i,j) = \exp(-\gamma|i-j|/n)$ with $\gamma = 0.1$.





1. Excellent convergence for Hilbert matrix.
2. Slow singular value decay impedes partial pivoting.

ACA for SPSD matrices

For symmetric positive semi-definite matrix $A \in \mathbb{R}^{n \times n}$:

- SVD becomes spectral decomposition.
- Can use trace instead of Frobenius norm to control error.
- Remainder R_k stays SPSD.
- Rows/columns can be chosen by largest diagonal element of R_k.
- ACA becomes
  - $=$ Cholesky (with diagonal pivoting); see [Higham'1990].
  - $=$ Nyström method [Williams/Seeger'2001].
- DEIM-like error bound [Harbrecht/Peters/Schneider'2012], [Cortinovis/DK/Massei'2020]:

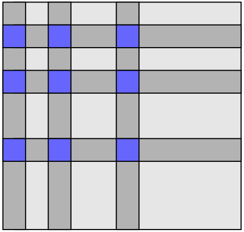
$$\|R_k\|_C \leq 4^k \sigma_{k+1}(A),$$

This is the only known situation (of practical relevance), for which a deterministic method only needs to see $O(nk)$ entries of A and still satisfies an error bound.

3. Stochastic sampling

Randomized column/row sampling

Aim: Obtain rank-r approximation from randomly selected rows and columns of A.



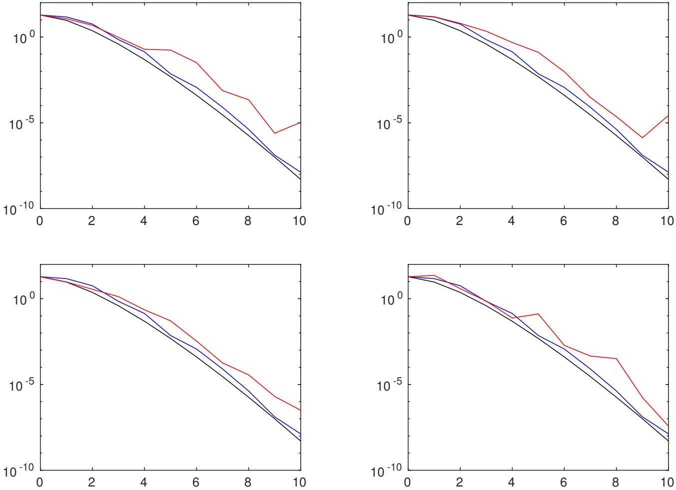
Popular sampling strategies:

- Uniform sampling.
- Sampling based on row/column norms.
- Sampling based on more complicated quantities (leverage scores).

Preliminaries on randomized sampling

Exponential function example from before.

Comparison between best approximation, greedy approximation, approximation obtained by randomly selecting rows.



Preliminaries on randomized sampling

A simple way to fool uniformly random row selection:

$$U = \begin{pmatrix} 0_{(n-r)\times r} \\ I_r \end{pmatrix}$$

for n very large and $r \ll n$.

Column sampling

Basic algorithm aiming at rank-r approximation:

1. Sample (and possibly rescale) $k > r$ columns of A
   $\rightsquigarrow m \times k$ matrix C.
2. Compute SVD $C = U\Sigma V^T$ and set $Q = U_r \in \mathbb{R}^{m\times r}$.
3. Return low-rank approximation $QQ^T A$.

- Can be combined with streaming algorithm [Liberty'2007] to limit memory/cost of working with C.
- Quality of approximation crucially depends on sampling strategy.

Column sampling

Lemma

For any matrix $C \in \mathbb{R}^{m \times r}$, let Q be the matrix computed above. Then

$$\|A - QQ^T A\|_2^2 \leq \sigma_{r+1}(A)^2 + 2\|AA^T - CC^T\|_2.$$

Proof. We have

$$\begin{aligned} & (A - QQ^T A)(A - QQ^T A)^T \\ = \quad & (I - QQ^T)CC^T(I - QQ^T) + (I - QQ^T)(AA^T - CC^T)(I - QQ^T) \end{aligned}$$

Hence,

$$\begin{aligned} \|A - QQ^T A\|_2^2 &= \lambda_{\max}\big((A - QQ^T A)(A - QQ^T A)^T\big) \\ &\leq \lambda_{\max}\big((I - QQ^T)CC^T(I - QQ^T)\big) + \|AA^T - CC^T\|_2 \\ &= \sigma_{r+1}(C)^2 + \|AA^T - CC^T\|_2. \end{aligned}$$

The proof is completed by applying Weyl's inequality:

$$\sigma_{r+1}(C)^2 = \lambda_{r+1}(CC^T) \leq \lambda_{r+1}(AA^T) + \|AA^T - CC^T\|_2.$$

Random column sampling

Using the lemma, the goal now becomes to approximate the matrix product AA^T using column samples of A.

Notation:

$$A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}, \quad C = \begin{bmatrix} c_1 & \cdots & c_k \end{bmatrix}$$

General sampling method:

Input: $A \in \mathbb{R}^{m\times n}$, probabilities $p_1, \ldots, p_n \neq 0$, integer k.
Output: $C \in \mathbb{R}^{m\times k}$ containing selected columns of A.

for $t = 1, \ldots, k$ **do**
    Pick $j_t \in \{1, \ldots, n\}$ with $\mathbb{P}[j_t = \ell] = p_\ell$, $\ell = 1, \ldots, n$, independently and with replacement.
    Set $c_t = a_{j_t}/\sqrt{kp_{j_t}}$.
end for

Random column sampling

One has

$$\begin{aligned}
\mathbb{E}[\|AA^T - CC^T\|_F^2] &= \sum_{ij} \mathbb{E}[(AA^T - CC^T)_{ij}^2] \\
&= \sum_{ij} \mathrm{Var}[(CC^T)_{ij}] \\
&= \frac{1}{k} \sum_{ij} \left(\sum_{\ell=1}^{n} \frac{a_{i\ell}^2 a_{j\ell}^2}{p_\ell} - \frac{1}{k} (AA^T)_{ij}^2 \right) \\
&= \frac{1}{k} \left[\sum_{\ell=1}^{n} \frac{1}{p_\ell} \|a_\ell\|_2^4 - \|AA^T\|_F^2 \right].
\end{aligned}$$

Lemma

The choice $p_\ell = \|a_\ell\|_2^2 / \|A\|_F^2$ minimizes $\mathbb{E}[\|AA^T - CC^T\|_F^2]$ and yields

$$\mathbb{E}[\|AA^T - CC^T\|_F^2] = \frac{1}{k} \left[\|A\|_F^4 - \|AA^T\|_F^2 \right]$$

Proof. Established by showing that this choice of p_ℓ satisfies first-order conditions of constrained optimization problem.

Random column sampling

Norm based sampling:

Input: $A \in \mathbb{R}^{m \times n}$, integer k.
Output: $C \in \mathbb{R}^{m \times k}$ containing selected columns of A.

```
Set p_ℓ = ‖a_ℓ‖_2^2 / ‖A‖_F^2 for ℓ = 1, ..., n.
for t = 1, ..., k do
    Pick j_t ∈ {1, ..., n} with P[j_t = ℓ] = p_ℓ, ℓ = 1, ..., n,
    independently and with replacement.
    Set c_t = a_{j_t} / sqrt(k p_{j_t}).
end for
Compute SVD C = UΣV^T and set Q = U_r ∈ R^{m×r}.
Return low-rank approximation QQ^T A.
```

Set $p_\ell = \|a_\ell\|_2^2 / \|A\|_F^2$ for $\ell = 1, \ldots, n$.
for $t = 1, \ldots, k$ **do**
  Pick $j_t \in \{1, \ldots, n\}$ with $\mathbb{P}[j_t = \ell] = p_\ell$, $\ell = 1, \ldots, n$, independently and with replacement.
  Set $c_t = a_{j_t} / \sqrt{k p_{j_t}}$.
end for
Compute SVD $C = U \Sigma V^T$ and set $Q = U_r \in \mathbb{R}^{m \times r}$.
Return low-rank approximation $QQ^T A$.

Random column sampling

By Azuma-Hoeffding inequality:

Theorem (Drineas/Kannan/Mahoney'2006)

For the matrix Q returned by the algorithm above it holds that

$$\mathbf{E}\left[\|A - QQ^T A\|_2^2\right] \leq \sigma_{r+1}^2(A) + \varepsilon\|A\|_F^2 \textit{ for } k \geq 4/\varepsilon^2.$$

With probability at least $1 - \delta$*,*

$$\|A - QQ^T A\|_2^2 \leq \sigma_{r+1}^2(A) + \varepsilon\|A\|_F^2 \textit{ for } k \geq 4(1 + \sqrt{8 \cdot \log(1/\delta)})^2/\varepsilon^2.$$

Proof. Follows from combining very first lemma with last two lemmas.

Remarks:

- Dependence of k on ε pretty bad. Unlikely to achieve something significantly better without assuming further properties of A (e.g., incoherence of singular vectors) with sampling based on row norms only.
- Simple "counter example":

$$A = \begin{pmatrix} \frac{1}{\sqrt{n}}e_1 & \frac{1}{\sqrt{n}}e_1 & \cdots & \frac{1}{\sqrt{n}}e_1 & \frac{1}{\sqrt{n}}e_2 \end{pmatrix} \in \mathbb{R}^{n\times(n+1)}.$$

Random column sampling

[Drineas/Mahoney/Muthukrishnan'2007]: Let V_k contain k dominant right singular vectors of A. Setting

$$p_\ell = \|V_k(\ell,:)\|_2^2/k, \qquad \ell = 1,\dots,n$$

and sampling $\mathcal{O}(k^2(\log 1/\delta)/\varepsilon^2)$ columns[13] yields

$$\|A - QQ^T A\|_F \le (1+\varepsilon)\|A - \mathcal{T}_k(A)\|_F$$

with probability $1-\delta$.

Relative error bound!

CUR decomposition can be obtained by applying ideas to rows and columns (yielding R and C, respectively) and choosing U appropriately.

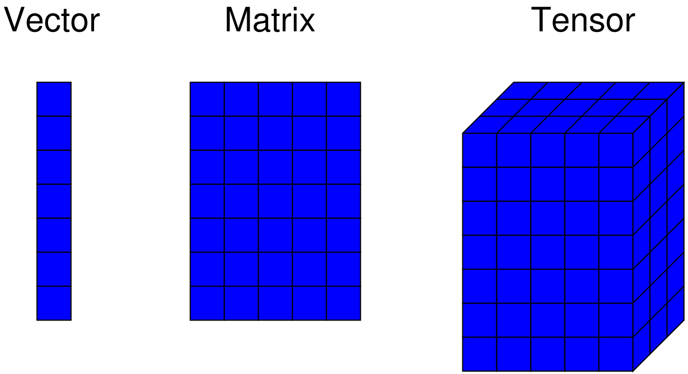
Many improvements: For example, it is enough to have a rough approximation of $\|V_k(\ell,:)\|_2$, which can be refined iteratively [Luan/Pan'2023].

[13] There are variants that improve this to $\mathcal{O}(k \log k \log(1/\delta)/\varepsilon^2)$.

4. Tensors

First steps with tensors

Vectors, matrices, and tensors



- scalar $=$ tensor of order 0
- (column) vector $=$ tensor of order 1
- matrix $=$ tensor of order 2
- tensor of order 3
  $= n_1 n_2 n_3$ numbers arranged in $n_1 \times n_2 \times n_3$ array

Tensors of arbitrary order

A d-th order tensor $\mathcal{X}$ of size $n_1 \times n_2 \times \cdots \times n_d$ is a d-dimensional array with entries

$$\mathcal{X}_{i_1, i_2, \ldots, i_d}, \qquad i_\mu \in \{1, \ldots, n_\mu\} \text{ for } \mu = 1, \ldots, d.$$

In the following, entries of $\mathcal{X}$ are usually real (for simplicity) $\rightsquigarrow$

$$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}.$$

Multi-index notation:

$$\mathfrak{I} = \{1, \ldots, n_1\} \times \{1, \ldots, n_2\} \times \cdots \times \{1, \ldots, n_d\}.$$

Then $i \in \mathfrak{I}$ is a tuple of d indices:

$$i = (i_1, i_2, \ldots, i_d).$$

Allows to write entries of $\mathcal{X}$ as $\mathcal{X}_i$ for $i \in \mathfrak{I}$.

Two important points

1. A matrix $A \in \mathbb{R}^{m \times n}$ has a natural interpretation as a linear operator in terms of matrix-vector multiplications:

$$A : \mathbb{R}^n \to \mathbb{R}^m, \quad A : x \mapsto A \cdot x.$$

   There is no such (unique and natural) interpretation for tensors!

   $\rightsquigarrow$ fundamental difficulty to define meaningful general notion of eigenvalues and singular values of tensors.

2. Number of entries in tensor grows exponentially with d $\rightsquigarrow$

   Curse of dimensionality.

   Example: Tensor of order 30 with $n_1 = n_2 = \cdots = n_d = 10$ has 10^{30} entries $= 8 \times 10^{12}$ Exabyte storage![14]

   For $d \gg 1$: Cannot afford to store tensor explicitly (in terms of its entries).

[14] Global data storage a few years ago calculated at 295 exabyte, see `http://www.bbc.co.uk/news/technology-12419672`.

Basic calculus

- Addition of two equal-sized tensors $\mathcal{X}, \mathcal{Y}$:

$$\mathcal{Z} = \mathcal{X} + \mathcal{Y} \quad \Leftrightarrow \quad \mathcal{Z}_i = \mathcal{X}_i + \mathcal{Y}_i \quad \forall i \in \mathfrak{I}.$$

- Scalar multiplication with $\alpha \in \mathbb{R}$:

$$\mathcal{Z} = \alpha\mathcal{X} \quad \Leftrightarrow \quad \mathcal{Z}_i = \alpha\mathcal{X}_i \quad \forall i \in \mathfrak{I}.$$

$\rightsquigarrow$ vector space structure.

- Inner product of two equal-sized tensors $\mathcal{X}, \mathcal{Y}$:

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i \in \mathfrak{I}} x_i y_i.$$

$\rightsquigarrow$ Induced norm

$$\|\mathcal{X}\| := \left(\sum_{i \in \mathfrak{I}} x_i^2 \right)^{1/2}$$

For a 2nd order tensor ($=$ matrix) this corresponds to the usual Euclidean geometry and *Frobenius norm*.

Vectorization

Tensor $\mathcal{X}$ of size $n_1 \times n_2 \times \cdots \times n_d$ has $n_1 \cdot n_2 \cdots n_d$ entries
$\leadsto$ many ways to stack entries in a (loooong) column vector.

One possible choice:

The vectorization of $\mathcal{X}$ is denoted by $\text{vec}(\mathcal{X})$, where

$$\text{vec} : \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \cdot n_2 \cdots n_d}$$

stacks the entries of a tensor in reverse lexicographical order into a long column vector.

Example: $d = 3$, $n_1 = 3$, $n_2 = 2$, $n_3 = 3$.

$$\text{vec}(\mathcal{X}) = \begin{bmatrix} x_{111} \\ x_{211} \\ x_{311} \\ x_{121} \\ \vdots \\ \vdots \\ x_{123} \\ x_{223} \\ x_{323} \end{bmatrix}$$

Matricization

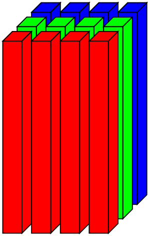
- A matrix has two modes (column mode and row mode).
- A dth-order tensor $\mathcal{X}$ has d modes ($\mu = 1$, $\mu = 2$, …, $\mu = d$).

Let us fix all but one mode, e.g., $\mu = 1$: Then

$$\mathcal{X}(:, i_2, i_3, \dots, i_d) \qquad \text{(abuse of MATLAB notation)}$$

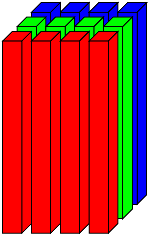
is a vector of length n_1 for each choice of $i_2, \dots, i_d$. These vectors are called fibers.

$\rightsquigarrow$ View tensor $\mathcal{X}$ as a bunch of column vectors:

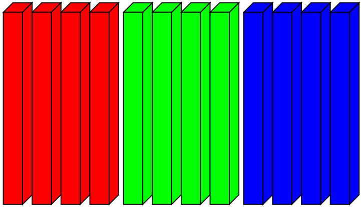


Matricization

Stack vectors into an $n_1 \times (n_2 \cdots n_d)$ matrix:



$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$



$X^{(1)} \in \mathbb{R}^{n_1 \times (n_2 n_3 \cdots n_d)}$

For $\mu = 1, \ldots, d$, the μ-mode matricization of $\mathcal{X}$ is a matrix

$$X^{(\mu)} \in \mathbb{R}^{n_\mu \times (n_1 \cdots n_{\mu-1} n_{\mu+1} \cdots n_d)}$$

with entries

$$\left(X^{(\mu)}\right)_{i_{\mu_1},(i_1,\ldots,i_{\mu-1},i_{\mu+1}\ldots i_d)} = \mathcal{X}_i \quad \forall i \in \mathfrak{I}.$$

Matricization

In MATLAB: `a = rand(2,3,4,5);`

- 1-mode matricization:

  ```
  reshape(a,2,3*4*5)
  ```

- 2-mode matricization:

  ```
  b = permute(a,[2 1 3 4]);
  reshape(b,3,2*4*5)
  ```

- 3-mode matricization:

  ```
  b = permute(a,[3 1 2 4]);
  reshape(b,4,2*3*5)
  ```

- 4-mode matricization:

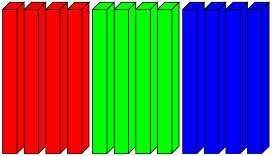
  ```
  b = permute(a,[4 1 2 3]);
  reshape(b,5,2*3*4)
  ```

For a matrix $A \in \mathbb{R}^{n_1 \times n_2}$:

$$A^{(1)} = A, \qquad A^{(2)} = A^T.$$

μ-mode matrix products

Consider 1-mode matricization $X^{(1)} \in \mathbb{R}^{n_1 \times (n_2 \cdots n_d)}$:



Seems to make sense to multiply an $m \times n_1$ matrix A from the left:

$$Y^{(1)} := A\,X^{(1)} \in \mathbb{R}^{m \times (n_2 \cdots n_d)}.$$

Can rearrange $Y^{(1)}$ back into an $m \times n_2 \times \cdots \times n_d$ tensor $\mathcal{Y}$.

This is called 1-mode matrix multiplication

$$\mathcal{Y} = A \circ_1 \mathcal{X} \qquad \Leftrightarrow \qquad Y^{(1)} = AX^{(1)}$$

More formally (and more ugly):

$$\mathcal{Y}_{i_1, i_2, \ldots, i_d} = \sum_{k=1}^{n_1} a_{i_1, k} \mathcal{X}_{k, i_2, \ldots, i_d}.$$

μ-mode matrix products

General definition of a μ-mode matrix product with $A \in \mathbb{R}^{m \times n_1}$:

$$\mathcal{Y} = A \circ_\mu \mathcal{X} \qquad \Leftrightarrow \qquad Y^{(\mu)} = AX^{(\mu)}.$$

More formally (and more ugly):

$$\mathcal{Y}_{i_1,i_2,\ldots,i_d} = \sum_{k=1}^{n_1} a_{i_\mu,k} \mathcal{X}_{i_1,\ldots,i_{\mu-1},k,i_{\mu+1},\ldots,i_d}.$$

For matrices:

- 1-mode multiplication $=$ multiplication from the left:

$$Y = A \circ_1 X = A\,X.$$

- 2-mode multiplication $=$ *transposed* multiplication from the right:

$$Y = A \circ_2 X = X\,A^T.$$

μ-mode matrix products and vectorization

By definition,

$$\mathrm{vec}(\mathcal{X}) = \mathrm{vec}\big(X^{(1)}\big).$$

Consequently, also

$$\mathrm{vec}(A \circ_1 \mathcal{X}) = \mathrm{vec}\big(A\, X^{(1)}\big).$$

$\rightsquigarrow$ Vectorized version of 1-mode matrix product:

$$\begin{aligned} \mathrm{vec}(A \circ_1 \mathcal{X}) &= (I_{n_2 \cdots n_d} \otimes A)\mathrm{vec}(\mathcal{X}) \\ &= (I_{n_d} \otimes \cdots \otimes I_{n_2} \otimes A)\, \mathrm{vec}(\mathcal{X}). \end{aligned}$$

Relation between μ-mode matrix product and matrix-vector product:

$$\mathrm{vec}(A \circ_\mu \mathcal{X}) = (I_{n_d} \otimes \cdots \otimes I_{n_{\mu+1}} \otimes A \otimes I_{n_{\mu-1}} \otimes \cdots \otimes I_{n_1})\, \mathrm{vec}(\mathcal{X})$$

Summary

- Tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a d-dimensional array.
- Various ways of reshaping entries of a tensor $\mathcal{X}$ into a vector or matrix.
- μ-mode matrix multiplication can be expressed with Kronecker products

Further reading:

- T. Kolda and B. W. Bader. Tensor decompositions and applications. SIAM Rev. 51 (2009), no. 3, 455–500.

Software:

- MATLAB (and all programming languages) offer basic functionality to work with d-dimensional arrays.
- MATLAB Tensor Toolbox: `http://www.tensortoolbox.org/`

Applications of tensors

Two classes of tensor problems

Class 1: function-related tensors

Consider a function $u(\xi_1, \ldots, \xi_d) \in \mathbb{R}$ in d variables $\xi_1, \ldots, \xi_d$.
Tensor $\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ represents discretization of u:

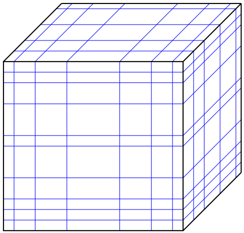
- $\mathcal{U}$ contains function values of u evaluated on a grid; **or**
- $\mathcal{U}$ contains coefficients of truncated expansion in tensorized basis functions:

$$u(\xi_1, \ldots, \xi_d) \approx \sum_{i \in \mathfrak{I}} \mathcal{U}_i \, \phi_{i_1}(\xi_1) \phi_{i_2}(\xi_2) \cdots \phi_{i_d}(\xi_d).$$

Typical setting:

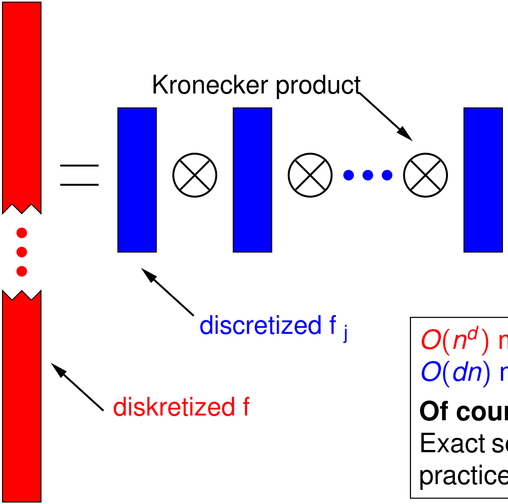
- $\mathcal{U}$ only given implicitly, e.g., as the solution of a discretized PDE;
- seek approximations to $\mathcal{U}$ with very low storage and tolerable accuracy.
- d may become very large.

Discretization of function in d variables
$\xi_1, \ldots, \xi_d \in [0, 1]$
$\rightsquigarrow$ #function values grows exponentially with d



Separability helps

Ideal situation:
Function f separable:
$f(\xi_1, \xi_2, \ldots, \xi_d) = f_1(\xi_1) f_2(\xi_2) \ldots f_d(\xi_d)$



$O(n^d)$ memory $\rightsquigarrow$
$O(dn)$ memory

Of course:
Exact separability rarely satisfied in practice.

Two classes of tensor problems

Class 2: data-related tensors

Tensor $\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ contains multi-dimensional data.

Example 1: $\mathcal{U}_{2011,3,2}$ denotes the number of papers published 2011 by author 3 in the mathematical journal 2.

Example 2: A video of 1000 frames with resolution 640×480 can be viewed as a $640 \times 480 \times 1000$ tensor.

Example 3: Hyperspectral images.

Example 4: Deep learning: Coefficients in each layer of deep NN stored as tensors (TensorFlow), Interpretation of RNNs as hierarchical tensor decomposition.

Typical setting (except for Example 4):

- entries of $\mathcal{U}$ often given explicitly (at least partially).
- extraction of dominant features from $\mathcal{U}$.
- usually moderate values for d.

Low-rank tensor techniques

- Emerged during last 15 years in scientific computing.
- Successfully applied to:
  - quantum many body problems;
  - parameter-dependent / multi-dimensional integrals;
  - electronic structure calculations: Hartree-Fock / DFT;
  - stochastic and parametric PDEs;
  - high-dimensional Boltzmann / chemical master / Fokker-Planck / Schrödinger equations;
  - micromagnetism;
  - rational approximation problems;
  - computational homogenization;
  - computational finance;
  - multivariate regression and machine learning;
  - ...

Classical references on applications

[1] M. Bachmayr, R. Schneider, and A. Uschmajew.
Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations.
Found. Comput. Math., 16(6):1423–1472, 2016.

[2] A. Cichocki.
Era of big data processing: A new approach via tensor networks and tensor decompositions.
arXiv:1403.2048, 2014.

[3] L. Grasedyck, D. Kressner, and C. Tobler.
A literature survey of low-rank tensor approximation techniques.
GAMM-Mitt., 36(1):53–78, 2013.

[4] W. Hackbusch.
Tensor Spaces and Numerical Tensor Calculus.
Springer, Heidelberg, 2012.

[5] V. Khrulkov, A. Novikov, and I. Oseledets.
Expressive power of recurrent neural networks, 2018.
ICLR: Sixth International Conference on Learning Representations.

[6] Anthony Nouy.
Low-rank methods for high-dimensional approximation and model order reduction.
In *Model reduction and approximation*, volume 15 of *Comput. Sci. Eng.*, pages 171–226. SIAM, Philadelphia, PA, 2017.

[7] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos.
Tensor decomposition for signal processing and machine learning.
IEEE Trans. Signal Process., 65(13):3551–3582, 2017.

The CP decomposition

CP decomposition

- Aim: Generalize concept of low rank from matrices to tensors.
- One possibility motivated by

$$
\begin{aligned}
X &= \left[a_1,\ a_2,\ \ldots,\ a_R\right]\left[b_1,\ b_2,\ \ldots,\ b_R\right]^T = \\
&= a_1 b_1^T + a_2 b_2^T + \cdots + a_R b_R^T.
\end{aligned}
$$

$\rightsquigarrow$ vectorization

$$
\mathrm{vec}(X) = b_1 \otimes a_1 + b_2 \otimes a_2 + \cdots + b_R \otimes a_R.
$$

Canonical **P**olyadic decomposition of tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ defined via

$$
\begin{aligned}
\mathrm{vec}(\mathcal{X}) &= c_1 \otimes b_1 \otimes a_1 + c_2 \otimes b_2 \otimes a_2 + \cdots + c_R \otimes b_R \otimes a_R \\
\mathcal{X} &= a_1 \circ b_1 \circ c_1 + a_2 \circ b_2 \circ c_2 + \cdots + a_R \circ b_R \circ c_R
\end{aligned}
$$

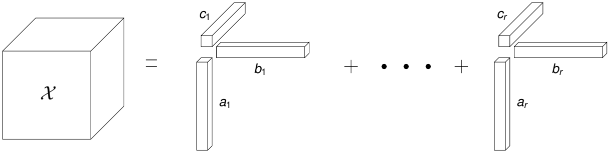
for vectors $a_j \in \mathbb{R}^{n_1}$, $b_j \in \mathbb{R}^{n_2}$, $c_j \in \mathbb{R}^{n_3}$.

CP directly corresponds to semi-separable approximation.
Tensor rank of $\mathcal{X}$ = minimal possible R

CP decomposition

Illustration of CP decomposition

$$\mathcal{X} = a_1 \circ b_1 \circ c_1 + a_2 \circ b_2 \circ c_2 + \cdots + a_R \circ b_R \circ c_R.$$



More compact notation:

$$\text{vec}(\mathcal{X}) = [\![A, B, C]\!],$$

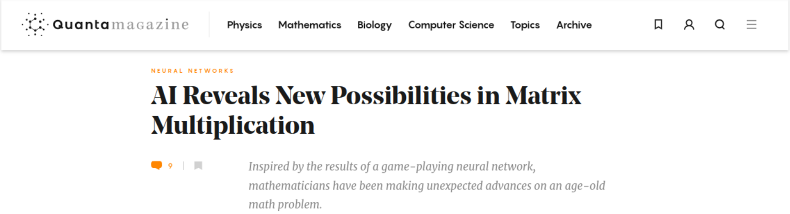
with

$$\begin{aligned} A &= [a_1, \ldots, a_R] \in \mathbb{R}^{n_1 \times R} \\ B &= [b_1, \ldots, b_R] \in \mathbb{R}^{n_2 \times R} \\ C &= [c_1, \ldots, c_R] \in \mathbb{R}^{n_3 \times R} \end{aligned}$$

Dismissal of CP decomposition

Despite its simplicity, the CP decomposition comes with a lot of problems [Silva/Lim’2008], [Kolda/Bader’2009]:

- Tensor rank can be extremely difficult to determine.



- Tensor rank is not lower semi-continuous.
- Real $\neq$ complex tensor rank.
- No simple quasi-optimal approximation algorithm known.

The Tucker decomposition

Tucker decomposition

- Alternative rank concept for tensors motivated by

$$A = U \cdot \Sigma \cdot V^T, \qquad U \in \mathbb{R}^{n_1 \times r}, \quad V \in \mathbb{R}^{n_2 \times r}, \quad \Sigma \in \mathbb{R}^{r \times r}.$$

$\rightsquigarrow$ vectorization

$$\mathrm{vec}(X) = \big(V \otimes U\big) \cdot \mathrm{vec}(\Sigma).$$

Ignore diagonal structure of Σ and call it C.

Tucker decomposition of tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ defined via

$$\mathrm{vec}(\mathcal{X}) = \big(W \otimes V \otimes U\big) \cdot \mathrm{vec}(\mathcal{C})$$

with $U \in \mathbb{R}^{n_1 \times r_1}$, $V \in \mathbb{R}^{n_2 \times r_2}$, $W \in \mathbb{R}^{n_3 \times r_3}$,
and core tensor $\mathcal{C} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$.

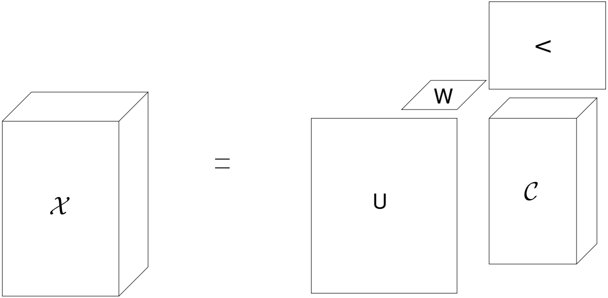
In terms of μ-mode matrix products:

$$\mathcal{X} = U \circ_1 V \circ_2 W \circ_3 \mathcal{C} =: (U, V, W) \circ \mathcal{C}.$$

Tucker decomposition

Illustration of Tucker decomposition

$$\mathcal{X} = (U, V, W) \circ \mathcal{C}$$



Tucker decomposition

Consider all three matricizations:

$$\begin{aligned} X^{(1)} &= U \cdot C^{(1)} \cdot \left(W \otimes V\right)^T, \\ X^{(2)} &= V \cdot C^{(2)} \cdot \left(W \otimes U\right)^T, \\ X^{(3)} &= W \cdot C^{(3)} \cdot \left(V \otimes U\right)^T. \end{aligned}$$

These are low rank decompositions $\rightsquigarrow$

$$\operatorname{rank}\left(X^{(1)}\right) \le r_1, \quad \operatorname{rank}\left(X^{(2)}\right) \le r_2, \quad \operatorname{rank}\left(X^{(3)}\right) \le r_3.$$

Multilinear rank of tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ defined by tuple

$$(r_1, r_2, r_3), \quad \text{with} \quad r_i = \operatorname{rank}\left(X^{(i)}\right).$$

Higher-order SVD (HOSVD)

Goal: Approximate given tensor $\mathcal{X}$ by Tucker decomposition with prescribed multilinear rank (r_1, r_2, r_3).

1. Calculate SVD of matricizations:

$$X^{(\mu)} = \widetilde{U}_\mu \widetilde{\Sigma}_\mu \widetilde{V}_\mu^T \quad \text{for } \mu = 1, 2, 3.$$

2. Truncate basis matrices:

$$U_\mu := \widetilde{U}_\mu(:, 1 : r_\mu) \quad \text{for } \mu = 1, 2, 3.$$

3. Form core tensor:

$$\mathcal{C} := U_1^T \circ_1 U_2^T \circ_2 U_3^T \circ_3 \mathcal{X}.$$

Truncated tensor produced by HOSVD [Lathauwer/De Moor/Vandewalle'2000]:

$$\widetilde{\mathcal{X}} := U_1 \circ_1 U_2 \circ_2 U_3 \circ_3 \mathcal{C}.$$

Remark:
Orthogonal projection $\widetilde{\mathcal{X}} := \big(\pi_1 \circ \pi_2 \circ \pi_3\big)\mathcal{X}$ with $\pi_\mu \mathcal{X} := U_\mu U_\mu^T \circ_\mu \mathcal{X}$.

Higher-order SVD (HOSVD)

Theorem. Tensor $\widetilde{\mathcal{X}}$ resulting from HOSVD satisfies quasi-optimality condition

$$\|\mathcal{X} - \widetilde{\mathcal{X}}\| \leq \sqrt{d}\|\mathcal{X} - \mathcal{X}_{\text{best}}\|,$$

where $\mathcal{X}_{\text{best}}$ is best approximation of $\mathcal{X}$ with multilinear ranks $(r_1, \dots, r_d)$.

Proof:

$$\begin{aligned}
\|\mathcal{X} - \widetilde{\mathcal{X}}\|^2 &= \|\mathcal{X} - (\pi_1 \circ \pi_2 \circ \pi_3)\mathcal{X}\|^2 \\
&= \|\mathcal{X} - \pi_1\mathcal{X}\|^2 + \|\pi_1\mathcal{X} - (\pi_1 \circ \pi_2)\mathcal{X}\|^2 + \cdots \\
&\qquad \cdots + \|(\pi_1 \circ \pi_2)\mathcal{X} - (\pi_1 \circ \pi_2 \circ \pi_3)\mathcal{X}\|^2 \\
&\leq \|\mathcal{X} - \pi_1\mathcal{X}\|^2 + \|\mathcal{X} - \pi_2\mathcal{X}\|^2 + \|\mathcal{X} - \pi_3\mathcal{X}\|^2
\end{aligned}$$

Using

$$\|\mathcal{X} - \pi_\mu\mathcal{X}\| \leq \|\mathcal{X} - \mathcal{X}_{\text{best}}\| \quad \text{for } \mu = 1, 2, 3$$

leads to

$$\|\mathcal{X} - \widetilde{\mathcal{X}}\|^2 \leq 3 \cdot \|\mathcal{X} - \mathcal{X}_{\text{best}}\|^2.$$

Approximation error obtained from HOSVD

Another direct consequence of the proof:

Corollary. Let $\sigma_k^{(\mu)}$ denote the kth singular of $X^{(\mu)}$. Then the approximation $\widetilde{\mathcal{X}}$ obtained from the HOSVD satisfies

$$\|\mathcal{X} - \widetilde{\mathcal{X}}\|^2 \leq \sum_{\mu=1}^{3} \sum_{k=r_\mu+1}^{n_\mu} (\sigma_k^{(\mu)})^2.$$

This also implies a lower bound for $\|\mathcal{X} - \mathcal{X}_{\text{best}}\|$ in terms of the singular values of the matricizations of $\mathcal{X}$.

- SVD can be replaced by any low-rank approximation technique discussed in this course. By triangular inequality, bound of Corollary still holds with an extra term accounting for the inexact SVD.
- Approximation error can be improved by alternativing optimization (HOOI), but often not worth bothering.

Tucker decomposition – Summary

For general tensors:

- multilinear rank r is upper semi-continuous $\rightsquigarrow$ closedness property.
- HOSVD – simple and robust algorithm to obtain quasi-optimal low-rank approximation.
- quasi-optimality good enough for most applications in scientific computing.
- robust black-box algorithms/software available (e.g., Tensor Toolbox).

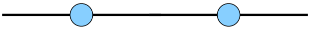
Drawback:

Storage of core tensor $\sim r^d$
$\rightsquigarrow$ curse of dimensionality

The Tensor Train decomposition

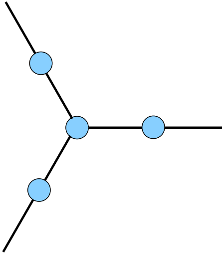
Tensor network diagrams

- Introduced by Roger Penrose.
- Heavily used in quantum mechanics (spin networks).
- Useful to gain intuition and guide design of algorithms.
- This is the matrix product $C = AB$:



$$C_{ij} = \sum_{k=1}^{r} A_{ik} B_{kj}$$

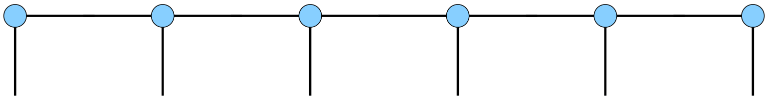
Tensor of order 3 in Tucker decomposition



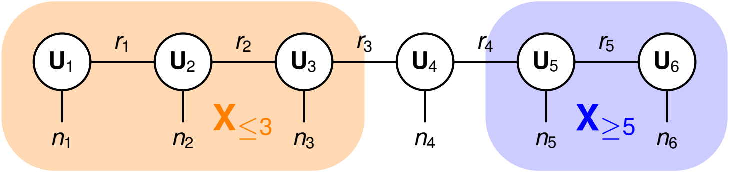
$$\mathcal{X}_{ijk} = \sum_{\ell_1=1}^{r_1} \sum_{\ell_2=1}^{r_2} \sum_{\ell_3=1}^{r_3} \mathcal{C}_{\ell_1 \ell_2 \ell_3} U_{i\ell_1} V_{j\ell_2} W_{k\ell_3}$$

- $r_1 \times r_2 \times r_3$ core tensor $\mathcal{C}$
- $n_1 \times r_1$ matrix U spans first mode
- $n_2 \times r_2$ matrix V spans second mode
- $n_3 \times r_3$ matrix W spans third mode.

Tensor of order 6 in TT decomposition



- $\mathcal{X}$ implicitly represented by four $r \times n \times r$ tensors and two $n \times r$ matrices
- More detailed picture:



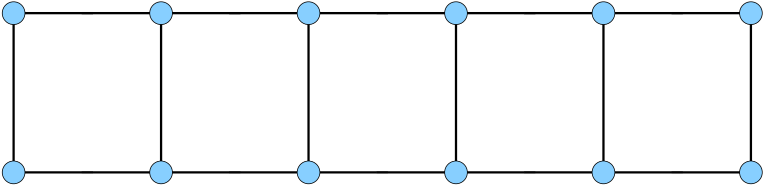
Tensor Train (TT) decomposition

A tensor $\mathcal{X}$ is in TT decomposition if it can be written as

$$\mathcal{X}(i_1, \dots, i_d) = \sum_{k_1=1}^{r_1} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} \mathcal{U}_1(1, i_1, k_1)\mathcal{U}_2(k_1, i_2, k_2)\cdots\mathcal{U}_d(k_{d-1}, i_d, 1).$$

- Smallest possible tuple $(r_1, \dots, r_{d-1})$ is called TT rank of $\mathcal{X}$.
- $\mathcal{U}_\mu \in \mathbb{R}^{r_{\mu-1}\times n_\mu \times r_\mu}$ (formally set $r_0 = r_d = 1$) are called TT cores for $\mu = 1, \dots, d$.
- If TT ranks are not large $\rightsquigarrow$ high compression ratio as d grows.
- TT decomposition multilinear wrt cores.
- TT decomposition connects to
  - matrix products $\rightsquigarrow$ Matrix Product States (MPS) in physics (see [Grasedyck/DK/Tobler'2013] for references)
  - simultaneous matrix factorizations $\rightsquigarrow$ SVD-based compression
  - contractions and tensor network diagrams $\rightsquigarrow$ design of efficient contraction-based algorithms

Inner product of two tensors in TT decomposition



- Carrying out contractions requires $O(dnr^4)$ instead of $O(n^d)$ operations for tensors of order d.

TT decomposition and matrix products

$$\mathcal{X}(i_1,\ldots,i_d) = \sum_{k_1=1}^{r_1}\cdots\sum_{k_{d-1}=1}^{r_{d-1}} \mathcal{U}_1(1,i_1,k_1)\mathcal{U}_2(k_1,i_2,k_2)\cdots\mathcal{U}_d(k_{d-1},i_d,1).$$

Let $U_\mu(i_\mu)$ be i_μth slice of μth core: $U_\mu(i_\mu) := \mathcal{U}_\mu(:,i_\mu,:) \in \mathbb{R}^{r_{\mu-1}\times r_\mu}$.
Then

$$\mathcal{X}(i_1,i_2,\ldots,i_d) = U_1(i_1)U_2(i_2)\cdots U_d(i_d).$$

Remark: Error analysis of matrix multiplication [Higham'2002] shows that TT decomposition may suffer from numerical instabilities if

$$\|U_1(i_1)\|_2\|U_2(i_2)\|_2\cdots\|U_d(i_d)\|_2 \gg |\mathcal{X}(i_1,i_2,\ldots,i_d)|.$$

See [Bachmayr/Kazeev: arXiv:1802.09062] for more details.

TT decomposition and matrix factorizations

$$\mathcal{X}(i_1, \ldots, i_d) = \sum_{k_1, k_2, \ldots, k_{d-1}} \mathcal{U}_1(1, i_1, k_1)\mathcal{U}_2(k_1, i_2, k_2) \cdots \mathcal{U}_d(k_{d-1}, i_d, 1).$$

For any $1 \leq \mu \leq d-1$ group first μ factors and last $d-\mu$ factors together:

$$\begin{aligned} & \mathcal{X}(i_1, \ldots, i_\mu, i_{\mu+1}, \ldots i_d) \\ = & \sum_{k_\mu=1}^{r_\mu} \Big(\sum_{k_1, \ldots, k_{\mu-1}} \mathcal{U}_1(1, i_1, k_1) \cdots \mathcal{U}_\mu(k_{\mu-1}, i_\mu, k_\mu) \Big) \\ & \cdot \Big(\sum_{k_{\mu+1}, \ldots, k_{d-1}} \mathcal{U}_{\mu+1}(k_\mu, i_{\mu+1}, k_{\mu+1}) \cdots \mathcal{U}_d(k_{d-1}, i_d, 1) \Big) \end{aligned}$$

This can be interpreted as a matrix-matrix product of two (large) matrices!

TT decomposition and matrix factorizations

The μth unfolding of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ is obtained by arranging the entries in a matrix

$$X^{<\mu>} \in \mathbb{R}^{(n_1 n_2 \cdots n_\mu) \times (n_{\mu+1} \cdots n_d)}$$

where the corresponding index map is given by

$$\iota : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \cdots n_\mu} \times \mathbb{R}^{n_{\mu+1} \cdots n_d}, \quad \iota(i_1, \ldots, i_d) = (i_{\text{row}}, i_{\text{col}}),$$

$$i_{\text{row}} = 1 + \sum_{\nu=1}^{\mu} (i_\nu - 1) \prod_{\tau=1}^{\nu-1} n_\tau, \quad i_{\text{col}} = 1 + \sum_{\nu=\mu+1}^{d} (i_\nu - 1) \prod_{\tau=\mu+1}^{\nu-1} n_\tau.$$

TT decomposition and matrix factorizations

Define interface matrices

$$X_{\le\mu} \in \mathbb{R}^{n_1 n_2 \cdots n_\mu \times r_\mu}, \quad X_{\ge\mu+1} \in \mathbb{R}^{r_\mu \times n_{\mu+1} n_{\mu+2} \cdots n_d}$$

as

$$\begin{aligned} X_{\le\mu}(i_{\text{row}}, j) &= \sum_{k_1,\ldots,k_{\mu-1}} \mathcal{U}_1(1, i_1, k_1) \cdots \mathcal{U}_{\mu-1}(k_{\mu-2}, i_{\mu-1}, k_{\mu-1}) \mathcal{U}_\mu(k_{\mu-1}, i_\mu, j) \\ X_{\ge\mu+1}(j, i_{\text{col}}) &= \sum_{k_{\mu+1},\ldots,k_{d-1}} \mathcal{U}_{\mu+1}(j, i_{\mu+1}, k_{\mu+1}) \mathcal{U}_{\mu+2}(k_{\mu+1}, i_{\mu+2}, k_{\mu+2}) \cdots \mathcal{U}_d(k_{d-1}, i_d, 1) \end{aligned}$$

Matrix factorizations

$$X^{<\mu>} = X_{\le\mu} X_{\ge\mu+1}, \quad \mu = 1, \ldots, d-1.$$

Lemma

The TT rank of a tensor is given by

$$\left(\operatorname{rank} X^{<1>}, \ldots, \operatorname{rank} X^{<d-1>} \right)$$

Truncation in TT format

Lemma follows from TT-SVD [Oseledets'2011]) for approximating a given tensor $\mathcal{X}$ in TT format:

Input: $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, target TT rank $(r_1, \ldots, r_{d-1})$.

Output: TT cores $\mathcal{U}_\mu \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times r_\mu}$ that define a TT decomposition approximating $\mathcal{X}$.

Set $r_0 = r_d = 1$. (and formally add leading singleton dimension to $\mathcal{X} \in \mathbb{R}^{1 \times n_1 \times \cdots \times n_d}$).
for $\mu = 1, \ldots, d-1$ **do**
    Reshape $\mathcal{X}$ into $X^{<2>} \in \mathbb{R}^{r_{\mu-1} n_\mu \times n_{\mu+1} \cdots n_d}$.
    Compute rank-r_μ approximation $X^{<2>} \approx U \Sigma V^T$ (e.g., via SVD)
    Reshape U into $\mathcal{U}_\mu \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times r_\mu}$.
    Update $\mathcal{X}$ via $X^{<2>} \leftarrow U^T X^{<2>} = \Sigma V^T$.
end for
Set $\mathcal{U}_d = \mathcal{X}$.

Truncation in TT format

Theorem

Let $\mathcal{X}_{\mathrm{SVD}}$ denote the tensor in TT decomposition obtained from TT-SVD. Then

$$\|\mathcal{X} - \mathcal{X}_{\mathrm{SVD}}\| \leq \sqrt{\varepsilon_1^2 + \cdots + \varepsilon_d^2},$$

where

$$\varepsilon_\mu^2 = \|X^{<\mu>} - \mathcal{T}_{r_\mu}(X^{<\mu>})\|_F^2 = \sigma_{r_\mu+1}(X^{<\mu>})^2 + \cdots .$$

Corollary

Let $\mathcal{X}_{\mathrm{best}}$ denote the best approximation of $\mathcal{X}$ with TT rank $(r_1, \ldots, r_{d-1})$. Then

$$\|\mathcal{X} - \mathcal{X}_{\mathrm{SVD}}\| \leq \sqrt{d-1}\|\mathcal{X} - \mathcal{X}_{\mathrm{best}}\|.$$

TT decomposition – Summary of operations

Easy:

- (partial) contractions
- multiplication with operators in suitable format (MPO)
- compression/recompression

Medium:

- entrywise products

Hard:

- almost everything else

Software:

- TT toolbox (Matlab, Python), ...

Ongoing research:
Effective randomized techniques [Ma/Solomonik'2022, Al Daas et al.'2023, DK/Vandereacken/Vorhaar'2023, ...].

5. Alternating Optimization

Alternating least-squares / linear scheme

General setting: Solve optimization problem

$$\min_X f(X),$$

where X is a (large) matrix or tensor and f is “simple” (e.g., convex).

Constrain X to $\mathcal{M}_r$, set of rank-r matrices or tensors and aim at solving

$$\min_{X \in \mathcal{M}_r} f(X),$$

Set

$$X = \mathrm{i}(U_1, U_2, \ldots, U_d).$$

(e.g., $X = U_1 U_2^T$). Low-rank formats are multilinear $\rightsquigarrow$ hope that optimizing for each component is simple:

$$\min_{U_\mu} f(\mathrm{i}(U_1, U_2, \ldots, U_d)).$$

Alternating least-squares / linear scheme

Set $f(U_1, \ldots, U_d) := f(\mathrm{i}(U_1, \ldots, U_d))$.

ALS:

while not converged **do**
$\quad U_1 \leftarrow \arg\min_{U_1} f(\mathrm{i}(U_1, U_2, \ldots, U_d))$
$\quad U_2 \leftarrow \arg\min_{U_1} f(\mathrm{i}(U_1, U_2, \ldots, U_d))$
$\quad \ldots$
$\quad U_d \leftarrow \arg\min_{U_1} f(\mathrm{i}(U_1, U_2, \ldots, U_d))$
end while

Examples:

- ALS for fitting CP decomposition
- Subspace iteration.

Closely related: Block Gauss-Seidel, Block Coordinate Descent.

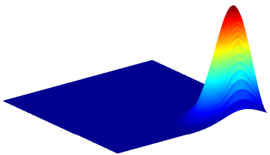
Difficulties:

- Representation $(U_1, U_2, \ldots, U_d)$ often non-unique, parameters may become unbounded.
- $\mathcal{M}_r$ not closed
- Convergence (analysis)

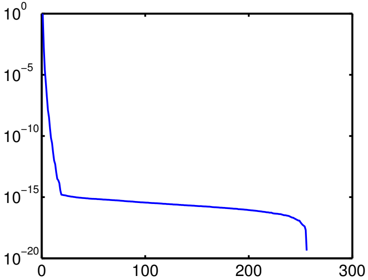
2D eigenvalue problem

- $-\triangle u(x) + V(x)u = \lambda u(x)$ in $\Omega = [0,1] \times [0,1]$ with Dirichlet b.c. and Henon-Heiles potential V
- Regular discretization
- Reshaped ground state into matrix

Ground state



Singular values



Excellent rank-10 approximation possible

Rayleigh quotients wrt low-rank matrices

Consider symmetric $n^2 \times n^2$ matrix $\mathcal{A}$. Then

$$\lambda_{\min}(\mathcal{A}) = \min_{x \neq 0} \frac{\langle x, \mathcal{A}x \rangle}{\langle x, x \rangle}.$$

We now...

- reshape vector x into $n \times n$ matrix X;
- reinterpret $\mathcal{A}x$ as linear operator $\mathcal{A} : X \mapsto \mathcal{A}(X)$.

Rayleigh quotients wrt low-rank matrices

Consider symmetric $n^2 \times n^2$ matrix $\mathcal{A}$. Then

$$\lambda_{\min}(\mathcal{A}) = \min_{X \neq 0} \frac{\langle X, \mathcal{A}(X) \rangle}{\langle X, X \rangle}$$

with matrix inner product $\langle \cdot, \cdot \rangle$. We now...

- restrict X to low-rank matrices.

Rayleigh quotients wrt low-rank matrices

Consider symmetric $n^2 \times n^2$ matrix $\mathcal{A}$. Then

$$\lambda_{\min}(\mathcal{A}) \approx \min_{X=UV^T\neq 0} \frac{\langle X, \mathcal{A}(X)\rangle}{\langle X, X\rangle}.$$

- Approximation error governed by low-rank approximability of X.
- Solved by Riemannian optimization techniques or ALS.

ALS for eigenvalue problem

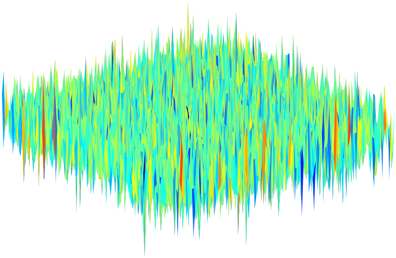
ALS for solving

$$\lambda_{\min}(\mathcal{A}) \approx \min_{X=UV^T \neq 0} \frac{\langle X, \mathcal{A}(X) \rangle}{\langle X, X \rangle}.$$

Initially:

- fix target rank r
- $U \in \mathbb{R}^{m \times r}$, $V^{n \times r}$ randomly, such that V is ONB

$\tilde{\lambda} - \lambda = 6 \times 10^3$
residual $= 3 \times 10^3$



ALS for eigenvalue problem

ALS for solving

$$\lambda_{\min}(\mathcal{A}) \approx \min_{X=UV^T\neq 0} \frac{\langle X, \mathcal{A}(X)\rangle}{\langle X, X\rangle}.$$

Fix V, optimize for U.

$$\begin{aligned}\langle X, \mathcal{A}(X)\rangle &= \mathrm{vec}(UV^T)^T \mathcal{A}\,\mathrm{vec}(UV^T) \\ &= \mathrm{vec}(U)^T (V \otimes I)^T \mathcal{A} (V \otimes I) \mathrm{vec}(U)\end{aligned}$$

$\rightsquigarrow$ Compute smallest eigenvalue of reduced matrix ($rn \times rn$) matrix

$$(V \otimes I)^T \mathcal{A} (V \otimes I).$$

Note: Computation of reduced matrix benefits from Kronecker structure of $\mathcal{A}$.

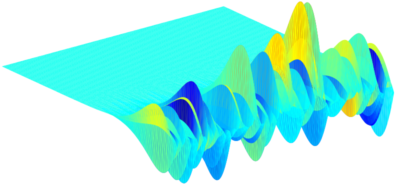
ALS for eigenvalue problem

ALS for solving

$$\lambda_{\min}(\mathcal{A}) \approx \min_{X = UV^T \neq 0} \frac{\langle X, \mathcal{A}(X) \rangle}{\langle X, X \rangle}.$$

Fix V, optimize for U.

$\tilde{\lambda} - \lambda = 2 \times 10^3$
residual $= 2 \times 10^3$



ALS for eigenvalue problem

ALS for solving

$$\lambda_{\min}(\mathcal{A}) \approx \min_{X=UV^T \neq 0} \frac{\langle X, \mathcal{A}(X) \rangle}{\langle X, X \rangle}.$$

Orthonormalize U, fix U, optimize for V.

$$\begin{aligned}\langle X, \mathcal{A}(X) \rangle &= \mathrm{vec}(UV^T)^T \mathcal{A}\, \mathrm{vec}(UV^T) \\ &= \mathrm{vec}(V^T)(I \otimes U)^T \mathcal{A}(I \otimes U)\mathrm{vec}(V^T)\end{aligned}$$

$\rightsquigarrow$ Compute smallest eigenvalue of reduced matrix ($rn \times rn$) matrix

$$(I \otimes U)^T \mathcal{A}(I \otimes U).$$

Note: Computation of reduced matrix benefits from Kronecker structure of $\mathcal{A}$.

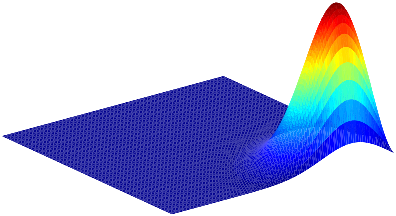
ALS for eigenvalue problem

ALS for solving

$$\lambda_{\min}(\mathcal{A}) \approx \min_{X = UV^T \neq 0} \frac{\langle X, \mathcal{A}(X) \rangle}{\langle X, X \rangle}.$$

Orthonormalize U, fix U, optimize for V.

$\tilde{\lambda} - \lambda = 1.5 \times 10^{-7}$
residual $= 7.7 \times 10^{-3}$



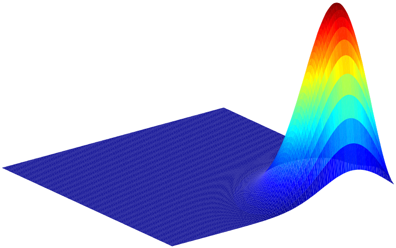
ALS

ALS for solving

$$\lambda_{\min}(\mathcal{A}) \approx \min_{X=UV^T \neq 0} \frac{\langle X, \mathcal{A}(X) \rangle}{\langle X, X \rangle}.$$

Orthonormalize V, fix V, optimize for U.

$\tilde{\lambda} - \lambda = 1 \times 10^{-12}$
residual $= 6 \times 10^{-7}$



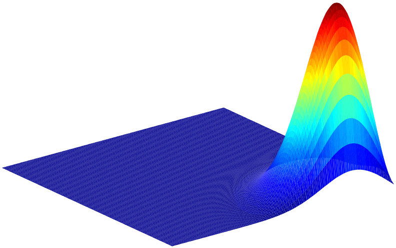
ALS for eigenvalue problem

ALS for solving

$$\lambda_{\min}(\mathcal{A}) \approx \min_{X=UV^T \neq 0} \frac{\langle X, \mathcal{A}(X) \rangle}{\langle X, X \rangle}.$$

Orthonormalize U, fix U, optimize for V.

$\tilde{\lambda} - \lambda = 7.6 \times 10^{-13}$
residual $= 7.2 \times 10^{-8}$



Extension of ALS to TT

Recall interface matrices

$$X_{\leq \mu-1} \in \mathbb{R}^{n_1 n_2 \cdots n_\mu \times r_{\mu-1}}, \quad X_{\geq \mu} \in \mathbb{R}^{n_{\mu+1} n_{\mu+2} \cdots n_d \times r_{\mu-1}}$$

yielding factorization

$$X^{<\mu>} = X_{\leq \mu-1} X_{\geq \mu}^T, \quad \mu = 1, \ldots, d-1.$$

Combined with recursion

$$X_{\geq \mu+1}^T = U_\mu^{\mathrm{R}} (X_{\geq \mu}^T \otimes I_{n_\mu}),$$

this yields

$$X^{<\mu>} = X_{\leq \mu-1} U_\mu^{\mathrm{R}} X_{\geq \mu+1}^T, \quad \mu = 1, \ldots, d-1.$$

Hence,

$$\mathrm{vec}(\mathcal{X}) = (X_{\geq \mu+1} \otimes X_{\leq \mu-1}) \, \mathrm{vec}(\mathcal{U}_\mu)$$

This formula allows us to pull out μth core!

Extension of ALS to TT

A TT decomposition is called μ-orthogonal if

$$(U_\nu^{\mathsf{L}})^T U_\nu^{\mathsf{L}} = I_{r_\nu}, \quad X_{\leq\nu}^T X_{\leq\nu} = I_{r_\nu} \quad \text{for} \quad \nu = 1, \dots, \mu - 1.$$

and

$$U_\nu^{\mathsf{R}} (U_\nu^{\mathsf{R}})^T = I_{r_\nu}, \quad X_{\geq\nu} X_{\geq\nu}^T = I_{r_\mu} \quad \text{for} \quad \nu = \mu + 1, \dots, d.$$

This implies that $X_{\geq\mu+1} \otimes X_{\leq\mu-1}$ has orthonormal columns!

Consider eigenvalue problem

$$\lambda_{\min}(\mathcal{A}) = \min_{\mathcal{X} \neq 0} \frac{\langle \mathcal{X}, \mathcal{A}(\mathcal{X}) \rangle}{\langle \mathcal{X}, \mathcal{X} \rangle}$$

Optimizing for μth core $\rightsquigarrow$

$$\min_{\mathcal{U}_\mu \neq 0} \frac{\langle \mathcal{X}, \mathcal{A}(\mathcal{X}) \rangle}{\langle \mathcal{X}, \mathcal{X} \rangle} = \min_{\mathcal{U}_\mu \neq 0} \frac{\langle \operatorname{vec} \mathcal{U}_\mu, \mathcal{A}_\mu \operatorname{vec} \mathcal{U}_\mu \rangle}{\langle \operatorname{vec} \mathcal{U}_\mu, \operatorname{vec} \mathcal{U}_\mu \rangle}$$

with $r_{\mu-1} n_\mu r_\mu \times r_{\mu-1} n_\mu r_\mu$ matrix

$$\mathcal{A}_\mu = (X_{\geq\mu+1} \otimes X_{\leq\mu-1})^T \mathcal{A} (X_{\geq\mu+1} \otimes X_{\leq\mu-1})$$

Extension of ALS to TT

- $\mathcal{U}_\mu$ is obtained as eigenvector belonging to smallest eigenvalue of $\mathcal{A}_\mu$.
- Computation of $\mathcal{A}_\mu$ for large d only feasible if $\mathcal{A}$ has low operator TT ranks (and is in operator TT decomposition).
- One microstep of ALS optimizes $\mathcal{U}_\mu$ and prepares for next core, by adjusting orthogonalization.
- One sweep of ALS consists of processing cores twice: once from left to right and once from right to left.

Extension of ALS to TT

Input: $\mathcal{X}$ in right-orthogonal TT decomposition.

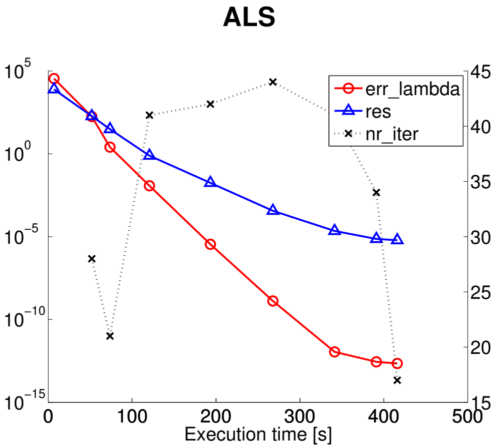
for $\mu = 1, 2, \ldots, d-1$ **do**
  Compute $\mathcal{A}_\mu$ and replace core $\mathcal{U}_\mu$ by an eigenvector belonging to smallest eigenvalue of $\mathcal{A}_\mu$.
  Compute QR decomposition $U_\mu^{\mathsf{L}} = QR$.
  Set $U_\mu^{\mathsf{L}} \leftarrow Q$.
  Update $U_{\mu+1} \leftarrow R \circ_1 U_{\mu+1}$.
end for
for $\mu = d, d-1, \ldots, 2$ **do**
  Compute $\mathcal{A}_\mu$ and replace core $\mathcal{U}_\mu$ by an eigenvector belonging to smallest eigenvalue of $\mathcal{A}_\mu$.
  Compute QR decomposition $(U_\mu^{\mathsf{R}})^T = QR$.
  Set $U_\mu^{\mathsf{R}} \leftarrow Q^T$.
  Update $U_{\mu-1} \leftarrow R \circ_3 U_{\mu-1}$.
end for

Extension of ALS to TT

Remarks:

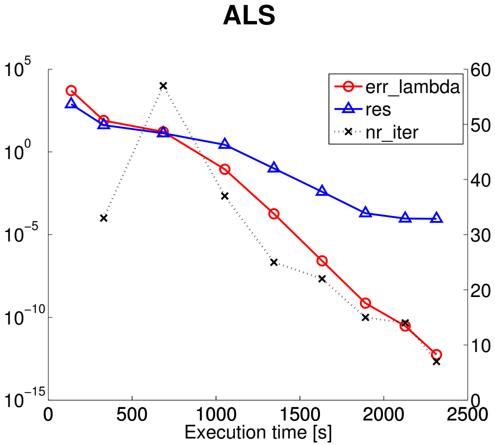
- "Small" matrix $\mathcal{A}_\mu$ quickly gets large as TT ranks increase $\rightsquigarrow$ Need to use iterative methods (e.g., Lanczos, LOBPCG), possibly combined with preconditioning [DK/Tobler'2011] for solving eigenvalue problems.
- In ALS TT ranks of $\mathcal{X}$ need to be chosen a priori. Adaptive choice of rank by merging neighbouring cores, optimizing for the merged core, and split the optimized merged core $\rightsquigarrow$ DMRG, modified ALS. Cheaper: AMEn [White'2005, Dolgov/Savostyanov'2013].
- Principles of ALS easily extend to other optimization problems, e.g., linear systems [Holtz/Rohwedder/Schneider'2012].

Numerical Experiments - Sine potential, $d = 10$



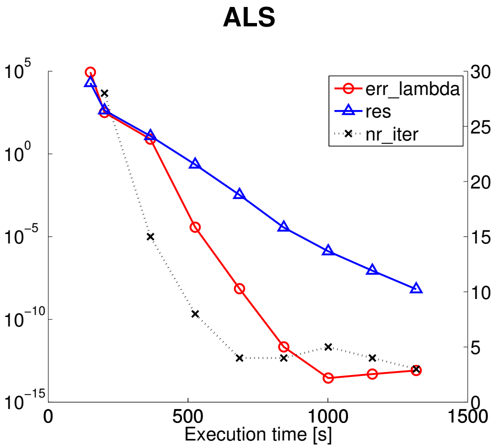
Size = $128^{10} \approx 10^{21}$. Maximal TT rank 40. See [Kressner/Steinlechner/Uschmajew'2014] for details.

Numerical Experiments - Henon-Heiles potential, $d = 20$



Size = $128^{20} \approx 10^{42}$. Maximal TT rank 40.

Numerical Experiments - $1/\|\xi\|_2$ potential, $d = 20$



Size = $128^{20} \approx 10^{42}$. Maximal TT rank 30.

Some ongoing work on low-rank approximation

- Dynamical low-rank approximation [Koch/Lubich'2007] with applications, e.g., to deep learning [Schotthöfer et al.'2022] and plasma physics [Einkemmer/Lubich'2018].
- Low-rank approximation $\rightsquigarrow$ entry-wise constraints and operations [Sarlos et al.'2023].
- Continuous limits and operator learning [Boullé/Townsend'2023].
- Representation/computation of high-dimensional pdfs through tensors [Dolgov et al. 2020–]
- Randomized techniuqes (stay tuned until Friday)