Near-optimal hierarchical matrix approximation from matrix-vector products

Tyler Chen

November 14, 2024

chen.pw/slides

Collaborators

Noah Amsel, Feyza Duman Keles, Diana Halikias,
David Persson, Chris Musco, Cameron Musco

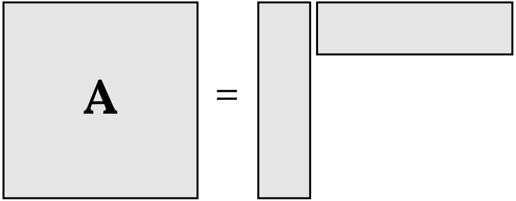
Paper to appear at SODA 2025. Available at: `https://arxiv.org/abs/2407.04686`.

Matrix recovery and approximation

Let S be some family of matrices parameterized by a few parameters.

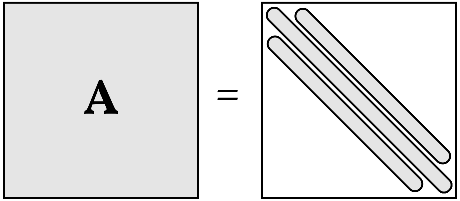
Matrix recovery and approximation

Let S be some family of matrices parameterized by a few parameters.



Matrix recovery and approximation

Let S be some family of matrices parameterized by a few parameters.



Matrix recovery and approximation

Let S be some family of matrices parameterized by a small number of parameters.

> **Recovery:** Promised $\mathbf{A} \in S$, learn parameterization of $\mathbf{A}$.

> **Approximation:** Arbitrary $\mathbf{A}$, learn (parameterization of) $\widetilde{\mathbf{A}} \in S$ such that
>
> $$\|\mathbf{A} - \widetilde{\mathbf{A}}\| \leq (1+\varepsilon) \min_{\mathbf{X} \in S} \|\mathbf{A} - \mathbf{X}\|.$$

How do we measure costs?

- number of arithmetic operations
- number of matrix-vector queries $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ or $\mathbf{y} \mapsto \mathbf{A}^\mathsf{T}\mathbf{y}$

Matrix recovery and approximation

Let S be some family of matrices parameterized by a small number of parameters.

> **Recovery:** Promised $\mathbf{A} \in S$, learn parameterization of $\mathbf{A}$.

> **Approximation:** Arbitrary $\mathbf{A}$, learn (parameterization of) $\widetilde{\mathbf{A}} \in S$ such that
>
> $$\|\mathbf{A} - \widetilde{\mathbf{A}}\| \leq (1+\varepsilon) \min_{\mathbf{X} \in S} \|\mathbf{A} - \mathbf{X}\|.$$

How do we measure costs?

- number of arithmetic operations
- number of matrix-vector queries $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ or $\mathbf{y} \mapsto \mathbf{A}^{\mathsf{T}}\mathbf{y}$

Matrix recovery and approximation

Let S be some family of matrices parameterized by a small number of parameters.

> **Recovery:** Promised $\mathbf{A} \in S$, learn parameterization of $\mathbf{A}$.

> **Approximation:** Arbitrary $\mathbf{A}$, learn (parameterization of) $\widetilde{\mathbf{A}} \in S$ such that
>
> $$\|\mathbf{A} - \widetilde{\mathbf{A}}\| \leq (1+\varepsilon) \min_{\mathbf{X} \in S} \|\mathbf{A} - \mathbf{X}\|.$$

How do we measure costs?

- number of arithmetic operations
- number of matrix-vector queries $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ or $\mathbf{y} \mapsto \mathbf{A}^\mathsf{T}\mathbf{y}$

Motivation 1: structured matrices are fast to work with

Suppose S is some family of easy to work with matrices.

Solve all your linear algebra problems with $\mathbf{A}$ in these three simple steps:

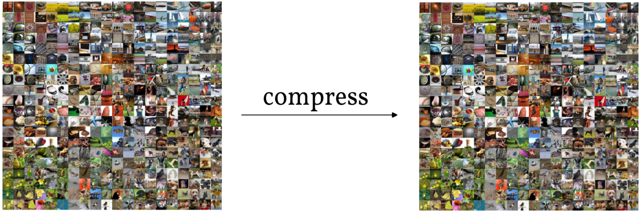
1. approximate $\mathbf{A}$ by $\widetilde{\mathbf{A}} \in S$
2. use structure of S to solve problem with $\widetilde{\mathbf{A}}$ quickly
3. pretend $\widetilde{\mathbf{A}}$ is $\mathbf{A}$ and declare success
   - or try to correct for the error

Motivation 1: structured matrices are fast to work with

Suppose S is some family of easy to work with matrices.

Solve all your linear algebra problems with $\mathbf{A}$ in these three simple steps:

1. approximate $\mathbf{A}$ by $\widetilde{\mathbf{A}} \in S$
2. use structure of S to solve problem with $\widetilde{\mathbf{A}}$ quickly
3. pretend $\widetilde{\mathbf{A}}$ is $\mathbf{A}$ and declare success
   - or try to correct for the error



very big

less very big

Motivation 1: structured matrices are fast to work with

Suppose S is some family of easy to work with matrices.

Solve all your linear algebra problems with $\mathbf{A}$ in these three simple steps:

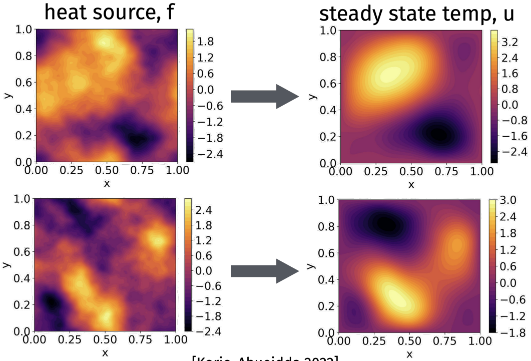
1. approximate $\mathbf{A}$ by $\widetilde{\mathbf{A}} \in S$
2. use structure of S to solve problem with $\widetilde{\mathbf{A}}$ quickly
3. pretend $\widetilde{\mathbf{A}}$ is $\mathbf{A}$ and declare success
   - or try to correct for the error

Examples of this framework:

- image-classification: $S =$ JPEG compressed images
- kernel spectral clustering: $S =$ low=rank matrices
- perform matrix products: $S =$ low-rank matrices, $S =$ sparse matrices, etc.
- solve regression problem, $S =$ low-rank matrices

Motivation 2: Operator Learning[1]

Physical processes often map a function f to a function u. I.e., implement some operator $\Phi(f) \mapsto u$.

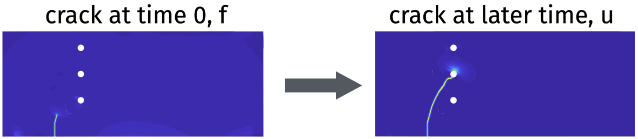


[Koric, Abueidda 2022]

[1] Boullé and Townsend 2024.

Motivation 2: Operator Learning[1]

Physical processes often map a function f to a function u. I.e., implement some operator $\Phi(f) \mapsto u$.



[Goswami, Anitescu, Rabczuk 2019]

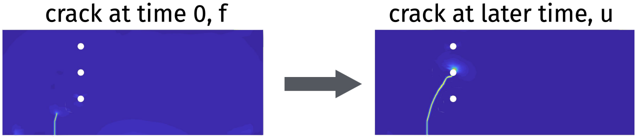
Goal: Learn mapping from input-output pairs: $(f_1, u_1), \ldots, (f_m, u_m)$.

Scientific ML: Assume S is some parameterized family (e.g. neural net as in DeepONet, DeepGreen, etc.)

[1] Boullé and Townsend 2024.

Motivation 2: Operator Learning[1]

Physical processes often map a function f to a function u. I.e., implement some operator $\Phi(f) \mapsto u$.



[Goswami, Anitescu, Rabczuk 2019]

Goal: Learn mapping from input-output pairs: $(f_1, u_1), \ldots, (f_m, u_m)$.

Scientific ML: Assume S is some parameterized family (e.g. neural net as in DeepONet, DeepGreen, etc.)

[1] Boullé and Townsend 2024.

What role can theory play?

Enginering: Come up with some algorithm and demonstrate it works empirically.

Applied math: Develop algorithms to provably solve the recovery problem.[2] Hope they work when $\mathbf{A}$ is not in S, but is very close to some matrix in S.

Theory: Guarantees for the approximation problem. Complexity lower bounds for the hardness of problems.

Low-rank approximation is has seen a lot of work from all of these perspectives. But other classes have relatively limited theory.

[2] Xia, Chandrasekaran, Gu, and Li 2010; Lin, Lu, and Ying 2011; Halikias and Townsend 2023; Levitt and Martinsson 2022a.

What role can theory play?

Enginering: Come up with some algorithm and demonstrate it works empirically.

Applied math: Develop algorithms to provably solve the recovery problem.[2] Hope they work when $\mathbf{A}$ is not in S, but is very close to some matrix in S.

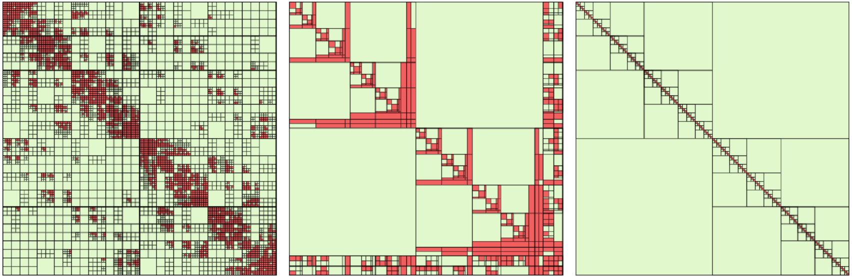
Theory: Guarantees for the approximation problem. Complexity lower bounds for the hardness of problems.

Low-rank approximation is has seen a lot of work from all of these perspectives. But other classes have relatively limited theory.

[2] Xia, Chandrasekaran, Gu, and Li 2010; Lin, Lu, and Ying 2011; Halikias and Townsend 2023; Levitt and Martinsson 2022a.

Hierarchical matrices

Today, S will be some family of hierarchical matrices.



Hierarchical matrices are useful for applications involving physical applications due to the presence of multiscale phenomena.

- example classes: hierarchical off-diagonal low-rank (HODLR), hierarchical semi-seperable (HSS), $\mathcal{H}^1$, $\mathcal{H}^2$, hierarchical off-diagonal butterfly, etc.

Why hierarchical matrices?

Argonne NATIONAL LABORATORY

BERKELEY LAB
Lawrence Berkeley National Laboratory

ButterflyPACK
Fast PDE solvers and transforms

HODLRlib

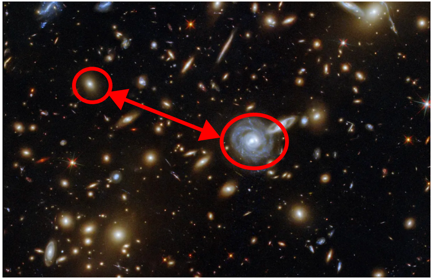
Motivating example: Suppose we're doing some n-body simulation and have the positions ($x_i \in \mathbb{R}^3$) of n celestial bodies in space.

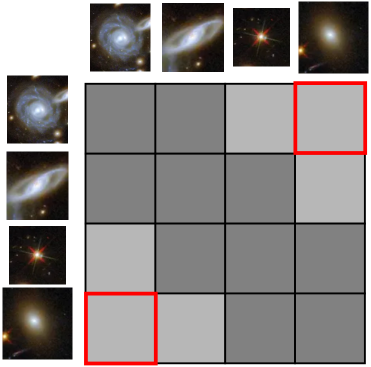
A relevant matrix is

$$\mathbf{A}_{i,j} = \|x_i - x_j\|^{-2}.$$

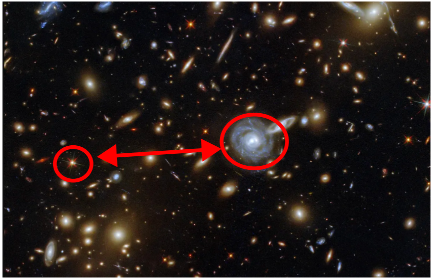
What does this matrix look like??

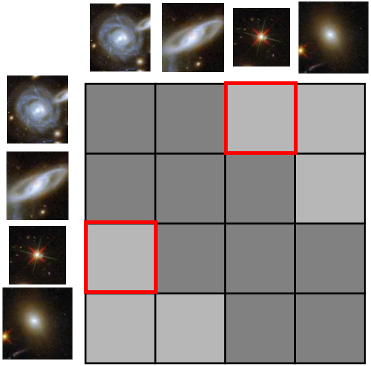
Motivating example



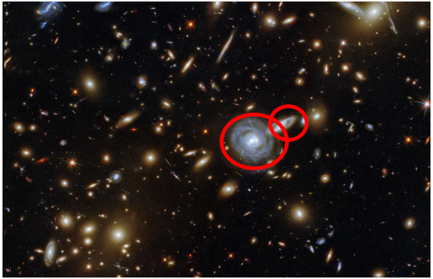


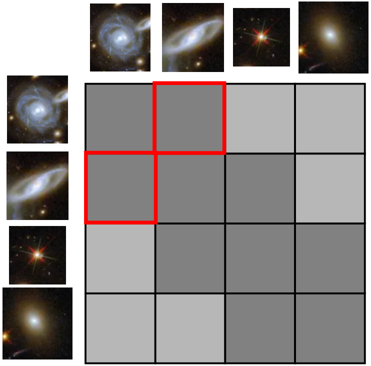
Motivating example



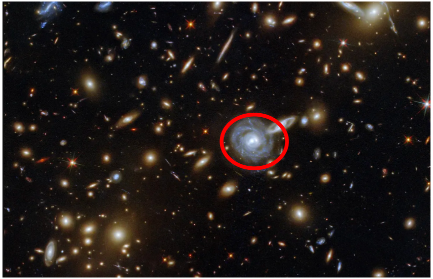


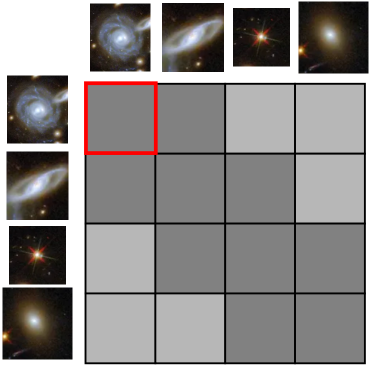
Motivating example



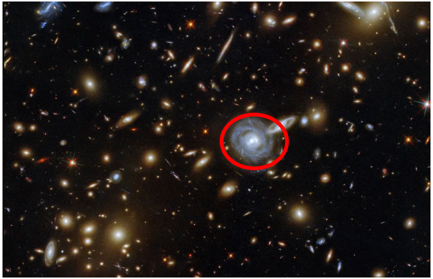


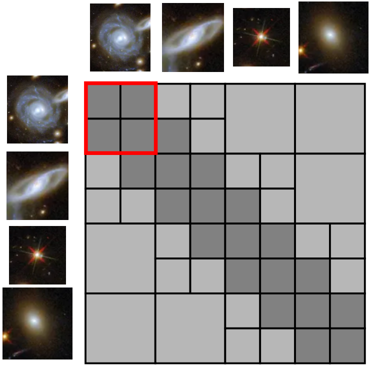
Motivating example



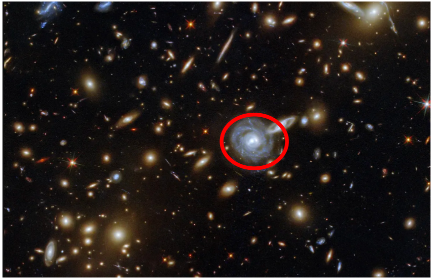


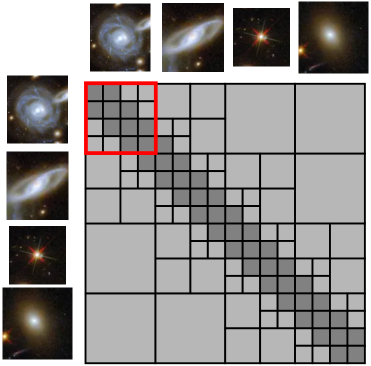
Motivating example



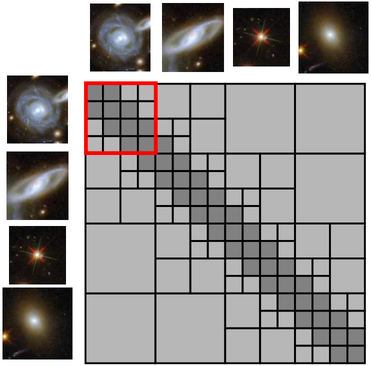


Motivating example

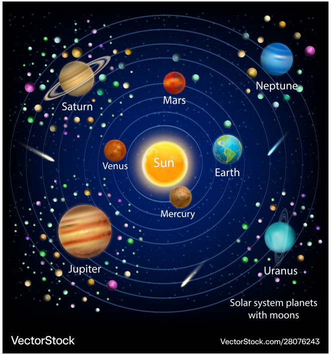


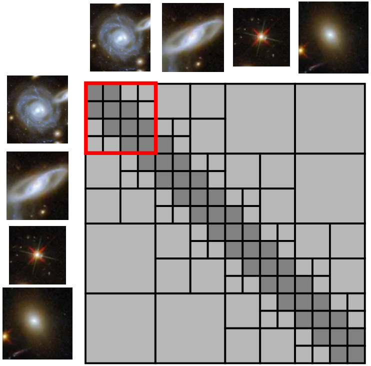


Motivating example

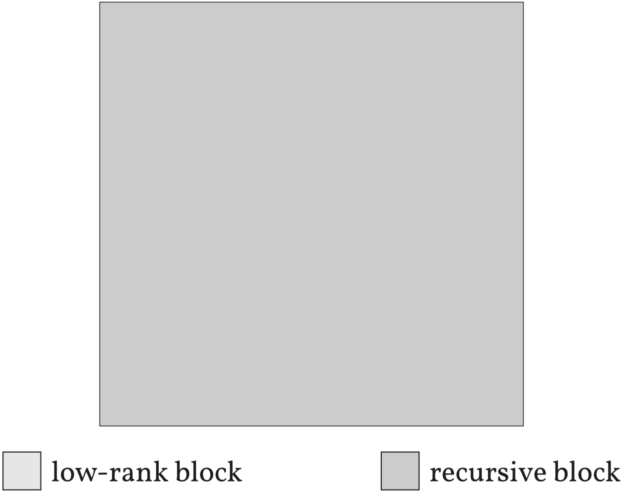


Motivating example

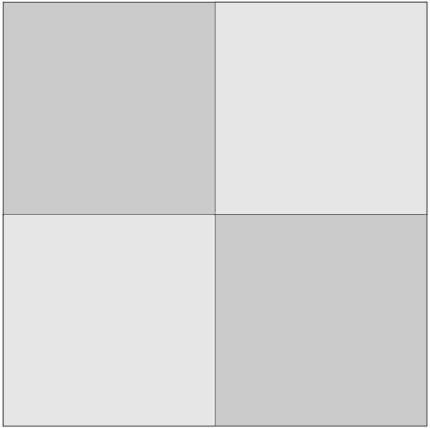




HODLR matrices



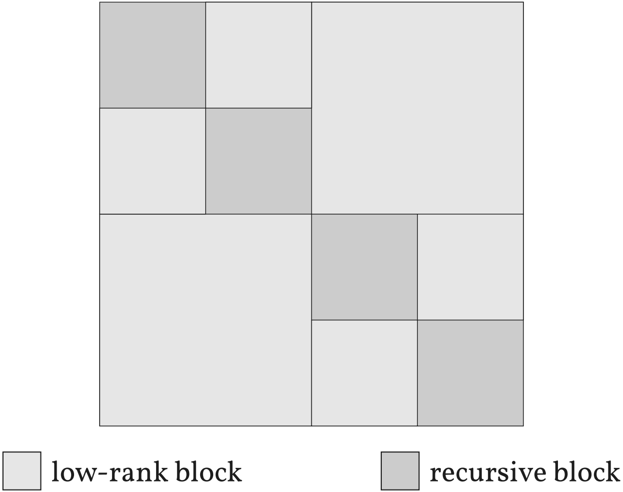
HODLR matrices



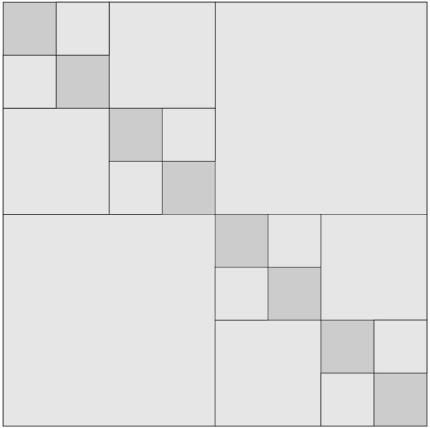
□ low-rank block

▨ recursive block

HODLR matrices



HODLR matrices



□ low-rank block

▨ recursive block

HODLR Matrices

Definition. Fix a rank parameter k. We say a $n \times n$ matrix $\mathbf{A}$ is HODLR(k) if $n \leq k$ or $\mathbf{A}$ can be partitioned into $(n/2) \times (n/2)$ blocks

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}$$

such that $\mathbf{A}_{1,2}$ and $\mathbf{A}_{2,1}$ are of rank at most k and $\mathbf{A}_{1,1}$ and $\mathbf{A}_{2,2}$ are each HODLR(k).

HODLR matries have $O(kn \log(n))$ parameters.

There are several matvec algorithms for the recovery problem.[3]

[3] Lin, Lu, and Ying 2011; Martinsson 2016; Levitt and Martinsson 2022b; Halikias and Townsend 2023.

Low-rank approximation from matrix-vector products

The Randomized SVD (RSVD) is a well-known algorithm for obtaining a low-rank approximation to a matrix $\mathbf{B}$:

1. Sample Gaussian matrix $\mathbf{\Omega}$
2. Form $\mathbf{Q} = \operatorname{orth}(\mathbf{B}\mathbf{\Omega})$
3. Compute $\mathbf{X} = \mathbf{Q}^\mathsf{T}\mathbf{B}$ (minimize: $\|\mathbf{B} - \mathbf{Q}\mathbf{X}\|_\mathsf{F}$)
4. Output $\mathbf{Q}[\![\mathbf{X}]\!]_k$

> **Theorem.** If $\mathbf{B}$ is rank-k, and $\mathbf{\Omega}$ has $O(k)$ columns, then $\mathbf{Q}[\![\mathbf{X}]\!]_k = \mathbf{B}$ (a.s.).

Peeling: an algorithm for the recovery problem[4]

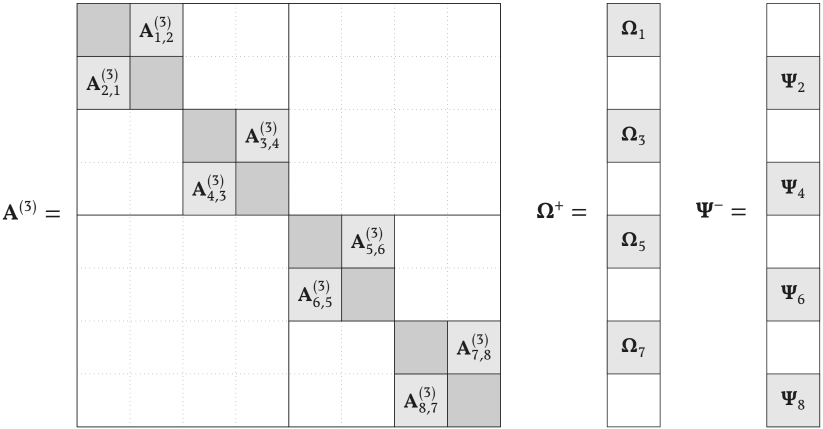
The algorithm works from the top layer down.

At each level, we simultaneosly apply the RSVD to the low-rank off-diagonal blocks.

We then “peel” off these blocks before proceeding to the next level

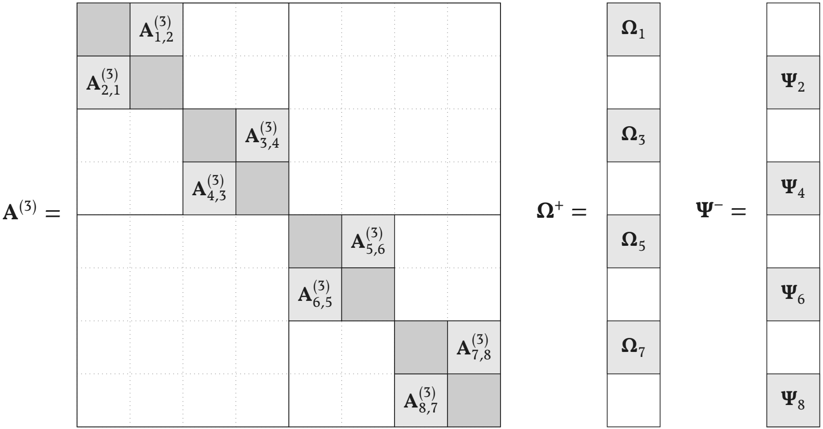
[4] Lin, Lu, and Ying 2011; Martinsson 2016.

Peeling: an algorithm for the recovery problem



From $\mathbf{A}^{(3)}\boldsymbol{\Omega}^{+}$ we get sketches: $\mathbf{A}^{(3)}_{2,1}\boldsymbol{\Omega}_1$, $\mathbf{A}^{(3)}_{4,3}\boldsymbol{\Omega}_3$, $\mathbf{A}^{(3)}_{6,5}\boldsymbol{\Omega}_5$, $\mathbf{A}^{(3)}_{8,7}\boldsymbol{\Omega}_7$.

Peeling: an algorithm for the recovery problem



From $\mathbf{A}^{(3)}\mathbf{\Omega}^{+}$ we get sketches: $\mathbf{A}^{(3)}_{2,1}\mathbf{\Omega}_1$, $\mathbf{A}^{(3)}_{4,3}\mathbf{\Omega}_3$, $\mathbf{A}^{(3)}_{6,5}\mathbf{\Omega}_5$, $\mathbf{A}^{(3)}_{8,7}\mathbf{\Omega}_7$.

Peeling: an algorithm for the recovery problem

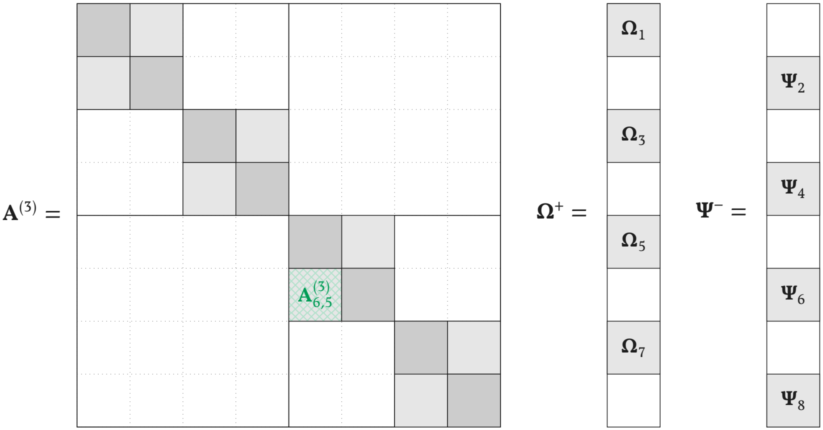
At each level we use $O(k)$ matrix-vector products with $\mathbf{A}$ and $\mathbf{A}^\top$.

There are $\log_2(n/k) \leq \log_2(n)$ levels until the blocks are of size k

- then we can directly recover them at once with k products

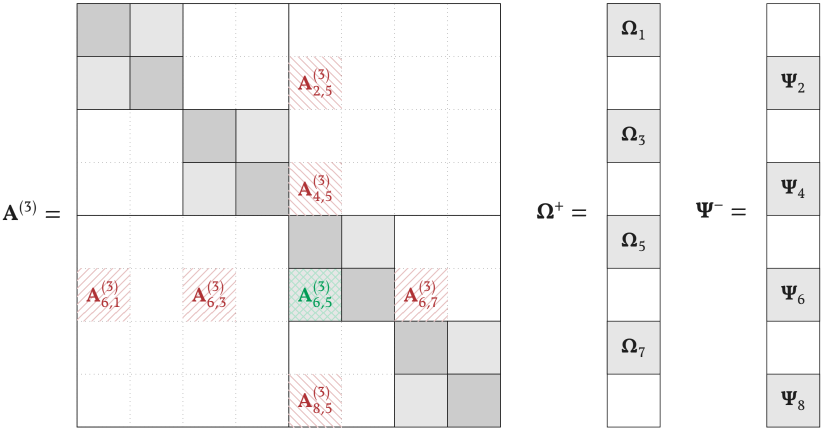
> **Theorem.** We can recover a HODLR matrix using $O\big(k \log_2(n)\big)$ matvecs.

Does peeling work on non-HODLR matrices?



$$\mathbf{A}^{(3)}_{6,5}\mathbf{\Omega}_5$$

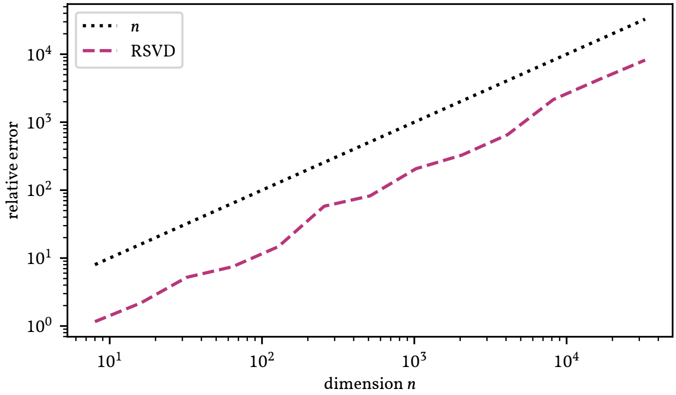
Does peeling work on non-HODLR matrices?



$$\mathbf{A}^{(3)}_{6,5}\boldsymbol{\Omega}_5 + \mathbf{A}^{(3)}_{6,1}\boldsymbol{\Omega}_1 + \mathbf{A}^{(3)}_{6,3}\boldsymbol{\Omega}_3 + \mathbf{A}^{(3)}_{6,7}\boldsymbol{\Omega}_7$$

Does peeling work on non-HODLR matrices?

If all the error at a level can propagate to the next level, then the total error doubles at each level. Exponential blow-up in the number of levels (linear in n)!



What’s going on? An illustration.

Suppose $\mathbf{X}$ and $\mathbf{Y}$ are rank k and $\mathbf{Y}$ is way bigger than $\mathbf{X}$. Consider

$$\mathbf{A} = \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{Y} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{Y} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right].$$

When we recover the low-rank blocks at the first level we will essentially get

$$\begin{bmatrix} \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

What’s going on? An illustration.

Next we subtract off these approximations:

$$\left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{Y} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{Y} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right] - \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{0} & \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{Y} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] = \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right].$$

What's going on? An illustration.

Now we sketch to learn the subspaces at the next level:

$$\left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \mathbf{\Omega}_1^\dagger \\ \mathbf{0} \\ \hdashline \mathbf{\Omega}_3^\dagger \\ \mathbf{0} \end{array}\right] = \left[\begin{array}{c} \mathbf{0} \\ \mathbf{X}(\mathbf{\Omega}_1^\dagger + \mathbf{\Omega}_3^\dagger) \\ \hdashline \mathbf{0} \\ \mathbf{X}(\mathbf{\Omega}_1^\dagger + \mathbf{\Omega}_3^\dagger) \end{array}\right].$$

We then compute $\mathbf{Q} = \operatorname{orth}(\mathbf{X}(\mathbf{\Omega}_1^\dagger + \mathbf{\Omega}_3^\dagger))$ and get the correct range for $\mathbf{X}$

What's going on? An illustration.

However, we run into problems at the projection stage:

$$\left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{Q}^\mathsf{T} & \mathbf{0} & \mathbf{Q}^\mathsf{T} \end{array}\right] \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right] = \left[\begin{array}{cc:cc} 2\mathbf{Q}^\mathsf{T}\mathbf{X} & \mathbf{0} & 2\mathbf{Q}^\mathsf{T}\mathbf{X} & \mathbf{0} \end{array}\right].$$

So our approximation to the off-diagonal blocks at this level is completely wrong…
We get $2\mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{X} = 2\mathbf{X}$ instead of $\mathbf{X}$.

All of the error from the first level propagated to the second level!

Accurate HODLR approximation?

This peeling type of algorithm is used in operator learning to approximate the solution operator of elliptic PDEs (2024 SIAM Linear Algebra Best Paper Prize winner).[5]

> **Boullé and Townsend 2022:** Is there a peeling-type algorithm that works for nearly-HODLR matrices?

[5] Boullé and Townsend 2022.

The HODLR approximation problem[6]

Problem. Given an $n \times n$ matrix $\mathbf{A}$, accessible only by matrix-vector products, a rank parameter k, and an accuracy parameter ε, find a HODLR(k) matrix $\widetilde{\mathbf{A}}$ such that

$$\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}} \leq (1+\varepsilon) \min_{\mathbf{H} \in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}.$$

Theorem. There is an efficient matvec algorithm for HODLR approximation.

Note: The best HODLR approximation to $\mathbf{A}$ is obtained by applying a rank-k SVD to each low-rank block of $\mathbf{A}$.

- This is too expensive in the matrix-vector product model (n products)

[6] Chen et al. 2025.

The HODLR approximation problem[6]

Problem. Given an $n \times n$ matrix $\mathbf{A}$, accessible only by matrix-vector products, a rank parameter k, and an accuracy parameter ε, find a HODLR(k) matrix $\widetilde{\mathbf{A}}$ such that

$$\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}} \le (1+\varepsilon) \min_{\mathbf{H} \in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}.$$

Theorem. There is an efficient matvec algorithm for HODLR approximation.

Note: The best HODLR approximation to $\mathbf{A}$ is obtained by applying a rank-k SVD to each low-rank block of $\mathbf{A}$.

- This is too expensive in the matrix-vector product model (n products)

[6] Chen et al. 2025.

The HODLR approximation problem[6]

> **Problem.** Given an $n \times n$ matrix $\mathbf{A}$, accessible only by matrix-vector products, a rank parameter k, and an accuracy parameter ε, find a HODLR(k) matrix $\widetilde{\mathbf{A}}$ such that
>
> $$\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}} \leq (1+\varepsilon) \min_{\mathbf{H} \in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}.$$

> **Theorem.** There is an efficient matvec algorithm for HODLR approximation.

Note: The best HODLR approximation to $\mathbf{A}$ is obtained by applying a rank-k SVD to each low-rank block of $\mathbf{A}$.

- This is too expensive in the matrix-vector product model (n products)

[6] Chen et al. 2025.

The HODLR approximation problem[6]

Problem. Given an $n \times n$ matrix $\mathbf{A}$, accessible only by matrix-vector products, a rank parameter k, and an accuracy parameter ε, find a HODLR(k) matrix $\widetilde{\mathbf{A}}$ such that

$$\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}} \le (1+\varepsilon) \min_{\mathbf{H} \in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}.$$

Theorem. There is an efficient matvec algorithm for HODLR approximation.

Note: The best HODLR approximation to $\mathbf{A}$ is obtained by applying a rank-k SVD to each low-rank block of $\mathbf{A}$.

- This is too expensive in the matrix-vector product model (n products)

[6] Chen et al. 2025.

Classical RSVD analysis[7]

Theorem. Let $\mathbf{Q} = \operatorname{orth}(\mathbf{B\Omega})$ and $\mathbf{X} = \mathbf{Q}^\top\mathbf{B}$. If $\mathbf{\Omega}$ has $O(k/\varepsilon)$ columns, then output of RSVD satisfies

$$\mathbb{E}\Big[\|\mathbf{B} - \mathbf{Q}[\![\mathbf{X}]\!]_k\|_\mathsf{F}^2\Big] \le (1+\varepsilon)\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_\mathsf{F}^2.$$

Structural perturbation bound:

$$\|\mathbf{B} - \mathbf{Q}[\![\mathbf{X}]\!]_k\|_\mathsf{F}^2 \le \|\mathbf{\Sigma}_{\mathrm{bot}}\|_\mathsf{F}^2 + \|\mathbf{\Sigma}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{top}}^\dagger\|_\mathsf{F}^2.$$

When $\mathbf{\Omega}$ is Gaussian and has $m \ge k+2$ columns:

$$\mathbb{E}\Big[\|\mathbf{\Sigma}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{top}}^\dagger\|_\mathsf{F}^2\Big] = \|\mathbf{\Sigma}_{\mathrm{bot}}\|_\mathsf{F}^2 \cdot \mathbb{E}\Big[\|\mathbf{\Omega}_{\mathrm{top}}^\dagger\|_\mathsf{F}^2\Big] = \frac{k}{m-k-1}\|\mathbf{\Sigma}_{\mathrm{bot}}\|_\mathsf{F}^2.$$

[7] Halko, Martinsson, and Tropp 2011; Tropp and Webber 2023.

Classical RSVD analysis[7]

Theorem. Let $\mathbf{Q} = \operatorname{orth}(\mathbf{B\Omega})$ and $\mathbf{X} = \mathbf{Q}^\top\mathbf{B}$. If $\mathbf{\Omega}$ has $O(k/\varepsilon)$ columns, then output of RSVD satisfies

$$\mathbb{E}\Big[\, \|\mathbf{B} - \mathbf{Q}[\![\mathbf{X}]\!]_k\|_{\mathsf{F}}^2 \Big] \le (1+\varepsilon)\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_{\mathsf{F}}^2.$$

Structural perturbation bound:

$$\|\mathbf{B} - \mathbf{Q}[\![\mathbf{X}]\!]_k\|_{\mathsf{F}}^2 \le \|\mathbf{\Sigma}_{\mathrm{bot}}\|_{\mathsf{F}}^2 + \|\mathbf{\Sigma}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{top}}^\dagger\|_{\mathsf{F}}^2.$$

When $\mathbf{\Omega}$ is Gaussian and has $m \ge k+2$ columns:

$$\mathbb{E}\Big[\, \|\mathbf{\Sigma}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{top}}^\dagger\|_{\mathsf{F}}^2 \Big] = \|\mathbf{\Sigma}_{\mathrm{bot}}\|_{\mathsf{F}}^2 \cdot \mathbb{E}\Big[\, \|\mathbf{\Omega}_{\mathrm{top}}^\dagger\|_{\mathsf{F}}^2 \Big] = \frac{k}{m-k-1}\|\mathbf{\Sigma}_{\mathrm{bot}}\|_{\mathsf{F}}^2.$$

[7] Halko, Martinsson, and Tropp 2011; Tropp and Webber 2023.

A perturbation bound for the RSVD

We prove a perturbation bound for the RSVD. This is likely of independent interest.

Theorem. Let $\mathbf{Q} = \operatorname{orth}(\mathbf{B\Omega} + \mathbf{E}_1)$ and $\mathbf{X} = \mathbf{Q}^\top\mathbf{B} + \mathbf{E}_2$. Then

$$\|\mathbf{B} - \mathbf{Q}[\![\mathbf{X}]\!]_k\|_\mathsf{F} \le \underbrace{\|\mathbf{E}_1\mathbf{\Omega}_{\mathrm{top}}^\dagger\|_\mathsf{F} + 2\|\mathbf{E}_2\|_\mathsf{F}}_{\text{perturbations}} + \underbrace{\left(\|\mathbf{\Sigma}_{\mathrm{bot}}\|_\mathsf{F}^2 + \|\mathbf{\Sigma}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{bot}}\mathbf{\Omega}_{\mathrm{top}}^\dagger\|_\mathsf{F}^2\right)^{1/2}}_{\text{classical RSVD bound}}.$$

When $\mathbf{\Omega}$ has $O(k/\varepsilon)$ columns, $\mathbf{\Omega}_{\mathrm{top}}$ is a $k \times O(k/\varepsilon)$ Gaussian matrix which has a small pseudoinverse:

$$\mathbb{E}\left[(\mathbf{\Omega}_{\mathrm{top}}^\dagger)^\top\mathbf{\Omega}_{\mathrm{top}}^\dagger\right] = \mathbb{E}\left[(\mathbf{\Omega}_{\mathrm{top}}\mathbf{\Omega}_{\mathrm{top}}^\top)^{-1}\right] = \varepsilon\mathbf{I}.$$

Takeaway: The pseudoinverse will help damp the perturbation $\mathbf{E}_1$, but (unsurprisingly) all of the perturbation $\mathbf{E}_2$ can propagate.

Generalized Nyström[8]

The RSVD tries to compute $\mathbf{Q}^\top\mathbf{B}$ directly; this is the solution to:

$$\min_{\mathbf{X}} \|\mathbf{A} - \mathbf{Q}\mathbf{X}\|_{\mathsf{F}}.$$

Instead, we can solve a sketched problem:

$$\min_{\mathbf{X}} \|\mathbf{\Psi}^\top\mathbf{A} - \mathbf{\Psi}^\top\mathbf{Q}\mathbf{X}\|_{\mathsf{F}}.$$

This means $\mathbf{X} = (\mathbf{\Psi}^\top\mathbf{Q})^\dagger\mathbf{\Psi}^\top\mathbf{A}$.

Observation. By adding columns to $\mathbf{\Psi}$, we can damp errors in the product $\mathbf{\Psi}^\top\mathbf{A}$.

The algorithm is also non-adaptive (we can do products with $\mathbf{\Psi}$ in advance)

[8] Clarkson and Woodruff 2009; Tropp, Yurtsever, Udell, and Cevher 2017; Nakatsukasa 2020.

Generalized Nyström[8]

The RSVD tries to compute $\mathbf{Q}^\top\mathbf{B}$ directly; this is the solution to:

$$\min_{\mathbf{X}} \|\mathbf{A} - \mathbf{Q}\mathbf{X}\|_\mathsf{F}.$$

Instead, we can solve a sketched problem:

$$\min_{\mathbf{X}} \|\mathbf{\Psi}^\top\mathbf{A} - \mathbf{\Psi}^\top\mathbf{Q}\mathbf{X}\|_\mathsf{F}.$$

This means $\mathbf{X} = (\mathbf{\Psi}^\top\mathbf{Q})^\dagger\mathbf{\Psi}^\top\mathbf{A}$.

Observation. By adding columns to $\mathbf{\Psi}$, we can damp errors in the product $\mathbf{\Psi}^\top\mathbf{A}$.

The algorithm is also non-adaptive (we can do products with $\mathbf{\Psi}$ in advance)

[8] Clarkson and Woodruff 2009; Tropp, Yurtsever, Udell, and Cevher 2017; Nakatsukasa 2020.

Generalized Nyström[8]

The RSVD tries to compute $\mathbf{Q}^\top \mathbf{B}$ directly; this is the solution to:

$$\min_{\mathbf{X}} \|\mathbf{A} - \mathbf{Q}\mathbf{X}\|_{\mathsf{F}}.$$

Instead, we can solve a sketched problem:

$$\min_{\mathbf{X}} \|\boldsymbol{\Psi}^\top \mathbf{A} - \boldsymbol{\Psi}^\top \mathbf{Q}\mathbf{X}\|_{\mathsf{F}}.$$

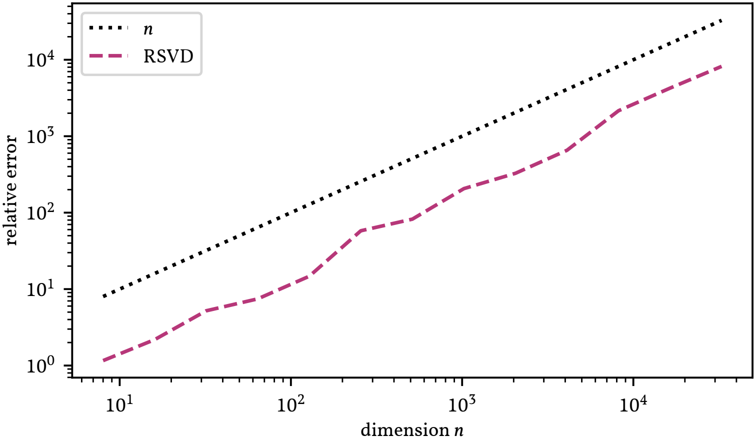
This means $\mathbf{X} = (\boldsymbol{\Psi}^\top \mathbf{Q})^\dagger \boldsymbol{\Psi}^\top \mathbf{A}$.

Observation. By adding columns to $\boldsymbol{\Psi}$, we can damp errors in the product $\boldsymbol{\Psi}^\top \mathbf{A}$.

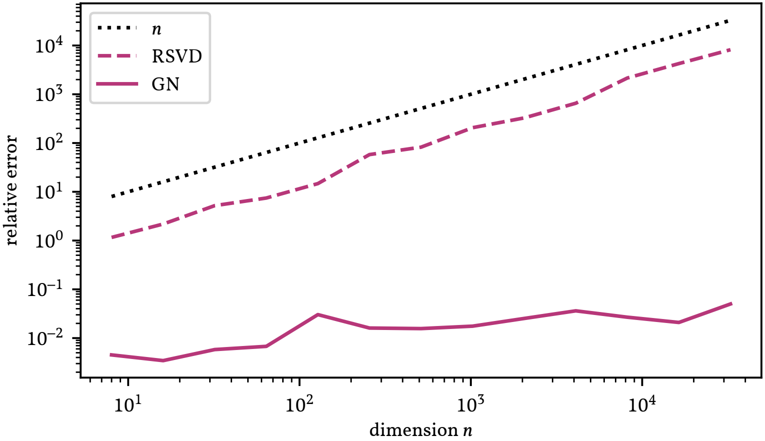
The algorithm is also non-adaptive (we can do products with $\boldsymbol{\Psi}$ in advance)

[8] Clarkson and Woodruff 2009; Tropp, Yurtsever, Udell, and Cevher 2017; Nakatsukasa 2020.

Back to the hard instance



Back to the hard instance

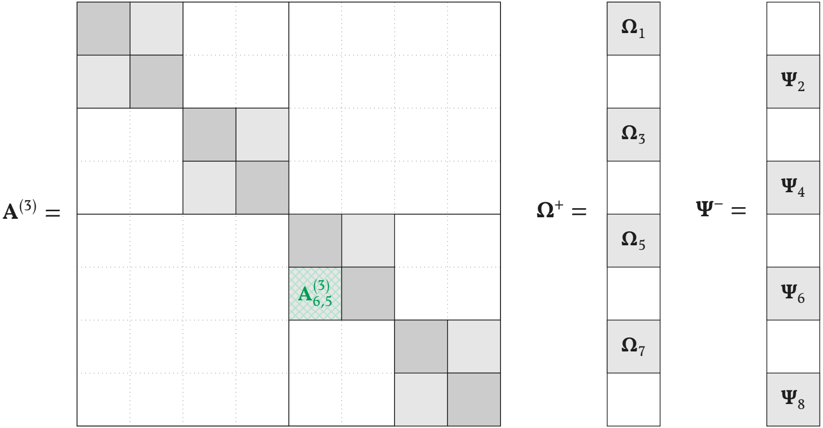


Another approach: perforated sketches

Because of the structure of peeling, the error happens when blocks of our sketch hit the error from our approximation of low-rank blocks at previous levels.

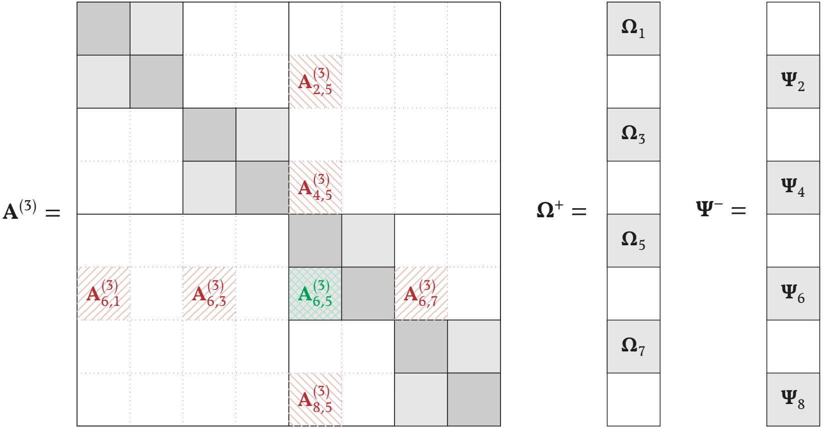
What if we just reduce how often this happens?

Perforated Block CountSketch



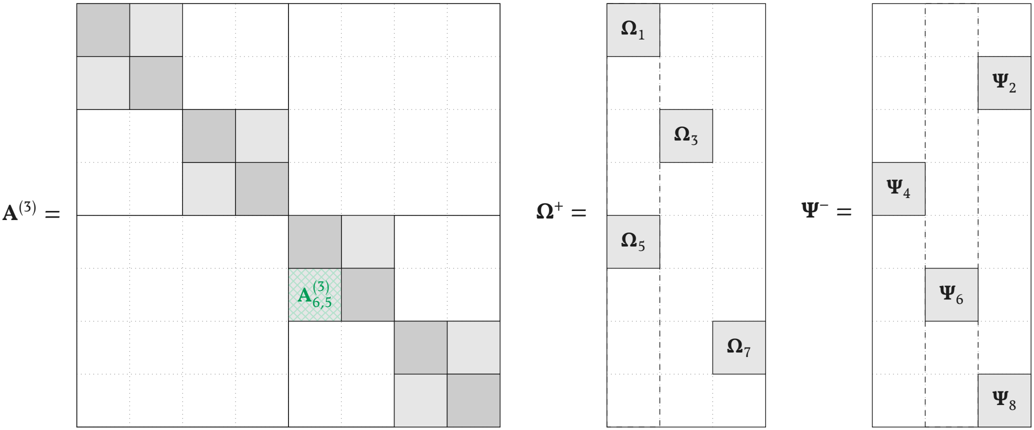
$$\mathbf{A}^{(3)}_{6,5}\mathbf{\Omega}_5$$

Perforated Block CountSketch



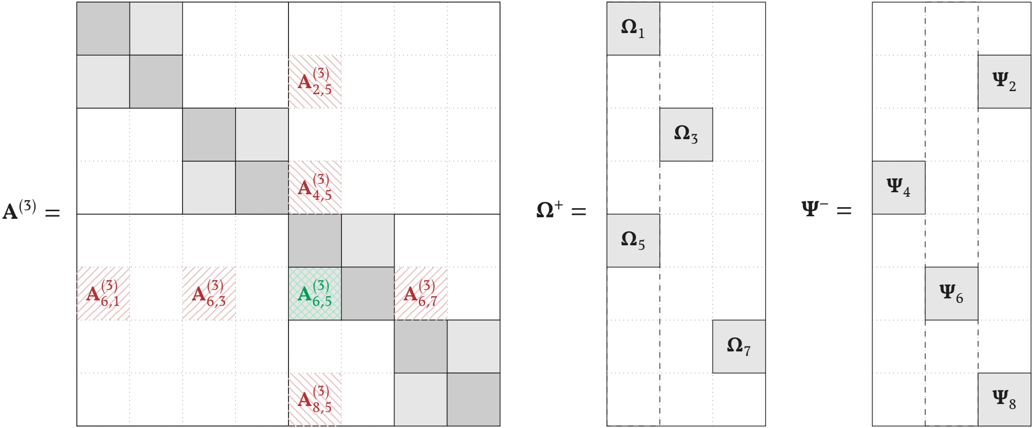
$$\mathbf{A}^{(3)}_{6,5}\mathbf{\Omega}_5 + \mathbf{A}^{(3)}_{6,1}\mathbf{\Omega}_1 + \mathbf{A}^{(3)}_{6,3}\mathbf{\Omega}_3 + \mathbf{A}^{(3)}_{6,7}\mathbf{\Omega}_7$$

Perforated Block CountSketch



$$\mathbf{A}^{(3)}_{6,5}\mathbf{\Omega}_5 + \mathbf{A}^{(3)}_{6,1}\mathbf{\Omega}_1$$

Perforated Block CountSketch



$$\mathbf{A}^{(3)}_{6,5}\mathbf{\Omega}_5 + \mathbf{A}^{(3)}_{6,1}\mathbf{\Omega}_1$$

Our main result

Theorem. There exist matvec algorithms which use $O\big(k \log(n)/\beta^3\big)$ products with $\mathbf{A}$ to obtain a HODLR(k) matrix $\widetilde{\mathbf{A}}$ satisfying

$$\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}} \le (1+\beta)^{\log_2(n)} \min_{\mathbf{H} \in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}.$$

Corollary. $(1+\varepsilon)$-optimal approximation with $O\big(k \log(n)^4/\varepsilon^3\big)$ matvecs

Corollary. $n^{0.01}$-optimal approximation with $O\big(k \log(n)\big)$ matvecs

Our main result

Theorem. There exist matvec algorithms which use $O\big(k\log(n)/\beta^3\big)$ products with $\mathbf{A}$ to obtain a HODLR(k) matrix $\widetilde{\mathbf{A}}$ satisfying

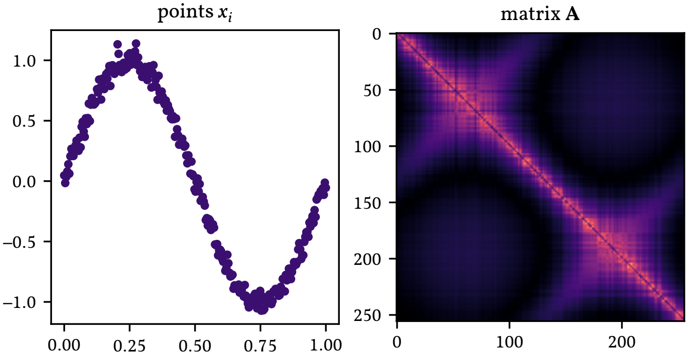
$$\|\mathbf{A}-\widetilde{\mathbf{A}}\|_{\mathsf{F}} \le (1+\beta)^{\log_2(n)} \min_{\mathbf{H}\in\mathrm{HODLR}(k)} \|\mathbf{A}-\mathbf{H}\|_{\mathsf{F}}.$$

Corollary. $(1+\varepsilon)$-optimal approximation with $O\big(k\log(n)^4/\varepsilon^3\big)$ matvecs

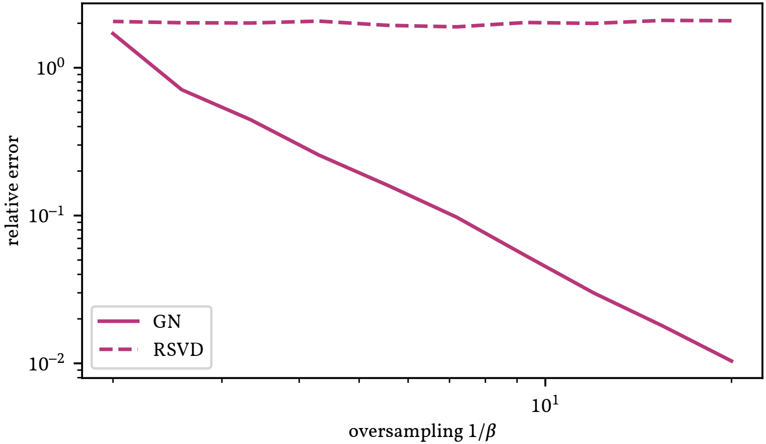
Corollary. $n^{0.01}$-optimal approximation with $O\big(k\log(n)\big)$ matvecs

Another experiment

Given points $x_i \in \mathbb{R}^2$, define $[\mathbf{A}]_{i,j} = -\log(\|x_i - x_j\|)$



Another experiment



Lower bounds?

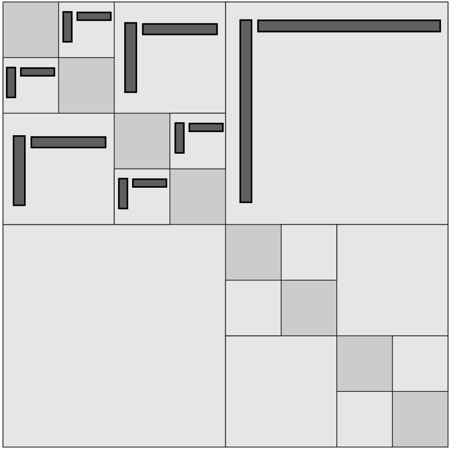
The matrix-vector query model often lets us prove lower-bounds against any matvec algorithm for a given task; i.e. study the complexity of a task.

This provides a very different approach for understanding how good algorithms are (compared to classical numerical analysis).

> **Theorem.** There is a constant $C > 0$ such that for any k, n, ε, there exists a matrix $\mathbf{A}$ such that getting a $(1+\varepsilon)$-optimal HODLR approximation requires at least $C\big(k\log_2(n/k) + k/\varepsilon\big)$ matvecs.

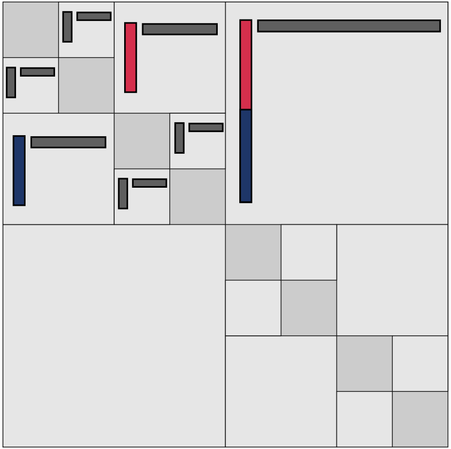
HSS matrices

The low-rank blocks of HSS matrices are related: $O(nk)$ parameters.



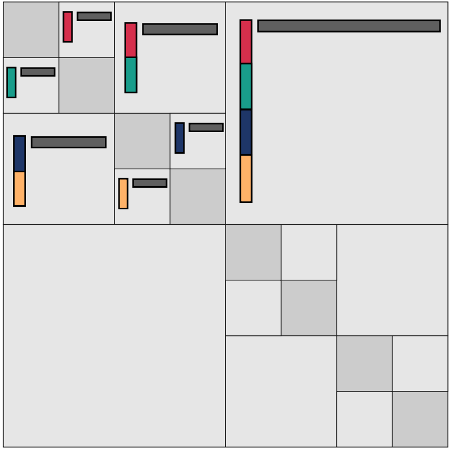
HSS matrices

The low-rank blocks of HSS matrices are related: $O(nk)$ parameters.



HSS matrices

The low-rank blocks of HSS matrices are related: $O(nk)$ parameters.



HSS is tricky!

Many papers study HSS recovery.[9]

The nestedness of column-spaces across levels adds lots of relations which make the approximation problem much harder.

- No known polynomial algorithm known for constant factor HSS approximation?!
- In fact, not even clear what to do in exponential time.

We prove:

> **Theorem.** Can get $O(\log(n))$ HSS approximation in $O(kn^2)$ time.

[9] Xia, Chandrasekaran, Gu, and Li 2010; Levitt and Martinsson 2022b; Halikias and Townsend 2023.

Some intuition for why HSS might be hard

> **Toy problem:** Fix matrices $\mathbf{A}_{i,j}$ for $i, j \in [q]$. Find matrices $\mathbf{U}_i$ and $\mathbf{V}_j$ with k orthonormal columns minimizing
>
> $$\sum_{i=1}^{q} \sum_{j=1}^{q} \|\mathbf{A}_{i,j} - \mathbf{U}_i \mathbf{X}_{i,j} \mathbf{V}_j^\mathsf{T}\|_\mathsf{F}^2, \qquad \mathbf{X}_{i,j} := \mathbf{U}_i^\mathsf{T} \mathbf{A}_{i,j} \mathbf{V}_j^\mathsf{T}.$$

Greedy approach: first find all the $\mathbf{U}_i$, then based on these, find the $\mathbf{V}_j$.

- gives 2-factor approximation

What's next?

Big goal: general theory for structured matrix approximation problem

- Correct $\log(n)$ and ε rates for the algorithms we study?
    - Limited by the best known bounds for Generalized Nyström: $O(k/\varepsilon^3)$
- True stability analysis (e.g. for floating point arithmetic)
    - Working on with students at NYU
- Adaptive algorithms
- Other hierarchical classes?
- Better understanding of (non-adaptive) low-rank approximation

Generalized Nyström (perturbation) analysis

Extend $\mathbf{Q}$ to an orthogonal matrix $[\mathbf{Q}\,\widehat{\mathbf{Q}}]$, and write $\boldsymbol{\Psi}_1 = \boldsymbol{\Psi}^\top\mathbf{Q}$ and $\boldsymbol{\Psi}_2 = \boldsymbol{\Psi}^\top\widehat{\mathbf{Q}}$.

By orthogonal invariance, $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$ are independent Gaussian matrices!

First observe:

$$\boldsymbol{\Psi}^\top\mathbf{B} = \boldsymbol{\Psi}^\top(\mathbf{Q}\mathbf{Q}^\top + \widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^\top)\mathbf{B} = \boldsymbol{\Psi}_1\mathbf{Q}^\top\mathbf{B} + \boldsymbol{\Psi}_2\widehat{\mathbf{Q}}^\top\mathbf{B}.$$

Therefore:

$$\mathbf{X} = (\boldsymbol{\Psi}^\top\mathbf{Q})^\dagger(\boldsymbol{\Psi}^\top\mathbf{B}) = \boldsymbol{\Psi}_1^\dagger\boldsymbol{\Psi}_1\mathbf{Q}^\top\mathbf{B} + \boldsymbol{\Psi}_1^\dagger\boldsymbol{\Psi}_2\widehat{\mathbf{Q}}^\top\mathbf{B} = \mathbf{Q}^\top\mathbf{B} + \boldsymbol{\Psi}_1^\dagger\boldsymbol{\Psi}_2\widehat{\mathbf{Q}}^\top\mathbf{B}.$$

Adding more columns to $\boldsymbol{\Psi}$ (and hence $\boldsymbol{\Psi}_1$) reduces the error in the second term.

Generalized Nyström (perturbation) analysis

Extend $\mathbf{Q}$ to an orthogonal matrix $[\mathbf{Q}\ \widehat{\mathbf{Q}}]$, and write $\boldsymbol{\Psi}_1 = \boldsymbol{\Psi}^\top\mathbf{Q}$ and $\boldsymbol{\Psi}_2 = \boldsymbol{\Psi}^\top\widehat{\mathbf{Q}}$.

By orthogonal invariance, $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$ are independent Gaussian matrices!

First observe:

$$\boldsymbol{\Psi}^\top\mathbf{B} + \mathbf{E} = \boldsymbol{\Psi}^\top(\mathbf{Q}\mathbf{Q}^\top + \widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^\top)\mathbf{B} + \mathbf{E} = \boldsymbol{\Psi}_1\mathbf{Q}^\top\mathbf{B} + \boldsymbol{\Psi}_2\widehat{\mathbf{Q}}^\top\mathbf{B} + \mathbf{E}.$$

Therefore:

$$\mathbf{X} = (\boldsymbol{\Psi}^\top\mathbf{Q})^\dagger(\boldsymbol{\Psi}^\top\mathbf{B} + \mathbf{E}) = \boldsymbol{\Psi}_1^\dagger\boldsymbol{\Psi}_1\mathbf{Q}^\top\mathbf{B} + \boldsymbol{\Psi}_1^\dagger\boldsymbol{\Psi}_2\widehat{\mathbf{Q}}^\top\mathbf{B} + \boldsymbol{\Psi}_1^\dagger\mathbf{E} = \mathbf{Q}^\top\mathbf{B} + \boldsymbol{\Psi}_1^\dagger\boldsymbol{\Psi}_2\widehat{\mathbf{Q}}^\top\mathbf{B} + \boldsymbol{\Psi}_1^\dagger\mathbf{E}.$$

Adding more columns to $\boldsymbol{\Psi}$ (and hence $\boldsymbol{\Psi}_1$) reduces the error in the second term.

References I

Boullé, Nicolas and Alex Townsend (Jan. 2022). "Learning Elliptic Partial Differential Equations with Randomized Linear Algebra". In: *Foundations of Computational Mathematics* 23.2, pp. 709–739.

— (2024). "A mathematical guide to operator learning". In: *Numerical Analysis Meets Machine Learning*. Elsevier, pp. 83–125.

Chen, Tyler et al. (2025). "Near-optimal hierarchical matrix approximation from matrix-vector products". In: *Symposium on Discrete Algorithms (SODA)*.

Clarkson, Kenneth L. and David P. Woodruff (May 2009). "Numerical linear algebra in the streaming model". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. STOC '09. ACM.

Halikias, Diana and Alex Townsend (Sept. 2023). "Structured matrix recovery from matrix-vector products". In: *Numerical Linear Algebra with Applications* 31.1.

Halko, N. , P. G. Martinsson, and J. A. Tropp (2011). "Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions". In: *SIAM Rev.* 53.2, pp. 217–288.

Levitt, James and Per-Gunnar Martinsson (2022a). *Linear-Complexity Black-Box Randomized Compression of Rank-Structured Matrices*.

— (2022b). *Randomized Compression of Rank-Structured Matrices Accelerated with Graph Coloring*.

Lin, Lin, Jianfeng Lu, and Lexing Ying (May 2011). "Fast construction of hierarchical matrix representation from matrix–vector multiplication". In: *Journal of Computational Physics* 230.10, pp. 4071–4087.

Martinsson, Per-Gunnar (Jan. 2016). "Compressing Rank-Structured Matrices via Randomized Sampling". In: *SIAM Journal on Scientific Computing* 38.4, A1959–A1986.

References II

Nakatsukasa, Yuji (2020). "Fast and stable randomized low-rank matrix approximation". In: *ArXiv* abs/2009.11392.

Tropp, Joel A. and Robert J. Webber (2023). *Randomized algorithms for low-rank matrix approximation: Design, analysis, and applications*.

Tropp, Joel A. et al. (Jan. 2017). "Practical Sketching Algorithms for Low-Rank Matrix Approximation". In: *SIAM Journal on Matrix Analysis and Applications* 38.4, pp. 1454–1485.

Xia, Jianlin et al. (Nov. 2010). "Fast algorithms for hierarchically semiseparable matrices". In: *Numerical Linear Algebra with Applications* 17.6, pp. 953–976.