Structured Matrix Learning from Matrix-Vector Products

Christopher Musco, New York University

COLLABORATORS

Noah Amsel, Praytush Avi, Tyler Chen, Feyza Duman Keles, Diana Halikias, Chinmay Hedge, Cameron Musco, David Persson.

Related papers in SIMAX 2026, SODA 2025, and a pending submission to COLT 2026.

PROBLEM WE ARE STUDYING

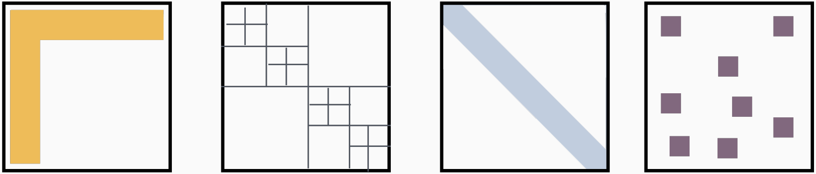
Problem: Let $\mathcal{F} \subset \mathbb{R}^{n \times n}$ be a family of $n \times n$ matrices. For tolerance parameter $\gamma > 1$, find a near-optimal approximation $\tilde{\mathsf{B}} \in \mathcal{F}$ satisfying:

$$\|\mathsf{A} - \tilde{\mathsf{B}}\|_F \leq \gamma \cdot \min_{\mathsf{B} \in \mathcal{F}} \|\mathsf{A} - \mathsf{B}\|_F.$$

Problem: Let $\mathcal{F} \subset \mathbb{R}^{n \times n}$ be a family of $n \times n$ matrices. For tolerance parameter $\gamma > 1$, find a near-optimal approximation $\tilde{B} \in \mathcal{F}$ satisfying:

$$\|A - \tilde{B}\|_F \leq \gamma \cdot \min_{B \in \mathcal{F}} \|A - B\|_F.$$

Example families:



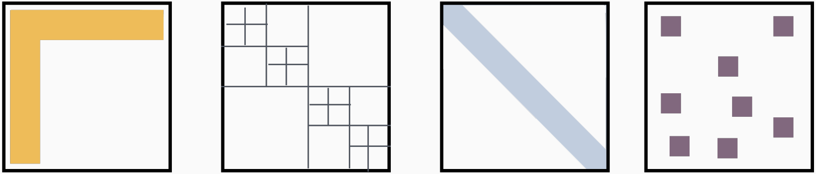
low-rank | hierarchically low-rank | diagonal | sparse

Banded, block diagonal, Toeplitz, butterly, diagonal + low-rank, sparse + low-rank, ...

Problem: Let $\mathcal{F} \subset \mathbb{R}^{n\times n}$ be a family of $n \times n$ matrices. For tolerance parameter $\gamma > 1$, find a near-optimal approximation $\tilde{\mathrm{B}} \in \mathcal{F}$ satisfying:

$$\|\mathrm{A} - \tilde{\mathrm{B}}\|_F \leq \gamma \cdot \min_{\mathrm{B}\in\mathcal{F}} \|\mathrm{A} - \mathrm{B}\|_F.$$

Example families:



low-rank | hierarchically low-rank | diagonal | sparse

Interesting choices of γ include constant, $\gamma = (1 + \epsilon)$, or even $\gamma = \log n$.

Problem: Let $\mathcal{F} \subset \mathbb{R}^{n\times n}$ be a family of $n \times n$ matrices. For tolerance parameter $\gamma > 1$, find a near-optimal approximation $\tilde{\mathbf{B}} \in \mathcal{F}$ satisfying:

$$\|\mathbf{A} - \tilde{\mathbf{B}}\|_F \leq \gamma \cdot \min_{\mathbf{B}\in\mathcal{F}} \|\mathbf{A} - \mathbf{B}\|_F.$$

We are specifically interested in the setting where $\mathbf{A}$ and $\mathbf{A}^T$ can only be accessed via black-box matrix-vector products.

I.e., return an approximation to $\mathbf{A}$ based only on

$$\mathbf{A}\mathbf{x}_1, \mathbf{A}^T\mathbf{x}_2, \mathbf{A}\mathbf{x}_3, \mathbf{A}^T\mathbf{x}_4, \ldots, \mathbf{A}\mathbf{x}_{m-1}, \mathbf{A}^T\mathbf{x}_m$$

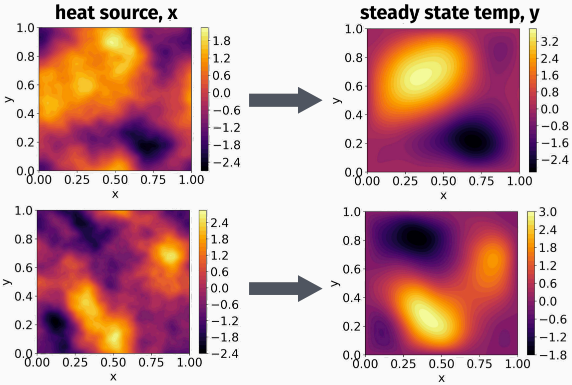
How many matrix-vector products, m, are needed to learn a near-optimal approximation from a given family $\mathcal{F}$?

- Compressed approximations of matrices that admit fast matvecs. E.g., rank-structured matrices that can be efficiently multiplied using Fast Multipole Method.
- Approximatiions of implicit matrices like Hessians, for which matvecs can be implemented via Automatic Differentiation or other techniques.
- Approximations to matrix functions like $\mathbf{A}^{-1}$. Can compute $\mathbf{A}^{-1}\mathbf{x}$ with iterative methods.
- Learning structured covariance matrices. Might receive samples of the form $\boldsymbol{\Sigma}^{-1/2}\mathbf{g}$, where $\mathbf{g}$ is standard Gaussian.

- Compressed approximations of matrices that admit fast matvecs. E.g., rank-structured matrices that can be efficiently multiplied using Fast Multipole Method.
- Approximatiions of implicit matrices like Hessians, for which matvecs can be implemented via Automatic Differentiation or other techniques.
- Approximations to matrix functions like $\mathbf{A}^{-1}$. Can compute $\mathbf{A}^{-1}\mathbf{x}$ with iterative methods.
- Learning structured covariance matrices. Might receive samples of the form $\mathbf{\Sigma}^{-1/2}\mathbf{g}$, where $\mathbf{g}$ is standard Gaussian.

Further theoretical motivation: One of the simplest interesting special cases of operator learning, a task of recent interest in scientific machine learning (SciML).

Physical processes often map a function/vector **x** to a function/vector **y**.



Goal in SciML: Learn neural network (DeepONet, Fourier Neural Operator, etc.) that can directly map inputs to outputs.

Train learned operator on input-output pairs,

$$(\mathsf{x}_1, \mathsf{y}_1), \ldots, (\mathsf{x}_m, \mathsf{y}_m),$$

obtained via simulation or physical experiments.

Matrix learning corresponds to the setting when the target operator is linear: $\mathsf{y}_i = \mathsf{A}\mathsf{x}_i$. Even in this setting, basic questions about the sample complexity of learning remain open.

OPERATOR LEARNING

Train learned operator on input-output pairs,

$$(x_1, y_1), \ldots, (x_m, y_m),$$

obtained via simulation or physical experiments.

Matrix learning corresponds to the setting when the target operator is linear: $y_i = Ax_i$. Even in this setting, basic questions about the sample complexity of learning remain open.

Learning Elliptic Partial Differential Equations with Randomized Linear Algebra

Nicolas Boullé[1] · Alex Townsend[2]

2024 SIAM Activity Group on Linear Algebra Best Paper Prize.

Train learned operator on input-output pairs,

$$(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m),$$

obtained via simulation or physical experiments.

Matrix learning corresponds to the setting when the target operator is linear: $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$. Even in this setting, basic questions about the sample complexity of learning remain open.

PNAS

BRIEF REPORT | APPLIED MATHEMATICS

Elliptic PDE learning is provably data-efficient

Nicolas Boullé[a,1], Diana Halikias[b], and Alex Townsend[b]

Problem: Let $\mathcal{F} \subset \mathbb{R}^{n\times n}$ be a family of $n \times n$ matrices. For tolerance parameter $\gamma > 1$, find $\tilde{\mathbf{B}} \in \mathcal{F}$ satisfying:

$$\|\mathbf{A} - \tilde{\mathbf{B}}\|_F \leq \gamma \cdot \min_{\mathbf{B}\in\mathcal{F}} \|\mathbf{A} - \mathbf{B}\|_F.$$

Take away from Gunnar's talk: Randomized methods like sketching are really useful for solving this problem, for structures well beyond low-rank matrices!

EXAMPLE: DIAGONAL APPROXIMATION

Let $\mathcal{F}$ be the class of diagonal matrices.

3	.1	.1	.1	.1
.1	**-2**	.1	.1	.1
.1	.1	**4**	.1	.1
.1	.1	.1	**6**	.1
.1	.1	.1	.1	**-1**

target matrix A

3				
	-2			
		4		
			6	
				-1

optimal diagonal approximation B*

0	.1	.1	.1	.1
.1	0	.1	.1	.1
.1	.1	0	.1	.1
.1	.1	.1	0	.1
.1	.1	.1	.1	0

error of optimal approximation

2.8				
	-2.1			
		4		
			6.1	
				-.9

near-optimal diagonal approximation $\tilde{B}$

.2	.1	.1	.1	.1
.1	.1	.1	.1	.1
.1	.1	0	.1	.1
.1	.1	.1	.1	.1
.1	.1	.1	.1	.1

error of near-optimal approximation

EXAMPLE: DIAGONAL APPROXIMATION

Let $\mathcal{F}$ be the class of diagonal matrices. Find $\tilde{\mathbf{B}} \in \mathcal{F}$ satisfying:

$$\|\mathbf{A} - \tilde{\mathbf{B}}\|_F \leq (1+\epsilon) \min_{\mathbf{B} \in \mathcal{F}} \|\mathbf{A} - \mathbf{B}\|_F.$$

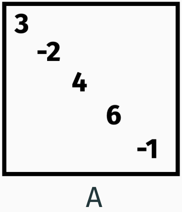
How many matvecs do I need to solve this problem?

Let $\mathcal{F}$ be the class of diagonal matrices. Find $\tilde{\mathbf{B}} \in \mathcal{F}$ satisfying:

$$\|\mathbf{A} - \tilde{\mathbf{B}}\|_F \leq (1+\epsilon) \min_{\mathbf{B}\in\mathcal{F}} \|\mathbf{A} - \mathbf{B}\|_F.$$

How many matvecs do I need to solve this problem?

Consider the case when $\mathbf{A}$ is exactly diagonal. I.e., $\min_{\mathbf{B}\in\mathcal{F}} \|\mathbf{A} - \mathbf{B}\|_F = 0$. **Hint:** You don't need randomness here.

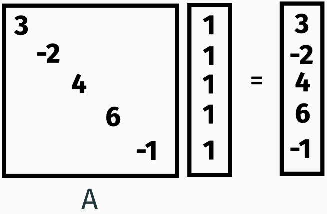


Let $\mathcal{F}$ be the class of diagonal matrices. Find $\tilde{\mathbf{B}} \in \mathcal{F}$ satisfying:

$$\|\mathbf{A} - \tilde{\mathbf{B}}\|_F \leq (1 + \epsilon) \min_{\mathbf{B} \in \mathcal{F}} \|\mathbf{A} - \mathbf{B}\|_F.$$

How many matvecs do I need to solve this problem?

Consider the case when $\mathbf{A}$ is exactly diagonal. I.e., $\min_{\mathbf{B} \in \mathcal{F}} \|\mathbf{A} - \mathbf{B}\|_F = 0$. **Hint:** You don't need randomness here.



EXAMPLE: DIAGONAL APPROXIMATION

Deterministic methods usually fail as soon as $A \notin \mathcal{F}$!

$$\begin{bmatrix} 3 & & & & \\ & -2 & & & \\ & & 4 & & \\ & & & 6 & \\ & & & & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ 4 \\ 6 \\ -1 \end{bmatrix} \qquad \begin{bmatrix} 3 & .1 & .1 & .1 & .1 \\ .1 & -2 & .1 & .1 & .1 \\ .1 & .1 & 4 & .1 & .1 \\ .1 & .1 & .1 & 6 & .1 \\ .1 & .1 & .1 & .1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3.4 \\ -1.6 \\ 4.4 \\ 6.4 \\ -.6 \end{bmatrix}$$

Goal is to ensure:

$$\|A - \tilde{B}\|_F \leq (1+\epsilon) \min_{B \in \mathcal{S}} \|A - B\|_F \approx .1 \cdot n.$$

Error of naive algorithm:

$$\lesssim \sqrt{\underbrace{.1^2 \cdot n^2}_{\text{off diag. error}} + \underbrace{n \cdot (.1 \cdot n)^2}_{\text{on diag. error}}} \approx .1 \cdot n^{1.5}.$$

Pick random sign vector $\mathbf{r} \in \{-1, 1\}^n$. Return $\mathbf{r} \circ (\mathbf{A}\mathbf{r})$ [Bekas, Kokiopoulou, Saad 2007].

$$\begin{bmatrix} 3 & & & & \\ & -2 & & & \\ & & 4 & & \\ & & & 6 & \\ & & & & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 4 \\ -6 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 3 & .1 & .1 & .1 & .1 \\ .1 & -2 & .1 & .1 & .1 \\ .1 & .1 & 4 & .1 & .1 \\ .1 & .1 & .1 & 6 & .1 \\ .1 & .1 & .1 & .1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 2.8 \\ 2 \\ 3.8 \\ -6 \\ 1 \end{bmatrix}$$

Error of randomized algorithm:

$$\sqrt{\underbrace{.1^2 \cdot n^2}_{\text{off diag. error}} + \underbrace{n \cdot (.1 \cdot \sqrt{n})^2}_{\text{on diag. error}}} \approx .14 \cdot n \leq 1.4 \cdot \|\mathbf{A} - \mathbf{B}^*\|_F.$$

Can improve error by repeating and averaging.

Theorem

Let $\mathcal{F}$ be the class of diagonal matrices. $O(1/\epsilon)$ matvecs with $\mathbf{A}$ are needed to find $\tilde{\mathbf{B}} \in \mathcal{F}$ satisfying:

$$\mathbb{E}[\|\mathbf{A} - \tilde{\mathbf{B}}\|_F] \leq (1+\epsilon) \min_{\mathbf{B} \in \mathcal{F}} \|\mathbf{A} - \mathbf{B}\|_F.$$

- Not hard to prove. See [Baston, Nakatsukasa 2022], [Dharangutte, Musco 2023], or [Amsel, Chen, Halikias, Duman Keles, Musco, Musco, 2026].
- Generalizes to $O(s/\epsilon)$ matvecs for approximation by any matrix with $\leq s$ non-zeros per row (e.g., banded or block diagonal with bandwidth s).
- This bound is tight. $\Omega(s/\epsilon)$ matvecs necessary in general.

RANDOMIZED ALGORTIHMS FOR MATRIX APPROXIMATION

Structure	# of matvecs to learn	reference
Rank k	$O(k/\epsilon^{1/3})$	Randomized SVD!
		[Bakshi et al., 2022]
Diagonal	$O(1/\epsilon)$	[Bekas et al., 2007]
		[Dharangutte, Musco 2023]
s-banded	$O(s/\epsilon)$	[Amsel et al., 2026]
s-sparse rows	$O(s/\epsilon)$	[Amsel et al., 2026]
rank-k HODLR	$O(k \log^4 n/\epsilon^3)$	[Lin,Lu,Ying, 2011]
		[Chen et al., 2025]
rank-k HSS	$O(k \log n)$	[Levitt, Martinsson, 2024]
		[Amsel et al., 2024]
rank-k butterfly	$O(k\sqrt{n})$	[Liu et al., 2021]
		[Le at al., 2026]
$\vdots$	$\vdots$	$\vdots$

Lots of gaps remain, and many natural families left unstudied!

Can keep writing papers on different matrix families... or ask:

Is there a general theory for the query complexity of structured matrix learning?

Can keep writing papers on different matrix families... or ask:

Is there a general theory for the query complexity of structured matrix learning?

In statistical learning theory, we have general tools for bounding the sample complexity of learning. **VC dimension**, **Pollard pseudodimension**, **fat-shattering dimension**, etc.

Can keep writing papers on different matrix families… or ask:

Is there a general theory for the query complexity of structured matrix learning?

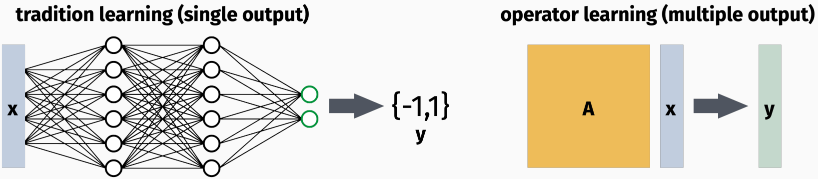
In statistical learning theory, we have general tools for bounding the sample complexity of learning. **VC dimension**, **Pollard pseudodimension**, **fat-shattering dimension**, etc.

Theorem (Informal)

Any hypothesis class $\mathcal{H}$ consisting of functions from $\mathbb{R}^n \rightarrow \{-1, 1\}$ with VC dimension C can be learned with:

$$O\left(C/\epsilon^2\right) \text{ samples.}$$

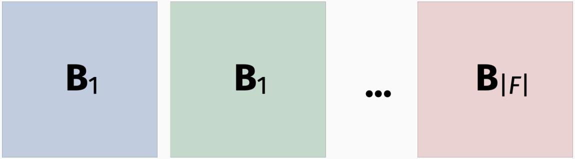
Existing tools do not directly apply to matrix learning.



You can potentially learn a lot more from a single sample in out setting than in a traditional statistical learning setting!

NATURAL FIRST STEP

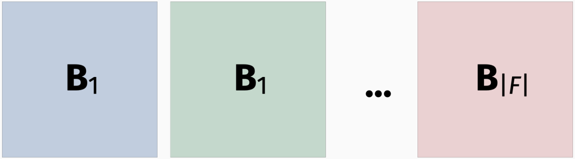
Start by considering finite-size matrix families. I.e., $|\mathcal{F}| < \infty$.



Goal is to find $\arg\min_{i \in 1,\ldots,|\mathcal{F}|} \|\mathbf{A} - \mathbf{B}_i\|_F$.

NATURAL FIRST STEP

Start by considering finite-size matrix families. I.e., $|\mathcal{F}| < \infty$.



Goal is to find $\arg\min_{i \in 1,\ldots,|\mathcal{F}|} \|\mathbf{A} - \mathbf{B}_i\|_F$.

Why is finite interesting? Most natural continuous families can be well-approximated by a finite family with size roughly

$$2^{O(\text{\# of parameters})}.$$

E.g., $2^{O(nk)}$ for rank-k matrices, $2^{O(s)}$ for s-sparse matrices, etc.

First result in learning theory: The VC-dimension of a finite hypothesis class $\mathcal{H}$ is upper bounded by $\log|\mathcal{H}|$, and the class can be learned to accuracy ϵ with:

$$O(\log|\mathcal{H}|/\epsilon^2) \text{ samples.}$$

First result in learning theory: The VC-dimension of a finite hypothesis class $\mathcal{H}$ is upper bounded by $\log|\mathcal{H}|$, and the class can be learned to accuracy ϵ with:

$$O(\log|\mathcal{H}|/\epsilon^2) \text{ samples.}$$

Claim

Let $\mathcal{F}$ be a finite matrix family. A near-optimal approximation to $\mathbf{A}$ from $\mathcal{F}$ can be learned up to accuracy $(1+\epsilon)$ with:

$$O(\log|\mathcal{F}|/\epsilon^2) \textit{ matrix-vector products.}$$

Claim

Let $\mathcal{F}$ be a finite matrix family. A near-optimal approximation to $\mathbf{A}$ from $\mathcal{F}$ can be learned up to accuracy $(1+\epsilon)$ with:

$$O(\log|\mathcal{F}|/\epsilon^2) \text{ matrix-vector products.}$$

Approach: Simply return $\arg\min_{i \in 1,\ldots,|\mathcal{F}|} \|\mathbf{A}\boldsymbol{\Pi} - \mathbf{B}_i\boldsymbol{\Pi}\|_F$, where $\boldsymbol{\Pi}$ is a random sign or Gaussian matrix with $O(\log|\mathcal{F}|/\epsilon^2)$ columns.

By standard analysis of Hutchinson's estimator, we have that with high probability, $\|\mathbf{A}\boldsymbol{\Pi} - \mathbf{B}_i\boldsymbol{\Pi}\|_F \in (1 \pm \epsilon)\|\mathbf{A} - \mathbf{B}_i\|_F$ for all i.

Theorem (Amsel, Avi, Chen, Duman Keles, Hegde, Musco, Musco, Persson, 2025)

Let $\mathcal{F}$ be a finite matrix family. An optimal approximation to A *from $\mathcal{F}$ can be learned up to accuracy $\gamma = 4$ with:*

$$\tilde{O}(\sqrt{\log|\mathcal{F}|}) \text{ matrix-vector products.}$$

I.e., find $\tilde{\mathsf{B}} \in \mathcal{F}$ satisfying $\|\mathsf{A} - \tilde{\mathsf{B}}\|_F \leq 4 \cdot \min_{\mathsf{B}\in\mathcal{F}} \|\mathsf{A} - \mathsf{B}\|_F$.

Theorem (Amsel, Avi, Chen, Duman Keles, Hegde, Musco, Musco, Persson, 2025)

Let $\mathcal{F}$ be a finite matrix family. An optimal approximation to $\mathbf{A}$ *from $\mathcal{F}$ can be learned up to accuracy $\gamma = 4$ with:*

$$\tilde{O}(\sqrt{\log |\mathcal{F}|}) \textit{ matrix-vector products.}$$

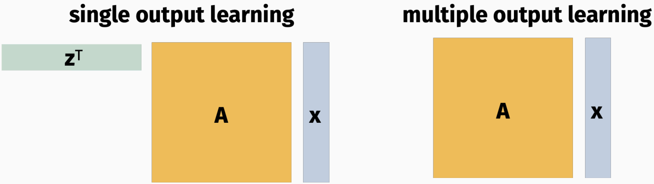
The multi-output nature of the problem allows for quadratic improvement in sample complexity!

Theorem (Amsel, Avi, Chen, Duman Keles, Hegde, Musco, Musco, Persson, 2025)

Let $\mathcal{F}$ be a finite matrix family. An optimal approximation to $\mathbf{A}$ *from $\mathcal{F}$ can be learned up to accuracy $\gamma = 4$ with:*

$$\tilde{O}(\sqrt{\log |\mathcal{F}|}) \text{ matrix-vector products.}$$

$O(\log |\mathcal{F}|)$ is optimal if we only allow vector-matrix-vector queries.



Theorem (Amsel, Avi, Chen, Duman Keles, Hegde, Musco, Musco, Persson, 2025)

Let $\mathcal{F}$ be a finite matrix family. An optimal approximation to $\mathbf{A}$ *from $\mathcal{F}$ can be learned up to accuracy* 4 *with:*

$$\tilde{O}(\sqrt{\log |\mathcal{F}|}) \text{ matrix-vector products.}$$

We can prove that the dependence on $\sqrt{\log |\mathcal{F}|}$ cannot be improved in general. It leads to tight results for some families, loose results for others:

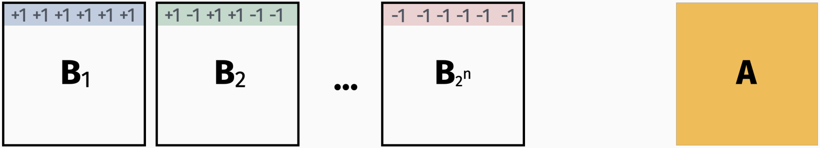
structure	size of family	query complexity
Constant rank butterfly	$2^{O(n)}$	$\tilde{O}(\sqrt{n})$
s-sparse matrices	$2^{O(s)}$	$\tilde{O}(\sqrt{s})$
Rank k matrices	$2^{O(nk)}$	$\tilde{O}(\sqrt{nk})$
$\vdots$	$\vdots$	$\vdots$

Either left <u>or</u> right queries must return a lot of information.

KEY IDEA

Either left <u>or</u> right queries must return a lot of information.

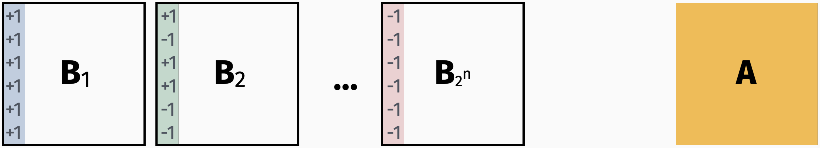
Consider approximation by the set of matrices that take ± 1 values in just their <u>first row</u>:



- Not hard to show that $\Omega(n)$ right queries of the form $\mathbf{A}\mathbf{x}$ are necessary.
- But a <u>single left query</u>, $\mathbf{A}^T\mathbf{x}$ suffices!

Either left or right queries must return a lot of information.

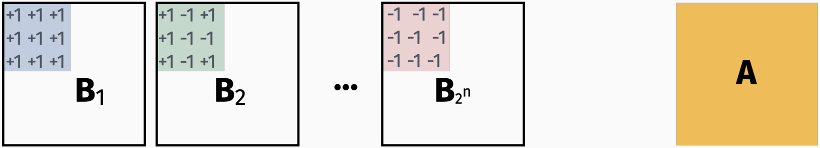
Consider approximation by the set of matrices that take ± 1 values in just their first column:



- Not hard to show that $\Omega(n)$ left queries of the form $\mathbf{A}^T\mathbf{x}$ are necessary.
- But a single right query, $\mathbf{Ax}$ suffices!

Either left <u>or</u> right queries must return a lot of information.

The hardest case is the set of matrices that take ± 1 values in just the top $\sqrt{n} \times \sqrt{n}$ block:



- Not hard to show that $\Omega(\sqrt{n})$ left queries or right queries are necessary, and $O(\sqrt{n})$ queries is of course sufficient.
- Having both doesn't help.

For experts: You can show that a permutation of this family is a <u>subset</u> of the rank-1 butterfly matrices. So Butterfly matrices also requires $\Omega(\sqrt{n})$ matrix-vector product queries to learn.

Please check out our paper *Query Efficient Structured Matrix Learning* for the general case: www.arxiv.org/pdf/2507.19290.

Fun exercise: Prove our result for the class of matrices that are ± 1 in s arbitrary locations. Assume $\mathbf{A} \in \mathcal{F}$. This family has size $\binom{n^2}{s} \cdot 2^s \approx 2^{O(s \log(n/s))}$. Prove that $O(\sqrt{s} \log n)$ matvecs suffice.

Tons of open questions:

- Obtain $(1+\epsilon)$ error instead of constant factor.
- Our method uses adaptive queries. Are they necessary?
- We have learned that "class size" does not fully characterize sample complexity: our result gives loose bounds for low-rank matrices, diagonal matrices, etc. What is the "right" complexity measure?

QUESTIONS?