Kronecker Matrix-Vector Complexity is Strange

Raphael Meyer, Haim Avron, William Swartworth, David P. Woodruff, Tyler Chen, Feyza Duman Keles

SIAM MDS 2024 – Minisymposium on

Randomized Matrix Computations for Large-scale Scientific and Machine Learning Problems

Caltech

RandNLA: Randomized Numerical Linear Algebra

RandNLA algorithms typically fit into one of a few paradigms:

- Matrix-Vector Products: Compute Ax for a few vectors $x_1, \dots, x_\ell$
- Entrywise Sampling: Compute $[A]_{i,j}$ for any i, j

Fast RandNLA methods designed by minimizing these computations

Today: Kronecker Matrix-Vector Complexity

Caltech

[Halko Martinsson Tropp SIAM Rev. '11, Simchowitz et al STOC '18, Sun et. al. ICALP '19, Braverman et. al. COLT '20, Bakshi et. al. STOC '22, Chen et. al. arxiv '24, many many many more……]

Motivation: Modeling Quantum Physics

[Feldman et. al. '22]

k is large

Noa is a Quantum Physicist studying a grid of k quantum particles, each particle "acts" in d dimensions

constant, like 2 or 8

Matrix $A \in \mathbb{R}^{d^k \times d^k}$ describes how these particles act

Want to compute Renyi Moments $tr(A^q)$ for integer q

Constraint: We can only efficiently compute Ax for <u>some</u> $x \in \mathbb{R}^{d^k}$

Can only efficiently compute Kronecker-Matrix-Vector Products!

Similar stories appear often. Linear algebraic structure of A unclear.

Kronecker Matrix-Vector Model vs. Normal Matrix-Vector

Before: $A \in \mathbb{R}^{d \times d}$. Can compute Ax for any $x \in \mathbb{R}^d$

Now: $A \in \mathbb{R}^{d^k \times d^k}$. Can compute Ax for any $x = x_1 \otimes \cdots \otimes x_k,\ x_i \in \mathbb{R}^d$

Can we still solve linear algebra problems efficiently?

poly$\left(k, d, \frac{1}{\varepsilon}\right)$?
Without strong assumptions on A?

Core Issue: d^k versus dk parameters

Caltech

[Avron et al. ‘14, Ahle et al. ‘20, Sun et al. ’21, Bujanovic and Kressner ‘21, Feldman et. al. ’22, many many more]

Part 1: Trace Estimation

Trace Estimation

Estimate $tr(A)$ from few matrix-vector products with PSD A

Find $\tilde{t}$ such that:

$$(1-\varepsilon)\,\mathrm{tr}(A) \le \tilde{t} \le (1+\varepsilon)\,\mathrm{tr}(A) \qquad \text{w.h.p.}$$

Classically, Hutchinson's Estimator uses $\ell = O(\frac{1}{\varepsilon^2})$ matvecs

$$\mathbb{E}[x^T A x] = \mathrm{tr}(A) \qquad if \qquad \mathbb{E}[xx^T] = I$$

$$H_\ell(A) = \tfrac{1}{\ell}\Sigma_{i=1}^{\ell} x^{(i)T} A x^{(i)} \qquad for \qquad x^{(i)} \sim \mathcal{D}$$

Kronecker case: We need a distribution where $x = x_1 \otimes x_2 \otimes \cdots \otimes x_k$

Caltech

[Girard '87, Hutchinson '89, Avron Toledo '11, Roosta-Khorasani Ascher '15, Cortinovis Kressne '21, Feldman et. al. '22, ...]

Kronecker-Hutchinson Estimator

Take $x = x_1 \otimes \cdots \otimes x_k$ where $x_i \sim \mathcal{D}_{small}$

Theorem:

Let $x_i \sim \mathcal{D}_{small}$ such that $Var\left[x_i^T B x_i\right] \leq C\left(tr(B)\right)^2$ for all PSD B

Then

$$Var[x^T A x] \leq (1+C)^k \left(tr(A)\right)^2$$

So $\ell = O\left(\frac{(1+C)^k}{\varepsilon^2}\right)$ samples suffice.

For $x_i \sim \mathcal{N}(0, I)$, $C = 2$ so $\ell = O(\frac{3^k}{\varepsilon^2})$.

[Ahle et. al. ‘24]

Addtl. Theorem: For $x_i \sim \mathcal{D} = \mathcal{N}(0, I)$, we know the exact variance.

Caltech

[Meyer Avron ‘24]

Kronecker-Hutchinson Estimator

Take $x = x_1 \otimes \cdots \otimes x_k$ where $x_i \sim \mathcal{D}_{small}$

For $\varepsilon = O(1)$,

$\boldsymbol{x_i} \sim \mathcal{D}_{\text{small}}$	**Worst Case**	$\boldsymbol{d} = \mathbf{2}$	$\boldsymbol{d} = \boldsymbol{\Omega}(\boldsymbol{k})$
$\mathcal{N}(0, I)$	3^k	3^k	3^k

Caltech

[Meyer Avron '24]

Kronecker-Hutchinson Estimator

Take $x = x_1 \otimes \cdots \otimes x_k$ where $x_i \sim \mathcal{D}_{small}$

For $\varepsilon = O(1)$,

$\boldsymbol{x_i} \sim \mathcal{D}_{\text{small}}$	**Worst Case**	$\boldsymbol{d = 2}$	$\boldsymbol{d = \Omega(k)}$
$\mathcal{N}(0, I)$	3^k	3^k	3^k
$\{\pm 1\}^d$	$\left(3 - \frac{2}{d}\right)^k$	2^k	3^k

Caltech

[Meyer Avron '24]

Kronecker-Hutchinson Estimator

Take $x = x_1 \otimes \cdots \otimes x_k$ where $x_i \sim \mathcal{D}_{small}$

For $\varepsilon = O(1)$,

$\boldsymbol{x_i} \sim \mathcal{D}_{\text{small}}$	**Worst Case**	$\boldsymbol{d = 2}$	$\boldsymbol{d = \Omega(k)}$
$\mathcal{N}(0, I)$	3^k	3^k	3^k
$\{\pm 1\}^d$	$\left(3 - \frac{2}{d}\right)^k$	2^k	3^k
$Unit\ Vector$	$\left(3 - \frac{6}{d+2}\right)^k$	1.5^k	3^k

Caltech

[Meyer Avron '24]

Kronecker-Hutchinson Estimator

Take $x = x_1 \otimes \cdots \otimes x_k$ where $x_i \sim \mathcal{D}_{small}$

For $\varepsilon = O(1)$,

	$\boldsymbol{x_i} \sim \mathcal{D}_{\text{small}}$	**Worst Case**	$\boldsymbol{d = 2}$	$\boldsymbol{d = \Omega(k)}$
	$\mathcal{N}(0, I)$	3^k	3^k	3^k
$\mathbb{R}^d$	$\{\pm 1\}^d$	$\left(3 - \frac{2}{d}\right)^k$	2^k	3^k
	Unit Vector	$\left(3 - \frac{6}{d+2}\right)^k$	1.5^k	3^k
$\mathbb{C}^d$	$\frac{1}{\sqrt{2}}\mathcal{N}(0, I) + \frac{i}{\sqrt{2}}\mathcal{N}(0, I)$	2^k	2^k	2^k

Caltech

[Meyer Avron '24]

Kronecker-Hutchinson Estimator

Take $x = x_1 \otimes \cdots \otimes x_k$ where $x_i \sim \mathcal{D}_{small}$

For $\varepsilon = O(1)$,

	$\boldsymbol{x_i \sim \mathcal{D}_{\text{small}}}$	**Worst Case**	$\boldsymbol{d = 2}$	$\boldsymbol{d = \Omega(k)}$
$\mathbb{R}^d$	$\mathcal{N}(0, I)$	3^k	3^k	3^k
	$\{\pm 1\}^d$	$\left(3 - \frac{2}{d}\right)^k$	2^k	3^k
	Unit Vector	$\left(3 - \frac{6}{d+2}\right)^k$	1.5^k	3^k
$\mathbb{C}^d$	$\frac{1}{\sqrt{2}}\mathcal{N}(0, I) + \frac{i}{\sqrt{2}}\mathcal{N}(0, I)$	2^k	2^k	2^k
	$\{\pm 1, \pm i\}^d$	$\left(2 - \frac{1}{d}\right)^k$	1.5^k	2^k
	$\mathbb{C}$ *Unit Vector*	$\left(2 - \frac{2}{d+1}\right)^k$	1.33^k	2^k

Caltech

[Meyer Avron '24]

Conclusions about Kron-Hutchinson

Faster than d^k Kronecker Matrix-Vector Products is possible

c^k complexity seems common

Conclusions about Kron-Hutchinson

Faster than d^k Kronecker Matrix-Vector Products is possible

c^k complexity seems common

Kronecker Model sensitive to things we used to not care about*

Distribution choices: Gaussian vs Rademacher vs Unit Vecs

Real versus Complex

Caltech

Conclusions about Kron-Hutchinson

Faster than d^k Kronecker Matrix-Vector Products is possible

c^k complexity seems common

Kronecker Model sensitive to things we used to not care about*

Distribution choices: Gaussian vs Rademacher vs Unit Vecs

Real versus Complex

Questions:

Are these sensitivities an artifact of Hutchinson?
Are they fundamental to the Kronecker Matvec Model?

Is $poly(d, k, \frac{1}{\varepsilon})$ possible?

Caltech

Part 2: Lower Bounds

Does Rademacher vs. Gaussian Matter?

Is there a natural linear algebra problem where using $x_i \in \{\pm 1\}^d$ is provably worse than using $x_i \in \mathbb{R}^d$?

Does Rademacher vs. Gaussian Matter?

Is there a natural linear algebra problem where using $x_i \in \{\pm 1\}^d$ is provably worse than using $x_i \in \mathbb{R}^d$?

Yes: Zero Testing *(Is the matrix $A = 0$?)*

Caltech

Does Rademacher vs. Gaussian Matter?

Is there a natural linear algebra problem where using $x_i \in \{\pm 1\}^d$ is provably worse than using $x_i \in \mathbb{R}^d$?

Yes: Zero Testing *(Is the matrix $A = 0$?)*

Sampling $x = x_1 \otimes \cdots \otimes x_k$ for $x_i \sim \mathcal{N}(0, I)$ works with 1 matvec

Caltech

Does Rademacher vs. Gaussian Matter?

Is there a natural linear algebra problem where using $x_i \in \{\pm 1\}^d$ is provably worse than using $x_i \in \mathbb{R}^d$?

Yes: Zero Testing *(Is the matrix $A = 0$?)*

Sampling $x = x_1 \otimes \cdots \otimes x_k$ for $x_i \sim \mathcal{N}(0, I)$ works with 1 matvec

Theorem: Any method with $x_i \in \{\pm 1\}^d$ needs $\Omega(2^k)$ matvecs

Caltech

Does Rademacher vs. Gaussian Matter?

Is there a natural linear algebra problem where using $x_i \in \{\pm 1\}^d$ is provably worse than using $x_i \in \mathbb{R}^d$?

Yes: Zero Testing *(Is the matrix $A = 0$?)*

Sampling $x = x_1 \otimes \cdots \otimes x_k$ for $x_i \sim \mathcal{N}(0, I)$ works with 1 matvec

Theorem: Any method with $x_i \in \{\pm 1\}^d$ needs $\Omega(2^k)$ matvecs

Implication: Sub-Gaussian does not matter

Does Rademacher vs. Gaussian Matter?

Is there a natural linear algebra problem where using $x_i \in \{\pm 1\}^d$ is provably worse than using $x_i \in \mathbb{R}^d$?

Yes: Zero Testing *(Is the matrix $A = 0$?)*

Sampling $x = x_1 \otimes \cdots \otimes x_k$ for $x_i \sim \mathcal{N}(0, I)$ works with 1 matvec

Theorem: Any method with $x_i \in \{\pm 1\}^d$ needs $\Omega(2^k)$ matvecs

Implication: Sub-Gaussian does not matter

$\boldsymbol{x_i \sim \mathcal{D}}$	**Worst Case**	$\boldsymbol{d = 2}$	$\boldsymbol{d = \Omega(k)}$
$\{\pm 1\}^d$	$\left(3 - \frac{2}{d}\right)^k$	2^k	3^k
$Unit\ Vector$	$\left(3 - \frac{6}{d+2}\right)^k$	1.5^k	3^k

Caltech

Is poly$\left(k, d, \frac{1}{\varepsilon}\right)$ possible?

Is there a natural linear algebra problem where using $poly(k, d, \frac{1}{\varepsilon})$ is impossible (with $x_i \in \mathbb{R}^d$)?

Yes: Planted Matrix Testing

Determine if $A = W$ or $A = W + \lambda uu^T$ from matrix-vector products

Theorem: With mild assumption, any method needs $\Omega(c^k)$ matvecs

Conclusions

Faster than d^k Kronecker Matrix-Vector Products is possible

c^k complexity seems common

Kronecker Model sensitive to things we used to not care about*

Subgaussianity does not matter

Open Questions:

Does Real vs Complex matter?

Is $poly(d, k, \frac{1}{\varepsilon})$ possible?

What assumptions on A can help us design fast algorithms?

Caltech