Krylov-aware low-rank approximation

Tyler Chen (joint work with Eric Hallman)

August 24, 2023

`chen.pw/slides`

What is a matrix function?

An $n \times n$ symmetric matrix $\mathbf{A}$ has real eigenvalues and orthonormal eigenvectors:

$$\mathbf{A} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top.$$

The matrix function $f(\mathbf{A})$ is defined as

$$f(\mathbf{A}) := \sum_{i=1}^{n} f(\lambda_i) \mathbf{u}_i \mathbf{u}_i^\top.$$

What are we doing with matrix functions?

Common matrix functions include:

- $f(x) = x^{-1}$
- $f(x) = \exp(-\beta x)$ for all β in some range
- $f(x) = \sqrt{x}$
- $f(x) = \mathrm{sign}(x)$

Goal. Compute a low-rank approximation to $f(\mathbf{A})$ and/or estimate $\mathrm{tr}(f(\mathbf{A}))$.

- We may wish to do this for several (related) functions $f(x)$

Computing with matrix functions

We can compute $f(\mathbf{A})$ via eigendecomposition of $\mathbf{A}$. However,

- this is slow: n^3 computation
- intractable n^2 storage costs
    - even if $\mathbf{A}$ is sparse, $f(\mathbf{A})$ typically is not
    - for $n = 2^{20}$, a $n \times n$ matrix of 64bit numbers requires 8.8 terabytes 🤯

Computing with matrix functions

We can compute $f(\mathbf{A})\mathbf{X}$ more cheaply. A standard approach is using Krylov Subspace Methods which produce an approximation using the information in

$$\mathcal{K}_k(\mathbf{A}, \mathbf{X}) = \operatorname{span}\{\mathbf{X}, \mathbf{A}\mathbf{X}, \ldots, \mathbf{A}^{k-1}\mathbf{X}\}.$$

The simplest approach is to just output $p(\mathbf{A})\mathbf{X}$, where $p(x)$ is some polynomial of degree $k-1$ and $p(x) \approx f(x)$ for $x \in [\lambda_{\min}, \lambda_{\max}]$.

- More powerful Lanczos-based methods more common

This essentially gives us a black-box method for approximating $f(\mathbf{A})\mathbf{X}$.

Randomized low-rank approximation

Suppose we wish to obtain a low-rank approximation to a symmetric matrix $\mathbf{B}$.

Algorithm 1 Randomized SVD (two-sided)

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form $\mathbf{K} = \mathbf{B\Omega}$                    ▷ $\ell$ matvecs with $\mathbf{B}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} = \mathbf{W}^\top \mathbf{BW}$          ▷ $\ell$ matvecs with $\mathbf{B}$
return $\mathbf{WXW}^\top$
```

The result $\mathbf{WXW}^\top$ is a rank ℓ approximation to $\mathbf{B}$ which is nearly as good as the best rank $\ell - p$ approximation.[1]

We can truncate to rank k if we desire a rank exactly k approximation.

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

Randomized low-rank approximation

Suppose we wish to obtain a low-rank approximation to a symmetric matrix $\mathbf{B}$.

Algorithm 2 Randomized SI (two-sided)

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form $\mathbf{K} = \mathbf{B}^q \mathbf{\Omega}$                ▷ $\ell q$ matvecs with $\mathbf{B}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} = \mathbf{W}^\top \mathbf{B} \mathbf{W}$       ▷ $\ell$ matvecs with $\mathbf{B}$
return $\mathbf{W}\mathbf{X}\mathbf{W}^\top$
```

The result $\mathbf{W}\mathbf{X}\mathbf{W}^\top$ is a rank ℓ approximation to $\mathbf{B}$ which is nearly as good as the best rank $\ell - p$ approximation.[1]

We can truncate to rank k if we desire a rank exactly k approximation.

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

Randomized low-rank approximation

Suppose we wish to obtain a low-rank approximation to a symmetric matrix $\mathbf{B}$.

Algorithm 3 Randomized BKI (two-sided)

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form $\mathbf{K} = [\mathbf{\Omega}, \mathbf{B\Omega}, \dots, \mathbf{B}^q\mathbf{\Omega}]$        ▷ $\ell q$ matvecs with $\mathbf{B}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} = \mathbf{W}^\top\mathbf{BW}$        ▷ $\ell(q+1)$ matvecs with $\mathbf{B}$
**return** $\mathbf{WXW}^\top$
```

The result $\mathbf{WXW}^\top$ is a rank $\ell(q+1)$ approximation to $\mathbf{B}$ which is nearly as good as the best rank $\ell - p$ approximation.[1]

We can truncate to rank k if we desire a rank exactly k approximation.

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

Matrix-vector product query model

The randomized SVD, SI, RBKI are matrix-free (access **B** only via matrix-vector products)

A natural way to analyze the runtime of such methods is by counting the number of matrix-vector products.

Pros:

- In many linear-algebra algorithms, matrix-vector products dominate the cost of computation
- We can hope to prove query complexity low-bounds to understand the hardness of linear algebra problems

Cons:

- Ignores arithmetic costs
- Matvecs with **B** may not be true core primitive

Randomized SVD for matrix functions (black-box version)

Algorithm 4 Low-rank approximation for matrix functions

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form $\mathbf{K} \approx f(\mathbf{A})\mathbf{\Omega}$ from $\mathcal{K}_s(\mathbf{A}, \mathbf{\Omega})$    ▷ $(s-1)\ell$ matvces with $\mathbf{A}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} \approx \mathbf{W}^\top f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W})$    ▷ $r\ell$ matvces with $\mathbf{A}$
return $\mathbf{W}\mathbf{X}\mathbf{W}^\top$
```

As we send $s, r \to \infty$, algorithm converges to the exact randomized SVD.

Randomized SVD for matrix functions (black-box version)

Algorithm 5 Low-rank approximation for matrix functions

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form $\mathbf{K} \approx f(\mathbf{A})\mathbf{\Omega}$ from $\mathcal{K}_s(\mathbf{A}, \mathbf{\Omega})$        ▷ $(s-1)\ell$ matvces with $\mathbf{A}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} \approx \mathbf{W}^\top f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W})$        ▷ $r\ell$ matvces with $\mathbf{A}$
**return** $\mathbf{W}\mathbf{X}\mathbf{W}^\top$
```

As we send $s, r \to \infty$, algorithm converges to the exact randomized SVD.

Observation. We can instead take: $\mathbf{K} \approx f(\mathbf{A})^q\mathbf{\Omega}$

Randomized SVD for matrix functions (black-box version)

Algorithm 6 Low-rank approximation for matrix functions

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form $\mathbf{K} \approx f(\mathbf{A})\mathbf{\Omega}$ from $\mathcal{K}_s(\mathbf{A}, \mathbf{\Omega})$        ▷ $(s-1)\ell$ matvces with $\mathbf{A}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} \approx \mathbf{W}^\top f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W})$        ▷ $r\ell$ matvces with $\mathbf{A}$
return $\mathbf{W}\mathbf{X}\mathbf{W}^\top$
```

As we send $s, r \to \infty$, algorithm converges to the exact randomized SVD.

Observation. We can instead take: $\mathbf{K} \approx f(\mathbf{A})^q\mathbf{\Omega}$ or $\mathbf{K} \approx [\mathbf{\Omega}, f(\mathbf{A})\mathbf{\Omega}, \ldots, f(\mathbf{A})^q\mathbf{\Omega}]$.

Randomized SVD for matrix functions (black-box version)

Algorithm 7 Low-rank approximation for matrix functions

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form $\mathbf{K} \approx f(\mathbf{A})\mathbf{\Omega}$ from $\mathcal{K}_s(\mathbf{A}, \mathbf{\Omega})$    ▷ $(s-1)\ell$ matvces with $\mathbf{A}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} \approx \mathbf{W}^\top f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W})$    ▷ $r\ell$ matvces with $\mathbf{A}$
**return** $\mathbf{W}\mathbf{X}\mathbf{W}^\top$
```

As we send $s, r \to \infty$, algorithm converges to the exact randomized SVD.

Observation. We can instead take: $\mathbf{K} \approx f(\mathbf{A})^q\mathbf{\Omega}$ or $\mathbf{K} \approx [\mathbf{\Omega}, f(\mathbf{A})\mathbf{\Omega}, \dots, f(\mathbf{A})^q\mathbf{\Omega}]$.

Best error if we use the whole Krylov subspace we generated: $\mathbf{K} = [\mathbf{\Omega}, \mathbf{A}\mathbf{\Omega}, \dots, \mathbf{A}^s\mathbf{\Omega}]$.

Krylov subspaces of Krylov subspaces

If $\mathbf{K}$ has more columns, approximating $\mathbf{X} \approx \mathbf{W}^\top f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W})$ is ostensibly more expensive.

[2] Meyer, Musco, and Musco 2023.

Krylov subspaces of Krylov subspaces

If $\mathbf{K}$ has more columns, approximating $\mathbf{X} \approx \mathbf{W}^\mathsf{T} f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W})$ is ostensibly more expensive.

Fact. Suppose $\mathbf{Q}_s = [\mathbf{\Omega}\ \mathbf{A\Omega}\ \cdots\ \mathbf{A}^{s-1}\mathbf{\Omega}]$. Then, $\mathcal{K}_{s+r}(\mathbf{A}, \mathbf{\Omega}) = \mathcal{K}_{r+1}(\mathbf{A}, \mathbf{Q}_s)$.

[2] Meyer, Musco, and Musco 2023.

Krylov subspaces of Krylov subspaces

If $\mathbf{K}$ has more columns, approximating $\mathbf{X} \approx \mathbf{W}^\top f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W})$ is ostensibly more expensive.

Fact. Suppose $\mathbf{Q}_s = [\boldsymbol{\Omega}\ \mathbf{A}\boldsymbol{\Omega}\ \cdots\ \mathbf{A}^{s-1}\boldsymbol{\Omega}]$. Then, $\mathcal{K}_{s+r}(\mathbf{A}, \boldsymbol{\Omega}) = \mathcal{K}_{r+1}(\mathbf{A}, \mathbf{Q}_s)$.

Proof.

$$\begin{aligned} \mathcal{K}_{r+1}(\mathbf{A}, \mathbf{Q}_s) &= \operatorname{range}\left(\left[\mathbf{Q}_s \quad \mathbf{A}\mathbf{Q}_s \quad \cdots \quad \mathbf{A}^r\mathbf{Q}_s\right]\right) \\ &= \operatorname{range}\big(\big[\ \boldsymbol{\Omega} \quad \mathbf{A}\boldsymbol{\Omega} \quad \cdots \quad \mathbf{A}^{s-1}\boldsymbol{\Omega} \\ &\qquad\qquad \mathbf{A}\boldsymbol{\Omega} \quad \mathbf{A}^2\boldsymbol{\Omega} \quad \cdots \quad \mathbf{A}^{s}\boldsymbol{\Omega} \\ &\qquad\qquad\quad \mathbf{A}^r\boldsymbol{\Omega} \quad \mathbf{A}^{r+1}\boldsymbol{\Omega} \quad \cdots \quad \mathbf{A}^{s+r-1}\boldsymbol{\Omega}\ \big]\big) \\ &= \operatorname{range}\left(\left[\boldsymbol{\Omega} \quad \mathbf{A}\boldsymbol{\Omega} \quad \cdots \quad \mathbf{A}^{s+r-1}\boldsymbol{\Omega}\right]\right) = \mathcal{K}_{s+r}(\mathbf{A}, \boldsymbol{\Omega}). \end{aligned}$$

Same observation independently used to analyze single-vector Lanczos for low-rank approximation[2]

[2] Meyer, Musco, and Musco 2023.

Krylov-aware low-rank approximation[3] (high level)

Algorithm 8 Low-rank approximation for matrix functions

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form $\mathbf{K} \approx f(\mathbf{A})\mathbf{\Omega}$ from $\mathcal{K}_s(\mathbf{A}, \mathbf{\Omega})$        ▷ $(s-1)\ell$ matvces with $\mathbf{A}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} \approx \mathbf{W}^\top f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W})$        ▷ $r\ell$ matvces with $\mathbf{A}$
return $\mathbf{WXW}^\top$
```

[3] Chen and Hallman 2023.

Krylov-aware low-rank approximation[3] (high level)

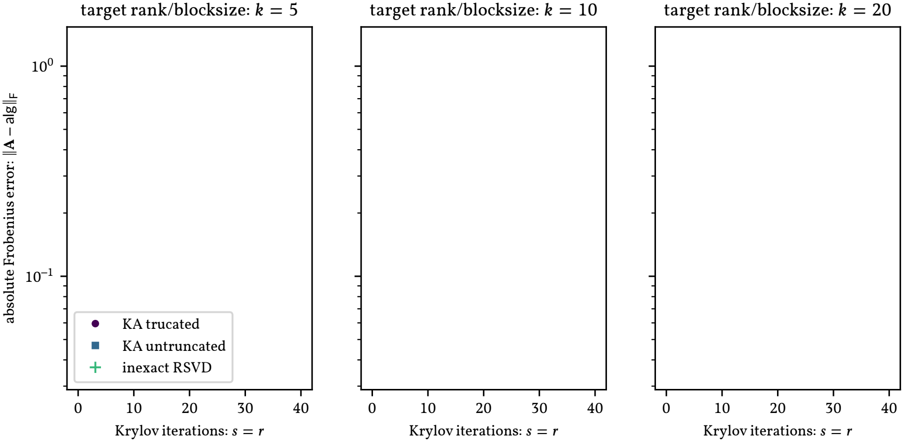
Algorithm 9 Krylov-aware low-rank approximation

```
Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
Form basis $\mathbf{K}$ for $\mathcal{K}_s(\mathbf{A}, \mathbf{\Omega})$    ▷ $(s-1)\ell$ matvces with $\mathbf{A}$
Compute $\mathbf{W} = \textsc{orth}(\mathbf{K})$
Form $\mathbf{X} \approx \mathbf{W}^\top f(\mathbf{A})\mathbf{W}$ from $\mathcal{K}_{r+1}(\mathbf{A}, \mathbf{W}) = \mathcal{K}_{s+r}(\mathbf{A}, \mathbf{\Omega})$    ▷ $r\ell$ matvces with $\mathbf{A}$
return $\mathbf{W}\mathbf{X}\mathbf{W}^\top$
```

[3] Chen and Hallman 2023.

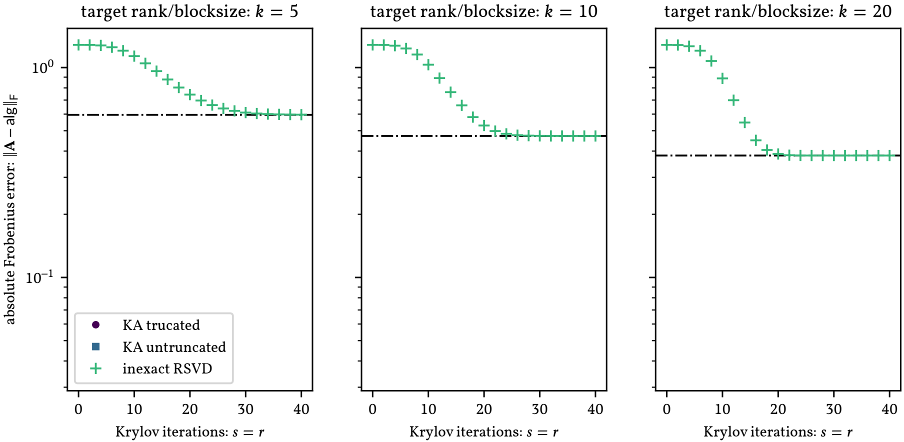
Numerical experiment: inverse function

Setup: $f(x) = 1/x$, $\mathbf{A}$ = 1000 uniform eigenvalues on $[1, 10^3]$



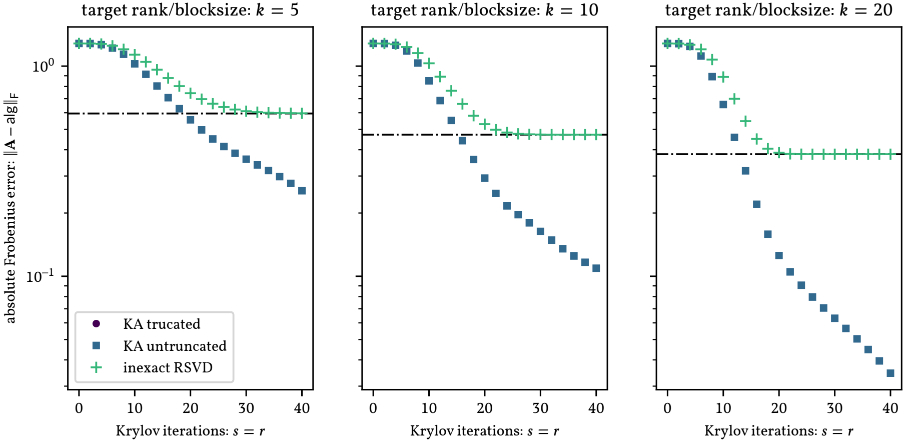
Numerical experiment: inverse function

Setup: $f(x) = 1/x$, $\mathbf{A}$ = 1000 uniform eigenvalues on $[1, 10^3]$



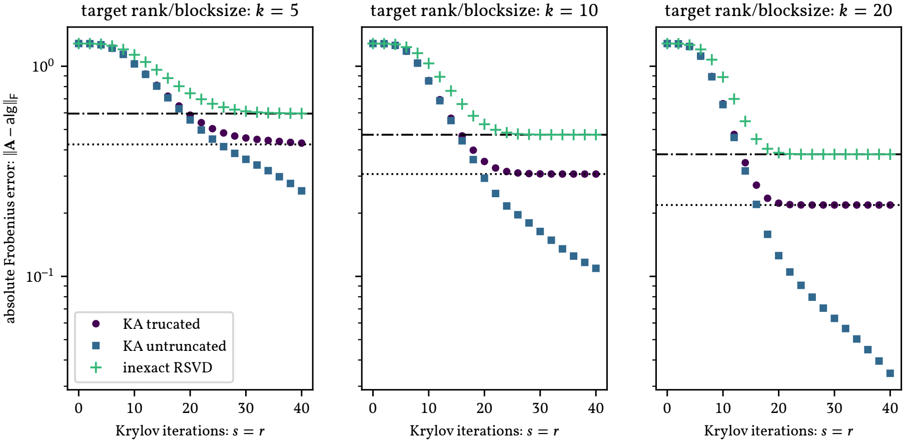
Numerical experiment: inverse function

Setup: $f(x) = 1/x$, $\mathbf{A}$ = 1000 uniform eigenvalues on $[1, 10^3]$



Numerical experiment: inverse function

Setup: $f(x) = 1/x$, $\mathbf{A}$ = 1000 uniform eigenvalues on $[1, 10^3]$



Implementation

Given $\mathbf{A}$, orthonormal $\mathbf{X}$, and $q > 0$, the block-Lanczos algorithm produces an orthonormal basis $\mathbf{Q}_q$ for $\mathcal{K}_q(\mathbf{A}, \mathbf{X})$ and a corresponding block-tridiagonal matrix $\mathbf{T}_k$ satisfying:

$$\mathbf{Q}_k = \begin{bmatrix} | & | & & | \\ \overline{\mathbf{Q}}_1 & \overline{\mathbf{Q}}_2 & \cdots & \overline{\mathbf{Q}}_q \\ | & | & & | \end{bmatrix}, \quad \mathbf{T}_k = \operatorname{tridiag}\begin{pmatrix} & \mathbf{R}_1^\top & \cdots & \mathbf{R}_{q-1}^\top & \\ \mathbf{M}_1 & \cdots & & \cdots & \mathbf{M}_q \\ & \mathbf{R}_1 & \cdots & \mathbf{R}_{q-1} & \end{pmatrix}.$$

These are related by the block-three-term recurrence

$$\mathbf{A}\mathbf{Q}_q = \mathbf{Q}_q\mathbf{T}_q + \overline{\mathbf{Q}}_q\mathbf{R}_{q+1}\mathbf{E}_q^\top,$$

Computational costs:

- q matvecs with $\mathbf{A}$
- $O(n)$ storage (or $O(nq)$ storage if $\mathbf{Q}_q$ is saved)
- $O(nq)$ arithmetic (or $O(nq^2)$ arithmetic if reorthogonalization is used)

Approximations to matrix functions

The Lanczos algorithm commonly is used to approximate quantities involving matrix functions:

$$f(\mathbf{A})\mathbf{X} \approx \mathbf{Q}_q f(\mathbf{T}_q)\mathbf{E}_1 = \mathbf{Q}_q f(\mathbf{T}_q)\mathbf{Q}_q^\top \mathbf{X} \tag{1}$$

$$\mathbf{X}^\top f(\mathbf{A})\mathbf{X} \approx \mathbf{E}_1^\top f(\mathbf{T}_q)\mathbf{E}_1 = \mathbf{X}^\top \mathbf{Q}_q f(\mathbf{T}_q)\mathbf{Q}_q^\top \mathbf{X} \tag{2}$$

If $f(x)$ is a polynomial of degree $q-1$ or $2q-1$ then (1) and (2) are respectively exact.

Note that (2) doesn't require knowledge of $\mathbf{Q}_q$!

If $\mathbf{X}$ is not orthonormal, apply $\mathbf{Q}$ factorization first.

Error guarantees

For any polynomial p of degree $\leq q-1$, $p(\mathbf{A})\mathbf{X} - \mathbf{Q}_k p(\mathbf{T}_k)\mathbf{E}_1$. Thus,

$$\begin{aligned}
\|f(\mathbf{A})\mathbf{X} - \mathbf{Q}_k f(\mathbf{T}_k)\mathbf{E}_1\| &= \|f(\mathbf{A})\mathbf{X} - p(\mathbf{A})\mathbf{X} - (\|\mathbf{Q}_k p(\mathbf{T}_k)\mathbf{E}_1 - \mathbf{Q}_k p(\mathbf{T}_k)\mathbf{E}_1)\| \\
&\leq \|f(\mathbf{A})\mathbf{X} - p(\mathbf{A})\mathbf{X}\| + \|\mathbf{Q}_k p(\mathbf{T}_k)\mathbf{E}_1 - \mathbf{Q}_k p(\mathbf{T}_k)\mathbf{E}_1\| \\
&\leq \|f(\mathbf{A}) - p(\mathbf{A})\| + \|f(\mathbf{T}_k - p(\mathbf{T})_k\| \\
&= \max_{x \in \Lambda(\mathbf{A})} |f(x) - p(x)| + \max_{x \in \Lambda(\mathbf{T}_k)} |f(x) - p(x)| \\
&\leq 2 \max_{x \in [\lambda_{\min}, \lambda_{\max}]} |f(x) - p(x)|.
\end{aligned}$$

Similar bounds for $\|\mathbf{X}^\top f(\mathbf{A})\mathbf{X} - \mathbf{E}_1^\top f(\mathbf{T}_k)\mathbf{E}_1\|$.

Remarkably, these bounds basically hold in finite precision arithmetic![4]

[4] Druskin and Knizhnerman 1992; Knizhnerman 1996.

Krylov-aware low-rank approximation[5]

Algorithm 10 Krylov-aware low-rank approximation

```
1: Sample a standard Gaussian $n \times \ell$ matrix $\mathbf{\Omega}$
2: Obtain $\mathbf{Q}_{s+r}, \mathbf{T}_{s+r} = \textsc{block-Lanczos}(\mathbf{A}, \mathbf{\Omega}, s+r)$    ▷ $(s+r)\ell$ matvces with $\mathbf{A}$
3: Set $\mathbf{W} = \mathbf{Q}_s = [\mathbf{Q}_{s+r}]_{:,1:s}$
4: Form $\mathbf{X} = [f(\mathbf{T}_{s+r})]_{1:s,1:s}$    ▷ repeat for different $f$ if you want
5: return $\mathbf{W}\mathbf{X}\mathbf{W}^\mathsf{T}$
```

In line 2:

- use full reorthogonalization for the first $s-1$ iterations
- do not save $[\mathbf{Q}_{s+r}]_{:,s+1:}$

[5] Chen and Hallman 2023.

Summary of Krylov-aware low-rank approximation

This "Krylov aware" idea is simple, but provides many benefits.

- use a (much) larger projection space "for free"
- algorithm is now agnostic to f
    - we can easily compute approximations to $\operatorname{tr}(f(\mathbf{A}))$ for multiple f without additional matrix products with $\mathbf{A}$.
- If memory or reorthogonalization costs are an issue, we can use restarting, and pick $\mathbf{Q}$ as an onb. for some subset of $\operatorname{span}\{\mathbf{\Omega}, \mathbf{A}\mathbf{\Omega}, \ldots, \mathbf{A}^{s-1}\mathbf{\Omega}\}$.

Related work on operator monotone functions[6]

- Better to sketch $\mathbf{A}$ than $\sqrt{\mathbf{A}}$

[6] Persson and Kressner 2023.

Implicit trace estimation[7]

It is well-known that if $\mathbb{E}[\mathbf{v}\mathbf{v}^\top] = \mathbf{I}$, then

$$\mathbb{E}[\mathbf{v}^\top\mathbf{B}\mathbf{v}] = \mathbb{E}[\operatorname{tr}(\mathbf{v}\mathbf{v}^\top\mathbf{B})] = \operatorname{tr}(\mathbb{E}[\mathbf{v}\mathbf{v}^\top]\mathbf{B}) = \operatorname{tr}(\mathbf{B}).$$

For many common distributions: $\mathbb{V}[\mathbf{v}^\top\mathbf{B}\mathbf{v}] \approx 2\|\mathbf{B}\|_\mathsf{F}^2$.

We can average iid copies of the estimator corresponding to iid copies $\mathbf{v}_i$ of $\mathbf{v}$. Variance is:

$$\mathbb{V}\left[\frac{1}{m}\sum_{i=1}^{m}\mathbf{v}_i^\top\mathbf{B}\mathbf{v}_i\right] = \frac{1}{m}\mathbb{V}[\mathbf{v}_1^\top\mathbf{B}\mathbf{v}_1] \approx \frac{2}{m}\|\mathbf{B}\|_\mathsf{F}^2.$$

Number of matvecs with $\mathbf{B}$ is: $2m$, so we get scaling

$$\text{accuracy} \sim (\#\ \text{matvecs})^{-2}$$

[7] Girard 1987; Hutchinson 1989; Skilling 1989.

Variance reduction

If we know $\widehat{\mathbf{B}} \approx \mathbf{B}$, we can use the variance reduced estimator:[8]

$$\operatorname{tr}(\mathbf{B}) = \operatorname{tr}(\widehat{\mathbf{B}}) + \operatorname{tr}(\mathbf{B} - \widehat{\mathbf{B}}) \approx \operatorname{tr}(\widehat{\mathbf{B}}) + \frac{1}{m}\sum_{i=1}^{m} \mathbf{v}_i^\mathsf{T}(\mathbf{B} - \widehat{\mathbf{B}})\mathbf{v}_i.$$

Variance is:

$$\mathbb{V}\left[\operatorname{tr}(\widehat{\mathbf{B}}) + \frac{1}{m}\sum_{i=1}^{m} \mathbf{v}_i^\mathsf{T}(\mathbf{B} - \widehat{\mathbf{B}})\mathbf{v}_i\right] = \frac{1}{m}\mathbb{V}[\mathbf{v}_1^\mathsf{T}(\mathbf{B} - \widehat{\mathbf{B}})\mathbf{v}_1] \approx \frac{2}{m}\|\mathbf{B} - \widehat{\mathbf{B}}\|_\mathsf{F}^2.$$

Take $\widehat{\mathbf{B}}$ as rank b approximation $\widehat{\mathbf{B}} = \mathbf{Q}(\mathbf{Q}^\mathsf{T}\mathbf{B}\mathbf{Q})\mathbf{Q}^\mathsf{T}$ obtained by sketching with a b-column random matrix. Number of matvecs with $\mathbf{B}$ is: $2b + m$, and if we set $b = m$, can get scaling[9]

$$\text{accuracy} \sim (\#\text{ matvecs})^{-1}$$

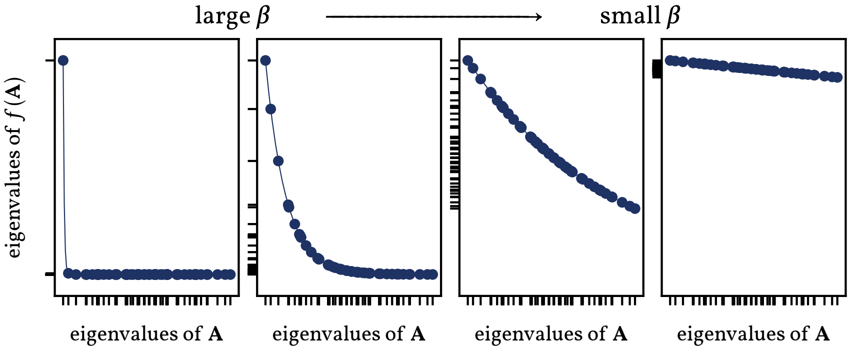
[8] Girard 1987; Weiße, Wellein, Alvermann, and Fehske 2006.

[9] Meyer, Musco, Musco, and Woodruff 2021.

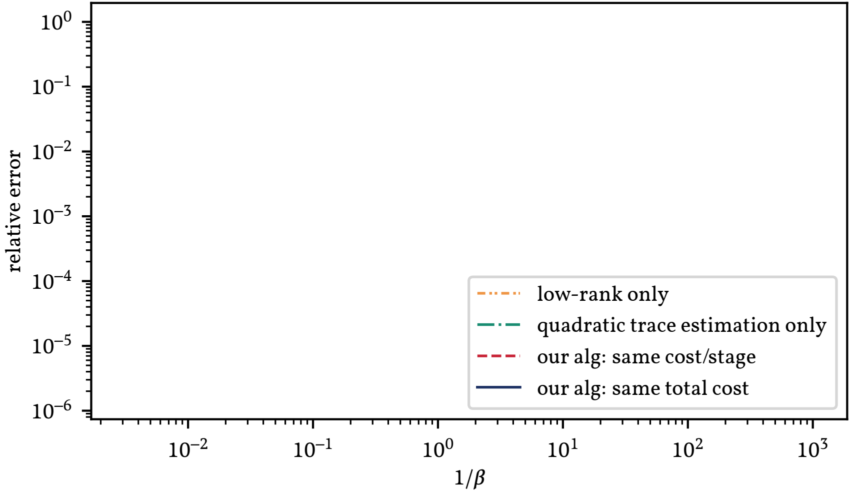
Example: equilibrium thermodynamics of quantum spin systems

In quantum physics, we often wish to compute $\operatorname{tr}(f(\mathbf{A})) = \operatorname{tr}(\exp(-\beta\mathbf{A}))$ for all $\beta > 0$.

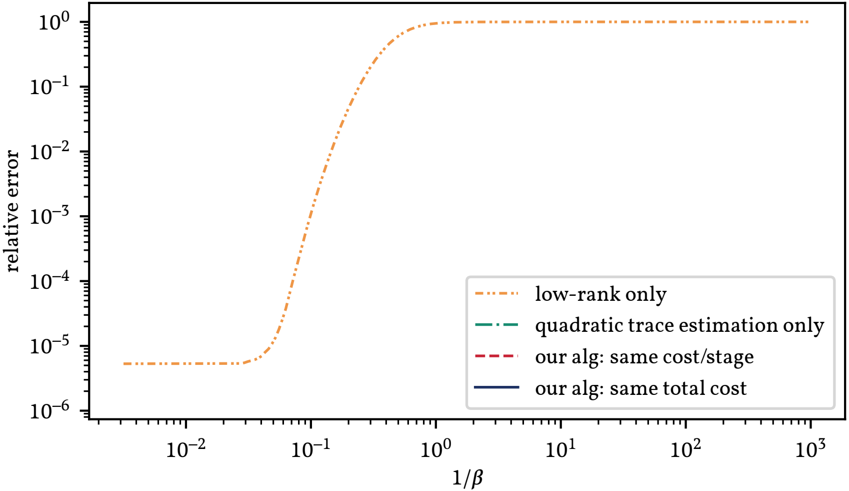
- if $\beta = \infty$ (zero temperature), then we only need ground state(s)
- if $\beta = 0$ (high temperature), then quadratic trace estimation works very well
- for intermediate beta, we might expect low-rank approaches to work well



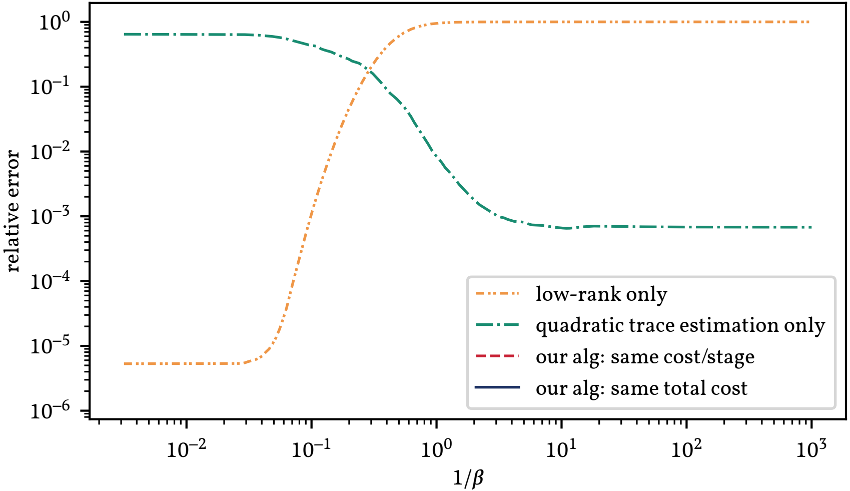
Example: quantum spin systems; $\mathrm{tr}(\exp(-\beta \mathbf{A}))$



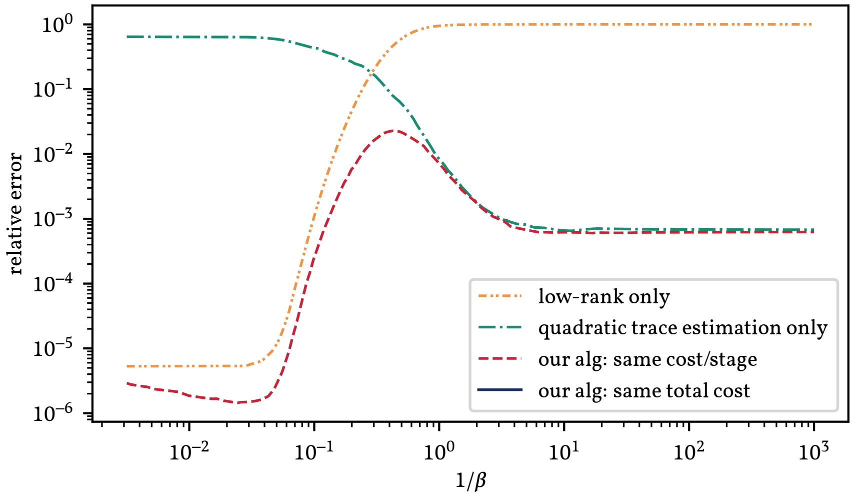
Example: quantum spin systems; $\mathrm{tr}(\exp(-\beta \mathbf{A}))$



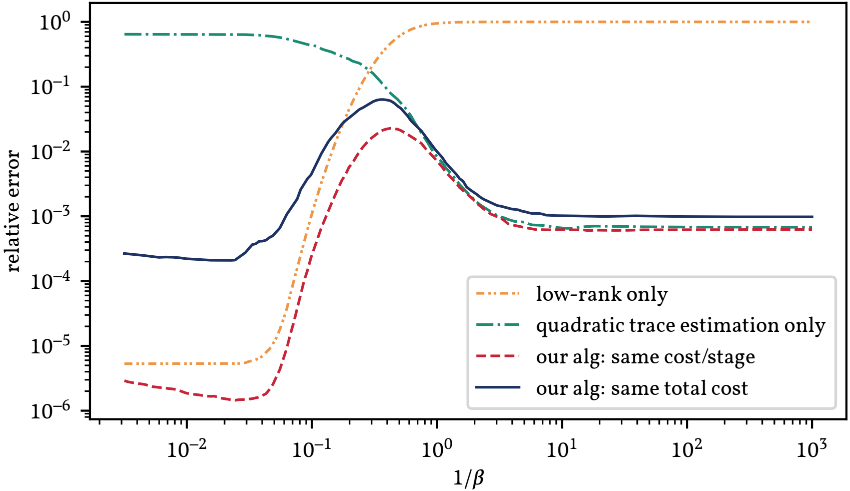
Example: quantum spin systems; $\mathrm{tr}(\exp(-\beta \mathbf{A}))$



Example: quantum spin systems; $\mathrm{tr}(\exp(-\beta\mathbf{A}))$



Example: quantum spin systems; $\mathrm{tr}(\exp(-\beta \mathbf{A}))$



Variants

We also have a number of modifications to make this idea more practical:

- Using the information in the space $\operatorname{span}\{\mathbf{\Omega}, \mathbf{A}\mathbf{\Omega}, \dots, \mathbf{A}^{q+n}\mathbf{\Omega}\}$ we can approximate

  $$\|(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathsf{T}} f(\mathbf{A})(\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathsf{T}})\|$$

  in order to determine a good value of q; see also[10]

[10] Persson, Cortinovis, and Kressner 2022.

Future work

- $\mathrm{tr}(\exp(-\beta(\mathbf{A} + h\mathbf{B})))$ for all $\beta > 0$, $h \in [-h_0, h_0]$.
- generalize low-rank algorithms to partial traces
- better understanding of stability
- lower bounds in matrix-vector query models

References I

Chen, Tyler and Eric Hallman (Aug. 2023). “Krylov-Aware Stochastic Trace Estimation”. In: *SIAM Journal on Matrix Analysis and Applications* 44.3, pp. 1218–1244.

Druskin, Vladimir and Leonid Knizhnerman (July 1992). “Error Bounds in the Simple Lanczos Procedure for Computing Functions of Symmetric Matrices and Eigenvalues”. In: *Comput. Math. Math. Phys.* 31.7, pp. 20–30.

Girard, Didier (1987). *Un algorithme simple et rapide pour la validation croisée généralisée sur des problèmes de grande taille*.

Halko, Nathan, Per-Gunnar Martinsson, and Joel A Tropp (2011). “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions”. In: *SIAM review* 53.2, pp. 217–288.

Hutchinson, M.F. (Jan. 1989). “A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines”. In: *Communications in Statistics - Simulation and Computation* 18.3, pp. 1059–1076.

Knizhnerman, L. A. (Jan. 1996). “The Simple Lanczos Procedure: Estimates of the Error of the Gauss Quadrature Formula and Their Applications”. In: *Comput. Math. Math. Phys.* 36.11, pp. 1481–1492.

Meyer, Raphael A., Cameron Musco, and Christopher Musco (2023). *On the Unreasonable Effectiveness of Single Vector Krylov Methods for Low-Rank Approximation*.

Meyer, Raphael A. et al. (Jan. 2021). “Hutch++: Optimal Stochastic Trace Estimation”. In: *Symposium on Simplicity in Algorithms (SOSA)*. Society for Industrial and Applied Mathematics, pp. 142–155.

Persson, David, Alice Cortinovis, and Daniel Kressner (July 2022). “Improved Variants of the Hutch++ Algorithm for Trace Estimation”. In: *SIAM Journal on Matrix Analysis and Applications* 43.3, pp. 1162–1185.

References II

Persson, David and Daniel Kressner (June 2023). “Randomized Low-Rank Approximation of Monotone Matrix Functions”. In: *SIAM Journal on Matrix Analysis and Applications* 44.2, pp. 894–918.

Skilling, John (1989). “The Eigenvalues of Mega-dimensional Matrices”. In: *Maximum Entropy and Bayesian Methods*. Springer Netherlands, pp. 455–466.

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

Weiße, Alexander et al. (Mar. 2006). “The kernel polynomial method”. In: *Reviews of Modern Physics* 78.1, pp. 275–306.