Krylov Subspace Methods for Matrix Function Trace Approximation

Tyler Chen

August 29, 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 $f(\mathbf{A})$, induced by $f : \mathbb{R} \to \mathbb{R}$ and $\mathbf{A}$, is defined as

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

In this talk, think of the dimension n as big! E.g. $n = 10^6$ or $n = 10^{10}$, etc.

What do we want?

Often, we don't need $f(\mathbf{A})$ itself. In this talk we will discuss:

$$f(\mathbf{A})\mathbf{v}, \qquad \mathbf{v}^\top f(\mathbf{A})\mathbf{v}, \qquad \operatorname{tr}(f(\mathbf{A})) = \sum_{i=0}^{n-1} f(\lambda_i)$$

What do we want?

Often, we don't need $f(\mathbf{A})$ itself. In this talk we will discuss:

$$f(\mathbf{A})\mathbf{v}, \qquad \mathbf{v}^{\top} f(\mathbf{A})\mathbf{v}, \qquad \operatorname{tr}(f(\mathbf{A})) = \sum_{i=0}^{n-1} f(\lambda_i)$$

Example. If $f(x) = x^{-1}$, then $f(\mathbf{A}) = \mathbf{A}^{-1}$ and $f(\mathbf{A})\mathbf{v} = \mathbf{A}^{-1}\mathbf{v}$ is the solution to the linear system $\mathbf{A}\mathbf{x} = \mathbf{v}$.

- More computationally efficient to compute an approximation to the solution $\mathbf{A}^{-1}\mathbf{v}$ rather than computing $\mathbf{A}^{-1}$ and then multiplying with $\mathbf{v}$.
  - Even if $\mathbf{A}$ is sparse, $f(\mathbf{A})$ is typically dense. Storing a $n \times n$ dense matrix might be intractable.
  - $n = 2^{20} \approx 1\text{M} \implies n \times n$ dense matrix requires 8.8 terrabytes of storage

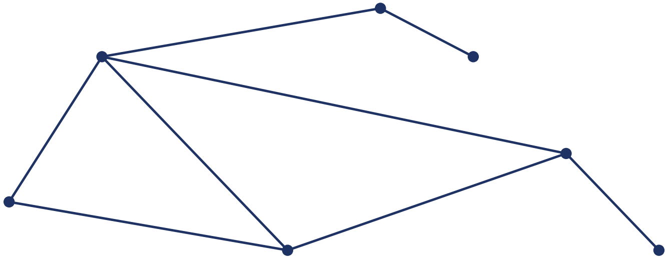
Applications

Applications in many fields: physics, chemistry, biology, statistics, high performance computing, machine learning, etc.

Common functions: inverse, exponential, square root, sign function.

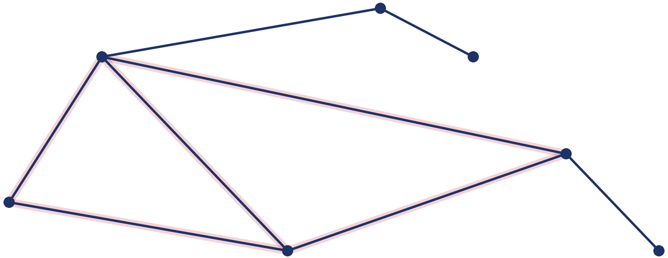
Example application: network science

Let G be a graph (nodes and edges). How many triangles are there?



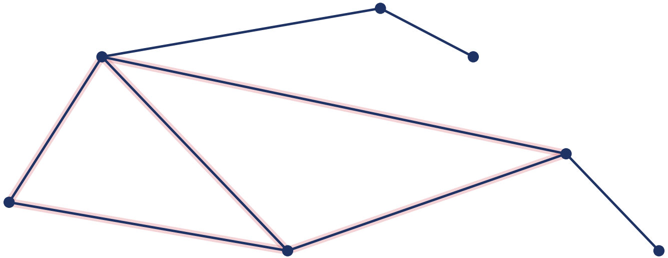
Example application: network science

Let G be a graph (nodes and edges). How many triangles are there?



Example application: network science

Let G be a graph (nodes and edges). How many triangles are there?

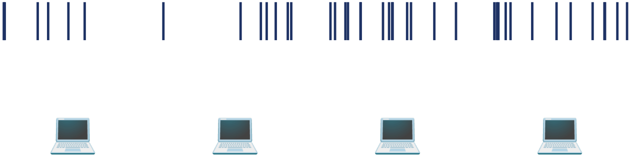


Fact. If $\mathbf{A}$ is the adjacency matrix for G, then

$$\#\text{ of triangles in } G = \frac{\operatorname{tr}(\mathbf{A}^3)}{6}.$$

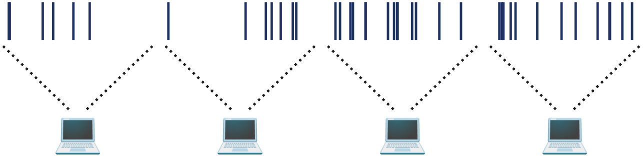
Example application: high performance computing

State of the art parallel eigensolvers such as FEAST and EVSL work by splitting the spectrum of **A** into pieces, which can each be solved on different machines in parallel.



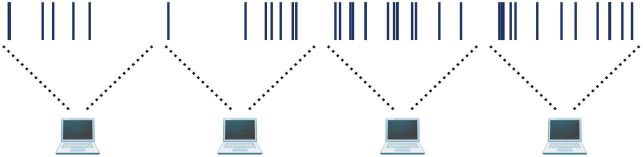
Example application: high performance computing

State of the art parallel eigensolvers such as FEAST and EVSL work by splitting the spectrum of $\mathbf{A}$ into pieces, which can each be solved on different machines in parallel.



Example application: high performance computing

State of the art parallel eigensolvers such as FEAST and EVSL work by splitting the spectrum of $\mathbf{A}$ into pieces, which can each be solved on different machines in parallel.

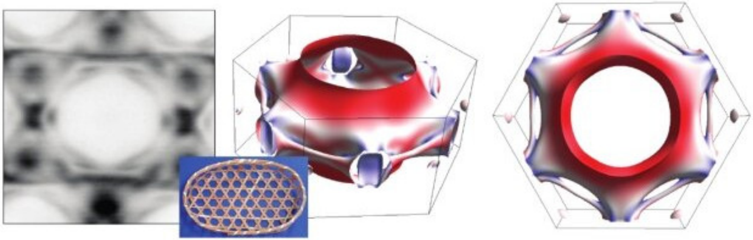


Let $\mathbb{1}[a \leq x \leq b] = 1$ if $x \in [a, b]$ and 0 otherwise. Then

$$\#\text{ of eigenvalues in } [a, b] = \operatorname{tr}(\mathbb{1}[a \leq \mathbf{A} \leq b]).$$

Example application: quantum thermodynamics

Let **A** be the Hamiltonian of a quantum system.



If the system is held in thermal equilibrium at inverse temperature $\beta = k_B/T$, then thermodynamic observables such as the specific heat, magnetization, heat capacity, etc. can be obtained from the partition function:

$$Z(\beta) = \operatorname{tr}(\exp(-\beta \mathbf{A})).$$

[0] https://phys.org/news/2023-06-quantum-materials-electron.html

Matrix polynomials

Given a scalar polynomial $p(x) = c_0 + c_1 x + \cdots + c_k x^k$, the matrix polynomial is

$$p(\mathbf{A}) = c_0 \mathbf{I} + c_1 \mathbf{A} + \cdots + c_k \mathbf{A}^k.$$

[1] Can compute $\mathbf{v}^\top p(\mathbf{A}) \mathbf{v}$ in a similar way. Symmetry allows us to double the degree.

Matrix polynomials

Given a scalar polynomial $p(x) = c_0 + c_1 x + \cdots + c_k x^k$, the matrix polynomial is

$$p(\mathbf{A}) = c_0 \mathbf{I} + c_1 \mathbf{A} + \cdots + c_k \mathbf{A}^k.$$

We can obtain $p(\mathbf{A})\mathbf{v}$ using with k matrix-vector products by computing[1]

$$\mathbf{v}, \mathbf{A}\mathbf{v}, \ldots, \mathbf{A}^k \mathbf{v}$$

and then taking a linear combination of the above vectors.

This is called the Krylov subspace:

$$\mathcal{K}_{k+1}(\mathbf{A}, \mathbf{v}) = \operatorname{span}\{\mathbf{v}, \mathbf{A}\mathbf{v}, \ldots, \mathbf{A}^k \mathbf{v}\} = \{p(\mathbf{A})\mathbf{v} : \deg(p) \le k\}.$$

[1] Can compute $\mathbf{v}^\top p(\mathbf{A})\mathbf{v}$ in a similar way. Symmetry allows us to double the degree.

Approximation with polynomials

Let p be a degree s polynomial approximation to f. Then,

$$\|f(\mathbf{A})\mathbf{v} - p(\mathbf{A})\mathbf{v}\|/\|\mathbf{v}\| \leq \|f(\mathbf{A}) - p(\mathbf{A})\|_2 = \|f - p\|_\Lambda.$$

$$|\mathbf{v}^\mathsf{T} f(\mathbf{A})\mathbf{v} - \mathbf{v}^\mathsf{T} p(\mathbf{A})\mathbf{v}|/\|\mathbf{v}\|_2^2 \leq \|f(\mathbf{A}) - p(\mathbf{A})\|_2 = \|f - p\|_\Lambda.$$

Error is determined at the eigenvalues of $\mathbf{A}$.

Approximation with polynomials

Let p be a degree s polynomial approximation to f. Then,

$$\|f(\mathbf{A})\mathbf{v} - p(\mathbf{A})\mathbf{v}\| / \|\mathbf{v}\| \leq \|f(\mathbf{A}) - p(\mathbf{A})\|_2 = \|f - p\|_\Lambda.$$

$$|\mathbf{v}^\mathsf{T} f(\mathbf{A})\mathbf{v} - \mathbf{v}^\mathsf{T} p(\mathbf{A})\mathbf{v}| / \|\mathbf{v}\|_2^2 \leq \|f(\mathbf{A}) - p(\mathbf{A})\|_2 = \|f - p\|_\Lambda.$$

Error is determined at the eigenvalues of $\mathbf{A}$.

However, we can reduce to a more classical setting:

$$\|f - p\|_\Lambda := \max_{\lambda \in \Lambda} |f(\lambda) - p(\lambda)| \leq \max_{\lambda \in \mathcal{I}} |f(\lambda) - p(\lambda)| =: \|f - p\|_{\mathcal{I}},$$

where $\mathcal{I} = [\lambda_{\min}, \lambda_{\max}]$.

Matrix-function trace approximation

The trace of a symmetric matrix $\mathbf{B}$ is the sum of the diagonal entries (equivalently, the sum of the eigenvalues)

How can we approximate $\operatorname{tr}(f(\mathbf{A}))$, given that we know $\mathbf{A}$ but not $f(\mathbf{A})$?

If we know $f(\mathbf{A})$, this task is trivial! But typically, we can't write down $f(\mathbf{A})$.

The matrix-vector query model

Suppose we have a black-box which, given a vector $\mathbf{v}$, outputs the vector $\mathbf{Bv}$.

- here $\mathbf{B}$ is some fixed matrix; e.g. $\mathbf{B} = f(\mathbf{A})$

How many times to we need to call this black box to perform basic linear algebra tasks? Some simple tasks include:

- Compute the trace of $\mathbf{B}$
- Estimate the Frobenius norm of $\mathbf{B}$
- Write down all of the entries of $\mathbf{B}$

A simple algorithm for trace estimation

Consider the matrix $\mathbf{B}$:

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\ b_{31} & b_{32} & b_{33} & \cdots & b_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & b_{n3} & \cdots & b_{nn} \end{bmatrix}$$

How can we obtain $\mathrm{tr}(\mathbf{B}) = b_{11} + b_{22} + b_{33} + \cdots + b_{nn}$ using only matrix-vector products with $\mathbf{B}$?

A simple algorithm for trace estimation

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\ b_{31} & b_{32} & b_{33} & \cdots & b_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & b_{n3} & \cdots & b_{nn} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

A simple algorithm for trace estimation

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\ b_{31} & b_{32} & b_{33} & \cdots & b_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & b_{n3} & \cdots & b_{nn} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} b_{12} \\ b_{22} \\ b_{32} \\ \vdots \\ b_{n2} \end{bmatrix}.$$

A simple algorithm for trace estimation

How can we obtain $\mathrm{tr}(\mathbf{B}) = b_{11} + b_{22} + b_{33} + \cdots + b_{nn}$ using only matrix-vector products with $\mathbf{B}$?

Multiply $\mathbf{B}$ with each of the standard basis vectors $\mathbf{e}_i = [0, 0, 1, 0, \dots, 0]^\top$, and read off the i-th entry of each result.

[2] see also Halikias and Townsend 2023

A simple algorithm for trace estimation

How can we obtain $\mathrm{tr}(\mathbf{B}) = b_{11} + b_{22} + b_{33} + \cdots + b_{nn}$ using only matrix-vector products with $\mathbf{B}$?

Multiply $\mathbf{B}$ with each of the standard basis vectors $\mathbf{e}_i = [0, 0, 1, 0, \dots, 0]^\mathsf{T}$, and read off the i-th entry of each result.

In fact, we can learn $\mathbf{B}$ completely using n matrix vector products.[2]

[2] see also Halikias and Townsend 2023

Can we do better?

Suppose we are willing to tolerate some error ϵ (e.g. $\epsilon = 10^{-3}$).

Can we approximate $\mathrm{tr}(\mathbf{B})$ with $\ll n$ matrix-vector product queries?

Can we do better?

Suppose we are willing to tolerate some error ϵ (e.g. $\epsilon = 10^{-3}$).

Can we approximate $\operatorname{tr}(\mathbf{B})$ with $\ll n$ matrix-vector product queries?

Yes!!! We can use randomized algorithms:

- deterministic: slow exact solution on all inputs
- randomized: fast approximate solution on most inputs

A simple randomized algorithm[3]

Suppose $\mathbf{v}$ is a length n vector where each entry v_i of $\mathbf{v}$ is an independent standard normal random variable.

$$\mathbb{E}[v_i] = \quad , \qquad \mathbb{E}[v_i v_j] =$$

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

A simple randomized algorithm[3]

Suppose $\mathbf{v}$ is a length n vector where each entry v_i of $\mathbf{v}$ is an independent standard normal random variable.

$$\mathbb{E}[v_i] = 0, \qquad \mathbb{E}[v_i v_j] =$$

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

A simple randomized algorithm[3]

Suppose $\mathbf{v}$ is a length n vector where each entry v_i of $\mathbf{v}$ is an independent standard normal random variable.

$$\mathbb{E}[v_i] = 0, \qquad \mathbb{E}[v_i v_j] = \begin{cases} & i = j \\ & i \neq j \end{cases}$$

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

A simple randomized algorithm[3]

Suppose $\mathbf{v}$ is a length n vector where each entry v_i of $\mathbf{v}$ is an independent standard normal random variable.

$$\mathbb{E}[v_i] = 0, \qquad \mathbb{E}[v_i v_j] = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

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

A simple randomized algorithm[3]

Suppose $\mathbf{v}$ is a length n vector where each entry v_i of $\mathbf{v}$ is an independent standard normal random variable.

$$\mathbb{E}[v_i] = 0, \qquad \mathbb{E}[v_i v_j] = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

In matrix form

$$\mathbb{E}[\mathbf{v}] = \mathbf{0}, \qquad \mathbb{E}[\mathbf{v}\mathbf{v}^\top] = \mathbf{I}.$$

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

A simple randomized algorithm[3]

Suppose $\mathbf{v}$ is a length n vector where each entry v_i of $\mathbf{v}$ is an independent standard normal random variable.

$$\mathbb{E}[v_i] = 0, \qquad \mathbb{E}[v_i v_j] = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

In matrix form

$$\mathbb{E}[\mathbf{v}] = \mathbf{0}, \qquad \mathbb{E}[\mathbf{v}\mathbf{v}^\top] = \mathbf{I}.$$

Recall that $\operatorname{tr}(\mathbf{XY}) = \operatorname{tr}(\mathbf{YX})$ and that the trace is linear. What is

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

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

A simple randomized algorithm[3]

Suppose $\mathbf{v}$ is a length n vector where each entry v_i of $\mathbf{v}$ is an independent standard normal random variable.

$$\mathbb{E}[v_i] = 0, \qquad \mathbb{E}[v_i v_j] = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

In matrix form

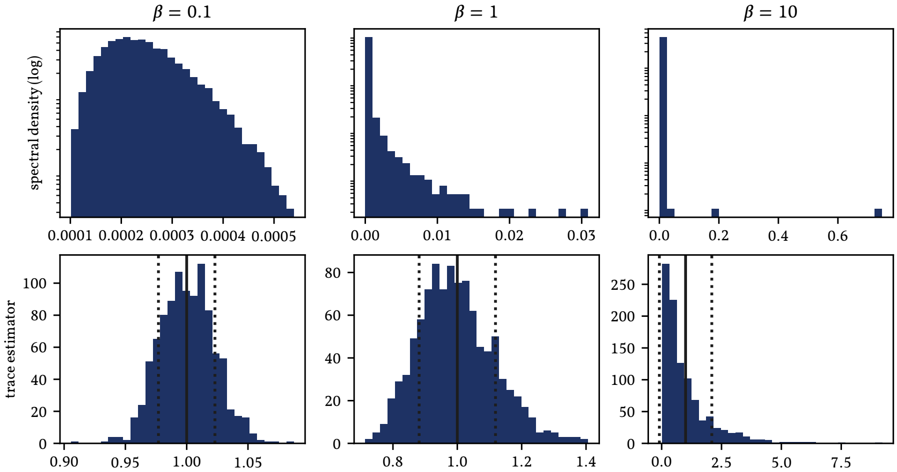
$$\mathbb{E}[\mathbf{v}] = \mathbf{0}, \qquad \mathbb{E}[\mathbf{v}\mathbf{v}^\top] = \mathbf{I}.$$

Recall that $\operatorname{tr}(\mathbf{X}\mathbf{Y}) = \operatorname{tr}(\mathbf{Y}\mathbf{X})$ and that the trace is linear. What is

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

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

Example: $f(x) = \exp(-\beta \mathbf{H})$, $f(\mathbf{A})$ scaled to unit trace



What about the variance?

We see $\mathbf{v}^\top \mathbf{B} \mathbf{v}$ is an unbiased estimator for $\mathbf{B}$. What is the variance?

What about the variance?

We see $\mathbf{v}^\top \mathbf{B}\mathbf{v}$ is an unbiased estimator for $\mathbf{B}$. What is the variance?

This is elementary but is super tedious, so let's assume (actually wlog) that $\mathbf{B}$ is diagonal. Then,

$$\mathbb{V}[\mathbf{v}^\top \mathbf{B}\mathbf{v}] = \mathbb{V}\left[\sum_{i=1}^{n} v_i^2 b_{ii}\right] = \sum_{i=1}^{n} \mathbb{V}[v_i^2 b_{ii}] = \sum_{i=1}^{n} b_{ii}^2 \mathbb{V}[v_i^2] = \sum_{i=1}^{n} 2b_{ii}^2 = 2\|\mathbf{B}\|_{\mathsf{F}}^2.$$

What about the variance?

We see $\mathbf{v}^\mathsf{T}\mathbf{B}\mathbf{v}$ is an unbiased estimator for $\mathbf{B}$. What is the variance?

This is elementary but is super tedious, so let's assume (actually wlog) that $\mathbf{B}$ is diagonal. Then,

$$\mathbb{V}[\mathbf{v}^\mathsf{T}\mathbf{B}\mathbf{v}] = \mathbb{V}\left[\sum_{i=1}^{n} v_i^2 b_{ii}\right] = \sum_{i=1}^{n} \mathbb{V}[v_i^2 b_{ii}] = \sum_{i=1}^{n} b_{ii}^2 \mathbb{V}[v_i^2] = \sum_{i=1}^{n} 2b_{ii}^2 = 2\|\mathbf{B}\|_\mathsf{F}^2.$$

So, if $\mathbf{v}_1, \dots, \mathbf{v}_m$ are independent and identically distributed copies of $\mathbf{v}$, then

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

In other words, to get accuracy ϵ, we need $m \approx \|\mathbf{B}\|_\mathsf{F}/\epsilon^2$ matrix-vector queries.

The rest of this talk

Stochastic trace estimation appeared around 1990[4], although similar ideas are older[5]

In the remainder of this talk, we will discuss developments based on stochastic trace estimation:

1. Spectral densities and spectral sums
2. Partial traces and variance reduction

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

[5] Alben, Blume, Krakauer, and Schwartz 1975.

Spectral densities and spectral sums

Define the cumulative empirical spectral measure (CSEM):

$$\Phi(x) = \sum_{i=1}^{n} \frac{1}{n} \mathbb{1}[\lambda_i \leq x], \qquad \frac{\mathrm{d}\Phi(x)}{\mathrm{d}x} = \sum_{i=1}^{n} \frac{1}{n} \delta(x - \lambda_i).$$

Note that we can write the spectral sum

$$\operatorname{tr}(f(\mathbf{A})) = \sum_{i=1}^{n} f(\lambda_i) = n \int f(x) \mathrm{d}\Phi(x).$$

So let's focus on the CESM $\Phi(x)$.

Approximating the CESM by moments

We can't compute Φ efficiently (why?), but maybe can we approximate Φ?

For the moment, let's suppose we know the moments

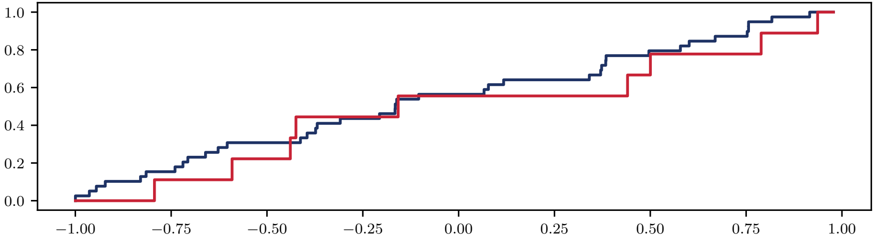
$$\int x^m \mathrm{d}\Phi(x) = n^{-1}\,\mathrm{tr}(p(\mathbf{A})), \qquad m = 0, 1, \dots, k.$$

We can obtain a distribution which has the same moments as Φ, and hope that it is near to Φ.

Measuring the similarity of distributions

The Wasserstein distance measures the similarity between distributions:

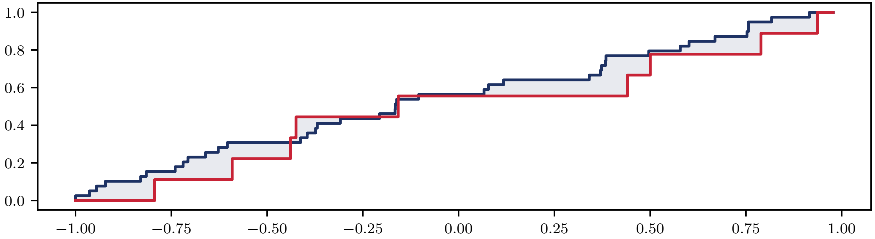
$$d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2) = \int |\Upsilon_1(x) - \Upsilon_2(x)| \mathrm{d}x.$$



Measuring the similarity of distributions

The Wasserstein distance measures the similarity between distributions:

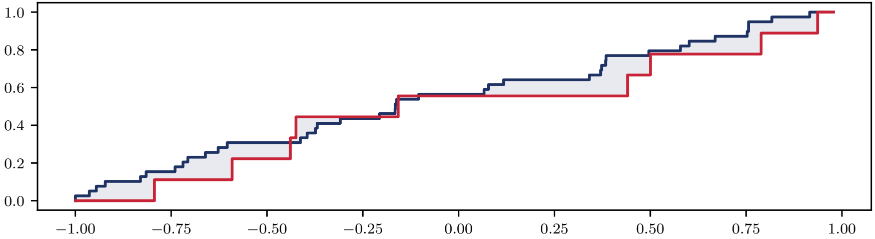
$$d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2) = \int |\Upsilon_1(x) - \Upsilon_2(x)| \mathrm{d}x.$$



Measuring the similarity of distributions

The Wasserstein distance measures the similarity between distributions:

$$d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2) = \int |\Upsilon_1(x) - \Upsilon_2(x)| \mathrm{d}x.$$



Fact. Suppose $\int x^m \mathrm{d}\Upsilon_1(x) = \int x^m \mathrm{d}\Upsilon_2(x)$ for all $m \leq k$. Then $d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2) = O(k^{-1})$.

But we don’t know the moments!

We don’t know the moments of Φ, and computing $\mathbf{A}^m$ is expensive.

What we can do, is approximate the moments with a stochatic trace estimator:

$$\int x^m \mathrm{d}\Phi(x) = n^{-1}\operatorname{tr}(\mathbf{A}^m) \approx n^{-1}\mathbf{v}^\top \mathbf{A}^m \mathbf{v}.$$

But we don't know the moments!

We don't know the moments of Φ, and computing $\mathbf{A}^m$ is expensive.

What we can do, is approximate the moments with a stochatic trace estimator:

$$\int x^m \mathrm{d}\Phi(x) = n^{-1}\operatorname{tr}(\mathbf{A}^m) \approx n^{-1}\mathbf{v}^\top \mathbf{A}^m \mathbf{v}.$$

Note that we can define the weighted CESM

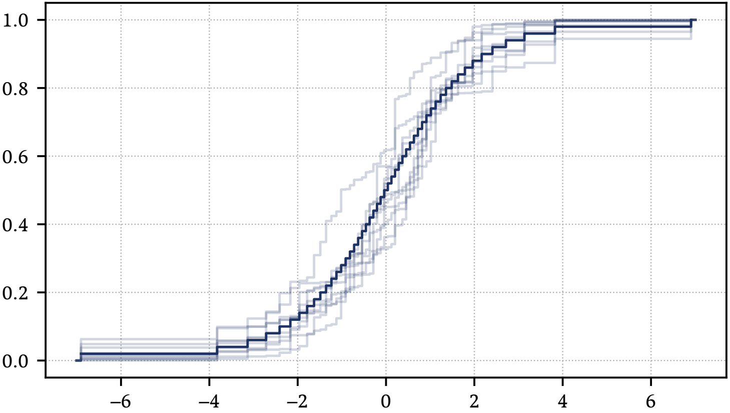
$$\Psi(x) = \sum_{i=1}^{n} |\mathbf{v}^\top \mathbf{u}_i|^2 \mathbb{1}[\lambda_i \leq x], \qquad \frac{\mathrm{d}\Psi(x)}{\mathrm{d}x} = \sum_{i=1}^{n} |\mathbf{v}^\top \mathbf{u}_i|^2 \delta(x - \lambda_i).$$

The weighted CESM is nice to work with:

$$\mathbb{E}[\Psi(x)] = \Phi(x), \qquad \int x^m \Psi(x) = \mathbf{v}^\top \mathbf{A}^m \mathbf{v}.$$

The weighted CESM

CESM (dark) and iid copies of the weighted CESM (light)



Gaussian quadrature: an applied math approach[6]

Consider a distribution of the form

$$\Upsilon(x) = \sum_{i=1}^{s} w_i \mathbb{1}[\theta_i \leq x], \qquad \frac{\mathrm{d}\Upsilon(x)}{\mathrm{d}x} = \sum_{i=1}^{s} w_i \delta(x - \theta_i).$$

This has $2s$ free parameters, so we can hope to match $k = 2s$ moments!

The gaussian quadrature for Ψ is closely related to the orthogonal polynomials of Ψ and can be computed with the Lanczos algorithm.

[6] Bai, Fahey, and Golub 1996.

The kernel polynomial method: a physics approach[7]

Fix a reference measure $\mu(x)$. This gives an inner product

$$\langle f, g\rangle_\mu = \int f(y) g(y) \mathrm{d}\mu(y).$$

Let p_i ($\deg p_i = i$) be the orthogonal polynomails of μ:

$$\|p_i\|_\mu^2 = \int |p_i(x)|^2 \mathrm{d}\mu(x) = 1, \qquad \langle p_i, p_j\rangle_\mu = \int p_i(x) p_j(x) \mathrm{d}\mu(x) = 0, \quad i \neq j.$$

We can decompose a function into the orthogonal polynomials as:

$$f(x) = \sum_{i=0}^{\infty} \langle f, p_i\rangle_\mu \, p_i(x) = \left(\int f(y) p_i(y) \mathrm{d}\mu(y) \right) p_i(x).$$

[7] Skilling 1989; Weiße, Wellein, Alvermann, and Fehske 2006.

The kernel polynomial method: a physics approach

Observe that

$$\frac{\mathrm{d}\Psi(x)}{\mathrm{d}\mu(x)} = \sum_{i=0}^{\infty} \left(\frac{\mathrm{d}\Psi(y)}{\mathrm{d}\mu(y)} p_i(y)\mathrm{d}\mu(y) \right) p_i(x) = \sum_{i=0}^{\infty} \left(p_i(y)\mathrm{d}\Psi(y)\right) p_i(x).$$

Thus,

$$\frac{\mathrm{d}\Psi(x)}{\mathrm{d}x} = \frac{\mathrm{d}\Psi(x)}{\mathrm{d}\mu(x)} \frac{\mathrm{d}\mu(x)}{\mathrm{d}x} = \frac{\mathrm{d}\mu(x)}{\mathrm{d}x} \sum_{i=0}^{\infty} \left(p_i(y)\mathrm{d}\Psi(y)\right) p_i(x).$$

The kernel polynomial method: a physics approach

Observe that

$$\frac{\mathrm{d}\Psi(x)}{\mathrm{d}\mu(x)} = \sum_{i=0}^{\infty} \left(\frac{\mathrm{d}\Psi(y)}{\mathrm{d}\mu(y)} p_i(y) \mathrm{d}\mu(y) \right) p_i(x) = \sum_{i=0}^{\infty} \left(p_i(y) \mathrm{d}\Psi(y) \right) p_i(x).$$

Thus,

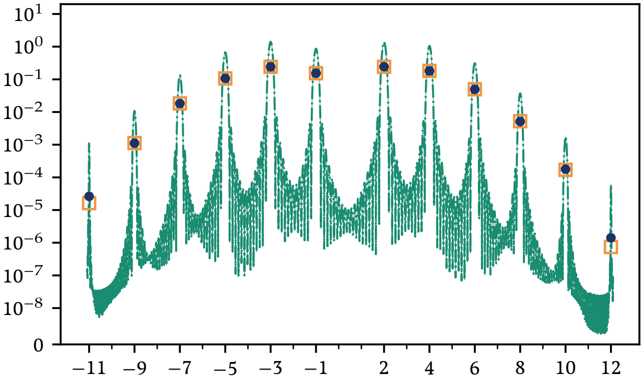
$$\frac{\mathrm{d}\Psi(x)}{\mathrm{d}x} = \frac{\mathrm{d}\Psi(x)}{\mathrm{d}\mu(x)} \frac{\mathrm{d}\mu(x)}{\mathrm{d}x} = \frac{\mathrm{d}\mu(x)}{\mathrm{d}x} \sum_{i=0}^{\infty} \left(p_i(y) \mathrm{d}\Psi(y) \right) p_i(x).$$

We can compute the modified moments $\int p_i(y) \mathrm{d}\Psi(y) = \mathbf{v}^\mathsf{T} p_i(\mathbf{A})\mathbf{v}$ through degree s, so truncate to get an approximation:

$$\frac{\mathrm{d}\Upsilon(x)}{\mathrm{d}x} = \frac{\mathrm{d}\mu(x)}{\mathrm{d}x} \sum_{i=0}^{s} (\mathbf{v}^\mathsf{T} p_i(\mathbf{A})\mathbf{v}) p_i(x).$$

Example: Kneser graph

The spectrum of Kneser graphs is discrete and anlytically known.



Yellow squares: true spectral density, blue dots: GQ, Green: KPM

Theoretical gurantees

How do we analyze these algorithms?

Early analyses[8] use triangle inequality:

$$\left| n^{-1}\operatorname{tr}(f(\mathbf{A})) - \int f \,\mathrm{d}\Upsilon \right| \leq \left| \int f \,\mathrm{d}\big(\Phi - \Psi\big) \right| + \left| \int f \,\mathrm{d}\big(\Psi - \Upsilon\big) \right|.$$

- First term: analyze by stochastic trace estimation bounds
- Second term: by classical quadrature analysis

Shortcomings: Only holds for one function

[8] Han, Malioutov, Avron, and Shin 2017; Ubaru, Chen, and Saad 2017; Cortinovis and Kressner 2021.

Uniform bounds

Recent analyses[9] use the fact:

$$d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2) = \int |\Upsilon_1(x) - \Upsilon_2(x)| \mathrm{d}x = \sup\left\{ \left| \int f \,\mathrm{d}\Upsilon_1 - \int f \,\mathrm{d}\Upsilon_2 \right| : f \text{ 1-Lipschitz} \right\}.$$

[9] Chen, Trogdon, and Ubaru 2021; Braverman, Krishnan, and Musco 2022; Chen, Trogdon, and Ubaru 2022.

[10] Trefethen 2019.

Uniform bounds

Recent analyses[9] use the fact:

$$d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2) = \int |\Upsilon_1(x) - \Upsilon_2(x)| \mathrm{d}x = \sup \left\{ \left| \int f \mathrm{d}\Upsilon_1 - \int f \mathrm{d}\Upsilon_2 \right| : f \text{ 1-Lipschitz} \right\}.$$

Proof sketch. Let p_s be the degree s Chebyshev approximant for $f(x)$. Then:

$$\left| \int f \mathrm{d}(\Phi - \Upsilon) \right| \leq 2 \| f - p_s \|_{[-1,1]} + 2 \sum_{k=1}^{s} \left| \int f T_k \mathrm{d}\mu_{-1,1}^{T} \right| \left| \int T_k \mathrm{d}(\Phi - \Upsilon) \right|.$$

- For families of functions f (e.g. analytic, Lipshitz, etc.) bounds for $\| f - p_s \|_{[-1,1]}$ and the Chebyshev coefficients $\int f T_k \mathrm{d}\mu_{-1,1}^{T}$ are well-known.[10]
- Union bound ensures the Chebyshev moments of Φ and Υ are close for all $k \leq s$.

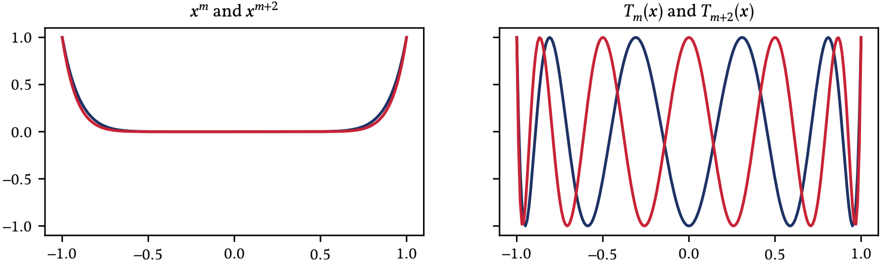
[9] Chen, Trogdon, and Ubaru 2021; Braverman, Krishnan, and Musco 2022; Chen, Trogdon, and Ubaru 2022.

[10] Trefethen 2019.

Chebyshev moments vs monomial moments

While two distribution functions with exactly the same first k moments have Wasserstein distance $O(k^{-1})$, if the monomial moments are even a little different, the Wasserstein distance can be big.

Instead, one should look at Chebyshev moments which are stable with respect to perturbations.



Other related ideas / research directions

- probing / structured test vectors[11]
- Faster trace estimation algorithms via low-rank structure[12]
    - randomized sketching of matrix functions[13]
- Theoretically justified implementations[14]
- Applications!

[11] Stathopoulos, Laeuchli, and Orginos 2013; Halikias and Townsend 2023.

[12] Saibaba, Alexanderian, and Ipsen 2017; Meyer, Musco, Musco, and Woodruff 2021; Epperly, Tropp, and Webber 2023.

[13] Persson and Kressner 2023; Chen and Hallman 2023.

[14] Chen, Trogdon, and Ubaru 2022; Chen 2023.

Quantum equilibrium thermodynamics

Consider a quantum system consisting of subsystems (s) and (b) with Hamiltonian

$$\mathbf{H} = \bar{\mathbf{H}}_\mathrm{s} + \bar{\mathbf{H}}_\mathrm{b} + \mathbf{H}_\mathrm{sb}, \qquad \bar{\mathbf{H}}_\mathrm{s} = \mathbf{H}_\mathrm{s} \otimes \mathbf{I}_\mathrm{b}, \quad \bar{\mathbf{H}}_\mathrm{b} = \mathbf{I}_\mathrm{s} \otimes \mathbf{H}_\mathrm{b}. \tag{1}$$

In thermal equilibrium at interver temperature β, the state of the system is described by a density matrix

$$\boldsymbol{\rho}_\mathrm{t}(\beta) = \frac{\exp(-\beta\mathbf{H})}{Z_\mathrm{t}(\beta)}, \qquad Z_\mathrm{t}(\beta) = \mathrm{tr}(\exp(-\beta\mathbf{H}); \tag{2}$$

The denisty matrix for subsystem (s) is given by

$$\boldsymbol{\rho}^*(\beta) = \mathrm{tr}_\mathrm{b}(\boldsymbol{\rho}_\mathrm{t}(\beta)) = \frac{\mathrm{tr}_\mathrm{b}(\exp(-\beta\mathbf{H}))}{\mathrm{tr}(\exp(-\beta\mathbf{H}))}, \tag{3}$$

where $\mathrm{tr}_b(\,\cdot\,)$ is the *partial trace* over subsystem (b).[15]

[15] Campisi, Zueco, and Talkner 2010; Ingold, Hänggi, and Talkner 2009; Talkner and Hänggi 2020.

Partial traces

Suppose $\mathbf{A}$ is a $d_\mathrm{s}d_\mathrm{b} \times d_\mathrm{s}d_\mathrm{b}$ matrtix partitioned as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,d_\mathrm{s}} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,d_\mathrm{s}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{d_\mathrm{s},1} & \mathbf{A}_{d_\mathrm{s},2} & \cdots & \mathbf{A}_{d_\mathrm{s},d_\mathrm{s}} \end{bmatrix},$$

Partial traces

Then the partial trace (wrt. this partitioning) is defined as:

$$\mathrm{tr}_{\mathrm{b}}(\mathbf{A}) = \begin{bmatrix} \mathrm{tr}(\mathbf{A}_{1,1}) & \mathrm{tr}(\mathbf{A}_{1,2}) & \cdots & \mathrm{tr}(\mathbf{A}_{1,d_{\mathrm{s}}}) \\ \mathrm{tr}(\mathbf{A}_{2,1}) & \mathrm{tr}(\mathbf{A}_{2,2}) & \cdots & \mathrm{tr}(\mathbf{A}_{2,d_{\mathrm{s}}}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{tr}(\mathbf{A}_{d_{\mathrm{s}},1}) & \mathrm{tr}(\mathbf{A}_{d_{\mathrm{s}},2}) & \cdots & \mathrm{tr}(\mathbf{A}_{d_{\mathrm{s}},d_{\mathrm{s}}}) \end{bmatrix}.$$

Partial traces

We can use a randomized estimator:[16]

$$(\mathbf{I}_{d_\mathrm{s}} \otimes \mathbf{v})^\mathsf{T} \mathbf{A} (\mathbf{I}_{d_\mathrm{s}} \otimes \mathbf{v}) = \begin{bmatrix} \mathbf{v}^\mathsf{T}\mathbf{A}_{1,1}\mathbf{v} & \mathbf{v}^\mathsf{T}\mathbf{A}_{1,2}\mathbf{v} & \cdots & \mathbf{v}^\mathsf{T}\mathbf{A}_{1,d_\mathrm{s}}\mathbf{v} \\ \mathbf{v}^\mathsf{T}\mathbf{A}_{2,1}\mathbf{v} & \mathbf{v}^\mathsf{T}\mathbf{A}_{2,2}\mathbf{v} & \cdots & \mathbf{v}^\mathsf{T}\mathbf{A}_{2,d_\mathrm{s}}\mathbf{v} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}^\mathsf{T}\mathbf{A}_{d_\mathrm{s},1}\mathbf{v} & \mathbf{v}^\mathsf{T}\mathbf{A}_{d_\mathrm{s},2}\mathbf{v} & \cdots & \mathbf{v}^\mathsf{T}\mathbf{A}_{d_\mathrm{s},d_\mathrm{s}}\mathbf{v} \end{bmatrix}.$$

[16] Chen and Cheng 2022.

Partial trace estimator: analysis

Define the varaince of a random matrix as:

$$\mathbb{V}[\mathbf{X}] = \mathbb{E}\left[\|\mathbf{X} - \mathbb{E}[\mathbf{X}]\|_{\mathsf{F}}^2\right] = \sum_i \sum_j \mathbb{V}[X_{i,j}]^2.$$

Then, since $\mathbb{V}[\mathbf{v}^\top \mathbf{A}_{i,j}\mathbf{v}] = 2\|\mathbf{A}_{i,j}\|_{\mathsf{F}}^2$,

$$\mathbb{V}\left[(\mathbf{I}_{d_s} \otimes \mathbf{v})^\top \mathbf{A}(\mathbf{I}_{d_s} \otimes \mathbf{v})\right] = \sum_{i=1}^{d_s}\sum_{j=1}^{d_s} \mathbb{V}[\mathbf{v}^\top \mathbf{A}_{i,j}\mathbf{v}] = \sum_{i=1}^{d_s}\sum_{j=1}^{d_s} 2\|\mathbf{A}_{i,j}\|_{\mathsf{F}}^2 = 2\|\mathbf{A}\|_{\mathsf{F}}^2.$$

As before, if $\mathbf{v}_1, \dots, \mathbf{v}_m$ are independent and identically distributed copies of $\mathbf{v}$, then

$$\mathbb{V}\left[\frac{1}{m}\sum_{i=1}^{m}(\mathbf{I}_{d_s} \otimes \mathbf{v}_i)^\top \mathbf{A}(\mathbf{I}_{d_s} \otimes \mathbf{v}_i)\right] = \frac{2}{m}\|\mathbf{A}\|_{\mathsf{F}}^2.$$

Partial trace estimator: variance reduction

For any matrix $\widetilde{\mathbf{A}}$,

$$\mathrm{tr}_{\mathrm{b}}(\mathbf{A}) = \mathrm{tr}_{\mathrm{b}}(\widetilde{\mathbf{A}}) + \mathrm{tr}_{\mathrm{b}}(\mathbf{A} - \widetilde{\mathbf{A}}).$$

So we might try to use the estimator

$$\mathrm{tr}_{\mathrm{b}}(\mathbf{A}) \approx \mathrm{tr}_{\mathrm{b}}(\widetilde{\mathbf{A}}) + \widehat{\mathrm{tr}}_{\mathrm{b}}^{m}(\mathbf{A} - \widetilde{\mathbf{A}}).$$

which will have reduced variance if $\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}}^2 \ll \|\mathbf{A}\|_{\mathsf{F}}^2$.

This residual trick is widely used in regular trace estimation.[17]

[17] Girard 1987; Weiße, Wellein, Alvermann, and Fehske 2006; Morita and Tohyama 2020; Meyer, Musco, Musco, and Woodruff 2021.

A cancellation issue

We could try to take $\widetilde{\mathbf{A}} = \mathbf{Q}\mathbf{Q}^\top\mathbf{A}\mathbf{Q}\mathbf{Q}^\top$, for some orthonormal $\mathbf{Q}$.

Recall, however, that in our seting $\mathbf{A} = \exp(-\beta\mathbf{H})$, and we must approxiamte products with $\mathbf{A}$. This can lead to cancellation issues in the term:

$$\widehat{\mathsf{tr}}_{\mathrm{b}}^{m}(\mathbf{A} - \widetilde{\mathbf{A}}).$$

A cancellation issue

We could try to take $\widetilde{\mathbf{A}} = \mathbf{Q}\mathbf{Q}^\top\mathbf{A}\mathbf{Q}\mathbf{Q}^\top$, for some orthonormal $\mathbf{Q}$.

Recall, however, that in our seting $\mathbf{A} = \exp(-\beta\mathbf{H})$, and we must approxiamte products with $\mathbf{A}$. This can lead to cancellation issues in the term:

$$(100042 \pm 0.01\%) - (100017 \pm 0.01\%) = (42 - 17) \pm 20 = \text{no accuracy.}$$

A cancellation issue

We could try to take $\widetilde{\mathbf{A}} = \mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{A}\mathbf{Q}\mathbf{Q}^\mathsf{T}$, for some orthonormal $\mathbf{Q}$.

Recall, however, that in our seting $\mathbf{A} = \exp(-\beta\mathbf{H})$, and we must approxiamte products with $\mathbf{A}$. This can lead to cancellation issues in the term:

$$(100042 \pm 0.01\%) - (100017 \pm 0.01\%) = (42 - 17) \pm 20 = \text{no accuracy}.$$

With normal traces, we can use the cyclic property to write

$$\operatorname{tr}(\mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{A}\mathbf{Q}\mathbf{Q}^\mathsf{T}) = \operatorname{tr}(\mathbf{A}\mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{Q}\mathbf{Q}^\mathsf{T}) = \operatorname{tr}(\mathbf{A}\mathbf{Q}\mathbf{Q}^\mathsf{T}).$$

Thus, we can avoid cancellation by using:

$$\operatorname{tr}(\mathbf{A} - \mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{A}\mathbf{Q}\mathbf{Q}^\mathsf{T}) = \operatorname{tr}(\mathbf{A}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^\mathsf{T})) = \operatorname{tr}((\mathbf{I} - \mathbf{Q}\mathbf{Q}^\mathsf{T})\mathbf{A}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^\mathsf{T})).$$

A fix[18]

Suppose $\mathbf{Q}$ contains only eigenvectors of $\mathbf{A} = \sum_i \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$. Then it can be shown,

$$\mathbf{A} - \mathbf{Q}\mathbf{Q}^\top \mathbf{A}\mathbf{Q}\mathbf{Q}^\top = (\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top)\mathbf{A}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top).$$

This avoids the cancellation issues.

[18] Chen, Chen, Li, Nzeuton, Pan, and Wang 2023.

A fix[18]

Suppose $\mathbf{Q}$ contains only eigenvectors of $\mathbf{A} = \sum_i \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$. Then it can be shown,

$$\mathbf{A} - \mathbf{Q}\mathbf{Q}^\top \mathbf{A}\mathbf{Q}\mathbf{Q}^\top = (\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top)\mathbf{A}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top).$$

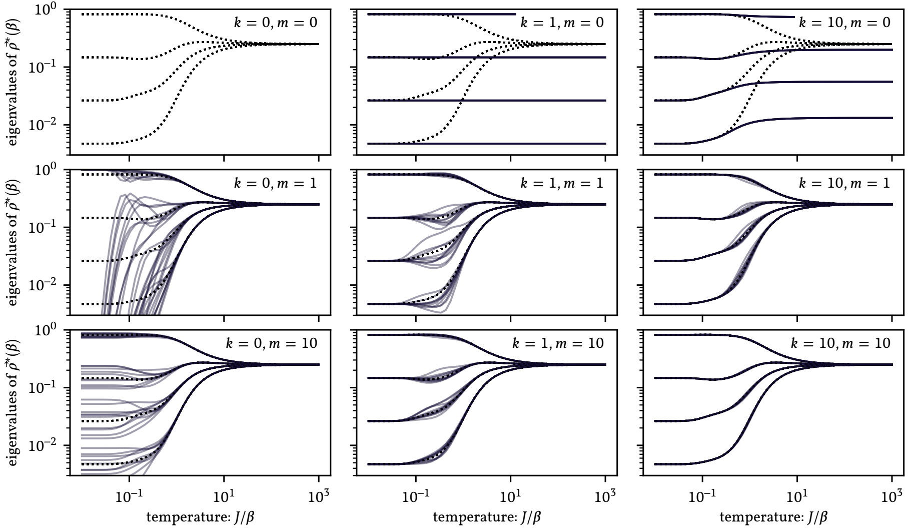
This avoids the cancellation issues.

Proof. WLOG assume $\mathbf{Q} = \mathbf{u}_j$. Note that

$$\begin{aligned}
\mathbf{A} - \mathbf{u}_j\mathbf{u}_j^\top \mathbf{A}\mathbf{u}_j\mathbf{u}_j &= \sum_{i=1}^{n} \lambda_i \left(\mathbf{u}_i\mathbf{u}_i^\top - \mathbf{u}_j\mathbf{u}_j^\top\mathbf{u}_i\mathbf{u}_i^\top\mathbf{u}_j\mathbf{u}_j^\top\right) \\
&= \sum_{i\neq j} \lambda_i (\mathbf{I} - \mathbf{u}_j\mathbf{u}_j^\top)\mathbf{u}_i\mathbf{u}_i^\top(\mathbf{I} - \mathbf{u}_j\mathbf{u}_j^\top) \\
&= (\mathbf{I} - \mathbf{u}_j\mathbf{u}_j^\top)\mathbf{A}(\mathbf{I} - \mathbf{u}_j\mathbf{u}_j^\top).
\end{aligned}$$

[18] Chen, Chen, Li, Nzeuton, Pan, and Wang 2023.

Eigenvalues of $\boldsymbol{\rho}^*(\beta)$: parameter test



von Neumann entropy

The von Neumann entropy $-\operatorname{tr}(\boldsymbol{\rho}^*(\beta)\ln(\boldsymbol{\rho}^*(\beta)))$ is a measure of the entanglement betweeen subsystems (s) and (b).

Understanding the von Neumann entropy for a range of a system with Hamiltonian $\mathbf{H}(\theta)$ at a range of parameter values θ and inverse temperatures β is of interest.

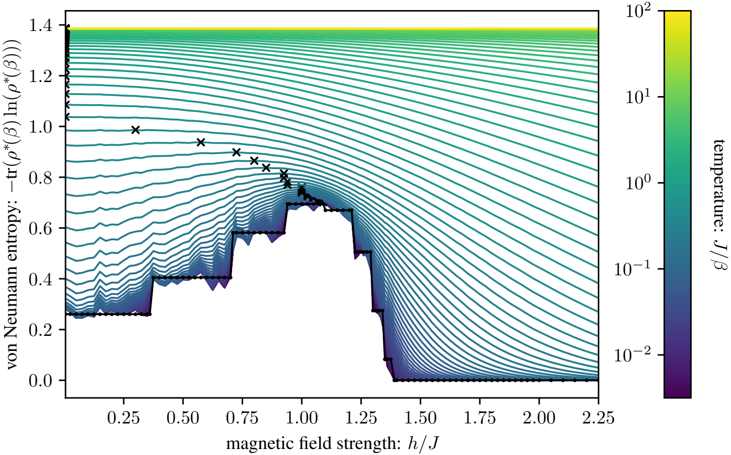
We will consider a special case

$$\mathbf{H} = \sum_{|i-j|=1} \left[J^{\mathrm{x}}_{i,j} \boldsymbol{\sigma}^{\mathrm{x}}_i \boldsymbol{\sigma}^{\mathrm{x}}_j + J^{\mathrm{y}}_{i,j} \boldsymbol{\sigma}^{\mathrm{y}}_i \boldsymbol{\sigma}^{\mathrm{y}}_j \right] + \frac{h}{2} \sum_{i=1}^{N} \boldsymbol{\sigma}^{\mathrm{z}}_i .$$

where h is the magnetic field strength.

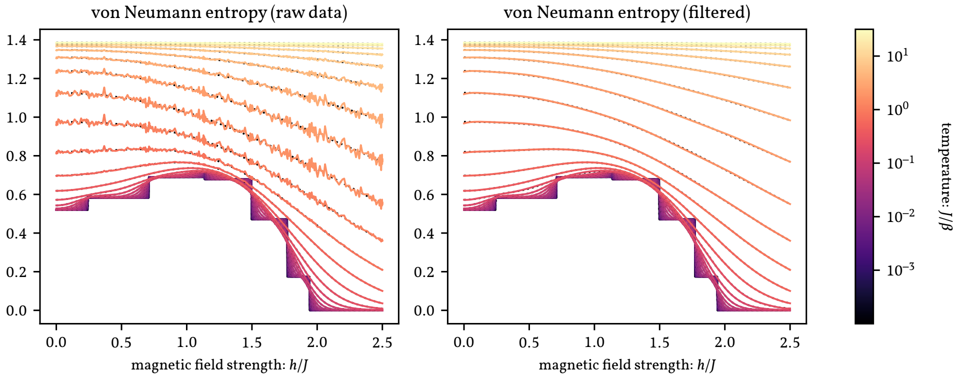
Subsystem (s) corresponds to $i = 1, 2$ and subsystem (b) corresponds to the rest of the spins.

von Neumann entropy phase plot[19]



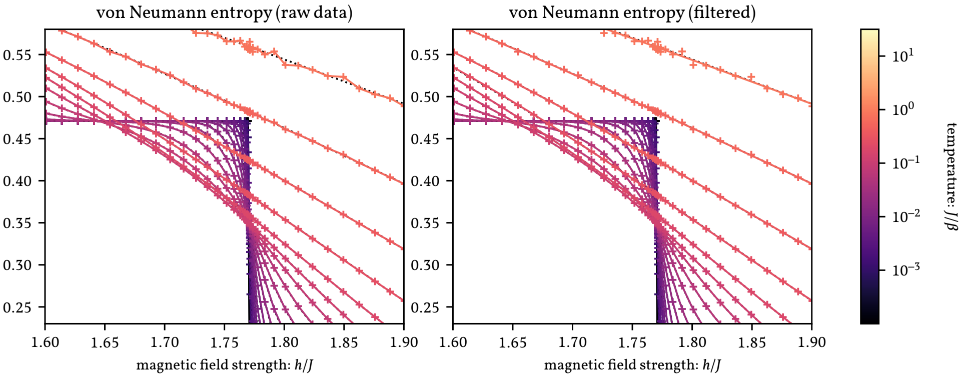
[19] Chen and Cheng 2022.

von Neumann entropy phase plot[20]



[20] Chen, Chen, Li, Nzeuton, Pan, and Wang 2023.

von Neumann entropy phase plot (cropped)[21]



[21] Chen, Chen, Li, Nzeuton, Pan, and Wang 2023.

References I

Alben, R. et al. (Nov. 1975). “Exact results for a three-dimensional alloy with site diagonal disorder: comparison with the coherent potential approximation”. In: *Physical Review B* 12.10, pp. 4090–4094.

Bai, Zhaojun, Gark Fahey, and Gene Golub (Nov. 1996). “Some large-scale matrix computation problems”. In: *Journal of Computational and Applied Mathematics* 74.1-2, pp. 71–89.

Braverman, Vladimir, Aditya Krishnan, and Christopher Musco (June 2022). *Sublinear time spectral density estimation*.

Campisi, Michele, David Zueco, and Peter Talkner (Oct. 2010). “Thermodynamic anomalies in open quantum systems: Strong coupling effects in the isotropic XY model”. In: *Chemical Physics* 375.2-3, pp. 187–194.

Chen, Tyler (2023). *A spectrum adaptive Kernel Polynomial Method*.

Chen, Tyler and Yu-Chen Cheng (2022). *Numerical computation of the equilibrium-reduced density matrix for strongly coupled open quantum systems*.

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.

Chen, Tyler, Thomas Trogdon, and Shashanka Ubaru (July 2021). “Analysis of stochastic Lanczos quadrature for spectrum approximation”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 1728–1739.

— (2022). *Randomized matrix-free quadrature for spectrum and spectral sum approximation*.

Chen, Tyler et al. (2023). *Faster randomized partial trace estimation*.

Cortinovis, Alice and Daniel Kressner (July 2021). “On Randomized Trace Estimates for Indefinite Matrices with an Application to Determinants”. In: *Foundations of Computational Mathematics*.

References II

Epperly, Ethan N. , Joel A. Tropp, and Robert J. Webber (2023). *XTrace: Making the most of every sample in stochastic trace estimation*.

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

Halikias, Diana and Alex Townsend (2023). *Structured matrix recovery from matrix-vector products*.

Han, Insu et al. (Jan. 2017). “Approximating Spectral Sums of Large-Scale Matrices using Stochastic Chebyshev Approximations”. In: *SIAM Journal on Scientific Computing* 39.4, A1558–A1585.

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.

Ingold, Gert-Ludwig, Peter Hänggi, and Peter Talkner (June 2009). “Specific heat anomalies of open quantum systems”. In: *Physical Review E* 79.6.

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.

Morita, Katsuhiro and Takami Tohyama (Feb. 2020). “Finite-temperature properties of the Kitaev-Heisenberg models on kagome and triangular lattices studied by improved finite-temperature Lanczos methods”. In: *Physical Review Research* 2.1.

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.

Saibaba, Arvind K, Alen Alexanderian, and Ilse CF Ipsen (2017). “Randomized matrix-free trace and log-determinant estimators”. In: *Numerische Mathematik* 137.2, pp. 353–395.

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

References III

Stathopoulos, Andreas, Jesse Laeuchli, and Kostas Orginos (Jan. 2013). “Hierarchical Probing for Estimating the Trace of the Matrix Inverse on Toroidal Lattices”. In: *SIAM Journal on Scientific Computing* 35.5, S299–S322.

Talkner, Peter and Peter Hänggi (Oct. 2020). “Colloquium : Statistical mechanics and thermodynamics at strong coupling: Quantum and classical”. In: *Reviews of Modern Physics* 92.4.

Trefethen, Lloyd N. (2019). *Approximation Theory and Approximation Practice, Extended Edition*. SIAM.

Ubaru, Shashanka, Jie Chen, and Yousef Saad (2017). “Fast Estimation of $tr(f(A))$ via Stochastic Lanczos Quadrature”. In: *SIAM Journal on Matrix Analysis and Applications* 38.4, pp. 1075–1099.

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