Spectral Estimation with Free Decompression

Team Members

Siavash Ameli

International Computer Science Institute

University of California, Berkeley

Chris van der Heide

The University of Melbourne

Liam Hodgkinson

The University of Melbourne

Michael W. Mahoney

International Computer Science Institute

Lawrence Berkeley National Laboratory

University of California, Berkeley

Computing Eigenvalues of Large Matrices

- Eigenvalues encode essential matrix information; empirical spectral distribution is useful for diagnostics, e.g. is the spectrum heavy-tailed?
- Particularly useful for computing spectral functions, including

$$\log\det A = \sum_i \log \lambda_i(A) \qquad \mathrm{tr}(A^{-k}) = \sum_i \lambda_i(A)^{-k}$$

$$\mathrm{cond}(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \quad \text{if } A \text{ is positive-definite.}$$

- These quantities are important e.g. for Gaussian processes, but need the *entire range* of eigenvalues, not just largest/smallest
- Standard eigenvalue solvers are $\mathcal{O}(n^3)$ complexity; expensive for large matrices!

Tiers of Matrix Difficulty

Explicit: the whole matrix fits in memory

Implicit: can make use of matrix-vector products (e.g. CG, SLQ)

Out-of-core: parts of the matrix can be loaded into memory a piece at a time

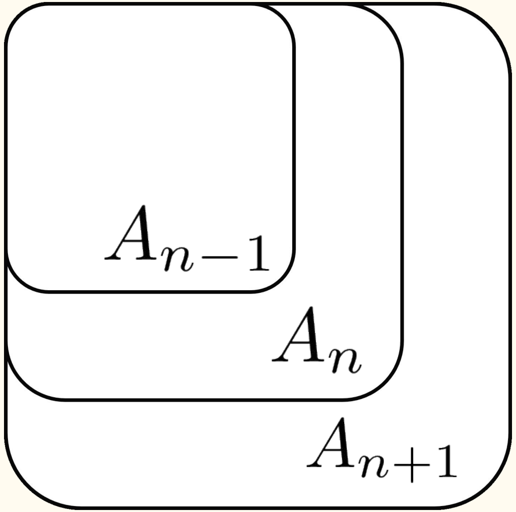
Impalpable: most matrix entries are inaccessible, matrix-vector products are unavailable (e.g. distributed or enormous datasets)

	Access in Memory		
Type	Matrix	Matrix–Vector Product	Any Subblock
Explicit	✓	✓	✓
Implicit	✗	✓	✗
Out-of-core	✗	$\sim$	✓
Impalpable	✗	✗	✗

Extrapolating Matrices

Suppose our matrix of interest is embedded in an infinite sequence of nested matrices

$$A_1, A_2, A_3, \ldots \qquad\qquad A_n \in \mathbb{R}^{n\times n}$$



so that $(A_n)_{ij} = (A_{n+1})_{ij}$

Objective: Find eigenspectrum of A_n using eigenspectrum of A_{n_s} for $n_s \ll n$

Free Probability

How do we ensure the eigenvalues of submatrices represent the whole matrix?

An important topic in random matrix theory involving random matrices with uniformly random eigenvectors, so that probability distributions of matrix dependents (including submatrices) *depend only on the eigenspectra*.

Theorem (Nica, 1993): Any sequence of matrices can be turned into an (asymptotically) free sequence of random matrices by applying random permutations σ to the rows and columns:

$$\tilde{A}_{ij} = A_{\sigma(i)\sigma(j)}$$

Stieltjes Transform

The spectral density of a matrix A is encoded in its Stieltjes transform:

$$m_n(z) = \frac{1}{n}\mathrm{tr}(A - zI)^{-1} \qquad A \in \mathbb{R}^{n\times n}$$

In the large matrix limit, when the eigenvalues are drawn from a density ρ, there is a one-to-one correspondence between ρ and the Stieltjes transform m.

$$m(z) = \int_{-\infty}^{\infty} \frac{\rho(x)}{x - z}\,\mathrm{d}x \qquad \rho(x) = \frac{1}{\pi}\lim_{\varepsilon\to 0^+} \Im[m(x + i\varepsilon)]$$

Free Decompression

Let $m(t, \cdot)$ be the Stieltjes transform of the enlargement of A by a factor of e^t
Under the large matrix limit, $m(t, \cdot)$ satisfies the *partial differential equation:*

$$\frac{\partial m}{\partial t} = -m + \frac{1}{m}\frac{\partial m}{\partial z}$$

Proof: Random matrix theory arguments involving the R-transform and the celebrated theorem of (Nica & Speicher, 1996).

To our knowledge, this operation has always been considered in reverse (*free compression*), finding eigenspectra of submatrices, given the eigenspectrum of the full matrix. We are the first to attempt ***free decompression***.

Free decompression of a random submatrix $\mathbf{A}_n$ to a larger matrix $\mathbf{A}$ requires:

1. **estimation** of its Stieltjes transform $m_{\mathbf{A}_n}$;
2. **evolution** of $m_{\mathbf{A}_n}$ in n using PDE;
3. **evaluation** of the spectral distribution of $\mathbf{A}$.

An Engineering Challenge

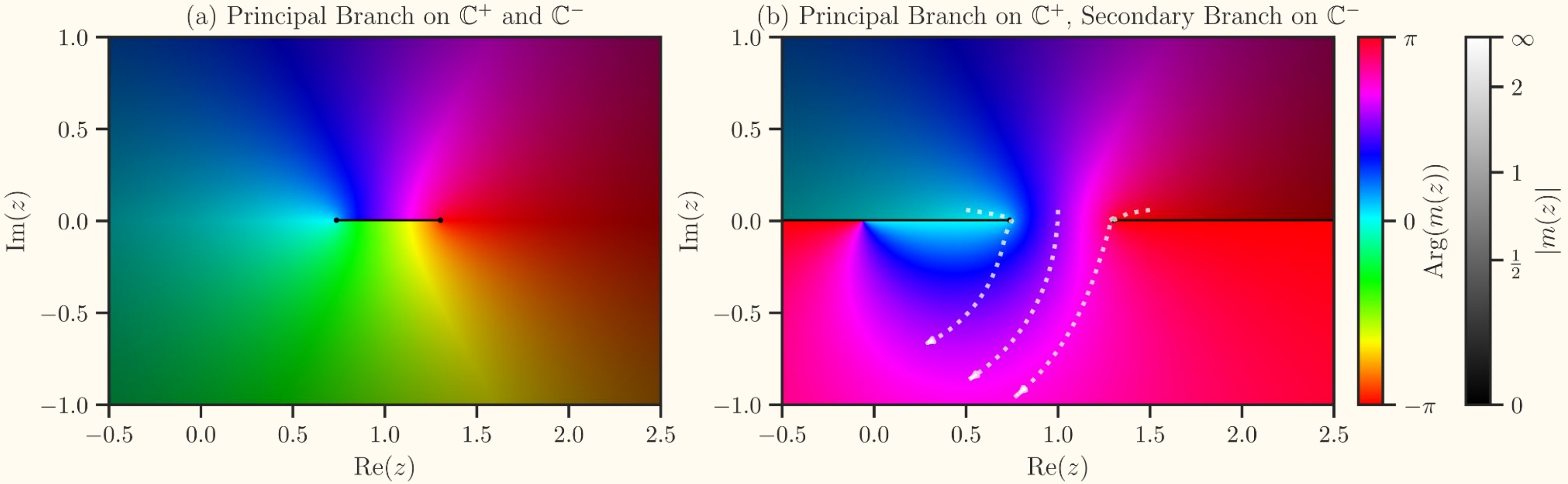
This is a very difficult equation to solve!

Solve the PDE using method of characteristics in the complex plane. But...

Proposition: All characteristic curves pass through the (discontinuous) branch cut for the principal branch of the Stieltjes transform.

- ➢ To solve the characteristic equations, a new *secondary* branch is required.
- ➢ Tantamount to (ill-posed) numerical analytic continuation.
- ➢ Naively solving the PDE fails: we need to directly tackle the analytic continuation problem.

Analytic Continuation of Stieltjes Transform



An Engineering Challenge

This is a very difficult equation to solve!

Theorem: The error grows **at most polynomially** in the matrix size.

Requires significantly more engineering than first glance:

- ➢ Multiple layers of polynomial approximation from eigenvalues (Lanczos iteration and Kernel Polynomial Method are not accurate enough)
- ➢ Construct a particular Padé approximant
- ➢ Solve characteristic curves using Newton iterations

Performed properly, in practice, error grows **at most logarithmically** in the matrix size.

Random Matrix Ensembles

Distribution	Free Corresp.	Abs. Cont. Density $\rho(x)$	Support $\lambda_\pm$	Number of Atoms
Wigner semicircle	Free Gaussian	$\frac{2\sqrt{r^2 - x^2}}{\pi r^2}$	$\pm r$	None
Marchenko–Pastur	Free Poisson	$\frac{\sqrt{(\lambda_+ - x)(x - \lambda_-)}}{2\pi\lambda x}$	$(1 \pm \sqrt{\lambda})^2$	$(1 - \frac{1}{\lambda})\delta(x)$ if $\lambda > 1$
Kesten–McKay	Free Binomial	$\frac{d\sqrt{4(d-1) - x^2}}{2\pi(d^2 - x^2)}$	$\pm 2\sqrt{d-1}$	None
Wachter	Free Jacobi	$\frac{(a+b)\sqrt{(\lambda_+ - x)(x - \lambda_-)}}{2\pi x(1-x)}$	$\left(\frac{\sqrt{b} \pm \sqrt{a(a+b-1)}}{a+b}\right)^2$	$x = 0, 1$
Meixner	Free Meixner	$\frac{c\sqrt{4b - (x-a)^2}}{2\pi((1-c)x^2 + acx + bc^2)}$	$a \pm 2\sqrt{b}$	At most two

Random Matrix Ensembles

Distribution	Matrix or Combinatorial Model	Parameters
Wigner semicircle	Gaussian orthogonal ensemble $\frac{1}{\sqrt{2}}(\mathbf{X}+\mathbf{X}^\intercal)$ Adjacency matrix of Erdős–Rényi graph $G(n,p)$	$r = 2\sqrt{n}$ $pn = \mathcal{O}(\log(n))$
Marchenko–Pastur	Sample covariance (Wishart) $\frac{1}{d}\mathbf{X}\mathbf{X}^\intercal$, $\mathbf{X} \in \mathbb{R}^{n\times d}$	$\lambda = \frac{n}{d}$
Kesten–McKay	Haar–orthogonal Hermitian sum $\sum_{i=1}^{k}(\mathbf{O}_i + \mathbf{O}_i^\intercal)$ Projection model $d\,\mathbf{PODO}^\intercal\mathbf{P}$ (Longoria & and, 2023) Adjacency matrix of a random d-regular graph	$d = 2k$ $d \geq 2$ $d \geq 2$
Wachter	Generalized eigenvalues of $(\mathbf{S}_1, \mathbf{S}_1 + \mathbf{S}_2)$, $\mathbf{S}_i = \frac{1}{d_i}\mathbf{X}_i\mathbf{X}_i^\intercal$ Arises in MANOVA problems	$a = \frac{d_1}{n}$, $b = \frac{d_2}{n}$
Meixner	Bordered Toeplitz tridiagonal with Jacobi coefficients α_1, β_1 Block–Gaussian ensembles (Lenczewski, 2015)	$a = \alpha_1, b = \beta_1 - 1$

Random Matrix Ensembles

$$\mathbf{J} = \left[\begin{array}{c|cccc} \alpha_0 & \beta_0 & & & \\ \hline \beta_0 & \alpha_1 & \beta_1 & & \\ & \beta_1 & \alpha_1 & \beta_1 & \\ & & \ddots & \ddots & \ddots \end{array}\right]$$

For Meixner family, the Jacobi matrix of orthogonal polynomial recursion is periodic.

$$m(z) = \cfrac{1}{z - \alpha_0 - \cfrac{\beta_0^2}{z - \alpha_1 - \cfrac{\beta_1^2}{z - \alpha_1 - \cfrac{\beta_1^2}{\ddots}}}}$$

Stieltjes transform, as continued fraction of Jacobi coefficients, becomes periodic.

Stieltjes transform can be solved by quadratic equation:

$$m(z) = \frac{P(z) + \sqrt{P(z)^2 - 4Q(z)}}{2Q(z)}$$

Random Matrix Ensembles

	Stieltjes and Hilbert Transforms		
Distribution	$P(z)$	$Q(z)$	R-Transform
Wigner semicircle	$-z$	$\frac{r^2}{4}$	$\frac{r^2}{4}z$
Marchenko–Pastur	$1-\lambda-z$	λz	$\frac{1}{1-\lambda z}$
Kesten–McKay	$\frac{(2-d)z}{d-1}$	$\frac{d^2-z^2}{d-1}$	$\frac{-d+d\sqrt{1+4z^2}}{2z}$
Wachter	$\frac{a-1-(a+b-2)z}{a+b-1}$	$\frac{z(1-z)}{a+b-1}$	$\frac{-(a+b)+z+\sqrt{(a+b)^2+2(a-b)z+z^2}}{2z}$
Meixner	$\frac{ac-(c-2)z}{2}$	$\frac{(1-c)z^2+acz+bc^2}{4}$	$\left(\frac{c}{1-c}\right)\frac{1-az+\sqrt{(1-az)^2-4b(1-c)z^2}}{2z}$

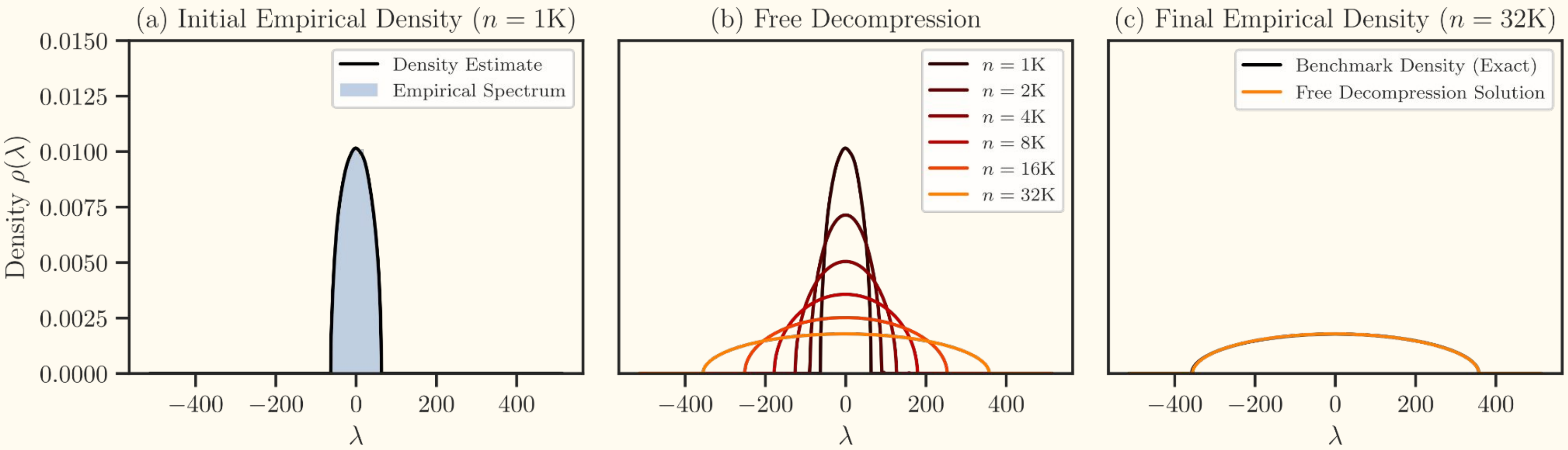
Experiments with Random Matrix Ensembles

These are convenient baselines, since we know the expected shape of the eigenspectrum in advance *for any matrix size* (computing eigenvalues is expensive!)

Under normally-distributed synthetic data, we expand

$$n = 1000 \xrightarrow{\text{free decompression}} n = 32,000$$

Matrices with iid Entries (Wigner Semicircle Law)

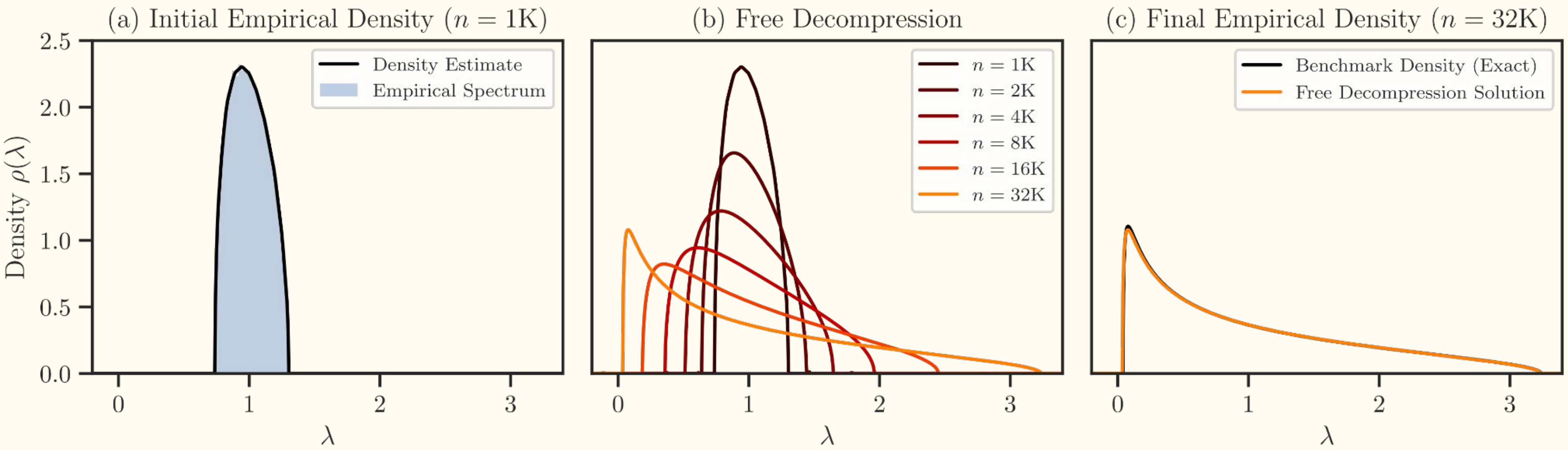


Histogram of eigenvalues of small matrix & density estimate

Densities under free decompression

Expected density & solution from free decompression

Wishart Matrices (Marchenko-Pastur Law)

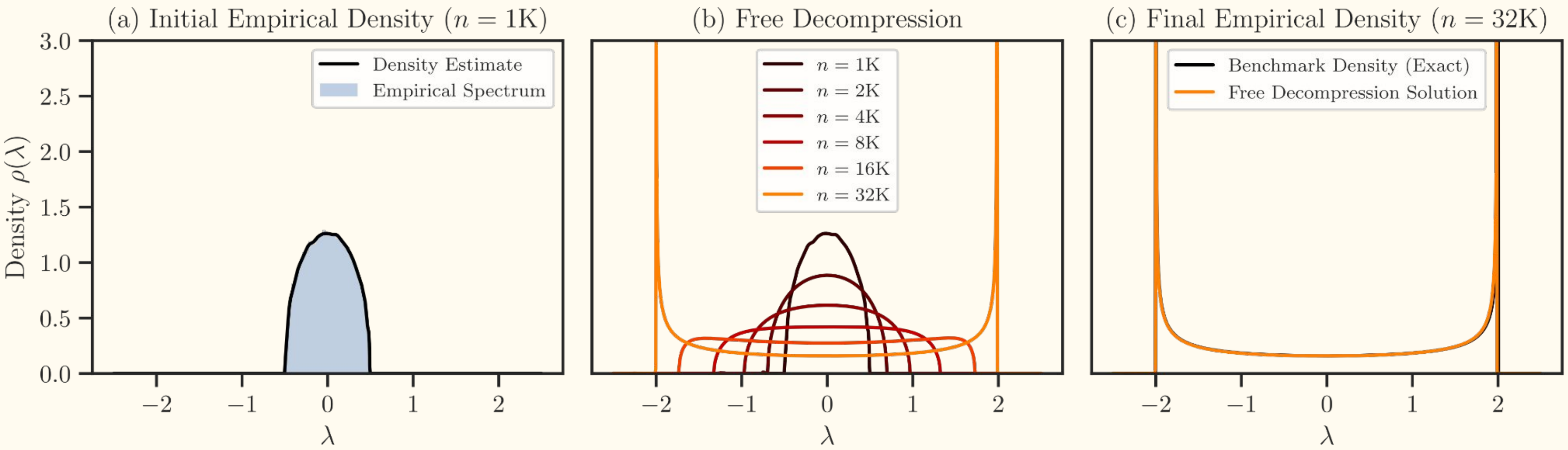


Histogram of eigenvalues of small matrix & density estimate

Densities under free decompression

Expected density & solution from free decompression

Random Projections (Kesten-McKay Law)

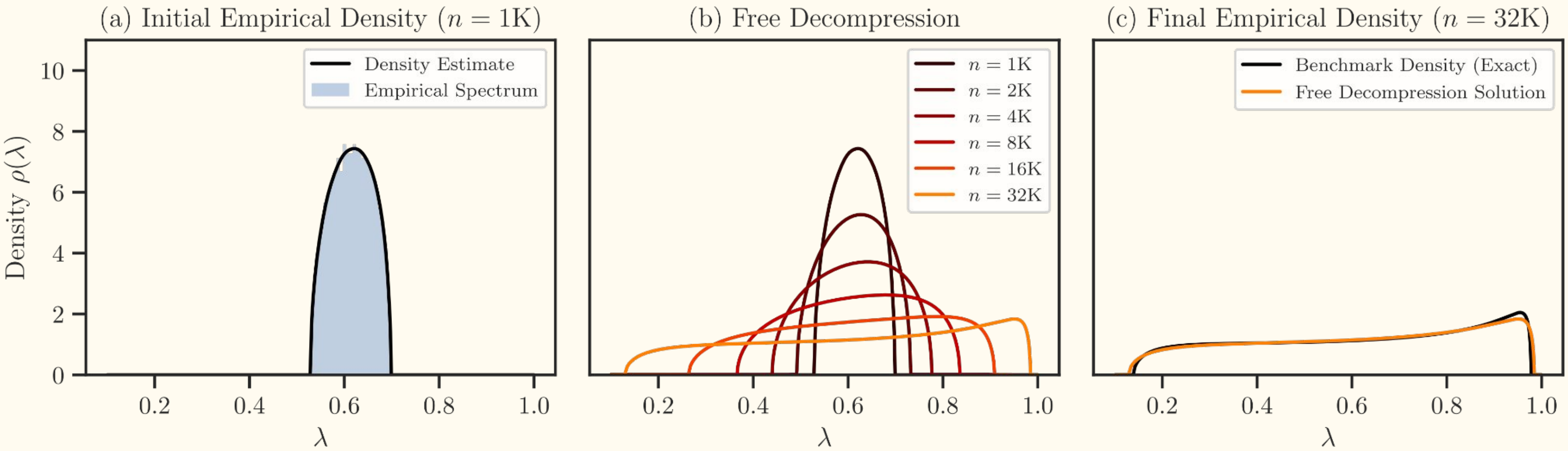


Histogram of eigenvalues of small matrix & density estimate

Densities under free decompression

Expected density & solution from free decompression

Generalized Eigenvalue Problems (Wachter Law)

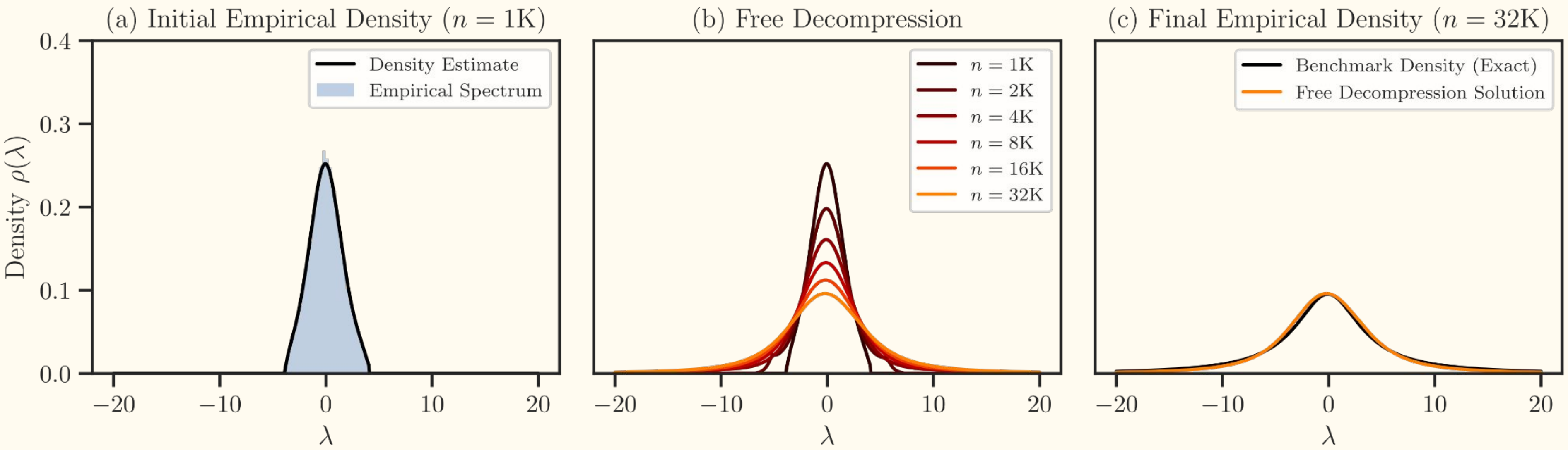


Histogram of eigenvalues of small matrix & density estimate

Densities under free decompression

Expected density & solution from free decompression

General Family of Meixner Law



Histogram of eigenvalues of small matrix & density estimate

Densities under free decompression

Expected density & solution from free decompression

Experiments with Real Data

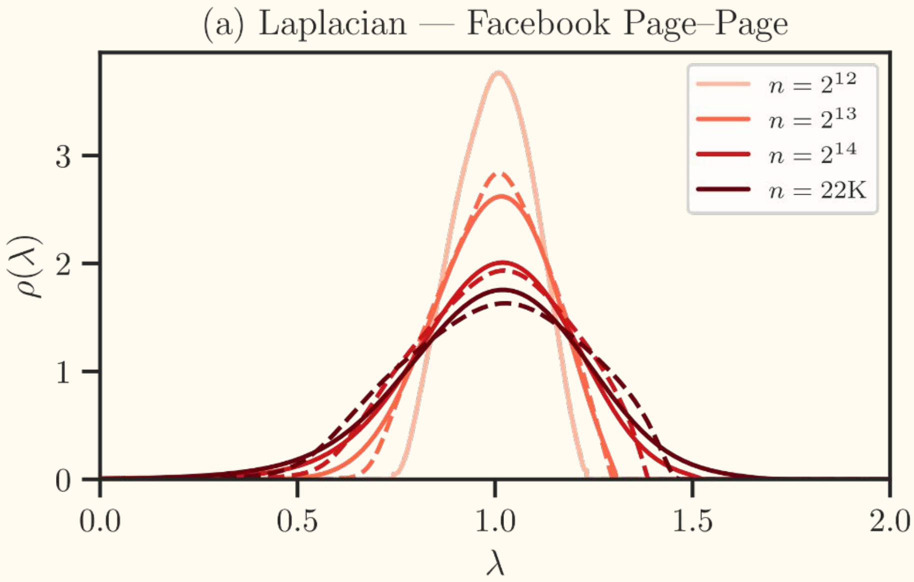
Large covariance and kernel matrices involving real data typically exhibit disconnected spectral densities with support over multiple orders of magnitude.

Density estimation remains a significant challenge here

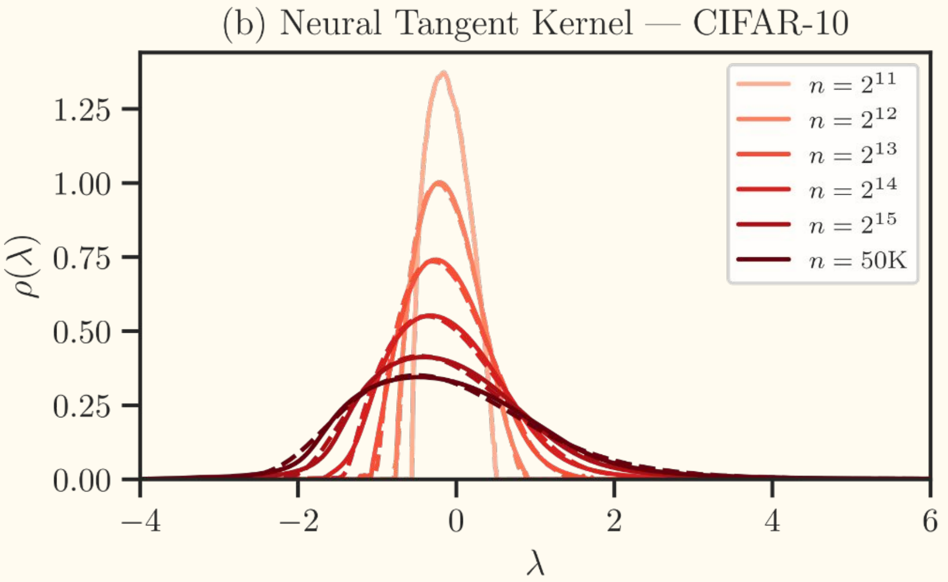
We consider two examples of real data matrices to demonstrate efficacy of our current procedure:

1. **Facebook SNAP Graph** Dataset (22,470 x 22,470 adjacency matrix) perturbed by an Erdős-Rényi graph to reduce leaf nodes.
2. Log-**neural tangent kernel** Gram matrix from ResNet50 trained on CIFAR-10 with low-rank components removed (50,000 x 50,000 dense matrix).

Experiments with Real Data



Symmetrically normalized Laplacian matrix of the SNAP Facebook dataset



log-NTK matrix computed from the CIFAR-10 dataset using a ResNet-50 model

Empirical spectral density (solid) vs. free decompression estimate from $n = 2^{11}$ (dashed)

Experiments with Real Data

Table: Comparison of runtime of direct computation of spectral density versus the free decompression of the NTK dataset, and accuracy in terms of statistical distance and moments.

Size	Process Time (sec)		Divergences		Rel. Error	
n_s	Direct	FD (ours)	TV	JS	μ_1	μ_2
2^{11}	10.2	10.2 + 0.00	0.0%	0.0%	0.0%	0.0%
2^{12}	50.9	10.2 + 54.2	1.2%	3.7%	0.4%	0.3%
2^{13}	358.9	10.2 + 56.6	1.9%	5.2%	0.9%	0.2%
2^{14}	2820.2	10.2 + 54.9	2.4%	5.8%	0.9%	0.1%
2^{15}	20451.2	10.2 + 61.9	2.6%	5.8%	1.2%	0.5%
50K	67331.1	10.2 + 16.2	2.9%	5.5%	2.4%	0.4%

freealg

freealg is our Python package that implements free decompression for estimating eigenspectra.

`pip install freealg`

(work in progress!)

Listing 1: A minimal usage example of the **freealg** package.

```
# Install freealg with "pip install freealg"
import freealg as fa

# Create an object for the Marchenko--Pastur distribution with the parameter λ = 1/50
mp = fa.distributions.MarchenkoPastur(1/50)

# Generate a matrix of size n_s = 1000 corresponding to this distribution
A = mp.matrix(size=1000)

# Create a free-form object for the matrix within the support I = [λ_-, λ_+]
ff = fa.FreeForm(A, support=(mp.lam_m, mp.lam_p))

# Fit the distribution using Jacobi polynomials of degree K = 20, with α = β = 1/2
# Also fit the glue function via Pade of degree [(p+q)/q] with p = 0, q = 1.
psi = ff.fit(method='jacobi', K=20, alpha=0.5, beta=0.5, reg=0.0, damp='jackson',
             pade_p=0, pade_q=1, optimizer='ls', plot=True)

# Decompress the spectral density corresponding to a larger matrix of size n = 2^5 × n_s,
rho_large = ff.decompress(size=32_000, plot=True)
```

arxiv

Siavash Ameli, Chris van der Heide, Liam Hodgkinson, Michael W. Mahoney. (2025) *Spectral Estimation with Free Decompression*. arxiv: 2506.11994