Randomized algorithms for trace estimation

Alice Cortinovis

Joint work with Daniel Kressner and David Persson

15 February 2022
Due giorni di Algebra Lineare Numerica – Napoli 2022

EPFL

Goal:

Compute (an approximation of) $\mathrm{trace}(A)$
for a large-scale symmetric matrix $A \in \mathbb{R}^{n \times n}$.

Only matrix-vector multiplications with A are available.

Outline

1. Motivation
2. Hutchinson's trace estimator + bounds
3. Hutch++
4. Our improved version of Hutch++ (A-Hutch++)

Motivation

Example 1: Trace of matrix functions / Determinant

Matrix functions: For symmetric B, given a spectral decomposition $B = Q \cdot \mathrm{diag}(\lambda_1, \dots, \lambda_n) \cdot Q^T$, the matrix function $f(B)$ is defined as $f(B) = Q \cdot \mathrm{diag}(f(\lambda_1), \dots, f(\lambda_n)) \cdot Q^T$.

Computing $Av = f(B)v$ is faster than computing $f(B)$!

- Trace(B^{-1}) (Uncertainty quantification, Lattice quantum chromodynamics)
- Network analysis ($\exp(B)$, Estrada index)
- Determinant of symmetric positive definite B via $\det(B) = \exp(\mathrm{trace}(\log B))$ (Statistical machine learning, Markov random fields models, graph theory (# spanning trees)

Example 2: Frobenius norm estimation $\|B\|_F^2 = \mathrm{trace}(B^T B)$ and other Schatten-p norms.

Hutchinson's trace estimator

Theorem

If X is random vector s.t. $\mathbb{E}[XX^T] = I$ then

$$\mathbb{E}[X^T A X] = \mathrm{trace}(A).$$

Proof: $\mathbb{E}[X^T A X] = \sum_{i,j} \mathbb{E}[X_i X_j] A_{ij} = \sum_i A_{ii} = \mathrm{trace}(A)$.

Most common choices for X:

- Gaussian vectors ($X \sim \mathcal{N}(0, I_n)$)
- Rademacher vectors (± 1 i.i.d. entries)

Hutchinson's trace estimator: Take N independent copies $X^{(1)}, \ldots, X^{(N)}$ of X and approximate

$$\mathrm{trace}(A) \approx \mathrm{trace}_N(A) := \frac{1}{N} \sum_{i=1}^{N} (X^{(i)})^T A X^{(i)}.$$

Hutchinson's trace estimator: Example

$$\text{trace}(A) \approx \text{trace}_N(A) := \frac{1}{N}\sum_{i=1}^{N}(X^{(i)})^T A X^{(i)}$$

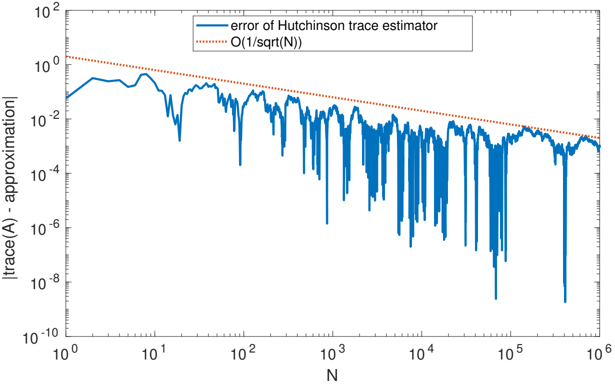


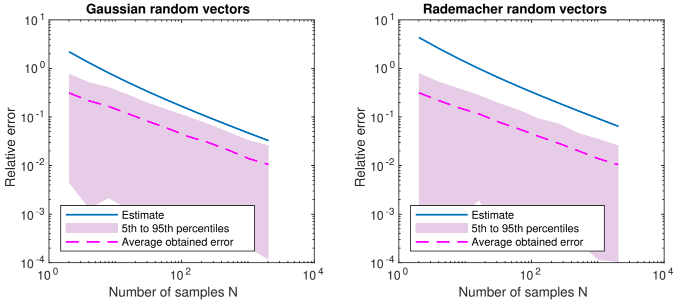
Figure: The behavior of $|\text{trace}(A) - \text{trace}_N(A)|$ when increasing N (# probe vectors).

Convergence of Hutchinson's trace estimator

Tail bounds:

$$\mathbb{P}\left(|\text{trace}(A) - \text{trace}_N(A)| \geq \varepsilon\right) \leq \delta \quad \text{for } N \geq \left(\frac{C_1}{\varepsilon^2}\|A\|_F^2 + \frac{C_2}{\varepsilon}\|A\|_2\right) \log\frac{2}{\delta},$$

with $(C_1, C_2) = (4, 4)$ for Gaussian random vectors, $(C_1, C_2) = (8, 16)$ for Rademacher random vectors [C./Kressner'2021].



What if A is approximately low-rank?

Remark: If A has rapidly decaying singular values, denote by A_r = best rank-r approximation of A, then $\mathrm{trace}(A_r)$ is a good approximation of $\mathrm{trace}(A)$!

Reminder: Best rank-r approximation is given by truncated singular value decomposition (SVD). Expensive to obtain!

Faster method: Randomized SVD [Halko/Martinsson/Tropp'2011].

Input: $A \in \mathbb{R}^{n \times n}$, integer r
Output: Rank-r approximation of A

1. Draw r random vectors and put them in an $n \times r$ matrix X ($r \ll n$).
2. Perform matrix-vectors multiplications $Y = AX$. (r matvecs)
3. Compute orthonormal basis Q of Y.
4. Return low-rank approximation $A \approx Q(Q^T A)$. (r matvecs)

Hutch++ [Meyer/Musco/Musco/Woodruff'2021]

Input: Matrix $A \in \mathbb{R}^{n \times n}$, number of matvecs $3r$
Output: Estimate of $\mathrm{trace}(A)$

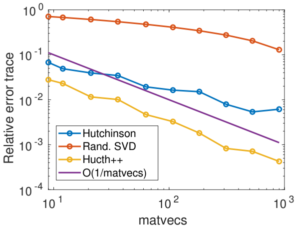
1. RandSVD phase: X random $n \times r$, compute Q_r = ONB of AX.
2. Hutchinson phase:

$$\begin{aligned}\mathrm{trace}(A) &= \mathrm{trace}(Q_r^T A Q_r) + \mathrm{trace}((I - Q_r Q_r^T)A(I - Q_r Q_r^T)) \\ &\approx \mathrm{trace}(Q_r^T A Q_r) + \mathrm{trace}_r((I - Q_r Q_r^T)A(I - Q_r Q_r^T)).\end{aligned}$$

(r matvecs for AX, r for $Q_r^T A$, r for Hutchinson)

For symmetric positive definite matrix A, number of matvecs needed to reach relative accuracy ε goes from $O\left(\frac{1}{\varepsilon^2}\right)$ to $O\left(\frac{1}{\varepsilon}\right)$.

Alg. also useful for indefinite A!



A-Hutch++

Our goal:
Given $A \in \mathbb{R}^{n\times n}$ (available through matvecs), find approx. of $\mathrm{trace}(A)$ with accuracy $\varepsilon > 0$, failure probability δ. Also, try to minimize #matvecs.

[From now on: Use Gaussian random vectors to simplify analysis.]

We do not know how many vectors we want to use for the randSVD phase!

But: Assume we have done randSVD with r vectors and we got a low-rank approximation $Q_r Q_r^T A$. Thanks to tail bounds on Hutchinson's trace estimator, we know that

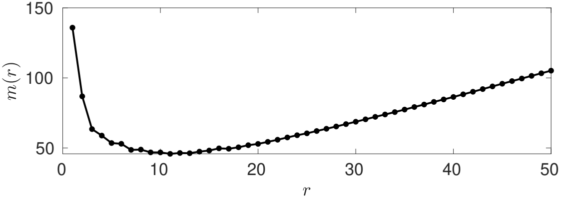
$$\approx \underbrace{\frac{4}{\varepsilon^2}\log\frac{2}{\delta}}_{C(\varepsilon,\delta)} \|(I - Q_r Q_r^T)A(I - Q_r Q_r^T)\|_F^2$$

samples are sufficient for reaching accuracy ε in the Hutchinson phase.

Counting the required matvecs

Therefore, the cost (= #matvecs) if we use r vectors in the randSVD phase is

$$m(r) := \text{\# matvecs needed} = \overbrace{2r}^{\text{for rand SVD}} + C(\varepsilon,\delta)\| \overbrace{(I - Q_rQ_r^T)A(I - Q_rQ_r^T)}^{\text{residual from randSVD}} \|_F^2$$
$$= \underbrace{2r - C(\varepsilon,\delta)(\|Q_r^TAQ_r\|_F^2 - 2\|Q_r^TA\|_F^2)}_{\widetilde{m}(r)} + C(\varepsilon,\delta)\|A\|_F^2.$$



If $Q_rQ_r^TA$ was the best rank-r approx then $m(r)$ would have a unique minimum (in practice, still well-behaved).

A-Hutch++

We start with $r = 1, 2, 3, \ldots$ and keep track of $\widetilde{m}(r)$. We stop the randSVD phase when

$$\widetilde{m}(r_*) \leq \widetilde{m}(r_* \pm 1).$$

Then we need

$$N := \lceil C(\varepsilon, \delta)(\|A\|_F^2 + \|Q_r^T A Q_r\|_F^2 - 2\|Q_r^T A\|_F^2) \rceil$$

matvecs for the Hutchinson phase.

We really need $\|A\|_F$! We approximate it by Hutchinson's trace estimator and use [Roosta-Khorasani/Székely/Ascher'2015] for convergence analysis.

Theorem: The accuracy is guaranteed.

Numerical experiments for A-Hutch++

Set ε, run A-Hutch++, get $m = \#$matvecs, run Hutch++ with input m.

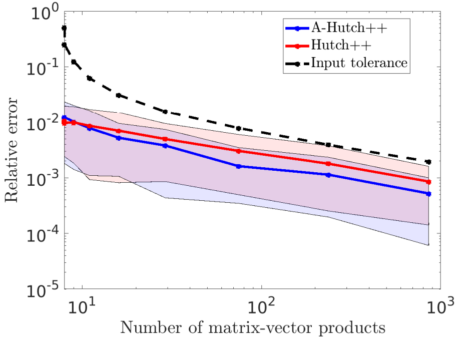


Figure: Eigenvalues of A are $1, 2^{-0.1}, 3^{-0.1}, \ldots, n^{-0.1}$.

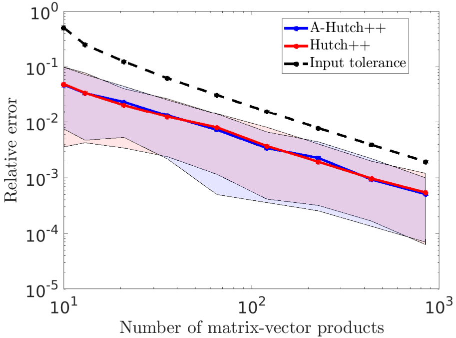


Figure: Eigenvalues of A are $1, 2^{-1}, 3^{-1}, \ldots, n^{-1}$.

Conclusions

Summary: A-Hutch++ outputs an estimate of $\mathrm{trace}(A)$ with accuracy ε with small failure probability.

Future directions: If $A = f(B)$, also need to approximate matvecs! One matvec with A corresponds to *many* matvecs with B: how to optimize #matvecs with B? What accuracy is needed?

AC, Daniel Kressner: On randomized trace estimates for indefinite matrices with an application to determinants. Foundations of Computational Mathematics, 2021.

David Persson, AC, Daniel Kressner. Improved variants of the Hutch++ algorithm for trace estimation. https://arxiv.org/abs/2109.10659