Trace Estimation III
– Making the most of every sample –
Lecture 7

F. M. Faulstich

01/30/2024

Implicit Trace Estimation Problem

- Given access to $\mathbf{A} \in \mathbb{F}^{n\times n}$ via the MatVec product $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$, estimate its trace:

$$\mathrm{Tr}(\mathbf{A}) = \sum_{i=i}^{n} (\mathbf{A})_{ii}$$

Implicit Trace Estimation Problem

- Given access to $\mathbf{A} \in \mathbb{F}^{n \times n}$ via the MatVec product $\mathbf{x} \mapsto \mathbf{Ax}$, estimate its trace:

$$\mathrm{Tr}(\mathbf{A}) = \sum_{i=i}^{n} (\mathbf{A})_{ii}$$

- [Girard–Hutchinson estimator] Let $\{\boldsymbol{\omega}_i\}$ be isotropic and i.i.d. then

$$\hat{\mathrm{tr}}_{\mathrm{GH}} := \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\omega}_i^* (A \boldsymbol{\omega}_i)$$

is an unbiased estimator of the trace

$$\mathbb{E}(\hat{\mathrm{tr}}_{\mathrm{GH}}) = \mathrm{Tr}(\mathbf{A}).$$

- We found

$$\mathbb{V}(\hat{\mathrm{tr}}_{\mathrm{GH}}) = \frac{1}{m} \mathbb{V}(\boldsymbol{\omega}^* (A \boldsymbol{\omega})) \in \mathcal{O}\left(\frac{1}{m}\right)$$

$\Rightarrow$ Converges as $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$ (Monte-Carlo)

Variance reduction (Hutch++)

Given m – a fixed number of MatVecs:

- Sample isotropic i.i.d. $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_{2m/3}$
- Sketch $\mathbf{Y} = \mathbf{A}[\boldsymbol{\omega}_{m/3+1}|\boldsymbol{\omega}_{m/3+2}|...|\boldsymbol{\omega}_{2m/3}]$
- Orthonormalize $\mathbf{Q} = \mathrm{orth}(\mathbf{Y})$
- Output estimator

$$\hat{\mathrm{tr}}_{H++} = \mathrm{Tr}(\mathbf{Q}^*(\mathbf{A}\mathbf{Q})) + \frac{1}{m/3}\sum_{i=1}^{m/3} \boldsymbol{\omega}_i^*(\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\big(\mathbf{A}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\boldsymbol{\omega}_i\big)$$

- Recall that $\hat{\mathbf{A}} = \mathbf{Q}\mathbf{Q}^*\mathbf{A}$ is a low rank approximation of $\mathbf{A}$
- Hutch++ computes the trace of this low-rank approximation

$$\mathrm{Tr}(\hat{\mathbf{A}}) = \mathrm{Tr}(\mathbf{Q}\mathbf{Q}^*\mathbf{A}) = \mathrm{Tr}(\mathbf{Q}^*(\mathbf{A}\mathbf{Q}))$$

  and then applies the Girard–Hutchinson estimator to the residual

$$\mathrm{Tr}(\mathbf{A} - \hat{\mathbf{A}}) = \mathrm{Tr}((\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{A}) = \mathrm{Tr}((\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{A}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^*))$$

- Hutch++ is an unbiased trace estimator of $\mathbf{A}$

$$\mathbb{E}(\hat{\mathrm{tr}}_{\mathrm{H++}}) = \mathrm{Tr}(\mathbf{A})$$

  and

$$\mathbb{V}(\hat{\mathrm{tr}}_{\mathrm{H++}}) \in \mathcal{O}\left(\frac{1}{m^2}\right)$$

HUTCH++ Pseudocode

- Input: $\mathbf{A} \in \mathbb{F}^{n \times n}$, m with $\text{mod}(m, 3) = 0$
- Output: $\hat{\text{tr}}_{\text{H++}}$
- Draw iid isotropic $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_{2m/3} \in \mathbb{F}^n$
- $\mathbf{Y} = \mathbf{A}[\boldsymbol{\omega}_{m/3+1} | \boldsymbol{\omega}_{m/3+2} | ... | \boldsymbol{\omega}_{2m/3}]$
- $\mathbf{Q} = \text{orth}(\mathbf{Y})$
- $\mathbf{G} = [\boldsymbol{\omega}_1 | \boldsymbol{\omega}_2 | ... | \boldsymbol{\omega}_{m/3}] - \mathbf{Q}\mathbf{Q}^*[\boldsymbol{\omega}_1 | \boldsymbol{\omega}_2 | ... | \boldsymbol{\omega}_{m/3}]$
- $\hat{\text{tr}}_{\text{H++}} = \text{Tr}(\mathbf{Q}^*(\mathbf{AQ})) - \frac{1}{m/3}\text{Tr}(\mathbf{G}^*(\mathbf{AG}))$

Indeed, we require m MatVecs

Exchangeable

- Exchangeability principle:
  If the test vectors $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_k$ are exchangeable, the "minimum-variance estimator" is always a symmetric function or $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_k$
  [invariant under application of the symmetric group $(\boldsymbol{\omega}_{\sigma(1)}, ..., \boldsymbol{\omega}_{\sigma(k)})$]
- An estimator is exchangeable, if it is invariant under under application of the symmetric group
- Exchangeability can be seen as a "robustness" property of probabilistic algorithms:

  > "Exchangeability implies that each element in the sequence of estimators contributes equally to the estimation" process

Exchangeable

- Exchangeability principle:
  If the test vectors $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_k$ are exchangeable, the "minimum-variance estimator" is always a symmetric function or $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_k$
  [invariant under application of the symmetric group $(\boldsymbol{\omega}_{\sigma(1)}, ..., \boldsymbol{\omega}_{\sigma(k)})$]
- An estimator is exchangeable, if it is invariant under under application of the symmetric group
- Exchangeability can be seen as a "robustness" property of probabilistic algorithms:

  > "Exchangeability implies that each element in the sequence of estimators contributes equally to the estimation" process

- The HUTCH++ estimator is not exchangeable
  [it uses some test vectors to perform low-rank approx]

$\Rightarrow$ Development of XTRACE estimator

XTrace Estimator

Idea: Use all but one test vector to form a low-rank approximation, and only use the remaining test vector to estimate the trace of the residual.

XTRACE Estimator

- Draw $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_{m/2}$ i.i.d. isotropic test vectors, and form

$$\boldsymbol{\Omega} := [\boldsymbol{\omega}_1 | ... | \boldsymbol{\omega}_{m/2}]$$

- Construct the orthonormal matrices

$$\mathbf{Q}_{(i)} = \mathrm{orth}(A\boldsymbol{\Omega}_{-i})$$

where $\boldsymbol{\Omega}_{-i}$ is the test matrix with the ith column removed.

- Compute the basic estimators

$$\hat{\mathrm{tr}}_i := \mathrm{Tr}(\mathbf{Q}_{(i)}^*(\mathbf{A}\mathbf{Q}_{(i)})) + \boldsymbol{\omega}_i^*(\mathbf{I} - \mathbf{Q}_{(i)}\mathbf{Q}_{(i)}^*)(\mathbf{A}(\mathbf{I} - \mathbf{Q}_{(i)}\mathbf{Q}_{(i)}^*)\boldsymbol{\omega}_i)$$

- m/2. The XTRCE estimator averages these basic estimators:

$$\hat{\mathrm{tr}}_X := \frac{1}{m/2}\sum_{i=1}^{m/2} \hat{\mathrm{tr}}_i$$

XTrace Estimator

- The XTRCE estimator is an unbiased estimator of $\mathrm{Tr}(\mathbf{A})$
- The XTRCE estimator is invariant under the action of the symmetry group

XTRACE Estimator Naïve Implementation

- Input: $\mathbf{A} \in \mathbb{F}^{n\times n}$, m with $\mathrm{mod}(m,2)=0$
- Output: $\hat{\mathrm{tr}}_X$ and trace error estimate
- Draw i.i.d isotropic $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_{m/2}$
- $\mathbf{Y} = \mathbf{A}[\boldsymbol{\omega}_1|...|\boldsymbol{\omega}_{m/2}]$
- for i=1 to m/2

  $\mathbf{Q}_{(i)} = \mathrm{ortho}(\mathbf{Y}_{-i})$

  $\hat{\mathrm{tr}}_{\mathrm{i}} = \mathrm{Tr}(\mathbf{Q}_{(i)}^*(\mathbf{A}\mathbf{Q}_{(i)})) + \boldsymbol{\omega}_i^*(\mathbf{I} - \mathbf{Q}_{(i)}\mathbf{Q}_{(i)}^*)(\mathbf{A}(\mathbf{I} - \mathbf{Q}_{(i)}\mathbf{Q}_{(i)}^*)\boldsymbol{\omega}_i)$
- $\hat{\mathrm{tr}} = \frac{1}{m/2}\sum_{i=1}^{m/2}\hat{\mathrm{tr}}_{\mathrm{i}}$
- $\hat{\mathrm{err}}^2 = \frac{1}{(m/2)(m/2-1)}\sum i = 1^{m/2}(\hat{\mathrm{tr}}_{\mathrm{i}} - \hat{\mathrm{tr}})^2$

XNysTrace Estimator

- The central idea of the variance improved estimators is to use a low-rank approximation of $\mathbf{A}$
- For an arbitrary matrix this requires
- What about $\mathbf{A} \in \mathbb{H}_n$ and $0 \preccurlyeq \mathbf{A}$?

$\Rightarrow$ Nyström approximation

Nyström approximation

- Let $\mathbf{A} \in \mathbb{H}_n$ and $0 \preccurlyeq \mathbf{A}$. Then

$$\mathbf{A}\langle\mathbf{X}\rangle = \mathbf{AX}(\mathbf{X}^*\mathbf{AX})^\dagger(\mathbf{AX})^* = \mathbf{Y}(\mathbf{X}^*\mathbf{Y})^\dagger\mathbf{Y}^*$$

  is the Nyström approximation for a test matrix $\mathbf{X} \in \mathbb{F}^{n\times s}$.
- Clearly $\operatorname{rank}(\mathbf{A}\langle\mathbf{X}\rangle) \leq s$.
- Note that we only need a single application of $\mathbf{AX}$ to compute the Nyström approximation.
- The randomized SVD requires two!

$\Rightarrow$ The Nyström approximation only requires k MatVecs whereas the randomized SVD requires $2k$ MatVecs.

Why does it work?

Proof for block matrix formulation

- Recall for

$$\mathbf{A} = \begin{pmatrix} \mathbf{W} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{C} \end{pmatrix}$$

  we have $\mathbf{A}/\mathbf{W} = \mathbf{C} - \mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T$

- The Nyström approximation is given by

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{W} \\ \mathbf{B} \end{pmatrix} \mathbf{W}^{-1} \begin{pmatrix} \mathbf{W} & \mathbf{B}^T \end{pmatrix} = \begin{pmatrix} \mathbf{W} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T \end{pmatrix}$$

- Let's look at

$$\mathbf{A} - \tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{W} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{C} \end{pmatrix} - \begin{pmatrix} \mathbf{W} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}/\mathbf{W} \end{pmatrix}$$

- Note: Nyström is a rough approximation

XNysTrace Estimator Naïve

- Input: $\mathbf{A} \in \mathbb{H}_n$ with $0 \preccurlyeq \mathbf{A}$, and $m \in \mathbb{N}$
- Output: $\hat{tr}_{\text{XN}}$, and trace error estimate
- Draw i.i.d. isotropic $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m$
- $\mathbf{\Omega} = [\boldsymbol{\omega}_1 | ... | \boldsymbol{\omega}_m]$
- $\mathbf{Y} = \mathbf{A\Omega}$
- for i = 1 to m

$$\mathbf{A}_i = \mathbf{Y}_{-i}(\mathbf{\Omega}^*_{-i}\mathbf{Y}_{-i})^\dagger \mathbf{Y}^*_{-i}$$

$$\hat{\text{tr}}_i = \text{Tr}(\mathbf{A}_i) + \boldsymbol{\omega}_i^*((A - A_i)\boldsymbol{\omega}_i)$$

$$\hat{\text{tr}}_{\text{XN}} = \frac{1}{m}\sum_{i=1}^m \hat{\text{tr}}_i$$

$$\hat{\text{err}}^2 = \frac{1}{m(m-1)}\sum_{i=1}^m (\hat{\text{tr}}_i - \hat{\text{tr}})^2$$

Computational Performance

Set up:

- Consider the matrix

$$\mathbf{A}(\boldsymbol{\lambda}) = \mathbf{U}\text{diag}(\boldsymbol{\lambda})\mathbf{U}^*$$

  where $\mathbf{U}$ is a Haar random orthogonal matrix.

- A Haar random orthogonal matrix is a matrix drawn uniformly from the set of all orthogonal matrices of a given size:
  i) Generate a $\mathbf{A} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  ii) $[\mathbf{Q}, \mathbf{R}] = \text{qr}(\mathbf{A})$
  iii) $\mathbf{D} = \text{diag}(\text{sign}(\text{diag}(R)))$
  iv) $\mathbf{Q} = \mathbf{QD}$

- For $\boldsymbol{\lambda}$ four choices are considered:
  i) flat: $\boldsymbol{\lambda} = (3 - 2(i-1)/(N-1) \ : \ i = 1, 2, ..., N)$
  ii) poly: $\boldsymbol{\lambda} = (i^{-2} \ : \ i = 1, 2, ..., N)$
  iii) <u>exp</u>: $\boldsymbol{\lambda} = (0.7^i \ : \ i = 0, 2, ..., N-1)$
  iv) step: $\boldsymbol{\lambda} = (\underbrace{1, ..., 1}_{50 \text{ times}}, \underbrace{10^{-3}, ..., 10^{-3}}_{N-50 \text{ times}})$

Computational Performance

- Apply the estimators to a random PSD matrix with exponentially decreasing eigenvalue
- Run 1000 trails for feasible m
- compute the averaged error of the trace per estimator

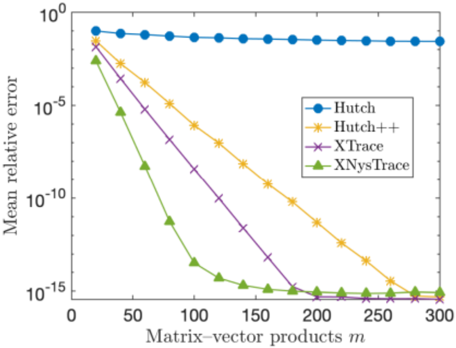

Figure: Computational performance of different trace estimators[1]

[1] Epperly, Tropp, Webber, SIMAX, 2024