Hutch++

Optimal Stochastic Trace Estimation

Raphael A. Meyer (New York University)

With Christopher Musco (New York University), Cameron Musco (University of Massachusetts Amherst), and David P. Woodruff (Carnegie Mellon University)

Trace Estimation

- Goal: Estimate trace of $d \times d$ matrix $\boldsymbol{A}$:

$$\mathrm{tr}(\boldsymbol{A}) = \sum_{i=1}^{d} \boldsymbol{A}_{ii} = \sum_{i=1}^{d} \lambda_i$$

Trace Estimation

- Goal: Estimate trace of $d \times d$ matrix $\boldsymbol{A}$:

$$\text{tr}(\boldsymbol{A}) = \sum_{i=1}^{d} \boldsymbol{A}_{ii} = \sum_{i=1}^{d} \lambda_i$$

- In Downstream Applications, $\boldsymbol{A}$ is not stored in memory.
- Instead, $\boldsymbol{B}$ is in memory and $\boldsymbol{A} = f(\boldsymbol{B})$:

No. Triangles	Estrada Index	Log-Determinant
$\text{tr}(\frac{1}{6}\boldsymbol{B}^3)$	$\text{tr}(e^{\boldsymbol{B}})$	$\text{tr}(\ln(\boldsymbol{B}))$

Trace Estimation

- Goal: Estimate trace of $d \times d$ matrix $\boldsymbol{A}$:

$$\text{tr}(\boldsymbol{A}) = \sum_{i=1}^{d} \boldsymbol{A}_{ii} = \sum_{i=1}^{d} \lambda_i$$

- In Downstream Applications, $\boldsymbol{A}$ is not stored in memory.
- Instead, $\boldsymbol{B}$ is in memory and $\boldsymbol{A} = f(\boldsymbol{B})$:

No. Triangles	Estrada Index	Log-Determinant
$\text{tr}(\frac{1}{6}\boldsymbol{B}^3)$	$\text{tr}(e^{\boldsymbol{B}})$	$\text{tr}(\ln(\boldsymbol{B}))$

- Computing $\boldsymbol{A} = \frac{1}{6}\boldsymbol{B}^3$ takes $O(n^3)$ time
- Computing $\boldsymbol{A}\mathbf{x} = \frac{1}{6}\boldsymbol{B}(\boldsymbol{B}(\boldsymbol{B}\mathbf{x}))$ takes $O(n^2)$ time
- If $\boldsymbol{A} = f(\boldsymbol{B})$, then we can often compute $\boldsymbol{A}\mathbf{x}$ quickly

Trace Estimation

- Goal: Estimate trace of $d \times d$ matrix $\boldsymbol{A}$:

$$\mathrm{tr}(\boldsymbol{A}) = \sum_{i=1}^{d} \boldsymbol{A}_{ii} = \sum_{i=1}^{d} \lambda_i$$

- In Downstream Applications, $\boldsymbol{A}$ is not stored in memory.
- Instead, $\boldsymbol{B}$ is in memory and $\boldsymbol{A} = f(\boldsymbol{B})$:

No. Triangles	Estrada Index	Log-Determinant
$\mathrm{tr}(\frac{1}{6}\boldsymbol{B}^3)$	$\mathrm{tr}(e^{\boldsymbol{B}})$	$\mathrm{tr}(\ln(\boldsymbol{B}))$

- Computing $\boldsymbol{A} = \frac{1}{6}\boldsymbol{B}^3$ takes $O(n^3)$ time
- Computing $\boldsymbol{A}\mathbf{x} = \frac{1}{6}\boldsymbol{B}(\boldsymbol{B}(\boldsymbol{B}\mathbf{x}))$ takes $O(n^2)$ time
- If $\boldsymbol{A} = f(\boldsymbol{B})$, then we can often compute $\boldsymbol{A}\mathbf{x}$ quickly
- Goal: Estimate $\mathrm{tr}(\boldsymbol{A})$ by computing $\boldsymbol{A}\mathbf{x}_1, \ldots \boldsymbol{A}\mathbf{x}_k$

Formal Problem Statement

Implicit Matrix Trace Estimation: Estimate $\mathrm{tr}(\boldsymbol{A})$ with as few Matrix-Vector products $\boldsymbol{A}\mathbf{x}_1, \dots, \boldsymbol{A}\mathbf{x}_m$ as possible.

$$(1-\varepsilon)\,\mathrm{tr}(\boldsymbol{A}) \leq \tilde{\mathrm{tr}}(\boldsymbol{A}) \leq (1+\varepsilon)\,\mathrm{tr}(\boldsymbol{A}) \qquad \text{w.p. } 1-\delta$$

Our Contributions

For PSD matrix trace estimation,

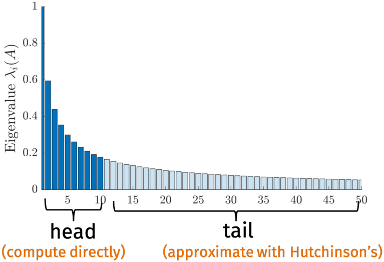
- Hutch++ algorithm, which uses $\tilde{O}(\frac{1}{\varepsilon})$ matrix-vector products.
  - Improves prior rate of $\tilde{O}(\frac{1}{\varepsilon^2})$
  - Empirically works well
  - Matching $\Omega(\frac{1}{\varepsilon})$ Lower Bound

Only 5 lines of code:

```
function T = hutchplusplus(A, m)
    S = 2*randi(2,size(A,1),m/3);
    G = 2*randi(2,size(A,1),m/3);
    [Q,~] = qr(A*S,0);
    G = G - Q*(Q'*G);
    T = trace(Q'*A*Q) + 1/size(G,2)*trace(G'*A*G);
end
```

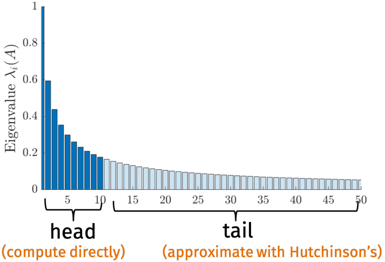
[1] $\tilde{O}$ notation only hide logarithmic dependence on the failure probability.

Hutch++ Intuition



Idea: Hutchinson's Estimator is **very** efficient unless $\boldsymbol{A}$ is almost low-rank.

Hutch++ Intuition



Idea: Hutchinson's Estimator is **very** efficient unless $\boldsymbol{A}$ is almost low-rank.

1. Find a good rank-k approximation $\tilde{\boldsymbol{A}}_k$
2. Compute $\tilde{T} \approx \mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k)$ with m steps of Hutchinson's
3. Return Hutch++$(\boldsymbol{A}) = \mathrm{tr}(\tilde{\boldsymbol{A}}_k) + \tilde{T}$

THANK YOU

Code available at

github.com/RaphaelArkadyMeyerNYU/hutchplusplus