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)

Collaborators

Christopher Musco
(NYU)

Cameron Musco
(UMass. Amherst)

David P. Woodruff
(CMU)

Implicit Trace Estimation

Basic problem in linear algebra:

- Given access to a $n \times n$ matrix $\boldsymbol{A}$ only through a Matrix-Vector Multiplication Oracle

$$\mathbf{x} \quad \xrightarrow{\textit{input}} \quad \text{ORACLE} \quad \xrightarrow{\textit{output}} \quad \boldsymbol{A}\mathbf{x}$$

- Goal is to approximate $\mathrm{tr}(\boldsymbol{A}) = \sum_{i=1}^{n} \boldsymbol{A}_{ii} = \sum_{i=1}^{n} \lambda_i$

Main Question: How many matrix-vector multiplication queries $\boldsymbol{A}\mathbf{x}_1, \ldots, \boldsymbol{A}\mathbf{x}_m$ are required to compute $\mathrm{tr}(\boldsymbol{A})$?[1]

[1] $\mathbf{x}_i$ can be chosen *adaptively*, based on the results $\boldsymbol{A}\mathbf{x}_1, \ldots, \boldsymbol{A}\mathbf{x}_{i-1}$

Background: Matrix-Vector Oracle

Application: Trace of a Function of a Matrix

- Suppose $\boldsymbol{B}$ is the adjacency matrix for graph G. Then $\frac{1}{6}\operatorname{tr}(\boldsymbol{B}^3)$ counts the number of triangles in G.
  - Computing $\boldsymbol{B}^3$ directly takes $O(n^3)$ time
  - Computing $\boldsymbol{B}^3\mathbf{x}$ takes $O(n^2)$ time

Background: Matrix-Vector Oracle

Application: Trace of a Function of a Matrix

- Suppose $\boldsymbol{B}$ is the adjacency matrix for graph G. Then $\frac{1}{6}\,\mathrm{tr}(\boldsymbol{B}^3)$ counts the number of triangles in G.
  - Computing $\boldsymbol{B}^3$ directly takes $O(n^3)$ time
  - Computing $\boldsymbol{B}^3\mathbf{x}$ takes $O(n^2)$ time
- Other functions of interest: $\mathrm{tr}(e^{\boldsymbol{B}})$, $\mathrm{tr}(\ln(\boldsymbol{\Sigma}))$, etc.

Application: Trace of a Function of a Matrix

- Suppose $\boldsymbol{B}$ is the adjacency matrix for graph G. Then $\frac{1}{6}\operatorname{tr}(\boldsymbol{B}^3)$ counts the number of triangles in G.
  - Computing $\boldsymbol{B}^3$ directly takes $O(n^3)$ time
  - Computing $\boldsymbol{B}^3\mathbf{x}$ takes $O(n^2)$ time
- Other functions of interest: $\operatorname{tr}(e^{\boldsymbol{B}})$, $\operatorname{tr}(\ln(\boldsymbol{\Sigma}))$, etc.
- Computing $f(\boldsymbol{B})\mathbf{x}$ is often much faster than computing $f(\boldsymbol{B})$ directly
  - Especially if we only need very few $\mathbf{x}$ vectors

Background: Matrix-Vector Oracle

Algorithms:

- Krylov Methods, Sketching Methods, Streaming Methods, etc.
- See also: *Implicit Matrix Methods*, *Matrix-Free Methods*
- Useful framework for algorithmic lower bounds
  - Allows us to prove optimality in a very general setting

Background: Hutchinson's Estimator

The classical approach to trace estimation:

Hutchinson 1991, Girard 1987

1. Draw $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$ with i.i.d. uniform $\{+1,\text{-}1\}$ entries
2. Return $\tilde{T} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i^\intercal \boldsymbol{A} \mathbf{x}_i$

Avron, Toledo 2011, Roosta, Ascher 2015

If $m = O(\frac{\log(1/\delta)}{\varepsilon^2})$, then with probability $1 - \delta$,

$$|\tilde{T} - \text{tr}(\boldsymbol{A})| \leq \varepsilon \|\boldsymbol{A}\|_F$$

Background: Hutchinson's Estimator

The classical approach to trace estimation:

Hutchinson 1991, Girard 1987

1. Draw $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$ with i.i.d. uniform $\{+1,\text{-}1\}$ entries
2. Return $\tilde{T} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i^\top \boldsymbol{A} \mathbf{x}_i$

Avron, Toledo 2011, Roosta, Ascher 2015

If $m = O(\frac{\log(1/\delta)}{\varepsilon^2})$, then with probability $1 - \delta$,

$$|\tilde{T} - \mathrm{tr}(\boldsymbol{A})| \leq \varepsilon \|\boldsymbol{A}\|_F$$

- If $\boldsymbol{A}$ is PSD, then $\|\boldsymbol{A}\|_F \leq \mathrm{tr}(\boldsymbol{A})$, so that

$$(1 - \varepsilon)\,\mathrm{tr}(\boldsymbol{A}) \leq \tilde{T} \leq (1 + \varepsilon)\,\mathrm{tr}(\boldsymbol{A})$$

Contribution: $O(1/\varepsilon)$ vectors is optimal

Theorems

1. For PSD $\boldsymbol{A}$ and $m = O(\frac{\log(1/\delta)}{\varepsilon})$, with probability $1-\delta$,

$$(1-\varepsilon)\,\mathrm{tr}(\boldsymbol{A}) \le \text{Hutch++}(\boldsymbol{A}) \le (1+\varepsilon)\,\mathrm{tr}(\boldsymbol{A})$$

```
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
```

For the rest of the talk, $\boldsymbol{A}$ is always PSD.

Contribution: $O(1/\varepsilon)$ vectors is optimal

Theorems

1. For PSD $\boldsymbol{A}$ and $m = O(\frac{\log(1/\delta)}{\varepsilon})$, with probability $1-\delta$,

$$(1-\varepsilon)\,\mathrm{tr}(\boldsymbol{A}) \leq \text{Hutch++}(\boldsymbol{A}) \leq (1+\varepsilon)\,\mathrm{tr}(\boldsymbol{A})$$

2. For any b-bit precision oracle, $\tilde{\Omega}(\frac{1}{\varepsilon b})$ possibly adaptive queries are necessary.

```
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
```

For the rest of the talk, $\boldsymbol{A}$ is always PSD.

Contribution: $O(1/\varepsilon)$ vectors is optimal

Theorems

1. For PSD $\boldsymbol{A}$ and $m = O(\frac{\log(1/\delta)}{\varepsilon})$, with probability $1 - \delta$,

$$(1-\varepsilon)\operatorname{tr}(\boldsymbol{A}) \le \text{Hutch++}(\boldsymbol{A}) \le (1+\varepsilon)\operatorname{tr}(\boldsymbol{A})$$

2. For any b-bit precision oracle, $\tilde{\Omega}(\frac{1}{\varepsilon b})$ possibly adaptive queries are necessary.
3. For any infinite precision oracle, $\Omega(\frac{1}{\varepsilon})$ non-adaptive queries are necessary.

```
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
```

For the rest of the talk, $\boldsymbol{A}$ is always PSD.

Hutchinson's Estimator
Versus the Top Few Eigenvalues

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$|\tilde{T} - \mathrm{tr}(\boldsymbol{A})| \leq O\left(\frac{1}{\sqrt{m}}\right)\|\boldsymbol{A}\|_F$$

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$|\tilde{T} - \operatorname{tr}(\boldsymbol{A})| \approx O\left(\frac{1}{\sqrt{m}}\right)\|\boldsymbol{A}\|_F$$

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$
\begin{aligned}
|\tilde{T} - \operatorname{tr}(\boldsymbol{A})| &\approx O\left(\tfrac{1}{\sqrt{m}}\right)\|\boldsymbol{A}\|_F \\
&\leq O\left(\tfrac{1}{\sqrt{m}}\right)\operatorname{tr}(\boldsymbol{A})
\end{aligned}
$$

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$
\begin{aligned}
|\tilde{T} - \operatorname{tr}(\boldsymbol{A})| &\approx O\left(\frac{1}{\sqrt{m}}\right)\|\boldsymbol{A}\|_F \\
&\leq O\left(\frac{1}{\sqrt{m}}\right)\operatorname{tr}(\boldsymbol{A}) \\
&= \varepsilon \operatorname{tr}(\boldsymbol{A})
\end{aligned}
$$

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$\begin{aligned} |\tilde{T} - \operatorname{tr}(\boldsymbol{A})| &\approx O\left(\tfrac{1}{\sqrt{m}}\right)\|\boldsymbol{A}\|_F \\ &\le O\left(\tfrac{1}{\sqrt{m}}\right)\operatorname{tr}(\boldsymbol{A}) \\ &= \varepsilon \operatorname{tr}(\boldsymbol{A}) \end{aligned}$$

- When does Hutchinson's Estimator truly need $O(\frac{1}{\varepsilon^2})$ queries?

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$
\begin{aligned}
|\tilde{T} - \mathrm{tr}(\boldsymbol{A})| &\approx O\left(\frac{1}{\sqrt{m}}\right)\|\boldsymbol{A}\|_F \\
&\leq O\left(\frac{1}{\sqrt{m}}\right)\mathrm{tr}(\boldsymbol{A}) \\
&= \varepsilon\, \mathrm{tr}(\boldsymbol{A})
\end{aligned}
$$

- When does Hutchinson's Estimator truly need $O(\frac{1}{\varepsilon^2})$ queries?
- When is the bound $\|\boldsymbol{A}\|_F \leq \mathrm{tr}(A)$ tight?

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$
\begin{aligned}
|\tilde{T} - \operatorname{tr}(\boldsymbol{A})| &\approx O(\tfrac{1}{\sqrt{m}})\|\boldsymbol{A}\|_F \\
&\leq O(\tfrac{1}{\sqrt{m}})\operatorname{tr}(\boldsymbol{A}) \\
&= \varepsilon \operatorname{tr}(\boldsymbol{A})
\end{aligned}
$$

- When does Hutchinson's Estimator truly need $O(\frac{1}{\varepsilon^2})$ queries?
- When is the bound $\|\boldsymbol{A}\|_F \leq \operatorname{tr}(A)$ tight?
- Let $\mathbf{v} = \begin{bmatrix} \lambda_1 & \dots & \lambda_n \end{bmatrix}$ be the eigenvalues of PSD $\boldsymbol{A}$

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$
\begin{aligned}
|\tilde{T} - \operatorname{tr}(\boldsymbol{A})| &\approx O\left(\tfrac{1}{\sqrt{m}}\right)\|\boldsymbol{A}\|_F \\
&\leq O\left(\tfrac{1}{\sqrt{m}}\right)\operatorname{tr}(\boldsymbol{A}) \\
&= \varepsilon \operatorname{tr}(\boldsymbol{A})
\end{aligned}
$$

- When does Hutchinson's Estimator truly need $O(\frac{1}{\varepsilon^2})$ queries?
- When is the bound $\|\boldsymbol{A}\|_F \leq \operatorname{tr}(A)$ tight?
- Let $\mathbf{v} = \begin{bmatrix} \lambda_1 & \dots & \lambda_n \end{bmatrix}$ be the eigenvalues of PSD $\boldsymbol{A}$
- When is the bound $\|\mathbf{v}\|_2 \leq \|\mathbf{v}\|_1$ tight?

Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$
\begin{aligned}
|\tilde{T} - \operatorname{tr}(\boldsymbol{A})| &\approx O(\tfrac{1}{\sqrt{m}}) \|\boldsymbol{A}\|_F \\
&\leq O(\tfrac{1}{\sqrt{m}}) \operatorname{tr}(\boldsymbol{A}) \\
&= \varepsilon \operatorname{tr}(\boldsymbol{A})
\end{aligned}
$$

- When does Hutchinson's Estimator truly need $O(\frac{1}{\varepsilon^2})$ queries?
- When is the bound $\|\boldsymbol{A}\|_F \leq \operatorname{tr}(A)$ tight?
- Let $\mathbf{v} = \begin{bmatrix} \lambda_1 & \dots & \lambda_n \end{bmatrix}$ be the eigenvalues of PSD $\boldsymbol{A}$
- When is the bound $\|\mathbf{v}\|_2 \leq \|\mathbf{v}\|_1$ tight?
  - Property of norms: $\|\mathbf{v}\|_2 \approx \|\mathbf{v}\|_1$ only if $\mathbf{v}$ is nearly sparse

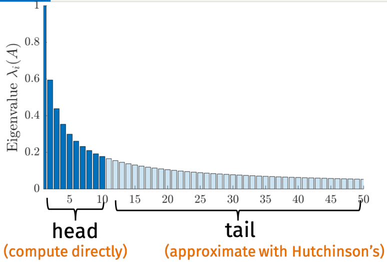
Hutchinson Analysis

Let's return to the result for Hutchinson's Estimator:

$$
\begin{aligned}
|\tilde{T} - \operatorname{tr}(\boldsymbol{A})| &\approx O\left(\tfrac{1}{\sqrt{m}}\right)\|\boldsymbol{A}\|_F \\
&\leq O\left(\tfrac{1}{\sqrt{m}}\right)\operatorname{tr}(\boldsymbol{A}) \\
&= \varepsilon \operatorname{tr}(\boldsymbol{A})
\end{aligned}
$$

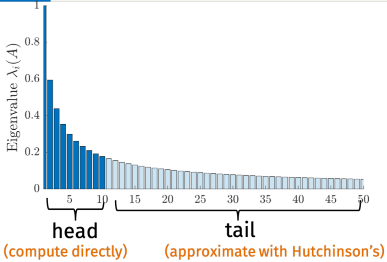
- When does Hutchinson's Estimator truly need $O(\frac{1}{\varepsilon^2})$ queries?
- When is the bound $\|\boldsymbol{A}\|_F \leq \operatorname{tr}(A)$ tight?
- Let $\mathbf{v} = \begin{bmatrix} \lambda_1 & \dots & \lambda_n \end{bmatrix}$ be the eigenvalues of PSD $\boldsymbol{A}$
- When is the bound $\|\mathbf{v}\|_2 \leq \|\mathbf{v}\|_1$ tight?
  - Property of norms: $\|\mathbf{v}\|_2 \approx \|\mathbf{v}\|_1$ only if $\mathbf{v}$ is nearly sparse
- Hutchinson only requires $O(\frac{1}{\varepsilon^2})$ queries if $\boldsymbol{A}$ has a few large eigenvalues

Helping Hutchinson's Estimator



Idea: Explicitly estimate the top few eigenvalues of $\boldsymbol{A}$. Use Hutchinson's for the rest.

Helping Hutchinson's Estimator



Idea: Explicitly estimate the top few eigenvalues of $\boldsymbol{A}$. Use Hutchinson's for the rest.

1. Find a good rank-k approximation $\tilde{\boldsymbol{A}}_k$
2. Notice that $\text{tr}(\boldsymbol{A}) = \text{tr}(\tilde{\boldsymbol{A}}_k) + \text{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k)$
3. Compute $\text{tr}(\tilde{\boldsymbol{A}}_k)$ exactly
4. Compute $\tilde{T} \approx \text{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k)$ with Hutchinson's Estimator
5. Return Hutch++$(\boldsymbol{A}) = \text{tr}(\tilde{\boldsymbol{A}}_k) + \tilde{T}$

Finding a Good Low-Rank Approximation

Let $\boldsymbol{A}_k$ be the best rank-k approximation of $\boldsymbol{A}$.

Lemma (Sarlos 2006, Woodruff 2014)

Let $\boldsymbol{S} \in \mathbb{R}^{n \times m}$ have i.i.d. uniform ± 1 entries, $\boldsymbol{Q} = \text{orth}(\boldsymbol{AS})$, and $\tilde{\boldsymbol{A}}_k = \boldsymbol{AQQ}^\intercal$. Then, with probability $1 - \delta$,

$$\|\boldsymbol{A} - \tilde{\boldsymbol{A}}_k\|_F \leq 2\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$$

so long as $\boldsymbol{S}$ has $m = O(k + \log(1/\delta))$ columns.

Let $\boldsymbol{A}_k$ be the best rank-k approximation of $\boldsymbol{A}$.

Lemma (Sarlos 2006, Woodruff 2014)

Let $\boldsymbol{S} \in \mathbb{R}^{n \times m}$ have i.i.d. uniform ± 1 entries, $\boldsymbol{Q} = \mathrm{orth}(\boldsymbol{AS})$, and $\tilde{\boldsymbol{A}}_k = \boldsymbol{AQQ}^\intercal$. Then, with probability $1 - \delta$,

$$\|\boldsymbol{A} - \tilde{\boldsymbol{A}}_k\|_F \leq 2\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$$

so long as $\boldsymbol{S}$ has $m = O(k + \log(1/\delta))$ columns.

We can compute the trace of $\tilde{\boldsymbol{A}}_k$ with m queries and $O(mn)$ space:

$$\mathrm{tr}(\tilde{\boldsymbol{A}}_k) = \mathrm{tr}(\boldsymbol{AQQ}^\intercal) = \mathrm{tr}(\boldsymbol{Q}^\intercal(\boldsymbol{AQ}))$$

Complete Analysis

Lemma: $\|\boldsymbol{A} - \boldsymbol{A}_k\|_F \leq \frac{1}{\sqrt{k}} \operatorname{tr}(\boldsymbol{A})$

Proof. Note that $\lambda_{k+1} \leq \frac{1}{k} \sum_{i=1}^{k} \lambda_i \leq \frac{1}{k} \operatorname{tr}(\boldsymbol{A})$.

Complete Analysis

Lemma: $\|\boldsymbol{A} - \boldsymbol{A}_k\|_F \leq \frac{1}{\sqrt{k}} \operatorname{tr}(\boldsymbol{A})$

Proof. Note that $\lambda_{k+1} \leq \frac{1}{k} \sum_{i=1}^{k} \lambda_i \leq \frac{1}{k} \operatorname{tr}(\boldsymbol{A})$. Then,

$$\|\boldsymbol{A} - \boldsymbol{A}_k\|_F^2 = \sum_{i=k+1}^{n} \lambda_i^2 \leq \lambda_{k+1} \sum_{i=k+1}^{n} \lambda_i \leq \left(\frac{1}{k} \operatorname{tr}(\boldsymbol{A})\right) \cdot \operatorname{tr}(\boldsymbol{A})$$

Complete Analysis

Lemma: $\|\boldsymbol{A} - \boldsymbol{A}_k\|_F \leq \frac{1}{\sqrt{k}} \operatorname{tr}(\boldsymbol{A})$

Proof. Note that $\lambda_{k+1} \leq \frac{1}{k} \sum_{i=1}^{k} \lambda_i \leq \frac{1}{k} \operatorname{tr}(\boldsymbol{A})$. Then,

$$\|\boldsymbol{A} - \boldsymbol{A}_k\|_F^2 = \sum_{i=k+1}^{n} \lambda_i^2 \leq \lambda_{k+1} \sum_{i=k+1}^{n} \lambda_i \leq \left(\tfrac{1}{k} \operatorname{tr}(\boldsymbol{A})\right) \cdot \operatorname{tr}(\boldsymbol{A})$$

- Formalizes our earlier intuition
- Replaces the earlier bound $\|\boldsymbol{A}\|_F \leq \operatorname{tr}(\boldsymbol{A})$
- Similar to standard compressed sensing result:

For all $\mathbf{v} \in \mathbb{R}^d$, *there exists k-sparse* $\tilde{\mathbf{v}}$ *such that*

$$\|\mathbf{v} - \tilde{\mathbf{v}}\|_2 \leq \frac{1}{\sqrt{k}} \|\mathbf{v}\|_1$$

Complete Analysis

Using rank-k approximation and ℓ sample for Hutchinson's.

Complete Analysis

Using rank-k approximation and ℓ sample for Hutchinson's.

1. We can only make an error in the Hutchinson's step:

$$|\mathrm{tr}(\boldsymbol{A}) - \text{Hutch++}(\boldsymbol{A})| = |\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k) - \tilde{T}|$$

Complete Analysis

Using rank-k approximation and ℓ sample for Hutchinson's.

1. We can only make an error in the Hutchinson's step:

$$|\text{tr}(\boldsymbol{A}) - \text{Hutch++}(\boldsymbol{A})| = |\text{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k) - \tilde{T}|$$

2. Guarantees for Hutchinson's and Low-Rank Approximation:

$$|\text{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k) - \tilde{T}| \leq O(\tfrac{1}{\sqrt{\ell}})\|\boldsymbol{A} - \tilde{\boldsymbol{A}}_k\|_F \leq O(\tfrac{1}{\sqrt{\ell}}) \cdot 2\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$$

Complete Analysis

Using rank-k approximation and ℓ sample for Hutchinson's.

1. We can only make an error in the Hutchinson's step:

$$|\mathrm{tr}(\boldsymbol{A}) - \text{Hutch++}(\boldsymbol{A})| = |\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k) - \tilde{T}|$$

2. Guarantees for Hutchinson's and Low-Rank Approximation:

$$|\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k) - \tilde{T}| \leq O(\tfrac{1}{\sqrt{\ell}})\|\boldsymbol{A} - \tilde{\boldsymbol{A}}_k\|_F \leq O(\tfrac{1}{\sqrt{\ell}}) \cdot 2\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$$

3. Use the lemma from the last slide:

$$|\mathrm{tr}(\boldsymbol{A}) - \text{Hutch++}(\boldsymbol{A})| \leq O(\tfrac{1}{\sqrt{k\ell}})\,\mathrm{tr}(\boldsymbol{A})$$

Complete Analysis

Using rank-k approximation and ℓ sample for Hutchinson's.

1. We can only make an error in the Hutchinson's step:

$$|\mathrm{tr}(\boldsymbol{A}) - \mathrm{Hutch{+}{+}}(\boldsymbol{A})| = |\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k) - \tilde{T}|$$

2. Guarantees for Hutchinson's and Low-Rank Approximation:

$$|\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k) - \tilde{T}| \leq O(\tfrac{1}{\sqrt{\ell}})\|\boldsymbol{A} - \tilde{\boldsymbol{A}}_k\|_F \leq O(\tfrac{1}{\sqrt{\ell}}) \cdot 2\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$$

3. Use the lemma from the last slide:

$$|\mathrm{tr}(\boldsymbol{A}) - \mathrm{Hutch{+}{+}}(\boldsymbol{A})| \leq O(\tfrac{1}{\sqrt{k\ell}})\,\mathrm{tr}(\boldsymbol{A})$$

4. If $k = \ell = O(\frac{1}{\varepsilon})$, then $|\mathrm{tr}(\boldsymbol{A}) - \mathrm{Hutch{+}{+}}(\boldsymbol{A})| \leq \varepsilon\,\mathrm{tr}(\boldsymbol{A})$ $\square$

Lower Bound: Communication Complexity

Communication Complexity

- Really rich area of theoretical computing

Gap-Hamming Problem

Let Alice and Bob each have vectors $\mathbf{s}, \mathbf{t} \in \{+1, -1\}^n$. Using as few bits of communication as possible, they must decide if

$$\langle \mathbf{s}, \mathbf{t} \rangle \geq \sqrt{n} \qquad \text{or if} \qquad \langle \mathbf{s}, \mathbf{t} \rangle \leq -\sqrt{n}$$

Chakrabarti, Regev 2012

Any (possibly adaptive) protocol between Alice and Bob must use $\Omega(n)$ bits to solve the Gap-Hamming problem with probability $\geq \frac{2}{3}$.

A Reduction from Gap-Hamming

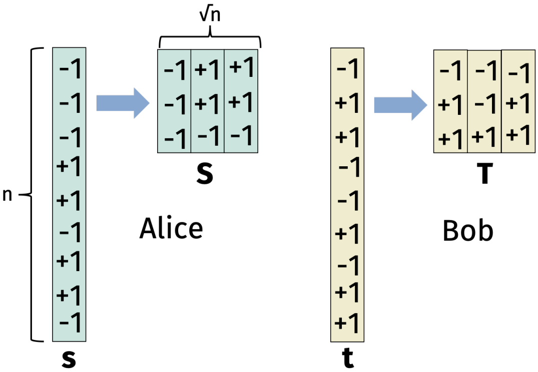
- Suppose the Matrix-Vector Oracle for $\boldsymbol{A}$ only accepts queries with entries that use b bits of precision
  - (e.g. the entries of $\mathbf{x}$ are integers between -2^b and 2^b).

Theorem

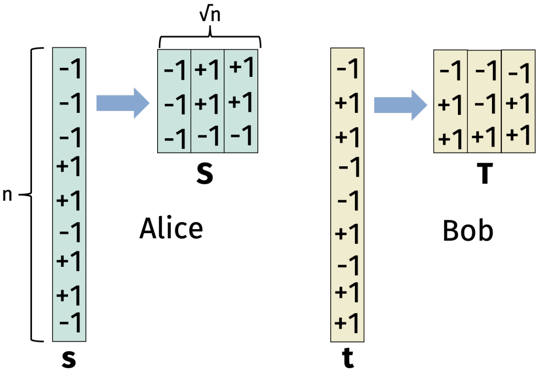
Any (possibly adaptive) algorithm that estimates $\mathrm{tr}(\boldsymbol{A})$ to relative error ε with probability $\geq \frac{2}{3}$ must use $\Omega(\frac{1}{\varepsilon(b+\log(1/\varepsilon))})$ queries.

Proof Idea: Simulate a m-query trace-estimation algorithm to solve a n-bit Gap-Hamming problem

A Reduction to Trace Estimation



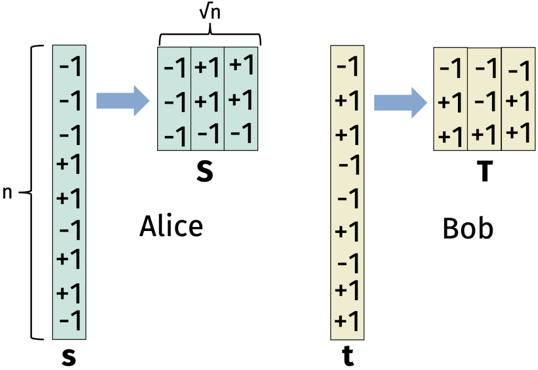
A Reduction to Trace Estimation



Let $\boldsymbol{Z} = \boldsymbol{S} + \boldsymbol{T}$ and $\boldsymbol{A} = \boldsymbol{Z}^T\boldsymbol{Z}$, so that

$$\mathrm{tr}(\boldsymbol{A}) = \|\boldsymbol{Z}\|_F^2 = \|\mathrm{s} + \mathrm{t}\|_2^2 = 2n - 2\langle \mathrm{s}, \mathrm{t} \rangle$$

A Reduction to Trace Estimation



Let $\boldsymbol{Z} = \boldsymbol{S} + \boldsymbol{T}$ and $\boldsymbol{A} = \boldsymbol{Z}^T\boldsymbol{Z}$, so that

$$\mathrm{tr}(\boldsymbol{A}) = \|\boldsymbol{Z}\|_F^2 = \|\mathrm{s} + \mathrm{t}\|_2^2 = 2n - 2\langle \mathrm{s}, \mathrm{t} \rangle$$

If Alice and Bob can estimate $\mathrm{tr}(\boldsymbol{A})$ to error $(1 \pm \frac{1}{\sqrt{n}})$, they can solve the Gap-Hamming problem (so $\varepsilon = \frac{1}{\sqrt{n}}$).

A Reduction to Trace Estimation

- For any precision b vector $\mathbf{x}$, Alice and Bob can compute $\boldsymbol{A}\mathbf{x}$ with $O(\sqrt{n}(\log(n) + b))$ bits of communication

A Reduction to Trace Estimation

- For any precision b vector $\mathbf{x}$, Alice and Bob can compute $\boldsymbol{A}\mathbf{x}$ with $O(\sqrt{n}(\log(n) + b))$ bits of communication
- They can simulate any m-query trace estimation algorithm with $O(m \cdot \sqrt{n}(\log(n) + b))$ bits of communication

A Reduction to Trace Estimation

- For any precision b vector $\mathbf{x}$, Alice and Bob can compute $\boldsymbol{A}\mathbf{x}$ with $O(\sqrt{n}(\log(n) + b))$ bits of communication
- They can simulate any m-query trace estimation algorithm with $O(m \cdot \sqrt{n}(\log(n) + b))$ bits of communication
- Gap-Hamming Lower bound: $m \geq \Omega(\frac{n}{\sqrt{n}(\log(n)+b)})$

A Reduction to Trace Estimation

- For any precision b vector $\mathbf{x}$, Alice and Bob can compute $\boldsymbol{A}\mathbf{x}$ with $O(\sqrt{n}(\log(n) + b))$ bits of communication
- They can simulate any m-query trace estimation algorithm with $O(m \cdot \sqrt{n}(\log(n) + b))$ bits of communication
- Gap-Hamming Lower bound: $m \geq \Omega(\frac{n}{\sqrt{n}(\log(n)+b)})$
- Substitute $\varepsilon = \frac{1}{\sqrt{n}}$: $m \geq \Omega(\frac{1}{\varepsilon(b+\log(1/\varepsilon))})$

Lower Bound:
Statistical Hypothesis Testing

Statistical Hypothesis Testing

Design distributions $\mathcal{P}_0$ and $\mathcal{P}_1$ over PSD matrices such that

1. A trace estimator can distinguish $\mathcal{P}_0$ from $\mathcal{P}_1$
   - If $\boldsymbol{A}_0 \sim \mathcal{P}_0$ and $\boldsymbol{A}_1 \sim \mathcal{P}_1$
   - With high probability, $\mathrm{tr}(\boldsymbol{A}_0) \leq (1-2\varepsilon)\,\mathrm{tr}(\boldsymbol{A}_1)$

Statistical Hypothesis Testing

Design distributions $\mathcal{P}_0$ and $\mathcal{P}_1$ over PSD matrices such that

1. A trace estimator can distinguish $\mathcal{P}_0$ from $\mathcal{P}_1$
   - If $\boldsymbol{A}_0 \sim \mathcal{P}_0$ and $\boldsymbol{A}_1 \sim \mathcal{P}_1$
   - With high probability, $\operatorname{tr}(\boldsymbol{A}_0) \leq (1-2\varepsilon)\operatorname{tr}(\boldsymbol{A}_1)$
2. No estimator can distinguish $\mathcal{P}_0$ from $\mathcal{P}_1$ with $\Omega(\frac{1}{\varepsilon})$ queries
   - Nature samples $i \sim \{0,1\}$, and $\boldsymbol{A} \sim \mathcal{P}_i$
   - Any estimator that correctly guesses i with probability $\geq \frac{3}{4}$ must use $\Omega(\frac{1}{\varepsilon})$ queries

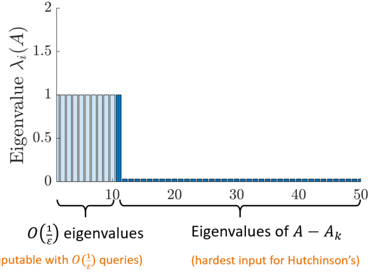
Statistical Hypothesis Testing

Design distributions $\mathcal{P}_0$ and $\mathcal{P}_1$ over PSD matrices such that

1. A trace estimator can distinguish $\mathcal{P}_0$ from $\mathcal{P}_1$
   - If $\boldsymbol{A}_0 \sim \mathcal{P}_0$ and $\boldsymbol{A}_1 \sim \mathcal{P}_1$
   - With high probability, $\operatorname{tr}(\boldsymbol{A}_0) \leq (1-2\varepsilon)\operatorname{tr}(\boldsymbol{A}_1)$
2. No estimator can distinguish $\mathcal{P}_0$ from $\mathcal{P}_1$ with $\Omega(\frac{1}{\varepsilon})$ queries
   - Nature samples $i \sim \{0,1\}$, and $\boldsymbol{A} \sim \mathcal{P}_i$
   - Any estimator that correctly guesses i with probability $\geq \frac{3}{4}$ must use $\Omega(\frac{1}{\varepsilon})$ queries

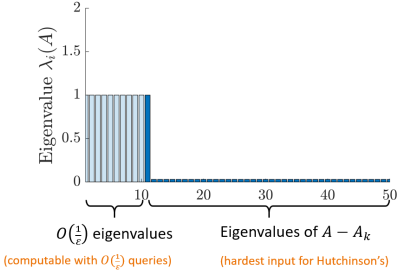
The design of $\mathcal{P}_0$ and $\mathcal{P}_1$ should reflect what structure makes trace estimation hard!

Designing a Hard Instance



What would the hardest input for Hutch++ be?

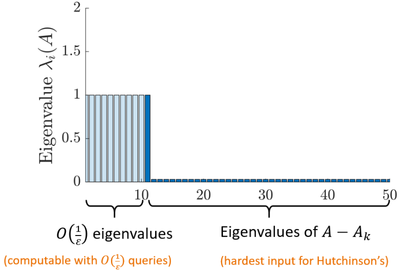
Designing a Hard Instance



What would the hardest input for Hutch++ be?

- Hutch++ only makes errors with Hutchinson's estimator on $\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k)$

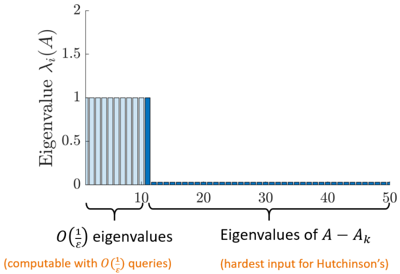
Designing a Hard Instance



What would the hardest input for Hutch++ be?

- Hutch++ only makes errors with Hutchinson's estimator on $\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k)$
- For what $\boldsymbol{A}$ would Hutchinson's estimator have difficulty estimating $\mathrm{tr}(\boldsymbol{A} - \boldsymbol{A}_k)$?

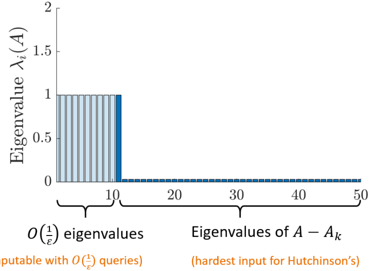
Designing a Hard Instance



What would the hardest input for Hutch++ be?

- Hutch++ only makes errors with Hutchinson's estimator on $\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k)$
- For what $\boldsymbol{A}$ would Hutchinson's estimator have difficulty estimating $\mathrm{tr}(\boldsymbol{A} - \boldsymbol{A}_k)$?
  - Hutchinson's estimator needs many samples when $\boldsymbol{A} - \boldsymbol{A}_k$ has concentrated eigenvalues

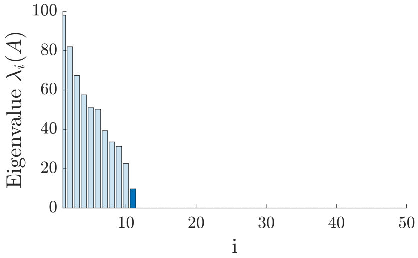
Designing a Hard Instance



What would the hardest input for Hutch++ be?

- Hutch++ only makes errors with Hutchinson's estimator on $\mathrm{tr}(\boldsymbol{A} - \tilde{\boldsymbol{A}}_k)$
- For what $\boldsymbol{A}$ would Hutchinson's estimator have difficulty estimating $\mathrm{tr}(\boldsymbol{A} - \boldsymbol{A}_k)$?
  - Hutchinson's estimator needs many samples when $\boldsymbol{A} - \boldsymbol{A}_k$ has concentrated eigenvalues
- $\boldsymbol{A}$ has $k = O(\frac{1}{\varepsilon})$ large eigenvalues. The rest are zero.

Designing a Hard Instance



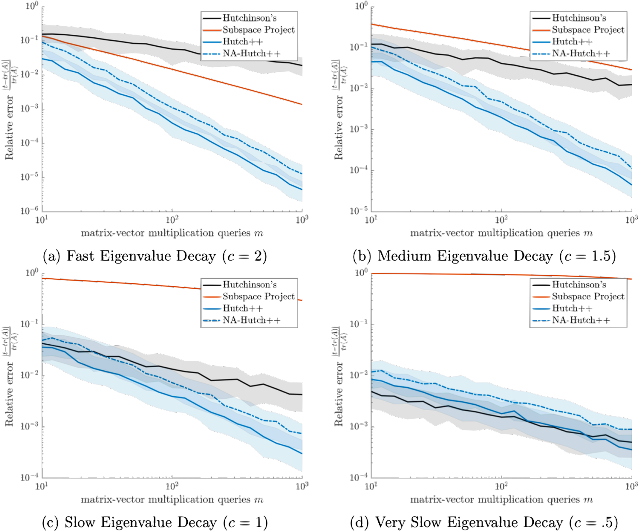
Formally, for large enough integer d,

$\mathcal{P}_0$	$\boldsymbol{A} = \boldsymbol{G}^T\boldsymbol{G}$	for	$\boldsymbol{G} \in \mathbb{R}^{d\times(\frac{1}{\varepsilon})}$	Gaussian
$\mathcal{P}_1$	$\boldsymbol{A} = \boldsymbol{G}^T\boldsymbol{G}$	for	$\boldsymbol{G} \in \mathbb{R}^{d\times(\frac{1}{\varepsilon}+1)}$	Gaussian

Experiments

Synthetic Experiments

Results on synthetic matrix $\boldsymbol{A}$ with spectrum $\lambda_i = i^{-c}$ for different values of c:



(a) Fast Eigenvalue Decay ($c = 2$)

(b) Medium Eigenvalue Decay ($c = 1.5$)

(c) Slow Eigenvalue Decay ($c = 1$)

(d) Very Slow Eigenvalue Decay ($c = .5$)

Non-PSD Experiments

Hutch++ works well empirically for many non-PSD matrices.

Let $\boldsymbol{B}$ be the (indefinite) adjacency matrix of an undirected graph G, $\frac{1}{6}\operatorname{tr}(\boldsymbol{B}^3)$ is exactly equal to the number of *triangles* in G.

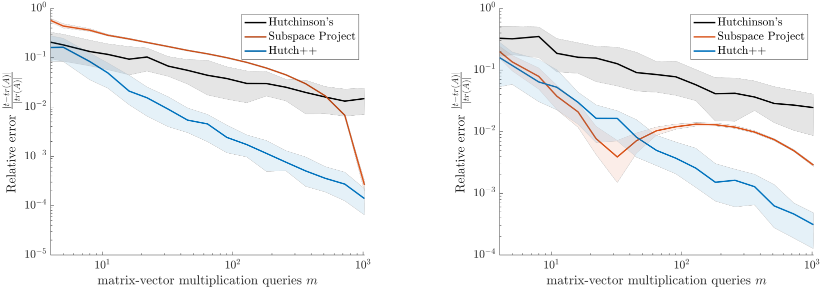


Figure: $\boldsymbol{A} = \boldsymbol{B}^3$ for arXiv.org citation network and Wikipedia voting network.

Open Questions

- **In progress:** Lower bounds for e.g. $\mathrm{tr}(\boldsymbol{A}^3)$, $\mathrm{tr}(e^{\boldsymbol{A}})$, $\mathrm{tr}(\boldsymbol{A}^{-1})$
- What about inexact oracles? We often approximate $f(\boldsymbol{A})\mathbf{x}$ with iterative methods. How accurate do these computations need to be?
- Extend to include row/column sampling? This would encapsulate e.g. SGD/SCD.

THANK
YOU

Code available at
github.com/RaphaelArkadyMeyerNYU/hutchplusplus