A (personally biased) introduction to randomized matrix computations

In honour of Olga Taussky-Todd

Ilse C.F. Ipsen

Based on joint work with
Jocelyn Chi, John Holodnak, Arnel Smith,
and Thomas Wentworth

North Carolina State University
Raleigh, NC, USA

Research supported by DOE, NSF-CCF, NSF-DMS

Olga Taussky-Todd

lived from 1906 to 1995

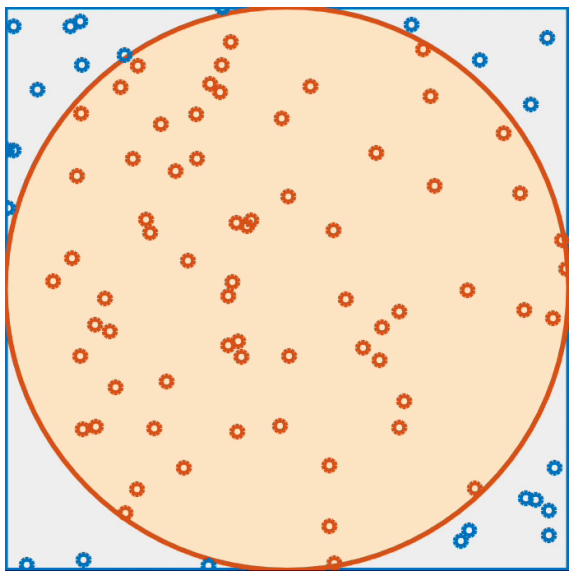
Algebraic number theory
Class field theory
Group theory
Matrix theory
Numerical analysis

https://mathshistory.st-andrews.ac.uk/Biographies/Taussky-Todd/

Randomized algorithms

Solution of a deterministic problem via statistical sampling

Example: Monte Carlo methods (von Neumann & Ulam, Los Alamos, 1946)



Play/Pause

$$\text{circle area} \;=\; 4\;\mathbb{E}\left[\frac{\#\text{hits}}{\#\text{darts}}\right]$$

Randomization for matrix computations

How to use it:

- Dimension reduction: Sample to solve a smaller subproblem
- Acceleration: Compute a 'preconditioner' via sampling
- Analysis: Iterative methods applied to random starting guesses

Advantages:

- Fast computation of approximate solutions
- Easy implementation
- Reduction of data movement (communication)
- Solution of large-scale problems
- `RandBLAS` and `RandLAPACK` under development[1]

[1] Murray, Demmel, Mahoney, . . ., Dongarra: *Randomized numerical linear algebra: A perspective on the field with an eye to software*

This talk

Caveat: This is not a comprehensive review

We focus on a few fundamental ideas, to convey insight

- What is sampling?
- Case study: Monte Carlo Gram matrix multiplication

  Why matrix multiplication???

  This is such a basic, old, worked-over problem

  Answer: It gives insight and is needed for the next case study
- Matrices with orthonormal columns

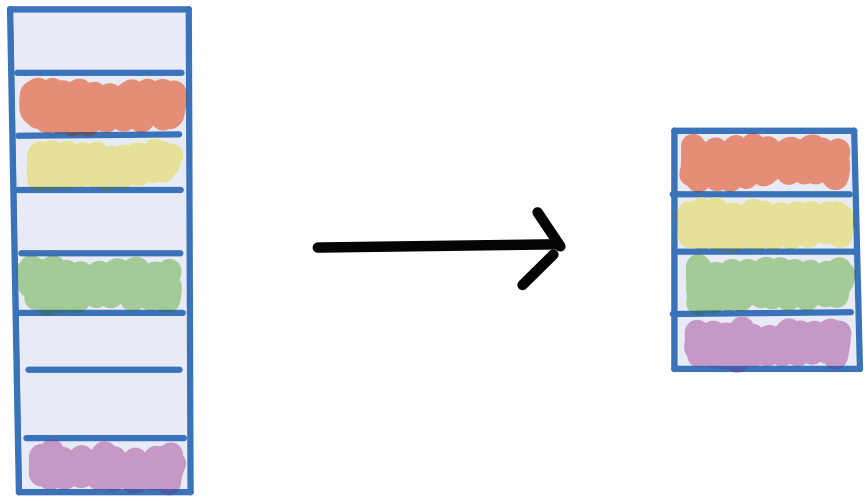
  Do we really need such a special case?

  Answer: Yes, we need this for the next case study
- Case study: Randomized least squares/regression problems
  1. Dimension reduction
  2. Convergence acceleration
- Summary and future research

What is Sampling?

Sampling rows from a matrix

Sampling rows = picking rows from a matrix



Dimension reduction: Sampled matrix has fewer rows

Data-oblivious sampling

- Matrix $\boldsymbol{S}$ samples, uniformly and independently, c rows from identity matrix

$$\boldsymbol{I}_m = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_1^T \\ \vdots \\ \boldsymbol{e}_m^T \end{bmatrix} \in \mathbb{R}^{m \times m} \qquad \boldsymbol{S} = \sqrt{\tfrac{m}{c}} \begin{bmatrix} \boldsymbol{e}_{k_1}^T \\ \vdots \\ \boldsymbol{e}_{k_c}^T \end{bmatrix} \in \mathbb{R}^{c \times m}$$

In expectation: $\mathbb{E}[\boldsymbol{S}^T \boldsymbol{S}] = \boldsymbol{I}_m$

- $\boldsymbol{SA}$ samples, uniformly and independently, c rows from $\boldsymbol{A}$

$$\boldsymbol{SA} = \sqrt{\tfrac{m}{c}} \begin{bmatrix} \boldsymbol{e}_{k_1}^T \boldsymbol{A} \\ \vdots \\ \boldsymbol{e}_{k_c}^T \boldsymbol{A} \end{bmatrix}$$

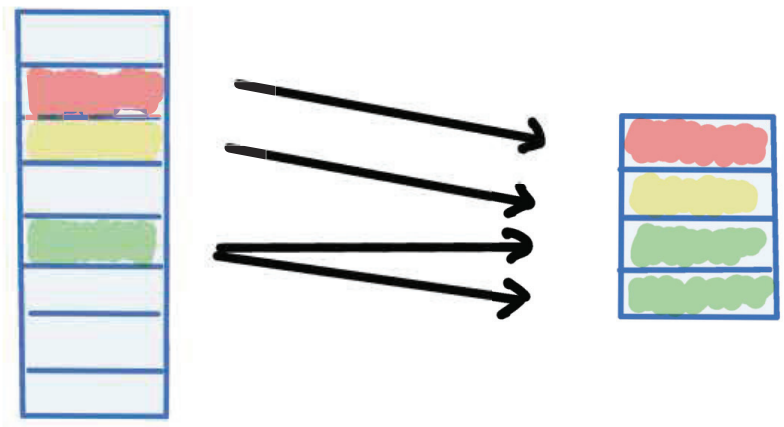
In expectation: $\mathbb{E}[(\boldsymbol{SA})^T(\boldsymbol{SA})] = \boldsymbol{A}^T\boldsymbol{A}$

$\{(\boldsymbol{SA})^T(\boldsymbol{SA})$ is unbiased estimator of $\boldsymbol{A}^T\boldsymbol{A}\}$

Uniform sampling, with replacement of c indices from $\{1, \ldots, m\}$

Repeat c times:
Sample k_t from $\{1, \ldots, m\}$ with probability $\frac{1}{m}$

Example: $m = 8$, $c = 4$



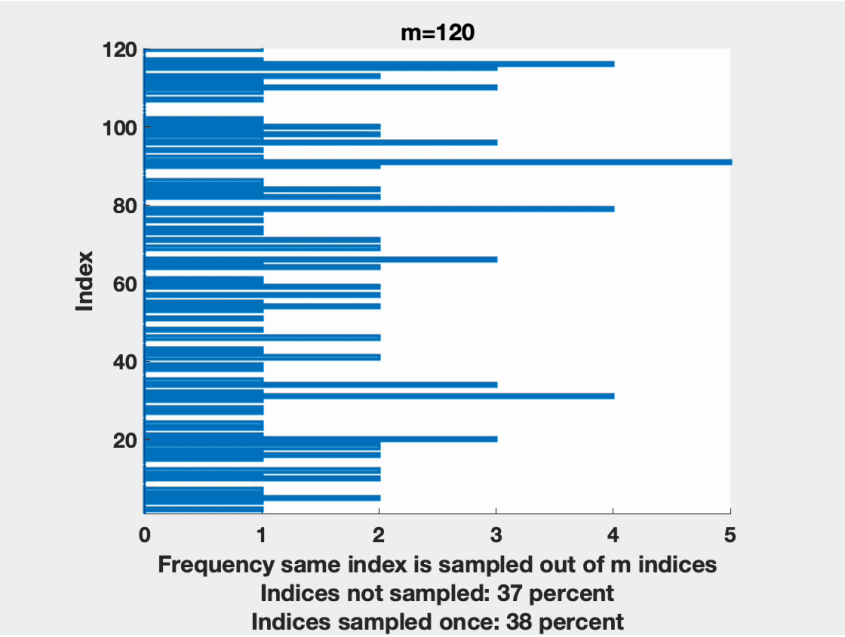
Implementation [Devroye]

$v =$ `rand` {uniform $[0, 1]$ random variable}
$k_t = \lfloor 1 + m v \rfloor$

Matlab: `randi`, `datasample` Julia, R: `sample` Python: `random.choice`

Uniform sampling with replacement, in action

Repeat m times:
    Sample index from $\{1, \ldots, m\}$ with probability $1/m$



Play/Pause

$\mathbb{E}[\#\text{ samples required to pick each index}] = m\,H_m = m \ln m + \mathcal{O}(m)$

Summary: Uniform sampling with replacement

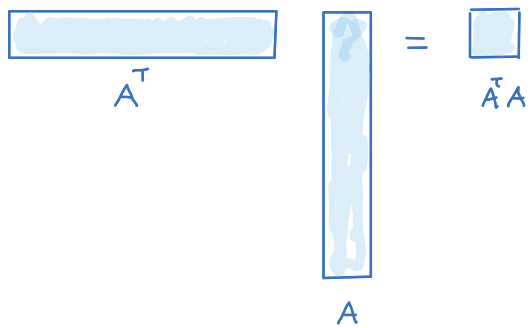
+ Simple implementation of data-oblivious sampling
+ Convenient probabilistic analysis (samples are independent)
– Repeated indices and rows
+ For practical (small to moderate) amounts of sampling:
  sampling with replacement as good as
  sampling without replacement

Case Study:
Monte Carlo Gram Matrix Multiplication

Idea behind Monte Carlo Matrix Multiplication

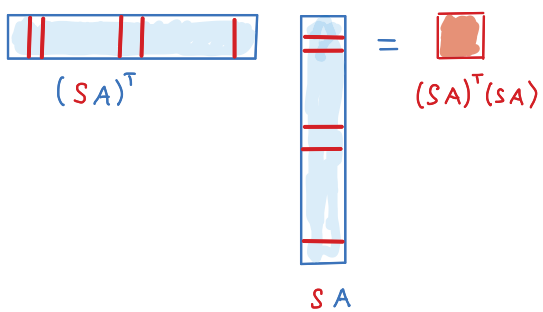
Given: Tall and skinny matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$

Instead of computing the Gram matrix $\boldsymbol{A}^T \boldsymbol{A}$



Do a dimension reduction:

Sample a few rows from $\boldsymbol{A}$, and compute an approximation



Monte Carlo Gram Matrix Multiplication

[Drineas, Kannan, Mahoney]

Input: $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $c \geq 1$ {sampling amount}

Output: $(\boldsymbol{SA})^T(\boldsymbol{SA}) \approx \boldsymbol{A}^T\boldsymbol{A}$

$\boldsymbol{S} = 0_{c \times m}$ {Initialize sampling matrix}

for $t = 1 : c$ do

  Sample k_t uniformly with replacement from $\{1, \ldots, m\}$

  $\boldsymbol{S}(t,:) = \sqrt{\frac{m}{c}}\boldsymbol{e}_{k_t}^T$ {row t of sampling matrix}

end for

$\boldsymbol{X} = \boldsymbol{SA}$ {Sample rows from $\boldsymbol{A}$}

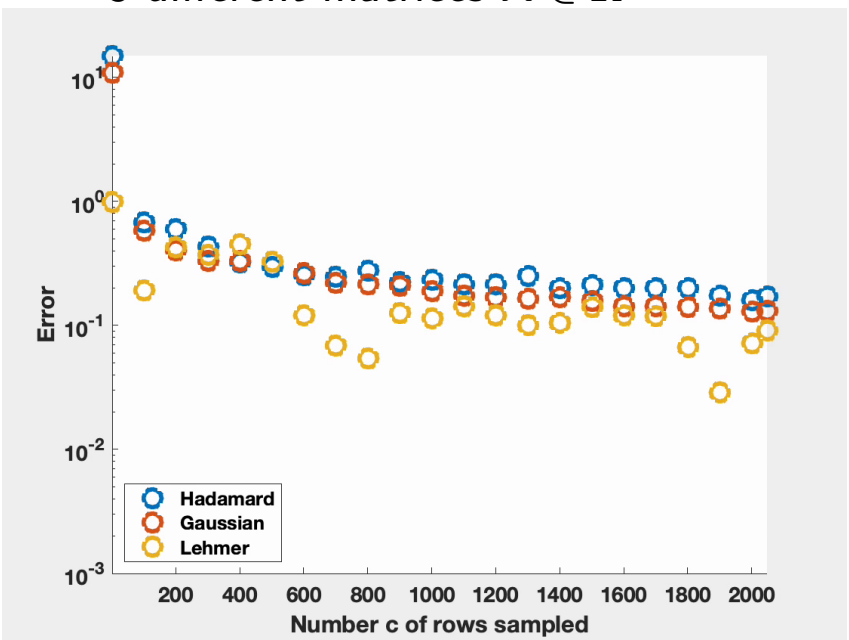
Return $\boldsymbol{X}^T\boldsymbol{X}$

Relative error: $\frac{\|(\boldsymbol{SA})^T(\boldsymbol{SA}) - \boldsymbol{A}^T\boldsymbol{A}\|_2}{\|\boldsymbol{A}^T\boldsymbol{A}\|_2}$ {in the spectral/two-norm}

What kind of accuracy to expect from Monte Carlo matrix multiplication

Relative error $\frac{\|(\boldsymbol{SA})^T(\boldsymbol{SA}) - \boldsymbol{A}^T\boldsymbol{A}\|_2}{\|\boldsymbol{A}^T\boldsymbol{A}\|_2}$ versus number of rows of $\boldsymbol{SA}$

3 different matrices $\boldsymbol{A} \in \mathbb{R}^{2048\times 16}$



Play/Pause

Error due to randomization for Monte Carlo Gram matrix multiplication

[Holodnak, Ipsen], [Ipsen, Smith]

Largest (squared) normalized row-norm of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$

$$\mu \equiv \max_{1 \leq i \leq m} \left(\frac{\|\boldsymbol{e}_i^T \boldsymbol{A}\|_2}{\|\boldsymbol{A}\|_2} \right)^2$$

Matrix $\boldsymbol{S}$ samples c rows uniformly with replacement

For any $0 < \delta < 1$ with probability at least $1 - \delta$

$$\frac{\|(\boldsymbol{SA})^T(\boldsymbol{SA}) - \boldsymbol{A}^T\boldsymbol{A}\|_2}{\|\boldsymbol{A}^T\boldsymbol{A}\|_2} \leq \tau + \sqrt{\tau\,(6+\tau)}$$

where

$$\tau \equiv \frac{m}{c}\,\mu\,\frac{\ln(n/\delta)}{3}$$

Interpretation of error bound

For any $0 < \delta < 1$ with probability at least $1 - \delta$

$$\underbrace{\frac{\|(\boldsymbol{SA})^T(\boldsymbol{SA}) - \boldsymbol{A}^T\boldsymbol{A}\|_2}{\|\boldsymbol{A}^T\boldsymbol{A}\|_2}}_{\text{Relative deviation from expectation}} \leq \tau + \sqrt{\tau\,(6+\tau)}$$

where

$$\tau \equiv \frac{m}{c}\,\mu\,\frac{\ln(n/\delta)}{3}$$

$\frac{m}{c}$: inverse proportion of # rows sampled

$\mu = \max_{1 \leq i \leq m} \left(\frac{\|\boldsymbol{e}_i^T \boldsymbol{A}\|_2}{\|\boldsymbol{A}\|_2} \right)^2$: 'distribution of mass' in the matrix

Foundation for the proof: Matrix Bernstein concentration inequality

[Recht], [Tropp]

Given

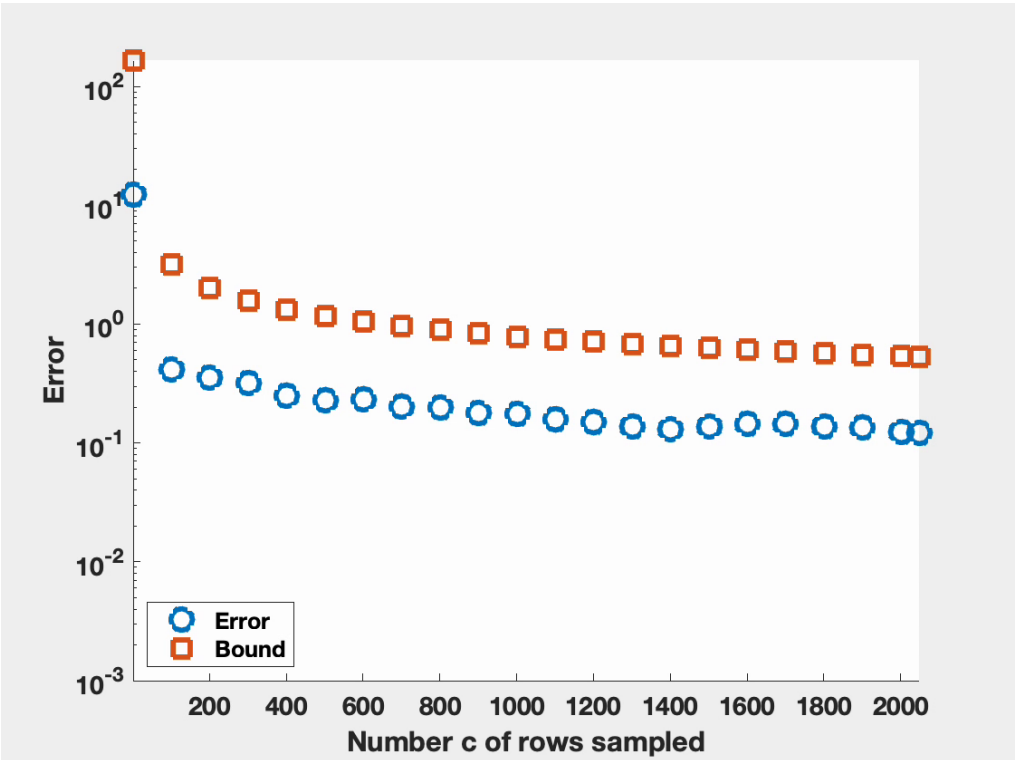
- Independent matrix-valued random variables $\boldsymbol{X}_j \in \mathbb{R}^{n \times n}$
- Symmetric: $\boldsymbol{X}_j^T = \boldsymbol{X}_j$
- Bounded norms: $\max_j \|\boldsymbol{X}_j\|_2 \leq \beta$
- Zero expectation: $\mathbb{E}[\boldsymbol{X}_j] = 0$
- Bounded 'variance': $\left\| \sum_j \mathbb{E}[X_j^2] \right\|_2 \leq v$

Then for any $\epsilon \geq 0$

$$\mathbb{P}\left[\left\| \sum_j \boldsymbol{X}_j \right\|_2 \geq \epsilon \right] \leq n \exp\left(-\frac{3\,\epsilon^2}{6\,v + 2\,\beta\,\epsilon} \right)$$

Error bound in action

Relative error $\frac{\|(\boldsymbol{SA})^T(\boldsymbol{SA})-\boldsymbol{A}^T\boldsymbol{A}\|_2}{\|\boldsymbol{A}^T\boldsymbol{A}\|_2}$ versus number of rows of $\boldsymbol{SA}$



Play/Pause

$\boldsymbol{A} \in \mathbb{R}^{2048\times 10}$ uniform $(0,1)$, $\kappa(\boldsymbol{A}) = 1.13$, $\mu = .0137$

Summary: Monte Carlo Gram matrix multiplication

– Low accuracy (1 digit for small matrix dimensions)

– More sampling does not help:
Error decreases slowly with increasing sampling amounts

– Appropriate for tall and skinny matrices
(many rows, few columns)

\+ Error bound reflects qualitative behaviour of error

\+ Error bound informative even for matrices of small dimension

– This is a risky approach for randomization:
Sampling produces loss of information and accuracy

Matrices with Orthonormal Columns

$$\boldsymbol{Q} \in \mathbb{R}^{m \times n} \text{ with } \boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}_n$$

Sampling rows from matrices with orthonormal columns

$\boldsymbol{Q} \in \mathbb{R}^{8\times 2}$ with $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_2$, $\quad \mu \equiv \max_{1\le i\le 8} \|\boldsymbol{e}_i^T\boldsymbol{Q}\|_2^2$

Sample 2 rows of $\boldsymbol{Q}$ so that $\operatorname{rank}(\boldsymbol{S}\boldsymbol{Q}) = 2$

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \boldsymbol{Q} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$\mu = 1$ $\qquad$ $\mu = \frac{1}{4}$

Sampling is hard $\qquad$ Sampling is easy

Largest row norm μ of $\boldsymbol{Q}$ quantifies difficulty of sampling

Coherence = Largest row norm of matrix with orthonormal columns

Coherence of $\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ with $\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}$ $\quad \{\text{orthonormal columns}\}$

$$\mu \equiv \max_{1 \le i \le m} \|\boldsymbol{e}_j^T \boldsymbol{Q}\|_2^2$$

- $n/m \le \mu(\boldsymbol{Q}) \le 1$
- Maximal coherence: $\mu(\boldsymbol{Q}) = 1$
  At least one column of $\boldsymbol{Q}$ is column of identity
- Minimal coherence: $\mu(\boldsymbol{Q}) = n/m$
  Columns of $\boldsymbol{Q}$ are columns of Hadamard matrix

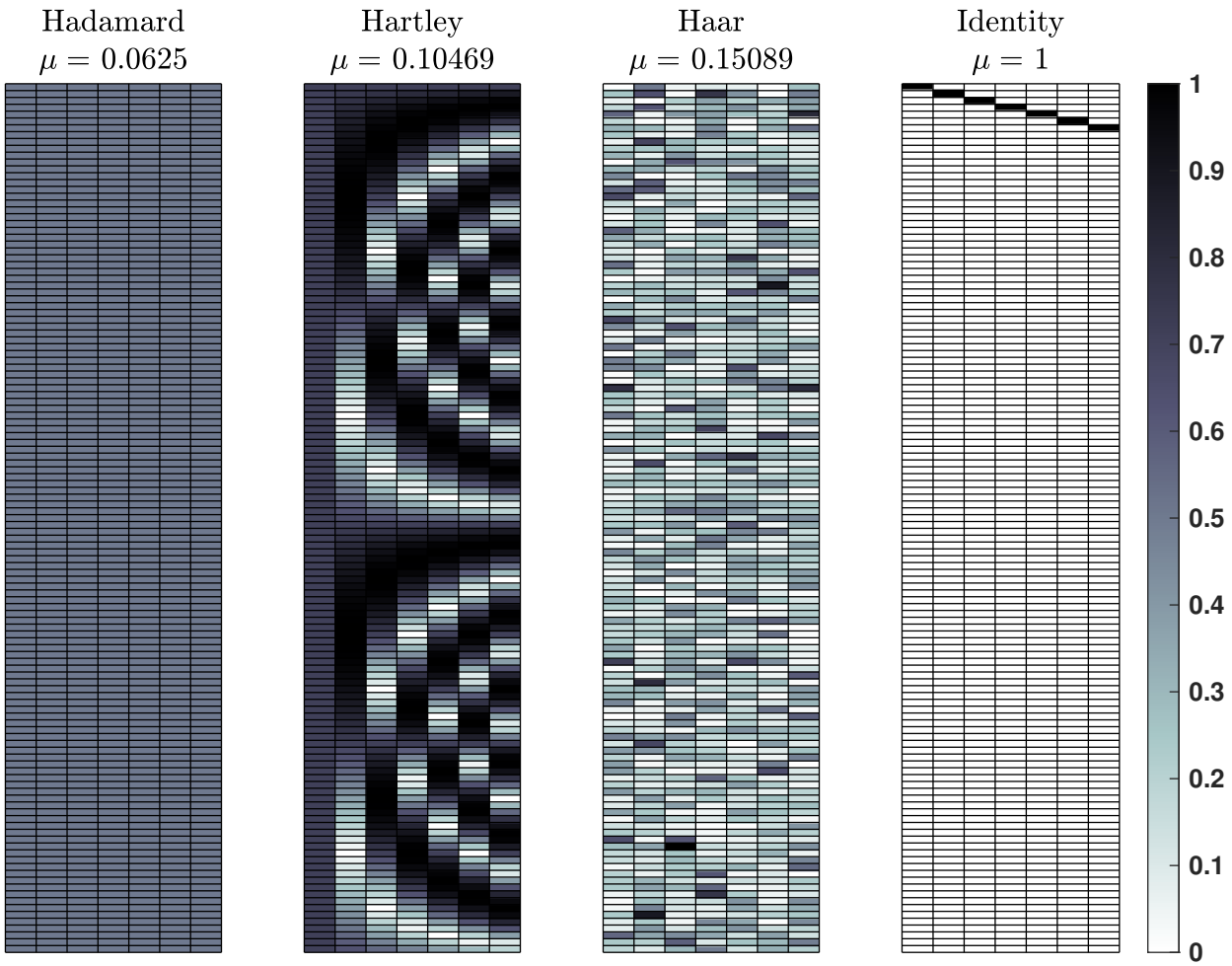
Coherence [Donoho, Huo]

- Measures correlation with standard basis
- Reflects difficulty of recovering matrix from sampling

Elements of different 128×7 matrices with orthonormal columns

good coherence

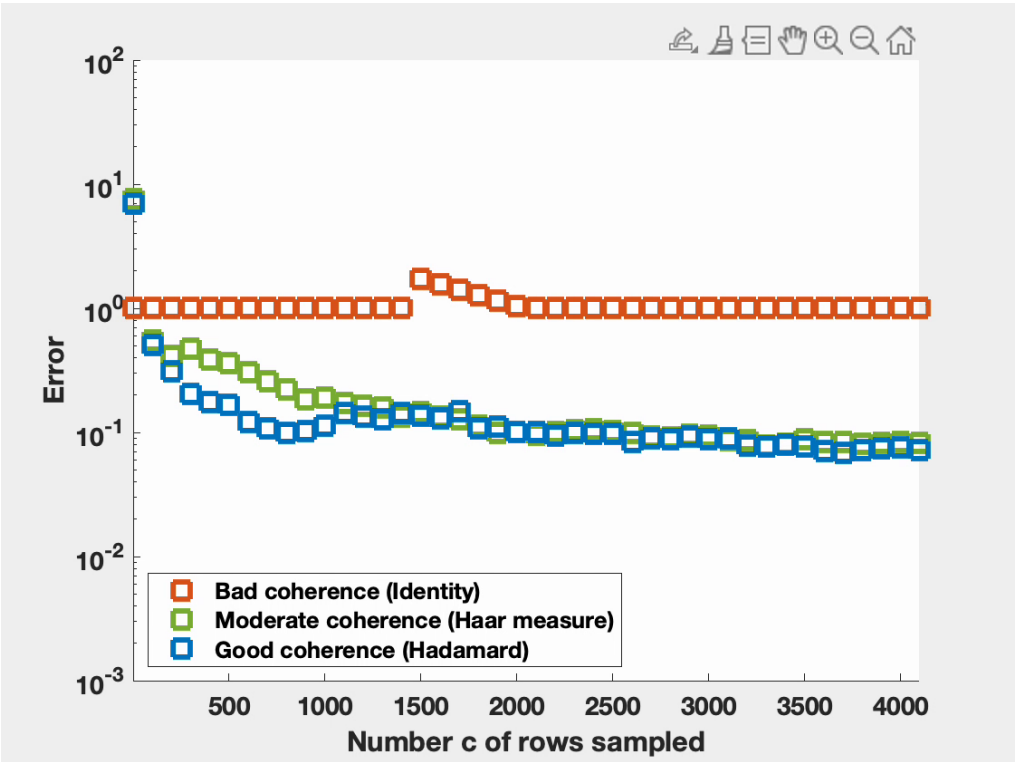
bad coherence



Monte Carlo Gram matrix multiplication for orthonormal matrices $\boldsymbol{Q}$ with different coherence

Error $\|(\boldsymbol{SQ})^T(\boldsymbol{SQ}) - \boldsymbol{I}_n\|_2$ versus number of rows of $\boldsymbol{SQ}$

Coherence: $\mu = 1$ $\mu = .0074$ $\mu = .002 = n/m$



Play/Pause

Error due to randomization for matrices with orthonormal columns

$\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ has orthonormal columns

Coherence $\mu \equiv \max_{1 \le i \le m} \|\boldsymbol{e}_i^T \boldsymbol{Q}\|_2^2$

$\boldsymbol{S}$ samples c rows uniformly with replacement

For any $0 < \delta < 1$ with probability at least $1 - \delta$

$$\|(\boldsymbol{SQ})^T(\boldsymbol{SQ}) - \boldsymbol{I}\|_2 \le \tau + \sqrt{\tau(6+\tau)} \quad \text{where} \quad \tau = m\,\mu\,\frac{\ln(n/\delta)}{3c}$$

Error depends on coherence of $\boldsymbol{Q}$

- Good coherence $\mu = n/m$ $\quad$ {mass of $\boldsymbol{Q}$ uniformly distributed}

  $\tau = n\,\frac{\ln(n/\delta)}{3c}$ depends on smallest matrix dimension

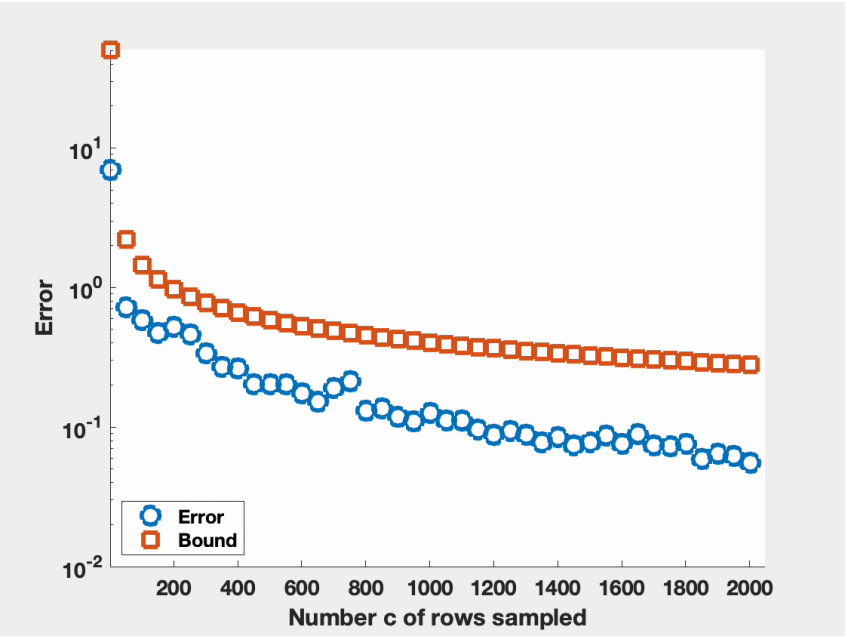
- Bad coherence $\mu = 1$ $\quad$ {mass of $\boldsymbol{Q}$ highly concentrated}

  $\tau = m\,\frac{\ln(n/\delta)}{3c}$ depends on largest matrix dimension

Probabilistic error bound in action

With probability at least $1 - \delta = .99$

$$\|(\boldsymbol{SQ})^T(\boldsymbol{SQ}) - \boldsymbol{I}\|_2 \leq \tau + \sqrt{\tau(6+\tau)} \qquad \text{where} \quad \tau = m\,\mu\,\frac{\ln(n/\delta)}{3c}$$



Play/Pause

$\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ Hadamard with $m = 2048$, $n = 8$, $\mu = n/m = .0039$

Summary: Matrices with orthonormal columns

$\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ has orthonormal columns, $\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}_n$

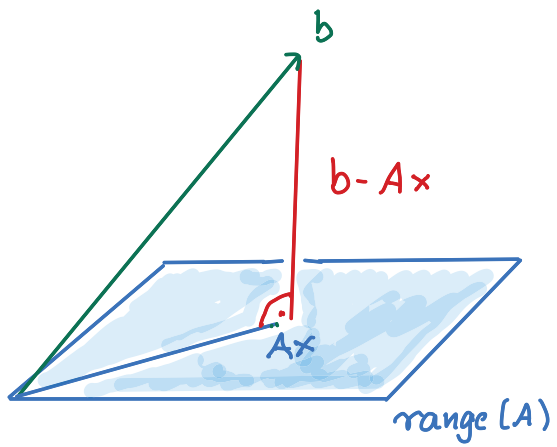
Monte Carlo matrix multiplication: $(\boldsymbol{SQ})^T(\boldsymbol{SQ}) \approx \boldsymbol{I}_n$

- Coherence $\mu \equiv \max_{1 \le i \le m} \|\boldsymbol{e}_i^T \boldsymbol{Q}\|_2^2$
  reflects distribution of 'mass' in the matrix $\boldsymbol{Q}$
  determines the difficulty of sampling rows from $\boldsymbol{Q}$
- Good coherence $\mu = n/m$: Sampling is easy
  Error depends on small matrix dimension n
- Bad coherence $\mu = 1$: Sampling is hard
  Error depends on large matrix dimension m
- Error bound reflects qualitative behaviour of error,
  and is informative even for matrices of small dimension

Case Study:
Randomized Solution of Least Squares/Regression Problems

Well-posed least squares/regression problems

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $\text{rank}(\boldsymbol{A}) = n$, $\boldsymbol{b} \in \mathbb{R}^m$
Solve $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$
Unique solution: $\boldsymbol{x}_* = \boldsymbol{A}^\dagger \boldsymbol{b}, \quad \boldsymbol{A}^\dagger \equiv (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T$



Two randomization approaches:

1. Dimension reduction
2. Convergence acceleration

First randomization approach: Dimension reduction [Drineas, Mahoney, Muthukrishnan, Sarlós]

Approach: Random sampling of rows from $\boldsymbol{A}$ and $\boldsymbol{b}$

- Matrix $\boldsymbol{S}$ samples c rows uniformly and with replacement
- Replace $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ by smaller-dimensional problem

$$\min_{\tilde{\boldsymbol{x}}} \|\boldsymbol{S}\,(\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b})\|_2$$

- Minimal norm solution $\tilde{\boldsymbol{x}}_* = (\boldsymbol{S}\boldsymbol{A})^\dagger(\boldsymbol{S}\boldsymbol{b})$

Perspective of regression:
How does row sampling affect the statistical model uncertainty?

Model uncertainty in regression problems

Gaussian linear model

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{\epsilon} \qquad \boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_4)$$

$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ has unique solution

$$\boldsymbol{x}_* = \boldsymbol{A}^\dagger \boldsymbol{b} \qquad \boldsymbol{A}^\dagger = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Maximum likelihood estimator

$$\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{x}_*] = \boldsymbol{A}^\dagger \, \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{b}] = \boldsymbol{A}^\dagger \boldsymbol{A}\boldsymbol{x}_0 = \boldsymbol{x}_0$$

with variance $\mathbb{V}\mathrm{ar}_{\boldsymbol{\epsilon}}[\boldsymbol{x}_*] = \sigma^2 \, (\boldsymbol{A}^T \boldsymbol{A})^{-1} = \sigma^2 \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}$

Effect of row sampling on model uncertainty

$$\min_{\tilde{\boldsymbol{x}}} \|\boldsymbol{S}\,(\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b})\|_2$$

has minimal norm solution $\tilde{\boldsymbol{x}}_* = (\boldsymbol{S}\boldsymbol{A})^\dagger(\boldsymbol{S}\boldsymbol{b})$

Example:

$$\boldsymbol{S}\boldsymbol{A} = \underbrace{\sqrt{2}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}}_{\boldsymbol{S}}\begin{bmatrix}1 & 0\\ 0 & 1\\ 1 & 0\\ 0 & 0\end{bmatrix} = \begin{bmatrix}\sqrt{2} & 0\\ 0 & \sqrt{2}\end{bmatrix}$$

- Unbiased estimator: $\mathbb{E}_{\boldsymbol{\epsilon}}[\tilde{\boldsymbol{x}}_*] = (\boldsymbol{S}\boldsymbol{A})^\dagger \boldsymbol{S}\,\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{b}] = \boldsymbol{x}_0$
- Variance changes:

$$\mathbb{V}\mathrm{ar}_{\boldsymbol{\epsilon}}[\tilde{\boldsymbol{x}}_*] = \sigma^2\begin{bmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{bmatrix} \neq \sigma^2\begin{bmatrix}\frac{1}{2} & 0\\ 0 & 1\end{bmatrix} = \mathbb{V}\mathrm{ar}_{\boldsymbol{\epsilon}}[\boldsymbol{x}_*]$$

Effect of row sampling on model uncertainty

[Chi, Ipsen]

$$\min_{\tilde{\boldsymbol{x}}} \|\boldsymbol{S}\,(\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b})\|_2$$

has minimal norm solution $\tilde{\boldsymbol{x}}_* = (\boldsymbol{S}\boldsymbol{A})^\dagger(\boldsymbol{S}\boldsymbol{b})$

- Define oblique projector: $\Pi \equiv \boldsymbol{A}(\boldsymbol{S}\boldsymbol{A})^\dagger \boldsymbol{S}$
- Expectation: $\mathbb{E}_\epsilon[\tilde{\boldsymbol{x}}_* \,|\boldsymbol{S}] = \boldsymbol{A}^\dagger \Pi \boldsymbol{A}\boldsymbol{x}_0$

  If $\mathrm{rank}(\boldsymbol{S}\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A})$, then $\mathbb{E}_\epsilon[\tilde{\boldsymbol{x}}_* \,|\boldsymbol{S}] = \boldsymbol{x}_0$ (unbiased estimator)
- Variance: $\mathbb{V}\mathrm{ar}_\epsilon[\tilde{\boldsymbol{x}}_* \,|\boldsymbol{S}] = \sigma^2\,(\boldsymbol{A}^\dagger \Pi)(\boldsymbol{A}^\dagger \Pi)^T$

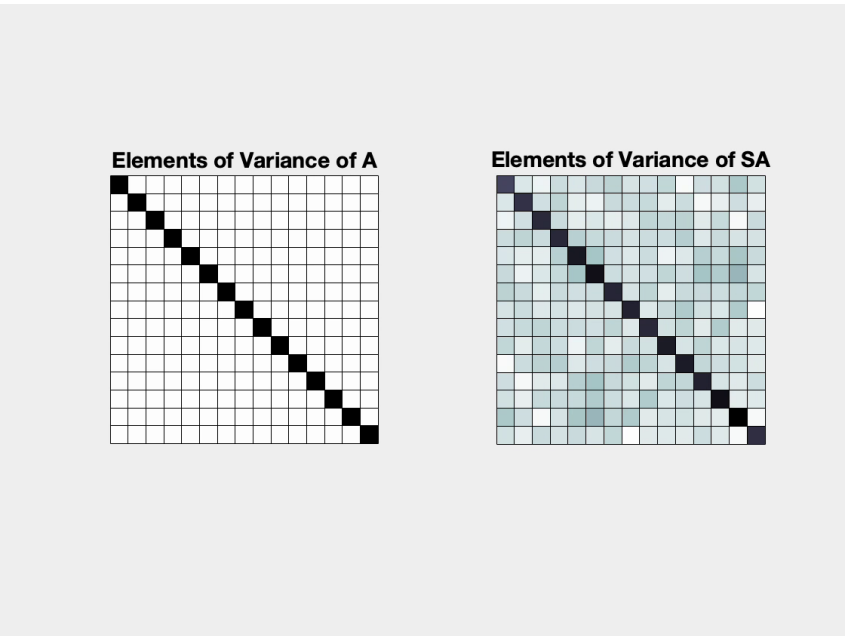
  If $\mathrm{rank}(\boldsymbol{S}\boldsymbol{A}) < \mathrm{rank}(\boldsymbol{A})$, then $\mathbb{V}\mathrm{ar}_\epsilon[\tilde{\boldsymbol{x}}_* \,|\boldsymbol{S}]$ singular

This is only a partial analysis: $\boldsymbol{S}$ is a fixed sampling matrix
No accounting for the uncertainty due to randomization

Change in variance under randomization

$\boldsymbol{A} \in \mathbb{R}^{2048 \times 16}$ has orthonormal columns
30 different sampling matrices $\boldsymbol{S}$ that sample $\frac{m}{10} = 204$ rows uniformly with replacement

Elements of $\mathbb{V}\mathrm{ar}_{\epsilon}[\boldsymbol{A}]$ versus elements of $\mathbb{V}\mathrm{ar}_{\epsilon}[\boldsymbol{SA}]$



Play/Pause

Summary: First randomization approach for least squares/regression problems

Dimension reduction: Random sampling of rows from

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$$

Statistical perspective:

- Row sampling changes statistical properties of underlying linear model
- Simultaneous analysis of uncertainties from both, model and randomization leads to results that are difficult to interpret

Computational perspective:

- If $\mathrm{rank}(\boldsymbol{S}\boldsymbol{A}) < \mathrm{rank}(\boldsymbol{A})$, the sampled least squares problem is ill-posed
- Even if $\mathrm{rank}(\boldsymbol{S}\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A})$, the sampled least squares problem can be more ill conditioned, due to
  worse matrix condition number: $\kappa(\boldsymbol{S}\boldsymbol{A}) > \kappa(\boldsymbol{A})$
  larger least squares residual: $\|\boldsymbol{S}(\boldsymbol{A}\tilde{\boldsymbol{x}}_* - \boldsymbol{b})\|_2 > \|\boldsymbol{A}\boldsymbol{x}_* - \boldsymbol{b}\|_2$

Conclusion: Row sampling is a risky approach for randomization

Second randomization approach: Convergence acceleration

[Avron, Maymounkov, Toledo], [Rokhlin, Tygert]

- Iterative Krylov space method LSQR [Paige, Saunders]

  Iterates x_k converge to $\boldsymbol{x}_*$ as

$$\underbrace{\|\boldsymbol{A}(\boldsymbol{x}_* - \boldsymbol{x}_k)\|_2^2}_{\text{error in iteration } k} \;\leq\; 2\left(\frac{\kappa(\boldsymbol{A})-1}{\kappa(\boldsymbol{A})+1}\right)^k \underbrace{\|\boldsymbol{A}(\boldsymbol{x}_* - \boldsymbol{x}_0)\|_2^2}_{\text{initial error}}$$

  Fast convergence, if condition number $\kappa(\boldsymbol{A}) \equiv \|\boldsymbol{A}\|_2 \|\boldsymbol{A}^\dagger\|_2 \approx 1$

- Accelerate convergence via right preconditioning

  {change of variables}

$$\min_{\boldsymbol{y}} \|\boldsymbol{A}\,\boldsymbol{P}^{-1}\,\underbrace{(\boldsymbol{P}\boldsymbol{x})}_{\boldsymbol{y}} - \boldsymbol{b}\|_2$$

The ideal preconditioner

QR factorization $\boldsymbol{A} = \boldsymbol{Q}\,\boldsymbol{R}$ {basis transformation to orthonormal basis}

$\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ has orthonormal columns: $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_n$
$\boldsymbol{R} \in \mathbb{R}^{n\times n}$ is triangular nonsingular

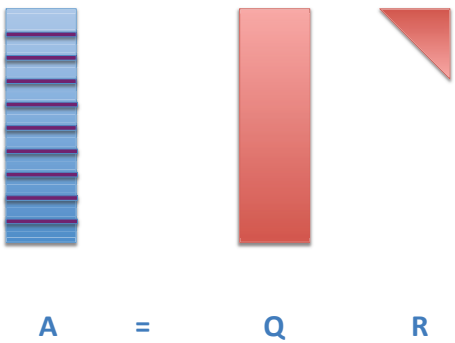
- Use $\boldsymbol{R}$ as preconditioner
- Preconditioned matrix $\boldsymbol{A}\boldsymbol{R}^{-1} = \boldsymbol{Q}$ has $\kappa(\boldsymbol{Q}) = 1$
- LSQR solves pre-conditioned problem in 1 iteration

But: Construction of preconditioner is too expensive
Operation count: $\mathcal{O}(mn^2)$ flops {floating point operations}

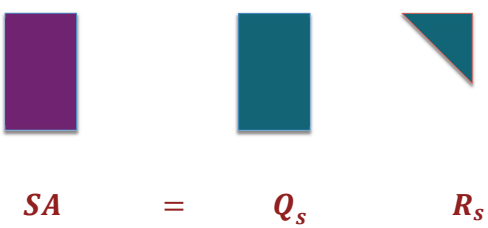
A cheaper randomized preconditioner

[Avron, Maymounkov, Toledo], [Rokhlin, Tygert]

QR factorization from a few rows of matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$



1. Sample $c \geq n$ rows of $\boldsymbol{A}$: $\boldsymbol{SA}$
2. Factor sampled matrix: $\boldsymbol{SA} = \boldsymbol{Q}_s \boldsymbol{R}_s$
3. Randomized preconditioner $\boldsymbol{R}_s$



Operation count: $\mathcal{O}(cn^2)$ flops

{independent of large dimension m}

Blendenpik (simplified version)

[Avron, Maymounkov, Toledo]

Input: $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(\boldsymbol{A}) = n$, $\boldsymbol{b} \in \mathbb{R}^{m \times n}$
Sampling amount $c \geq n$
Output: Solution $\boldsymbol{x}_*$ to $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$

{Construct preconditioner}
Sample c rows of $\boldsymbol{A}$ $\rightarrow \boldsymbol{S}\boldsymbol{A}$ {fewer rows}
Factor $\boldsymbol{S}\boldsymbol{A} = \boldsymbol{Q}_s \boldsymbol{R}_s$

{Solve preconditioned problem}
Solve $\min_{\boldsymbol{y}} \|\boldsymbol{A}\boldsymbol{R}_s^{-1}\boldsymbol{y} - \boldsymbol{b}\|_2$ with LSQR
Solve $\boldsymbol{R}_s \boldsymbol{x}_* = \boldsymbol{y}$ {$\triangle$ system}

Convergence analysis of preconditioned LSQR

[Avron, Maymounkov, Toledo], [Ipsen, Wentworth]

$$\min_{\boldsymbol{y}} \|\boldsymbol{A}\,\boldsymbol{R}_s^{-1}\boldsymbol{y} - \boldsymbol{b}\|_2$$

1. Two QR factorizations
   - Conceptual factorization of original matrix: $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$
   - Computed factorization of sampled matrix: $\boldsymbol{S}\boldsymbol{A} = \boldsymbol{Q}_s\boldsymbol{R}_s$
2. Condition number of preconditioned matrix = Condition number of sampled orthonormal matrix

$$\kappa(\boldsymbol{A}\boldsymbol{R}_s^{-1}) = \kappa(\boldsymbol{S}\boldsymbol{Q})$$

3. Next: Bound $\kappa(\boldsymbol{S}\boldsymbol{Q})$ with Monte Carlo matrix multiplication

From matrix multiplication error to condition numbers

- Given: $\boldsymbol{X}$ with $\boldsymbol{X}^T\boldsymbol{X} \approx \boldsymbol{I}$
- Eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$: $\lambda_{max} \geq \cdots \geq \lambda_{min} > 0$
- Condition number: $\kappa(\boldsymbol{X})^2 = \kappa(\boldsymbol{X}^T\boldsymbol{X}) = \frac{\lambda_{max}}{\lambda_{min}}$

1. Monte Carlo matrix multiplication:
   With probability at least $1 - \delta$
   $$\|\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{I}\|_2 \leq \epsilon$$
2. Weyl's monotonicity theorem:
   $$|\lambda_{min} - 1| \leq \epsilon \quad \text{and} \quad |\lambda_{max} - 1| \leq \epsilon$$
3. Condition number:
   With probability at least $1 - \delta$
   $$\kappa(\boldsymbol{X}) = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}} \leq \sqrt{\frac{1+\epsilon}{1-\epsilon}}$$

Condition number bound for row-sampled matrices with orthonormal columns

[Avron, Maymounkov, Toledo], [Ipsen, Wentworth], [Ipsen, Smith]

Assume:

$\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ with $\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}_n$ and $\mu \equiv \max_{1 \le i \le m} \|\boldsymbol{e}_i^T \boldsymbol{Q}\|_2^2$

$\boldsymbol{S}$ samples c rows uniformly with replacement

Condition number of sampled matrix:

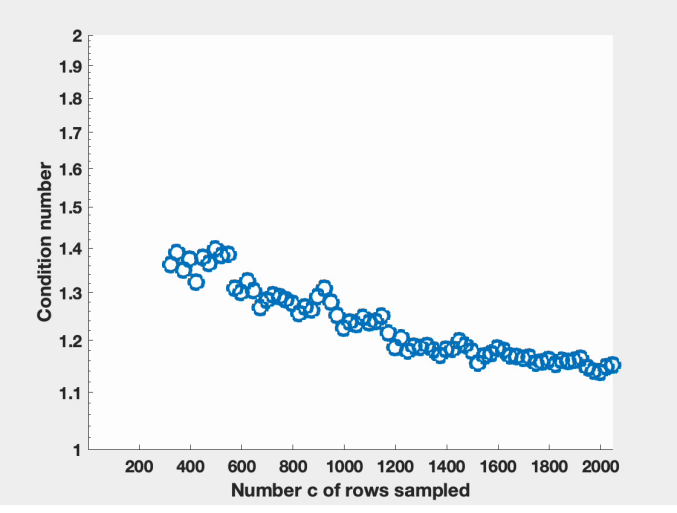
For any $0 < \delta < 1$ and $0 < \epsilon < 1$, if $c \geq m\mu \frac{\ln(n/\delta)}{\epsilon^2}$ then with probability at least $1 - \delta$

$$\kappa(\boldsymbol{S}\boldsymbol{Q}) \leq \frac{1+\epsilon}{1-\epsilon}$$

Condition number bound in action

Orthonormal matrix $\boldsymbol{Q} \in \mathbb{R}^{2048\times 16}$, perfect coherence $\mu = \frac{n}{m} = .0162$
Success probability $\delta = .99$

Choose $\epsilon = 99/101$ so that $\kappa(\boldsymbol{SQ}) \leq \frac{1+\epsilon}{1-\epsilon} = 10$



Play/Pause

Need $c \geq 2m\mu \frac{\ln(n/\delta)}{\epsilon^2} \approx 322 = 16\%$ of rows

Summary: Second randomization approach for least squares/regression problems

Convergence acceleration:
Randomized construction of preconditioner

+ Only need correct magnitude of condition number
+ Actual condition number close to optimal
– Need good coherence for small sampling amounts
+ Improve coherence by fast premultiplication with random orthogonal matrix [Avron, Maymounkov, Toledo]
+ Safe approach for randomization, no loss of information

Summary

- Sampling without replacement
  Viable in practice, and 'easy' to analyse
- Monte Carlo Gram matrix multiplication
  Dimension reduction to smaller-dimensional matrices
  Risky randomization: Loss of information and accuracy
- Matrices with orthonormal columns
  Coherence reflects the difficulty of sampling rows
- Randomized least squares/regression problems
  Dimension reduction to smaller-dimensional matrices
  Risky randomization: Can change statistical properties, destroy well-posedness, and worsen numerical accuracy
  Convergence acceleration via randomized preconditoner
  Safe randomization: Faster speed, no loss of accuracy

Future Research

- Tighter, more practical matrix concentration inequalities
- Analysis of randomized algorithms in finite precision:
  - Numerical error due to randomization
  - Notion of conditioning (e.g. coherence)
  - Notion of numerical stability of randomized algorithms
- Robust implementations (e.g. choice of failure probability)
- Opportunities for safe randomization:
  Dynamic randomized preconditioners in inner loops
- Randomization in mixed/lower arithmetic precision
- Propagating errors due to randomization
  through computational pipelines (probabilistic numerics)

Thank you very much
for your attention