Finding Structure with Randomness

Joel A. Tropp

Applied & Computational Mathematics
California Institute of Technology
jtropp@acm.caltech.edu

Joint with P.-G. Martinsson and N. Halko
Applied Mathematics, Univ. Colorado at Boulder

Research supported in part by NSF, DARPA, and ONR

Top 10 Scientific Algorithms

list (here, the list is in chronological order; however, the articles appear in no particular order):

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

With each of these algorithms or approaches, there is a person or group receiving credit for inventing or

Source: Dongarra and Sullivan, *Comput. Sci. Eng.*, 2000.

The Decompositional Approach

"The underlying principle of the decompositional approach to matrix computation is that it is not the business of the matrix algorithmicists to solve particular problems but to construct computational platforms from which a variety of problems can be solved."

- A decomposition solves not one but many problems
- Often expensive to compute but can be reused
- Shows that apparently different algorithms produce the same object
- Facilitates rounding-error analysis
- Can be updated efficiently to reflect new information
- Has led to highly effective black-box software

Source: Stewart 2000.

Low-Rank Matrix Approximation

$$\underset{m \times n}{\boldsymbol{A}} \approx \underset{m \times k}{\boldsymbol{B}} \; \underset{k \times n.}{\boldsymbol{C}},$$

Benefits:

- Exposes structure of the matrix
- Allows efficient storage
- Facilitates multiplication with vectors or other matrices

Applications:

- Principal component analysis
- Low-dimensional embedding of data
- Approximating continuum operators with exponentially decaying spectra
- Model reduction for PDEs with rapidly oscillating coefficients

Approximation of Massive Data

- **Problem:** Major cost for numerical algorithms is data transfer
- Cost scales, roughly, with number of passes **not** amount of arithmetic
- Random access to data is expensive, so classical algorithms may fail
- **Assume** a matrix–matrix product with data matrix takes one pass
- Matrix multiplication is efficient in many architectures:
  - Graphics processing units
  - Multi-core processors
  - Parallel computers
  - Distributed systems

Model Problem

Given:

- An $m \times n$ matrix $\boldsymbol{A}$ with $m \geq n$
- Target rank k
- Oversampling parameter p

Construct an $n \times (k + p)$ matrix $\boldsymbol{Q}$ with orthonormal columns s.t.

$$\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^* \boldsymbol{A}\| \approx \min_{\mathrm{rank}(\boldsymbol{B}) \leq k} \|\boldsymbol{A} - \boldsymbol{B}\|,$$

- $\boldsymbol{Q}\boldsymbol{Q}^*$ is the orthogonal projector onto the range of $\boldsymbol{Q}$
- The basis $\boldsymbol{Q}$ can be used to construct matrix decompositions

From Basis to Decomposition

Problem: Given the basis $\boldsymbol{Q}$, where do we get a factorization?

Example: Singular value decomposition

Assume $\boldsymbol{A}$ is $m \times n$ and $\boldsymbol{Q}$ is $n \times k$ where $\boldsymbol{A} \approx \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}$.

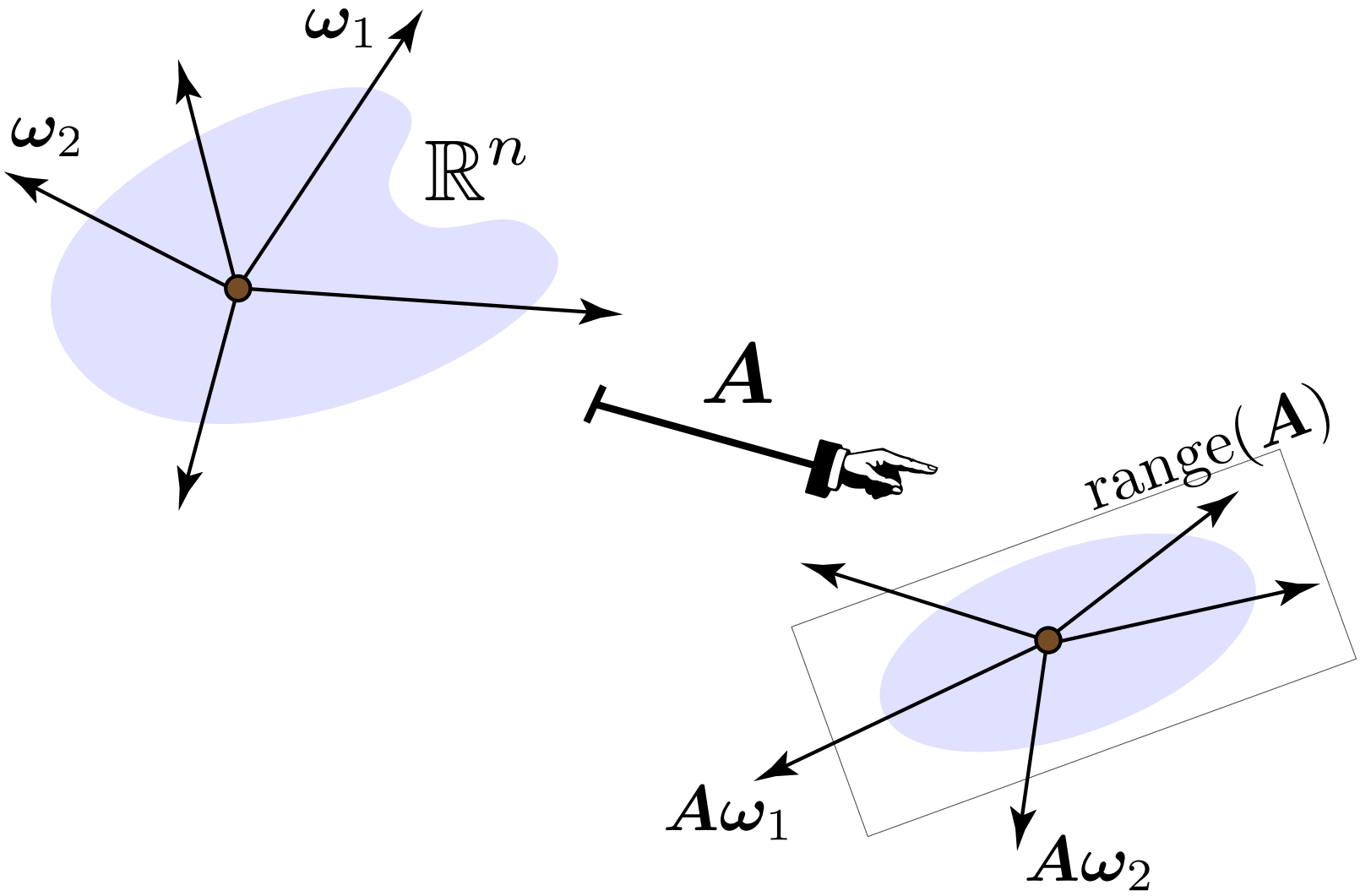
1. Form $k \times n$ matrix $\boldsymbol{B} = \boldsymbol{Q}^*\boldsymbol{A}$ with one pass

2. Factor $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ at cost $\mathrm{O}(k^2 n)$

3. Conclude $\boldsymbol{A} \approx (\boldsymbol{Q}\boldsymbol{U})\boldsymbol{\Sigma}\boldsymbol{V}^*$.

Total Cost: One pass + one multiply $(m \times n \times k) + \mathrm{O}(k^2 n)$ flops

Random Sampling: Intuition



Proto-Algorithm for Model Problem

- Converting this intuition into a computational procedure...

Input: An $m \times n$ matrix $\boldsymbol{A}$, a target rank k, an oversampling parameter p

Output: An $m \times (k + p)$ matrix $\boldsymbol{Q}$ with orthonormal columns

1. Draw an $n \times (k + p)$ random matrix $\boldsymbol{\Omega}$.
2. Form the matrix product $\boldsymbol{Y} = \boldsymbol{A\Omega}$.
3. Construct an orthonormal basis $\boldsymbol{Q}$ for the range of $\boldsymbol{Y}$.

Major Players: Deshpande, Drineas, Frieze, Kannan, Mahoney, Martinsson, Papadimitriou, Rokhlin, Sarlos, Tygert, Vempala (1998–2009)

Implementation Issues

Q: How much oversampling?

A: Remarkably, $p = 5$ or $p = 10$ is usually adequate!

Q: What random matrix?

A: For this application, standard Gaussian works nicely.

Q: How do we do the matrix–matrix multiply?

A: Exploit the computational architecture.

Q: How do we compute the orthonormal basis?

A: Carefully... Double Gram–Schmidt or Householder reflectors.

Q: How do we pick k?

A: Can be done adaptively using a randomized error estimator.

Total Costs for Approximate k-SVD

Proto-Algorithm:

$$\text{1 pass} \quad + \quad \text{2 multiplies } (m \times n \times k) \quad + \quad k^2(m+n) \quad \text{flops}$$

Classical Sparse Methods:

$$k \text{ passes} \quad + \quad k \text{ multiplies } (m \times n \times 1) \quad + \quad k^2(m+n) \quad \text{flops}$$

Classical Dense Methods:

$$k \text{ passes (or more)} \quad + \quad mnk \quad \text{flops}$$

Proto-Algorithm + Power Scheme

Problem: The singular values of the data matrix often decay slowly

Remedy: Apply the proto-algorithm to $(\boldsymbol{A}\boldsymbol{A}^*)^q\boldsymbol{A}$ for small q

---

Input: An $m \times n$ matrix $\boldsymbol{A}$, a target rank k, an oversampling parameter p

Output: An $m \times (k+p)$ matrix $\boldsymbol{Q}$ with orthonormal columns

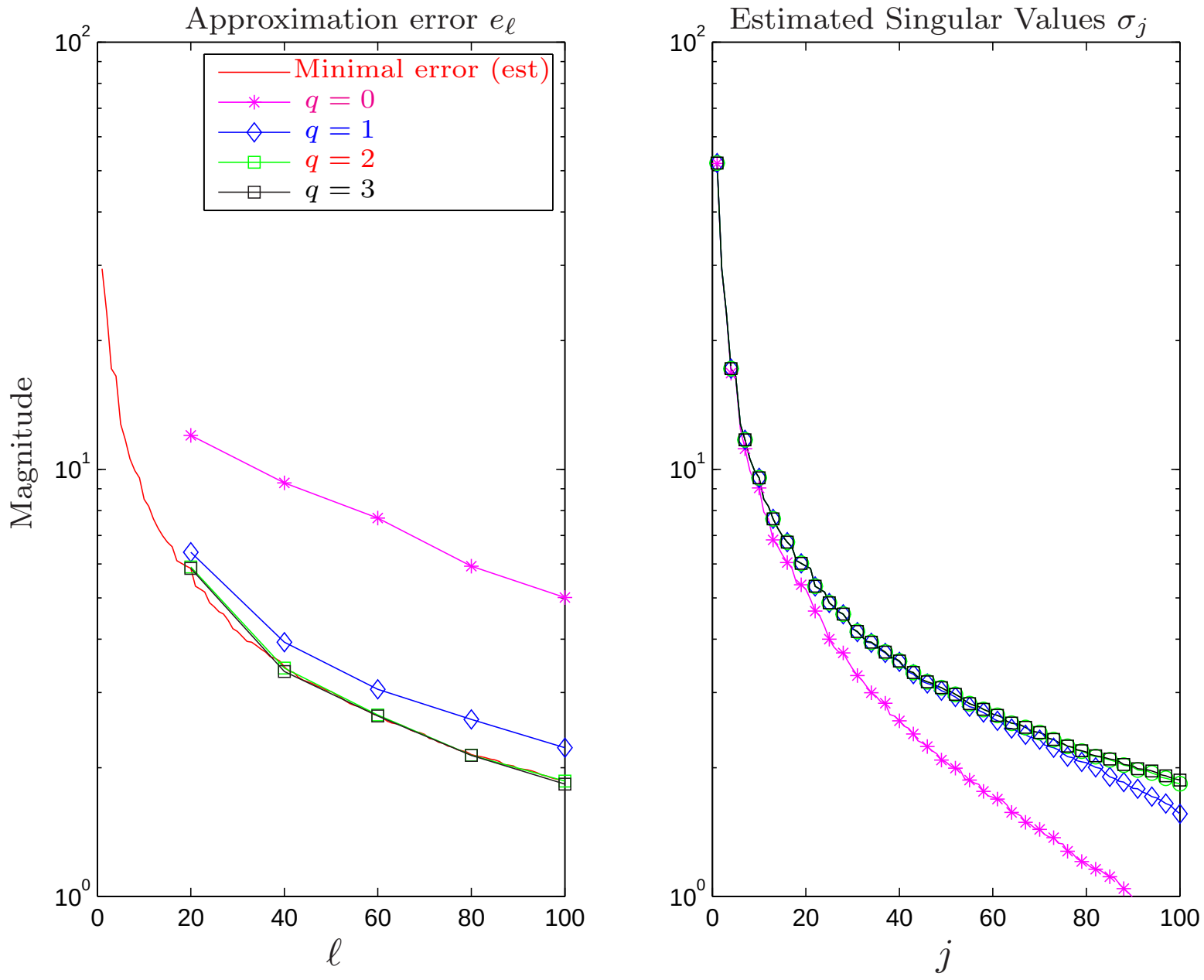
1. Draw an $n \times (k+p)$ random matrix $\boldsymbol{\Omega}$.
2. Form the matrix product $\boldsymbol{Y} = (\boldsymbol{A}\boldsymbol{A}^*)^q\boldsymbol{A}\boldsymbol{\Omega}$ by sequential multiplication.
3. Construct an orthonormal basis $\boldsymbol{Q}$ for the range of $\boldsymbol{Y}$.

Total Cost: q passes + q multiplies $(m \times n \times k)$ + $\mathrm{O}(kmn)$ flops.

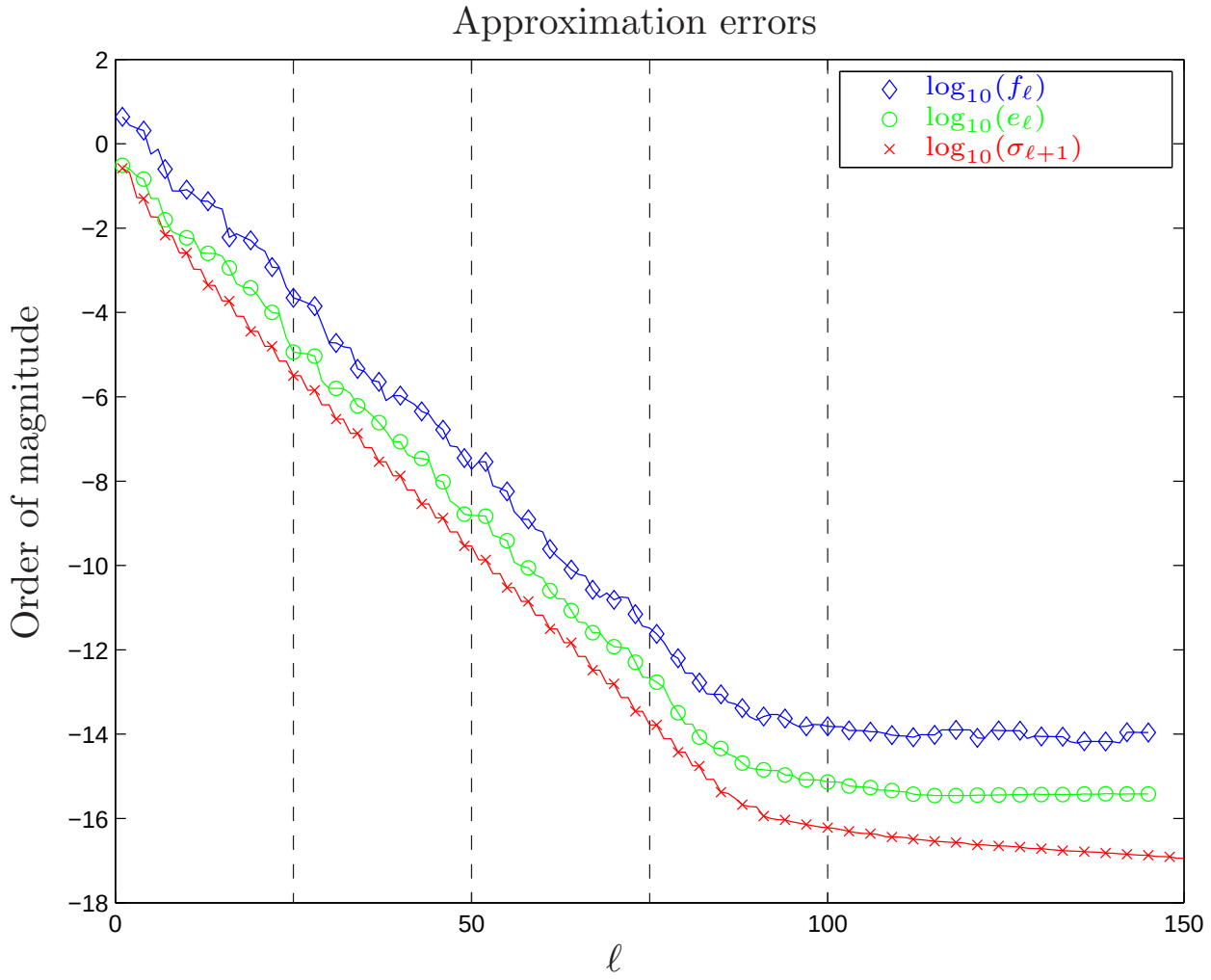
Eigenfaces

- Database consists of $7,254$ photographs with $98,304$ pixels each
- Form $98,304 \times 7,254$ data matrix $\widetilde{\boldsymbol{A}}$
- **Total storage:** 5.4 Gigabytes (uncompressed)
- Center each column and scale to unit norm to obtain $\boldsymbol{A}$
- The dominant left singular vectors are called **eigenfaces**
- Attempt to compute first 100 eigenfaces using power scheme

Image: Scholarpedia article "Eigenfaces," 12 October 2009


Approximation error e_ℓ
Minimal error (est)
$q = 0$
$q = 1$
$q = 2$
$q = 3$
Magnitude
10^2
10^1
10^0
0
20
40
60
80
100
ℓ
Estimated Singular Values σ_j
10^2
10^1
10^0
0
20
40
60
80
100
j

Approximating a Helmholtz Integral Operator



Error Bound for Proto-Algorithm

Theorem 1. [HMT 2009] **Assume**

- *the matrix* $\boldsymbol{A}$ *is* $m \times n$ *with* $m \geq n$*;*
- *the optimal error* $\sigma_{k+1} = \min_{\mathrm{rank}(\boldsymbol{B}) \leq k} \|\boldsymbol{A} - \boldsymbol{B}\|$*;*
- *the test matrix* $\boldsymbol{\Omega}$ *is* $n \times (k+p)$ *standard Gaussian.*

Then *the basis* $\boldsymbol{Q}$ *computed by the proto-algorithm satisfies*

$$\mathbb{E}\,\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\| \leq \left[1 + \frac{4\sqrt{k+p}}{p-1} \cdot \sqrt{n}\right] \sigma_{k+1}.$$

The probability of a substantially larger error is negligible.

Error Bound for Power Scheme

Theorem 2. [HMT 2009] **Assume**

- *the matrix* $\boldsymbol{A}$ *is* $m \times n$ *with* $m \geq n$;
- *the optimal error* $\sigma_{k+1} = \min_{\mathrm{rank}(\boldsymbol{B}) \leq k} \|\boldsymbol{A} - \boldsymbol{B}\|$;
- *the test matrix* $\boldsymbol{\Omega}$ *is* $n \times (k+p)$ *standard Gaussian.*

Then *the basis* $\boldsymbol{Q}$ *computed by the proto-algorithm satisfies*

$$\mathbb{E}\,\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\| \leq \left[1 + \frac{4\sqrt{k+p}}{p-1} \cdot \sqrt{n}\right]^{1/q} \sigma_{k+1}.$$

The probability of a substantially larger error is negligible.

- The power scheme drives the extra factor to one exponentially fast!

Inner Workings I

Assume

- $\boldsymbol{A}$ is $m \times n$ with SVD

$$\boldsymbol{A} = \boldsymbol{U} \begin{array}{c} \begin{matrix} k & n-k \end{matrix} \\ \begin{bmatrix} \boldsymbol{\Sigma}_1 & \\ & \boldsymbol{\Sigma}_2 \end{bmatrix} \end{array} \begin{array}{c} \begin{matrix} \\ \end{matrix} \\ \begin{bmatrix} \boldsymbol{V}_1^* \\ \boldsymbol{V}_2^* \end{bmatrix} \end{array} \begin{array}{c} \\ \begin{matrix} k \\ n-k \end{matrix} \end{array}$$

- Let $\boldsymbol{\Omega}$ be a test matrix, decomposed as

$$\boldsymbol{\Omega}_1 = \boldsymbol{V}_1^* \, \boldsymbol{\Omega} \quad \text{and} \quad \boldsymbol{\Omega}_2 = \boldsymbol{V}_2^* \, \boldsymbol{\Omega}.$$

- Construct the sample matrix $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\Omega}$.

Theorem 3. [BMD09, HMT09] *When $\boldsymbol{\Omega}_1$ has full row rank,*

$$\left\| (\mathbf{I} - \boldsymbol{P}_{\boldsymbol{Y}}) \boldsymbol{A} \right\|^2 \leq \left\| \boldsymbol{\Sigma}_2 \right\|^2 + \left\| \boldsymbol{\Sigma}_2 \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1^\dagger \right\|^2.$$

Inner Workings II

- When $\boldsymbol{\Omega}$ is Gaussian, $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ are independent.
- Taking the expectation w.r.t. $\boldsymbol{\Omega}_2$ first...

$$\mathbb{E}_2 \left\| \boldsymbol{\Sigma}_2 \boldsymbol{\Omega}_2 \boldsymbol{\Omega}_1^\dagger \right\| \leq \|\boldsymbol{\Sigma}_2\| \left\| \boldsymbol{\Omega}_1^\dagger \right\|_{\mathrm{F}} + \|\boldsymbol{\Sigma}_2\|_{\mathrm{F}} \left\| \boldsymbol{\Omega}_1^\dagger \right\|.$$

- The expectations of the norms w.r.t. $\boldsymbol{\Omega}_1$ satisfy

$$\mathbb{E} \left\| \boldsymbol{\Omega}_1^\dagger \right\|_{\mathrm{F}} \leq \sqrt{\frac{k}{p-1}} \quad \text{and} \quad \mathbb{E} \left\| \boldsymbol{\Omega}_1^\dagger \right\| \leq \frac{\mathrm{e}\sqrt{k+p}}{p}.$$

- Conclude

$$\mathbb{E} \left\| (\mathbf{I} - \boldsymbol{P}_{\boldsymbol{Y}}) \boldsymbol{A} \right\| \leq \left[1 + \sqrt{\frac{k}{p-1}} \right] \sigma_{k+1} + \frac{\mathrm{e}\sqrt{k+p}}{p} \left(\sum\nolimits_{j=k+1}^{\infty} \sigma_j^2 \right)^{1/2}$$

Result for Structured Random Matrices

Theorem 4. [HMT09] *Suppose that $\mathbf{\Omega}$ is an $n \times \ell$ SRFT matrix where*

$$\ell \gtrsim (k + \log n) \log k.$$

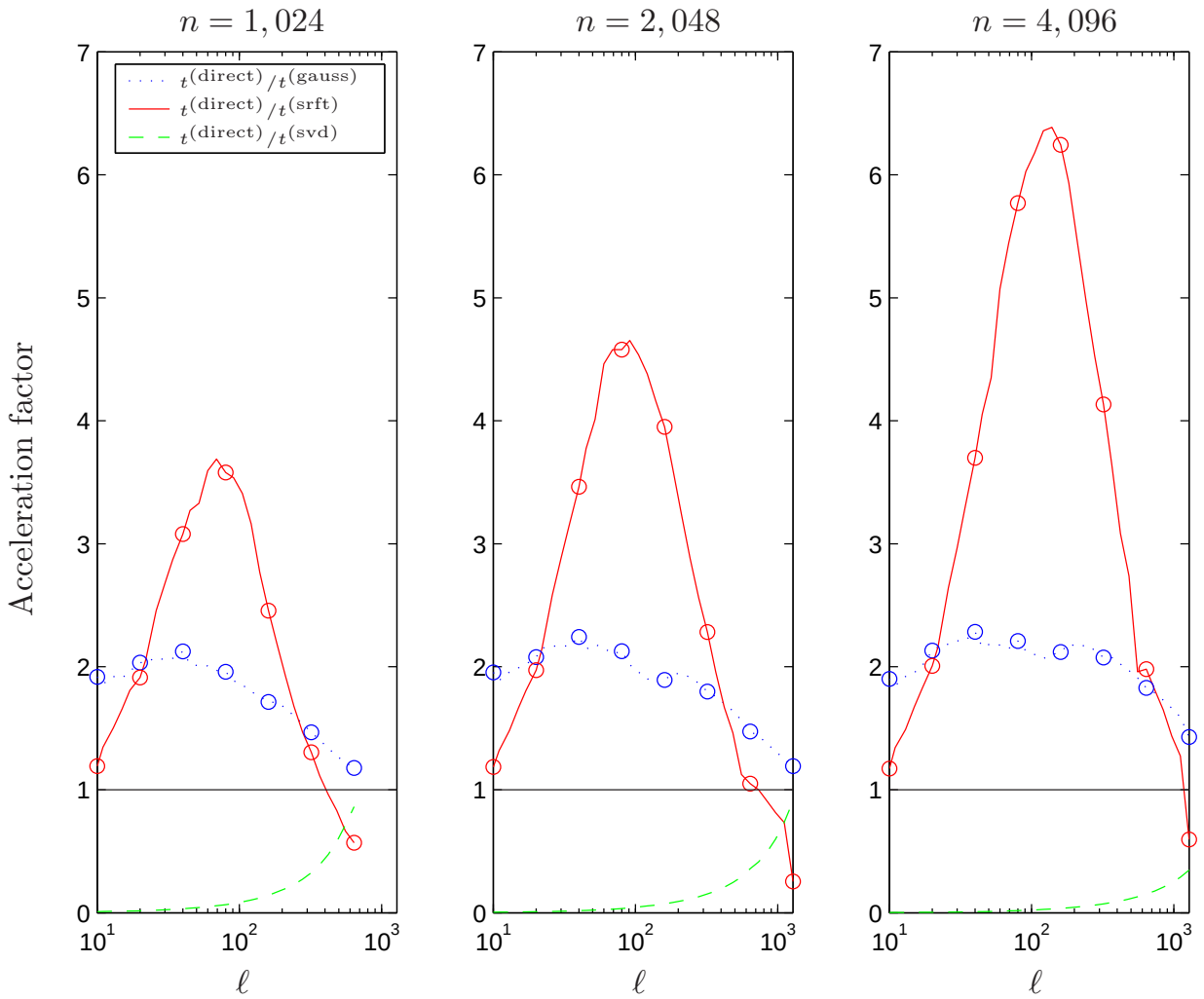
Then

$$\|(\mathbf{I} - \boldsymbol{P}_{\boldsymbol{Y}})\boldsymbol{A}\| \leq \sqrt{1 + \frac{\mathrm{C}n}{\ell}} \cdot \sigma_{k+1},$$

except with probability $k^{-\mathrm{c}}$.

- Follows from same approach
- Uses Rudelson's lemma to show that random rows from a randomized Fourier transform form a well-conditioned set

Faster SVD with Structured Randomness



To learn more...

E-mail:

- jtropp@acm.caltech.edu

Web: http://www.acm.caltech.edu/~jtropp

Papers:

- HMT, "Finding Structure with Randomness: Stochastic Algorithms for Computing Approximate Matrix Decompositions," submitted 2009