Randomised algorithms for computing low rank approximations of matrices

Per-Gunnar Martinsson

Dept. of Mathematics & Oden Inst. for Computational Sciences and Engineering

University of Texas at Austin

Students, postdocs, collaborators: Tracy Babb, Ke Chen, Robert van de Geijn, Abinand Gopal, Nathan Halko, Nathan Heavner, James Levitt, Yijun Liu, Gregorio Quintana-Ortí, Joel Tropp, Bowei Wu, Sergey Voronin, Anna Yesypenko, Patrick Young.

Slides: `http://users.oden.utexas.edu/~pgm/main_talks.html`

Research support by:

Randomised SVD:

Objective: Given an $m \times n$ matrix $\mathbf{A}$, find an approximate rank-k partial SVD:

$$\begin{array}{ccccc} \mathbf{A} & \approx & \mathbf{U} & \mathbf{D} & \mathbf{V}^* \\ m \times n & & m \times k & k \times k & k \times n \end{array}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and $\mathbf{D}$ is diagonal. (We assume $k \ll \min(m, n)$.)

Applications:

- Plane fitting ("principal component analysis").
- Model reduction in scientific computing.
- Spectral algorithms in data analysis.
- "Fast" algorithms of various types: Fast Multipole Methods, generalizations of the Fast Fourier Transform, Fast direct solvers, etc.
- Many, many, many more.

Randomised SVD:

Objective: Given an $m \times n$ matrix $\mathbf{A}$, find an approximate rank-k partial SVD:

$$\begin{array}{ccccc} \mathbf{A} & \approx & \mathbf{U} & \mathbf{D} & \mathbf{V}^* \\ m \times n & & m \times k & k \times k & k \times n \end{array}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and $\mathbf{D}$ is diagonal. (We assume $k \ll \min(m, n)$.)

(A) *Randomised sketching:*

A.1 Draw an $n \times k$ Gaussian random matrix $\mathbf{R}$. `R = randn(n,k)`

A.2 Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$. `Y = A * R`

A.3 Form an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\text{ran}(\mathbf{Q}) = \text{ran}(\mathbf{Y})$. `[Q, ~] = qr(Y)`

(B) *Deterministic post-processing:*

B.1 Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$. `B = Q' * A`

B.2 Form SVD of the matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$. `[Uhat, Sigma, V] = svd(B,'econ')`

B.3 Form the matrix $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$. `U = Q * Uhat`

Randomised SVD:

Objective: Given an $m \times n$ matrix $\mathbf{A}$, find an approximate rank-k partial SVD:

$$\begin{array}{ccccc} \mathbf{A} & \approx & \mathbf{U} & \mathbf{D} & \mathbf{V}^* \\ m \times n & & m \times k & k \times k & k \times n \end{array}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and $\mathbf{D}$ is diagonal. (We assume $k \ll \min(m, n)$.)

(A) *Randomised sketching:*

A.1 Draw an $n \times k$ Gaussian random matrix $\mathbf{R}$. `R = randn(n,k)`

A.2 Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$. `Y = A * R`

A.3 Form an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\text{ran}(\mathbf{Q}) = \text{ran}(\mathbf{Y})$. `[Q, ~] = qr(Y)`

(B) *Deterministic post-processing:*

B.1 Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$. `B = Q' * A`

B.2 Form SVD of the matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$. `[Uhat, Sigma, V] = svd(B,'econ')`

B.3 Form the matrix $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$. `U = Q * Uhat`

The objective of Stage A is to compute an ON-basis that approximately spans the column space of $\mathbf{A}$. The matrix $\mathbf{Q}$ holds these basis vectors and $\mathbf{A} \approx \mathbf{QQ}^*\mathbf{A}$.

Randomised SVD:

Objective: Given an $m \times n$ matrix $\mathbf{A}$, find an approximate rank-k partial SVD:

$$\begin{array}{ccccc} \mathbf{A} & \approx & \mathbf{U} & \mathbf{D} & \mathbf{V}^* \\ m \times n & & m \times k & k \times k & k \times n \end{array}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and $\mathbf{D}$ is diagonal. (We assume $k \ll \min(m, n)$.)

(A) *Randomised sketching:*

A.1 Draw an $n \times k$ Gaussian random matrix $\mathbf{R}$. `R = randn(n,k)`

A.2 Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$. `Y = A * R`

A.3 Form an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\text{ran}(\mathbf{Q}) = \text{ran}(\mathbf{Y})$. `[Q, ~] = qr(Y)`

(B) *Deterministic post-processing:*

B.1 Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$. `B = Q' * A`

B.2 Form SVD of the matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$. `[Uhat, Sigma, V] = svd(B,'econ')`

B.3 Form the matrix $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$. `U = Q * Uhat`

The objective of Stage A is to compute an ON-basis that approximately spans the column space of $\mathbf{A}$. The matrix $\mathbf{Q}$ holds these basis vectors and $\mathbf{A} \approx \mathbf{QQ}^*\mathbf{A}$.

Stage B is exact: $\|\mathbf{A} - \mathbf{Q}\underbrace{\mathbf{Q}^*\mathbf{A}}_{=\mathbf{B}}\| = \|\mathbf{A} - \mathbf{Q}\underbrace{\mathbf{B}}_{=\hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*}\| = \|\mathbf{A} - \underbrace{\mathbf{Q}\hat{\mathbf{U}}}_{=\mathbf{U}}\mathbf{D}\mathbf{V}^*\| = \|\mathbf{A} - \mathbf{U}\mathbf{D}\mathbf{V}^*\|$.

Randomised SVD:

Objective: Given an $m \times n$ matrix $\mathbf{A}$, find an approximate rank-k partial SVD:

$$\begin{array}{ccccc} \mathbf{A} & \approx & \mathbf{U} & \mathbf{D} & \mathbf{V}^* \\ m \times n & & m \times k & k \times k & k \times n \end{array}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and $\mathbf{D}$ is diagonal. (We assume $k \ll \min(m, n)$.)

(A) *Randomised sketching:*

A.1 Draw an $n \times k$ Gaussian random matrix $\mathbf{R}$. `R = randn(n,k)`

A.2 Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$. `Y = A * R`

A.3 Form an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\text{ran}(\mathbf{Q}) = \text{ran}(\mathbf{Y})$. `[Q, ~] = qr(Y)`

(B) *Deterministic post-processing:*

B.1 Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$. `B = Q' * A`

B.2 Form SVD of the matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$. `[Uhat, Sigma, V] = svd(B,'econ')`

B.3 Form the matrix $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$. `U = Q * Uhat`

How does it work? To develop intuition, it helps to first consider the case $\text{rank}(\mathbf{A}) = k$. Then $\text{ran}(\mathbf{Y}) = \text{ran}(\mathbf{A})$ holds with probability 1, so the output is *exactly the SVD* of $\mathbf{A}$. In the general case, contributions from the singular modes beyond the first k will shift $\text{ran}(\mathbf{Y})$ away from the desired space spanned by the dominant k left singular vectors.

Randomised SVD:

Objective: Given an $m \times n$ matrix $\mathbf{A}$, find an approximate rank-k partial SVD:

$$\begin{array}{ccccc} \mathbf{A} & \approx & \mathbf{U} & \mathbf{D} & \mathbf{V}^* \\ m \times n & & m \times k & k \times k & k \times n \end{array}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and $\mathbf{D}$ is diagonal. (We assume $k \ll \min(m, n)$.)

(A) *Randomised sketching:*

A.1 Draw an $n \times k$ Gaussian random matrix $\mathbf{R}$. `R = randn(n,k)`

A.2 Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$. `Y = A * R`

A.3 Form an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\text{ran}(\mathbf{Q}) = \text{ran}(\mathbf{Y})$. `[Q, ~] = qr(Y)`

(B) *Deterministic post-processing:*

B.1 Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$. `B = Q' * A`

B.2 Form SVD of the matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$. `[Uhat, Sigma, V] = svd(B,'econ')`

B.3 Form the matrix $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$. `U = Q * Uhat`

Distortions in the randomised projections are fine, since all we need is a subspace that captures "the essential" part of the range. Pollution from unwanted singular modes is harmless, as long as we capture the dominant ones. By drawing p extra samples (for, say, $p = 5$ or $p = 10$), we make the risk of missing anything important essentially zero.

Randomised SVD:

Input: An $m \times n$ matrix $\mathbf{A}$, and a target rank k.	
Output: Rank-k factors $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times k$ random matrix $\mathbf{R}$.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Randomised SVD:

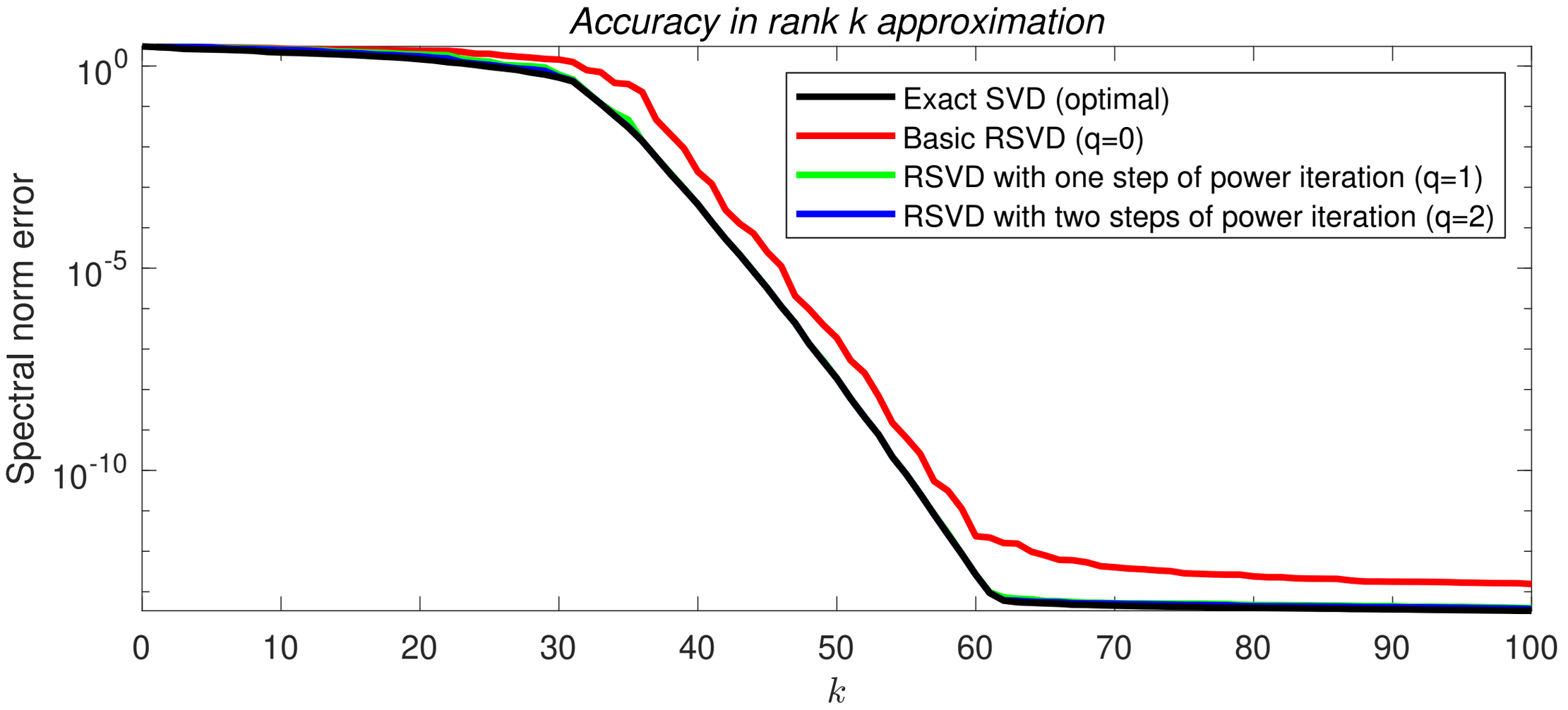
Input: An $m \times n$ matrix $\mathbf{A}$, and a target rank k.	
Output: Rank-k factors $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times k$ random matrix $\mathbf{R}$.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

- It is simple to adapt the scheme to the situation where the *tolerance is given,* and the rank has to be determined adaptively.

- Analogous schemes exist for computing "structure preserving" factorizations where a number of the columns/rows are chosen to serve as a basis for the column/row space. "Interpolative decomposition" / "CUR decomposition" / "skeletonization" / ...

- Accuracy of the basic scheme is good when the singular values decay reasonably fast. When they do not, the scheme can be combined with Krylov-type ideas: *Taking one or two steps of subspace iteration vastly improves the accuracy.* For instance, use the sampling matrix $\mathbf{Y} = \mathbf{AA}^*\mathbf{AG}$ instead of $\mathbf{Y} = \mathbf{AG}$.

Randomised SVD:



The plot shows the errors from the randomised range finder. To be precise, we plot

$$e_k = \|\mathbf{A} - \mathbf{P}_k\mathbf{A}\|,$$

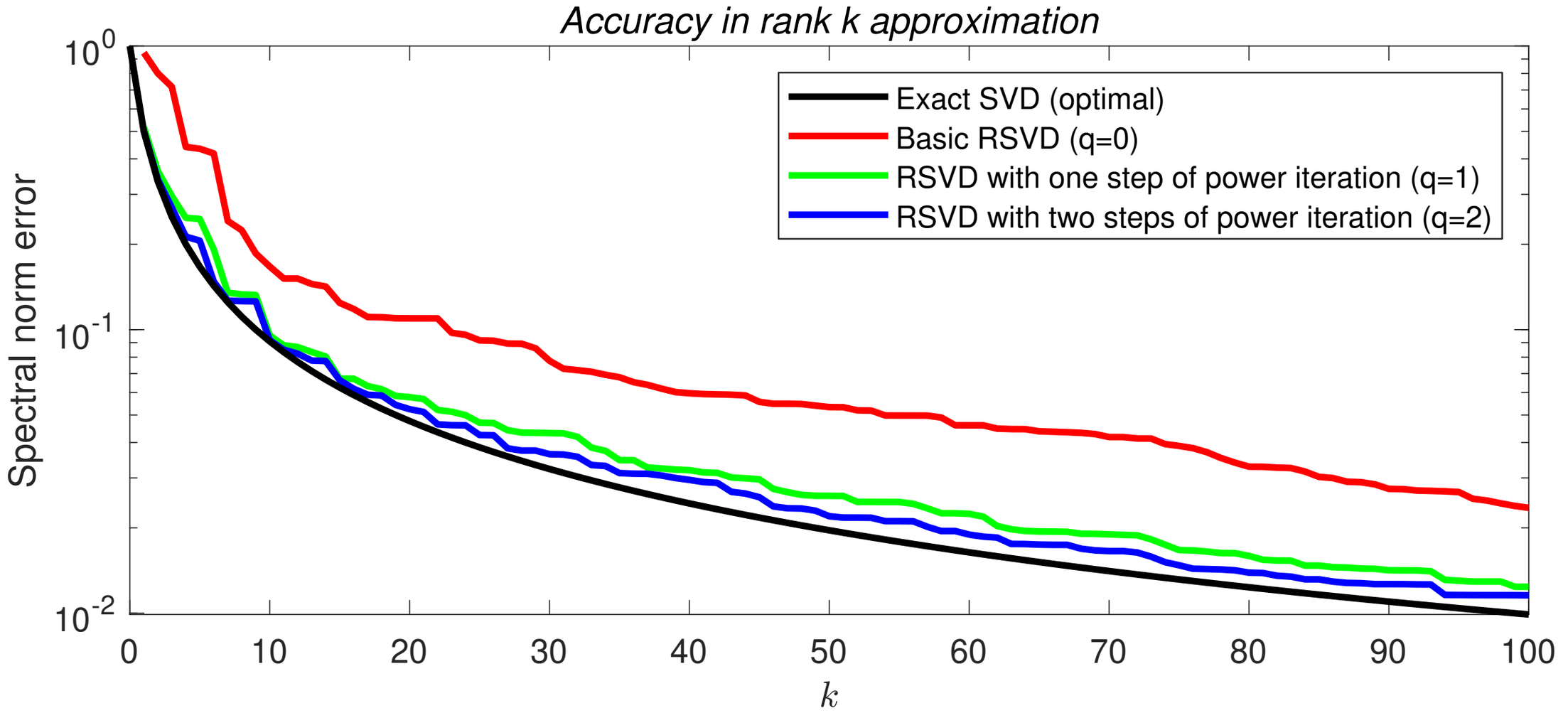
where $\mathbf{P}_k$ is the orthogonal projection onto the first k columns of

$$\mathbf{Y} = \left(\mathbf{A}\mathbf{A}^*\right)^q\mathbf{A}\mathbf{G},$$

and where $\mathbf{G}$ is a Gaussian random matrix.

The matrix $\mathbf{A}$ is an approximation to a scattering operator for a Helmholtz problem.

Randomised SVD:



The plot shows the errors from the randomised range finder. To be precise, we plot

$$e_k = \|\mathbf{A} - \mathbf{P}_k\mathbf{A}\|,$$

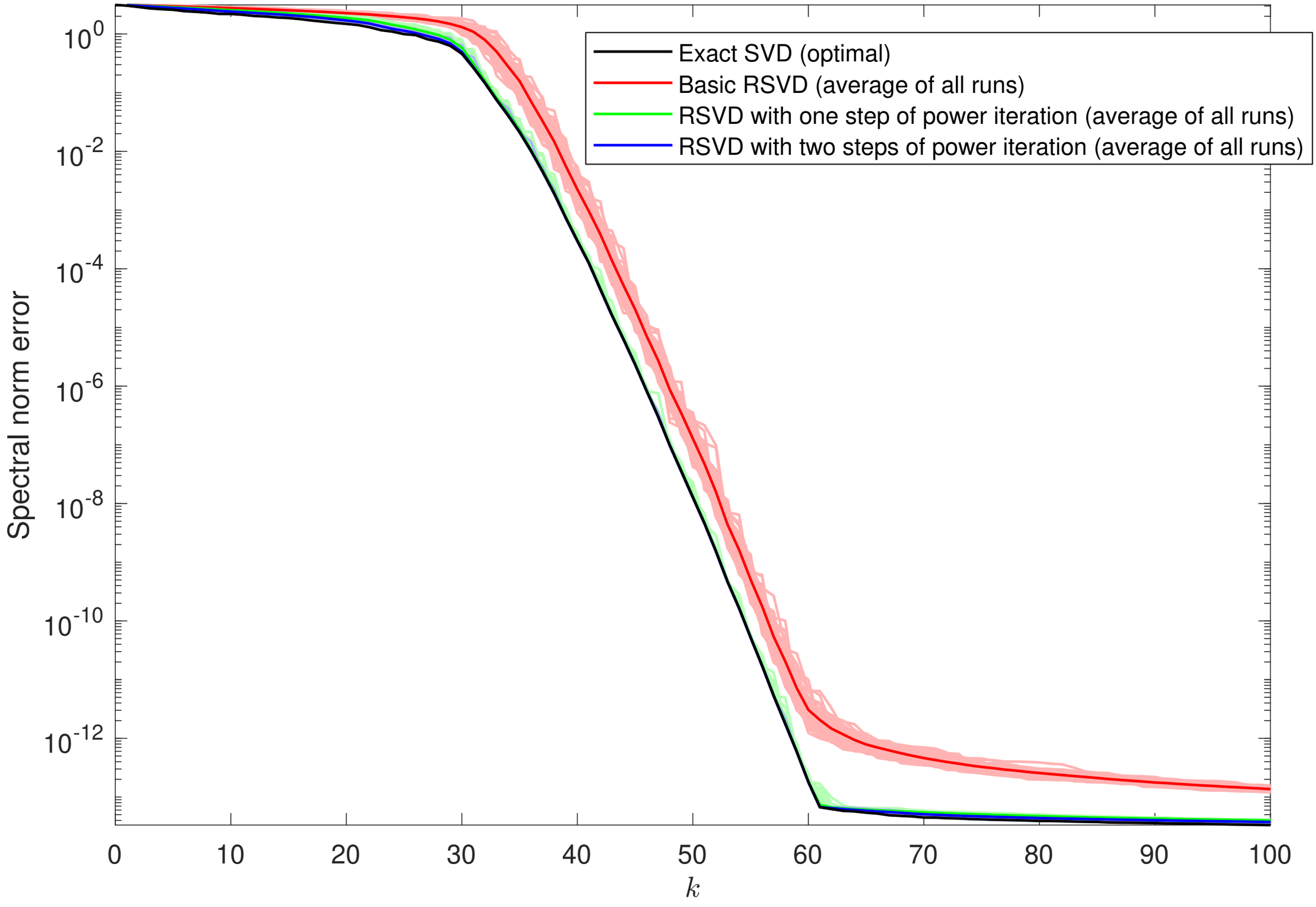
where $\mathbf{P}_k$ is the orthogonal projection onto the first k columns of

$$\mathbf{Y} = \left(\mathbf{A}\mathbf{A}^*\right)^q\mathbf{A}\mathbf{G},$$

and where **G** is a Gaussian random matrix.

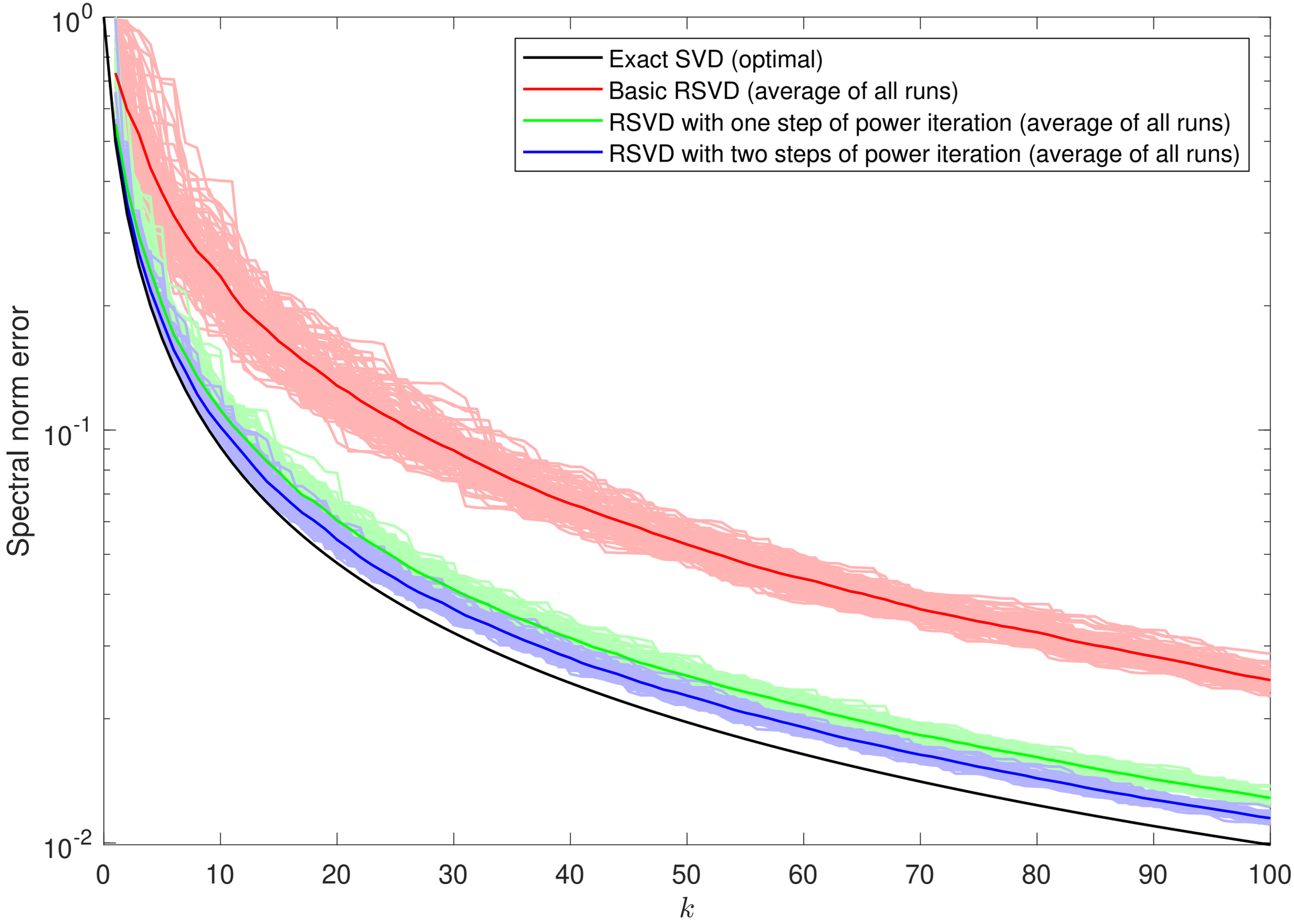
The matrix **A** now has singular values that decay slowly.

Randomised SVD: The same plot as before, but now showing 100 instantiations.



The darker lines show the mean errors across the 100 experiments.

Randomised SVD: The same plot as before, but now showing 100 instantiations.



The darker lines show the mean errors across the 100 experiments.

Cost of randomised methods vs. classical (deterministic) methods:

Case 1 — A is given as an array of numbers that fits in RAM ("small matrix"):

Classical methods (e.g. Golub-Businger) have cost $O(mnk)$. The basic randomised method described also has $O(mnk)$ cost, but with a lower pre-factor (and sometimes lower accuracy). However, the cost can be reduced to $O(mn\log(k))$ if a structured random matrix is used. For instance, $\mathbf{R}$ can be a sub-sampled randomised Fourier transform (SRFT), which can be applied rapidly using variations of the FFT.

Cost of randomised methods vs. classical (deterministic) methods:

Case 1 — A is given as an array of numbers that fits in RAM ("small matrix"):

Classical methods (e.g. Golub-Businger) have cost $O(mnk)$. The basic randomised method described also has $O(mnk)$ cost, but with a lower pre-factor (and sometimes lower accuracy). However, the cost can be reduced to $O(mn\log(k))$ if a structured random matrix is used. For instance, **R** can be a sub-sampled randomised Fourier transform (SRFT), which can be applied rapidly using variations of the FFT.

- The algorithm must be modified a bit beside replacing the random matrix.
- The SRFT leads to large speed-ups *for moderate matrix sizes.*
  For instance, for $m = n = 4000$, and $k \sim 10^2$, we observe about $\times 5$ speedup.
- In practice, accuracy is very similar to what you get from Gaussian random matrices.
- Theory is still quite weak.
- Many different "structured random projections" have been proposed: sub-sampled Hadamard transform, chains of Givens rotations, sparse projections, etc.

References: Ailon & Chazelle (2006); Liberty, Rokhlin, Tygert, and Woolfe (2006); Halko, Martinsson, Tropp (2011); Clarkson & Woodruff (2013).
Much subsequent work — "Fast Johnson-Lindenstrauss transform."

Cost of randomised methods vs. classical (deterministic) methods:

Case 1 — A is given as an array of numbers that fits in RAM ("small matrix"):

Classical methods (e.g. Golub-Businger) have cost $O(mnk)$. The basic randomised method described also has $O(mnk)$ cost, but with a lower pre-factor (and sometimes lower accuracy). However, the cost can be reduced to $O(mn\log(k))$ if a structured random matrix is used. For instance, $\mathbf{R}$ can be a sub-sampled randomised Fourier transform (SRFT), which can be applied rapidly using variations of the FFT.

Cost of randomised methods vs. classical (deterministic) methods:

Case 1 — A is given as an array of numbers that fits in RAM ("small matrix"):

Classical methods (e.g. Golub-Businger) have cost $O(mnk)$. The basic randomised method described also has $O(mnk)$ cost, but with a lower pre-factor (and sometimes lower accuracy). However, the cost can be reduced to $O(mn\log(k))$ if a structured random matrix is used. For instance, **R** can be a sub-sampled randomised Fourier transform (SRFT), which can be applied rapidly using variations of the FFT.

Case 2 — A is given as an array of numbers on disk ("large matrix"):

In this case, the relevant metric is memory access. Randomised methods access **A** via sweeps over the entire matrix. With slight modifications, the randomised method can be executed in a *single pass* over the matrix. High accuracy can be attained with a small number of passes (say two, three, four).

(In contrast, classical (deterministic) methods require "random" access to matrix elements...)

Cost of randomised methods vs. classical (deterministic) methods:

Case 1 — A is given as an array of numbers that fits in RAM ("small matrix"):

Classical methods (e.g. Golub-Businger) have cost $O(mnk)$. The basic randomised method described also has $O(mnk)$ cost, but with a lower pre-factor (and sometimes lower accuracy). However, the cost can be reduced to $O(mn\log(k))$ if a structured random matrix is used. For instance, **R** can be a sub-sampled randomised Fourier transform (SRFT), which can be applied rapidly using variations of the FFT.

Case 2 — A is given as an array of numbers on disk ("large matrix"):

In this case, the relevant metric is memory access. Randomised methods access **A** via sweeps over the entire matrix. With slight modifications, the randomised method can be executed in a *single pass* over the matrix. High accuracy can be attained with a small number of passes (say two, three, four).

(In contrast, classical (deterministic) methods require "random" access to matrix elements...)

Case 3 — A and $\mathbf{A}^*$ can be applied fast ("structured matrix"):

Think of **A** sparse, or sparse in the Fourier domain, or amenable to the Fast Multipole Method, etc. The classical competitor is in this case "Krylov methods". Randomised methods tend to be more robust, and easier to implement in massively parallel environments. They are more easily blocked to reduce communication. However, Krylov methods sometimes lead to higher accuracy.

Analysis

Input: An $m \times n$ matrix $\mathbf{A}$ and a target rank k.	
Output: Rank-k factors $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times k$ random matrix $\mathbf{R}$.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question: What is the error $e_k = \|\mathbf{A} - \mathbf{UDV}^*\|$? (Recall that $e_k = \|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|$.)

Input: An $m \times n$ matrix $\mathbf{A}$ and a target rank k.	
Output: Rank-k factors $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times k$ random matrix $\mathbf{R}$.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question: What is the error $e_k = \|\mathbf{A} - \mathbf{UDV}^*\|$? (Recall that $e_k = \|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|$.)

Eckart-Young theorem: e_k is bounded from below by the singular value σ_{k+1} of $\mathbf{A}$.

Input: An $m \times n$ matrix $\mathbf{A}$ and a target rank k.	
Output: Rank-k factors $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times k$ random matrix $\mathbf{R}$.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question: What is the error $e_k = \|\mathbf{A} - \mathbf{UDV}^*\|$? (Recall that $e_k = \|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|$.)

Eckart-Young theorem: e_k is bounded from below by the singular value σ_{k+1} of $\mathbf{A}$.

Question: Is e_k close to σ_{k+1}?

Input: An $m \times n$ matrix $\mathbf{A}$ and a target rank k.	
Output: Rank-k factors $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times k$ random matrix $\mathbf{R}$.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question: What is the error $e_k = \|\mathbf{A} - \mathbf{UDV}^*\|$? (Recall that $e_k = \|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|$.)

Eckart-Young theorem: e_k is bounded from below by the singular value σ_{k+1} of $\mathbf{A}$.

Question: Is e_k close to σ_{k+1}?

Answer: Lamentably, no. The expectation of $\frac{e_k}{\sigma_{k+1}}$ is large, and has very large variance.

Input: An $m \times n$ matrix **A** and a target rank k.	
Output: Rank-k factors **U**, **D**, and **V** in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times k$ random matrix **R**.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix **Q** s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question: What is the error $e_k = \|\mathbf{A} - \mathbf{UDV}^*\|$? (Recall that $e_k = \|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|$.)

Eckart-Young theorem: e_k is bounded from below by the singular value σ_{k+1} of **A**.

Question: Is e_k close to σ_{k+1}?

Answer: Lamentably, no. The expectation of $\frac{e_k}{\sigma_{k+1}}$ is large, and has very large variance.

Remedy: Over-sample *slightly*. Compute $k+p$ samples from the range of **A**.

It turns out that $p = 5$ or 10 is often sufficient. $p = k$ is almost always more than enough.

Input: An $m \times n$ matrix **A**, a target rank k, *and an over-sampling parameter p (say $p = 5$).*	
Output: Rank-$(k+p)$ factors **U**, **D**, and **V** in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times (k+p)$ random matrix **R**.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times (k+p)$ sample matrix $\mathbf{Y} = \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix **Q** s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Bound on the expectation of the error for Gaussian test matrices

Let $\mathbf{A}$ denote an $m \times n$ matrix with singular values $\{\sigma_j\}_{j=1}^{\min(m,n)}$.

Let k denote a target rank and let p denote an over-sampling parameter.

Let $\mathbf{R}$ denote an $n \times (k+p)$ Gaussian matrix.

Let $\mathbf{Q}$ denote the $m \times (k+p)$ matrix $\mathbf{Q} = \texttt{orth}(\mathbf{AR})$.

If $p \geq 2$, then

$$\mathbb{E}\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|_{\text{Frob}} \leq \left(1 + \frac{k}{p-1}\right)^{1/2} \left(\sum_{j=k+1}^{\min(m,n)} \sigma_j^2\right)^{1/2},$$

and

$$\mathbb{E}\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\| \leq \left(1 + \sqrt{\frac{k}{p-1}}\right) \sigma_{k+1} + \frac{e\sqrt{k+p}}{p} \left(\sum_{j=k+1}^{\min(m,n)} \sigma_j^2\right)^{1/2}.$$

Ref: Halko, Martinsson, Tropp, 2009 & 2011

Large deviation bound for the error for Gaussian test matrices

Let $\mathbf{A}$ denote an $m \times n$ matrix with singular values $\{\sigma_j\}_{j=1}^{\min(m,n)}$.

Let k denote a target rank and let p denote an over-sampling parameter.

Let $\mathbf{R}$ denote an $n \times (k+p)$ Gaussian matrix.

Let $\mathbf{Q}$ denote the $m \times (k+p)$ matrix $\mathbf{Q} = \mathtt{orth}(\mathbf{AR})$.

If $p \geq 4$, and u and t are such that $u \geq 1$ and $t \geq 1$, then

$$\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\| \leq \left(1 + t\sqrt{\frac{3k}{p+1}} + u\,t\,\frac{e\sqrt{k+p}}{p+1}\right)\sigma_{k+1} + \frac{t\,e\sqrt{k+p}}{p+1}\left(\sum_{j>k}\sigma_j^2\right)^{1/2}$$

except with probability at most $2\,t^{-p} + e^{-u^2/2}$.

Ref: Halko, Martinsson, Tropp, 2009 & 2011; Martinsson, Rokhlin, Tygert (2006)

---

u and t parameterize "bad" events — large u, t is bad, but unlikely.

Certain choices of t and u lead to simpler results. For instance,

$$\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\| \leq \left(1 + 16\sqrt{1 + \frac{k}{p+1}}\right)\sigma_{k+1} + 8\frac{\sqrt{k+p}}{p+1}\left(\sum_{j>k}\sigma_j^2\right)^{1/2},$$

except with probability at most $3\,e^{-p}$.

Large deviation bound for the error for Gaussian test matrices

Let $\mathbf{A}$ denote an $m \times n$ matrix with singular values $\{\sigma_j\}_{j=1}^{\min(m,n)}$.

Let k denote a target rank and let p denote an over-sampling parameter.

Let $\mathbf{R}$ denote an $n \times (k+p)$ Gaussian matrix.

Let $\mathbf{Q}$ denote the $m \times (k+p)$ matrix $\mathbf{Q} = \mathtt{orth}(\mathbf{AR})$.

If $p \geq 4$, and u and t are such that $u \geq 1$ and $t \geq 1$, then

$$\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\| \leq \left(1 + t\sqrt{\frac{3k}{p+1}} + u\,t\,\frac{e\sqrt{k+p}}{p+1}\right)\,\sigma_{k+1} + \frac{t\,e\sqrt{k+p}}{p+1}\left(\sum_{j>k}\sigma_j^2\right)^{1/2}$$

except with probability at most $2\,t^{-p} + e^{-u^2/2}$.

Ref: Halko, Martinsson, Tropp, 2009 & 2011; Martinsson, Rokhlin, Tygert (2006)

---

u and t parameterize "bad" events — large u, t is bad, but unlikely.

Certain choices of t and u lead to simpler results. For instance,

$$\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\| \leq \left(1 + 6\sqrt{(k+p)\cdot p\log p}\right)\,\sigma_{k+1} + 3\sqrt{k+p}\left(\sum_{j>k}\sigma_j^2\right)^{1/2},$$

except with probability at most $3\,p^{-p}$.

Proofs — Overview:

Let **A** denote an $m \times n$ matrix with singular values $\{\sigma_j\}_{j=1}^{\min(m,n)}$.
Let k denote a target rank and let p denote an over-sampling parameter. Set $\ell = k + p$.
Let **R** denote an $n \times \ell$ "test matrix", and let **Q** denote the $m \times \ell$ matrix $\mathbf{Q} = \texttt{orth}(\mathbf{AR})$.

We seek to bound the error $e_k = e_k(\mathbf{A}, \mathbf{R}) = \|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|$, which is a random variable.

1. Make no assumption on **R**. Construct a deterministic bound of the form

$$\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\| \leq \cdots \mathbf{A} \cdots \mathbf{R} \cdots$$

2. Assume that **R** is drawn from a standardized Gaussian distribution.
   Take expectations of the deterministic bound to attain a bound of the form

$$\mathbb{E}\big[\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|\big] \leq \cdots \mathbf{A} \cdots$$

3. Assume that **R** is drawn from a standardized Gaussian distribution.
   Take expectations of the deterministic bound conditioned on "bad behavior" in **R** to get that

$$\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\| \leq \cdots \mathbf{A} \cdots$$

   holds with probability at least $\cdots$.

Part 1 (out of 3) — deterministic bound:

Let **A** denote an $m \times n$ matrix with singular values $\{\sigma_j\}_{j=1}^{\min(m,n)}$.

Let k denote a target rank and let p denote an over-sampling parameter. Set $\ell = k + p$.

Let **R** denote an $n \times \ell$ "test matrix", and let **Q** denote the $m \times \ell$ matrix $\mathbf{Q} = \texttt{orth}(\mathbf{AR})$.

Partition the SVD of **A** as follows:

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

Define $\mathbf{R}_1$ and $\mathbf{R}_2$ via

$$\underset{k \times (k+p)}{\mathbf{R}_1} = \underset{k \times n}{\mathbf{V}_1^*} \; \underset{n \times (k+p)}{\mathbf{R}} \quad \text{and} \quad \underset{(n-k) \times (k+p)}{\mathbf{R}_2} = \underset{(n-k) \times n}{\mathbf{V}_2^*} \; \underset{n \times (k+p)}{\mathbf{R}}.$$

Theorem: [HMT2009,HMT2011] Assuming that $\mathbf{R}_1$ is not singular, it holds that

$$|||\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}|||^2 \leq \underbrace{|||\mathbf{D}_2|||^2}_{\text{theoretically minimal error}} + |||\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger|||^2.$$

Here, $|||\cdot|||$ represents either ℓ^2-operator norm, or the Frobenius norm.

Note: A similar (but weaker) result appears in Boutsidis, Mahoney, Drineas (2009).

Recall: $\mathbf{A} = \mathbf{U}\begin{bmatrix}\mathbf{D}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_2\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^*\\ \mathbf{V}_2^*\end{bmatrix}$, $\begin{bmatrix}\mathbf{R}_1\\ \mathbf{R}_2\end{bmatrix} = \begin{bmatrix}\mathbf{V}_1^*\mathbf{R}\\ \mathbf{V}_2^*\mathbf{R}\end{bmatrix}$, $\mathbf{Y} = \mathbf{AR}$, $\mathbf{P}$ proj$^{\mathrm{n}}$ onto $\mathrm{Ran}(\mathbf{Y})$.

Thm: Suppose $\mathbf{D}_1\mathbf{R}_1$ has full rank. Then $\|\mathbf{A} - \mathbf{PA}\|^2 \leq \|\mathbf{D}_2\|^2 + \|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\|^2$.

Proof: The problem is rotationally invariant $\Rightarrow$ We can assume $\mathbf{U} = \mathbf{I}$ and so $\mathbf{A} = \mathbf{DV}^*$.

Simple calculation: $\|(\mathbf{I} - \mathbf{P})\mathbf{A}\|^2 = \|\mathbf{A}^*(\mathbf{I} - \mathbf{P})^2\mathbf{A}\| = \|\mathbf{D}(\mathbf{I} - \mathbf{P})\mathbf{D}\|$.

$$\mathrm{Ran}(\mathbf{Y}) = \mathrm{Ran}\left(\begin{bmatrix}\mathbf{D}_1\mathbf{R}_1\\ \mathbf{D}_2\mathbf{R}_2\end{bmatrix}\right) = \mathrm{Ran}\left(\begin{bmatrix}\mathbf{I}\\ \mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\mathbf{D}_1\end{bmatrix}\mathbf{D}_1\mathbf{R}_1\right) = \mathrm{Ran}\left(\begin{bmatrix}\mathbf{I}\\ \mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\mathbf{D}_1\end{bmatrix}\right)$$

Set $\mathbf{F} = \mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\mathbf{D}_1^{-1}$. Then $\mathbf{P} = \begin{bmatrix}\mathbf{I}\\ \mathbf{F}\end{bmatrix}(\mathbf{I} + \mathbf{F}^*\mathbf{F})^{-1}[\mathbf{I}\ \mathbf{F}^*]$. (Compare to $\mathbf{P}_{\mathrm{ideal}} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$.)

Use properties of psd matrices: $\mathbf{I} - \mathbf{P} \preccurlyeq \cdots \preccurlyeq \begin{bmatrix}\mathbf{F}^*\mathbf{F} & -(\mathbf{I} + \mathbf{F}^*\mathbf{F})^{-1}\mathbf{F}^*\\ -\mathbf{F}(\mathbf{I} + \mathbf{F}^*\mathbf{F})^{-1} & \mathbf{I}\end{bmatrix}$

Conjugate by $\mathbf{D}$ to get $\mathbf{D}(\mathbf{I} - \mathbf{P})\mathbf{D} \preccurlyeq \begin{bmatrix}\mathbf{D}_1\mathbf{F}^*\mathbf{F}\mathbf{D}_1 & -\mathbf{D}_1(\mathbf{I} + \mathbf{F}^*\mathbf{F})^{-1}\mathbf{F}^*\mathbf{D}_2\\ -\mathbf{D}_2\mathbf{F}(\mathbf{I} + \mathbf{F}^*\mathbf{F})^{-1}\mathbf{D}_1 & \mathbf{D}_2^2\end{bmatrix}$

Diagonal dominance: $\|\mathbf{D}(\mathbf{I} - \mathbf{P})\mathbf{D}\| \leq \|\mathbf{D}_1\mathbf{F}^*\mathbf{F}\mathbf{D}_1\| + \|\mathbf{D}_2^2\| = \|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\|^2 + \|\mathbf{D}_2\|^2$.

Part 2 (out of 3) — bound on expectation of error when R is Gaussian:

Let **A** denote an $m \times n$ matrix with singular values $\{\sigma_j\}_{j=1}^{\min(m,n)}$.
Let k denote a target rank and let p denote an over-sampling parameter. Set $\ell = k + p$.
Let **R** denote an $n \times \ell$ "test matrix", and let **Q** denote the $m \times \ell$ matrix $\mathbf{Q} = \mathtt{orth}(\mathbf{AR})$.

Recall: $|||\mathbf{A} - \mathbf{QQ}^*\mathbf{A}|||^2 \leq |||\mathbf{D}_2|||^2 + |||\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger|||^2$, where $\mathbf{R}_1 = \mathbf{V}_1^*\,\mathbf{R}$ and $\mathbf{R}_2 = \mathbf{V}_2^*\,\mathbf{R}$.

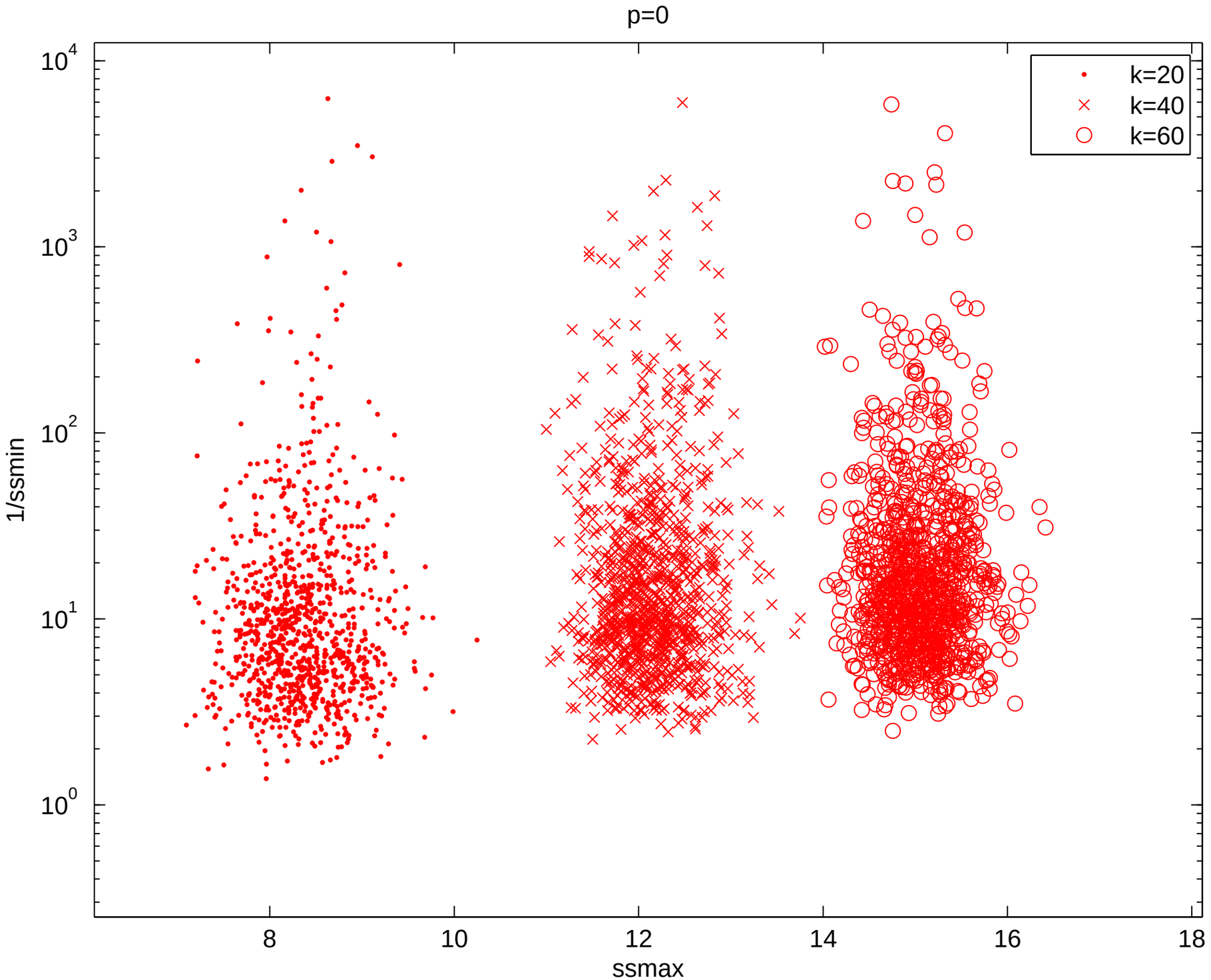
Assumption: **R** is drawn from a standardized Gaussian distribution.

Since the Gaussian distribution is rotationally invariant, the matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ also have a Gaussian distribution. (As a consequence, the matrices **U** and **V** do not enter the analysis and one could simply assume that **A** is diagonal, $\mathbf{A} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots)$.)

What is the distribution of $\mathbf{R}_1^\dagger$ when $\mathbf{R}_1$ is a $k \times (k+p)$ Gaussian matrix?

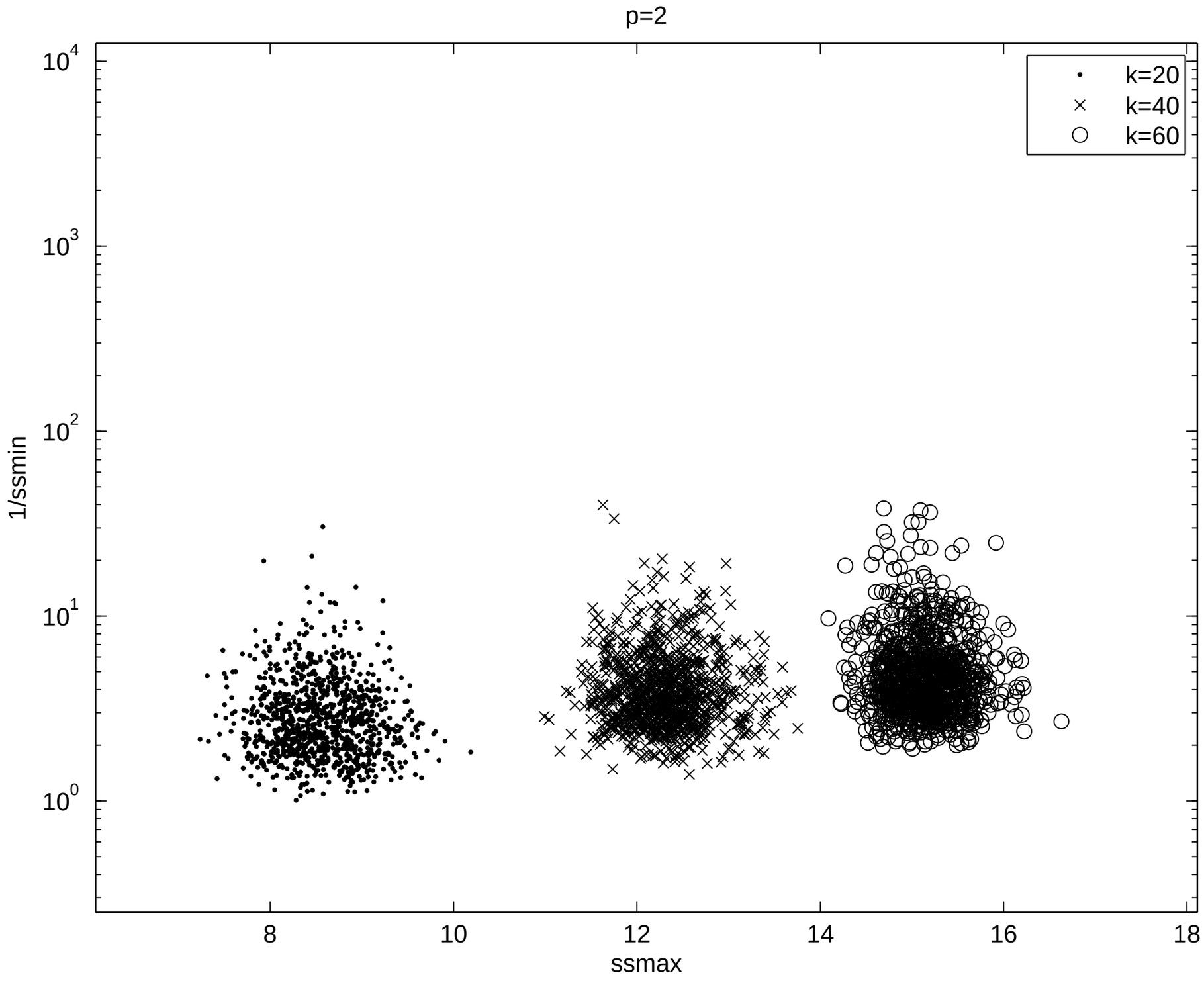
If $p = 0$, then $\|\mathbf{R}_1^\dagger\|$ is typically large, and is very unstable.

Scatter plot showing distribution of $1/\sigma_{\min}$ for $k \times (k+p)$ Gaussian matrices. $p = 0$



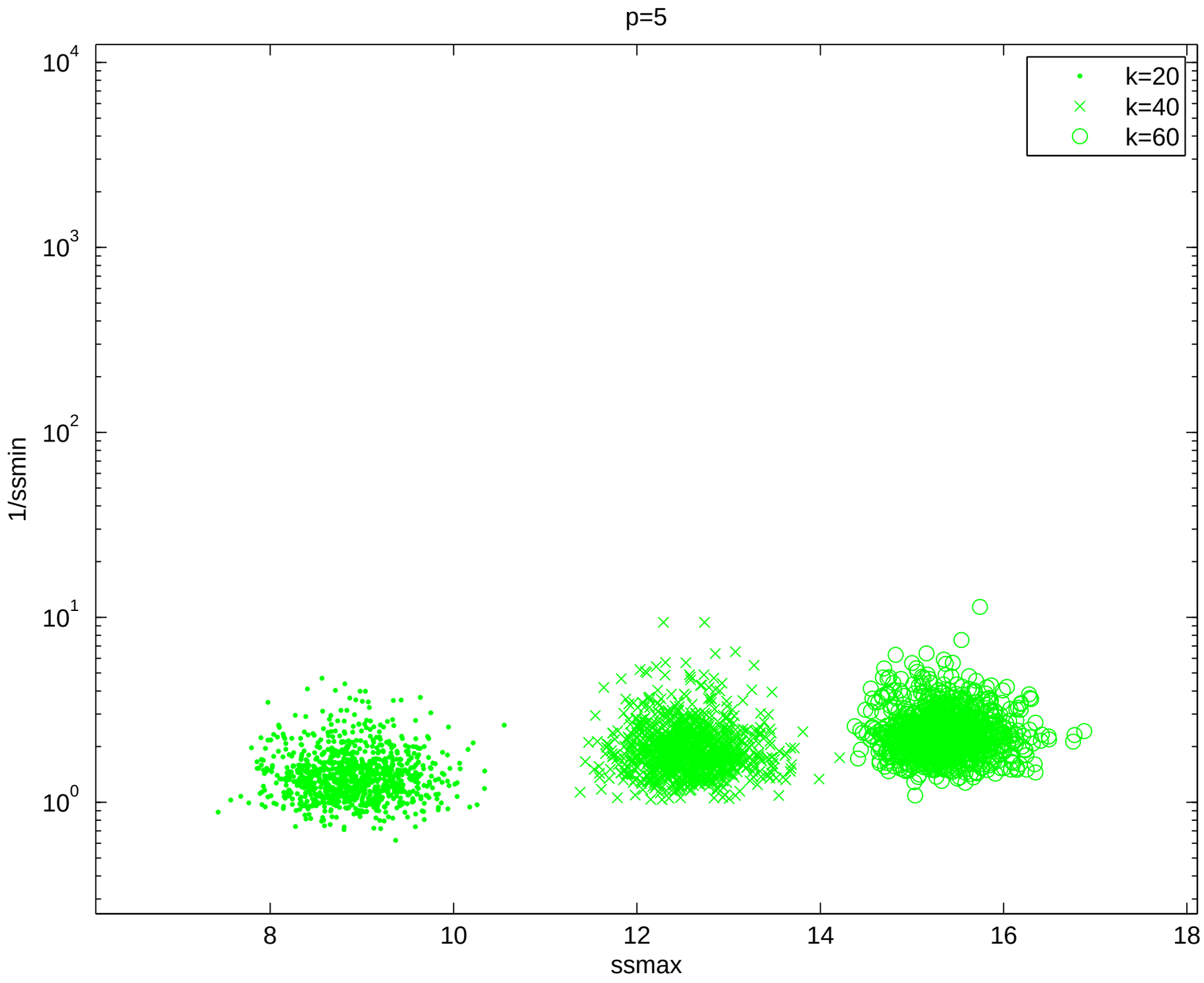
$1/\sigma_{\min}$ *is plotted against* $\sigma_{\max}$.

Scatter plot showing distribution of $1/\sigma_{\min}$ for $k \times (k+p)$ Gaussian matrices. $p = 2$



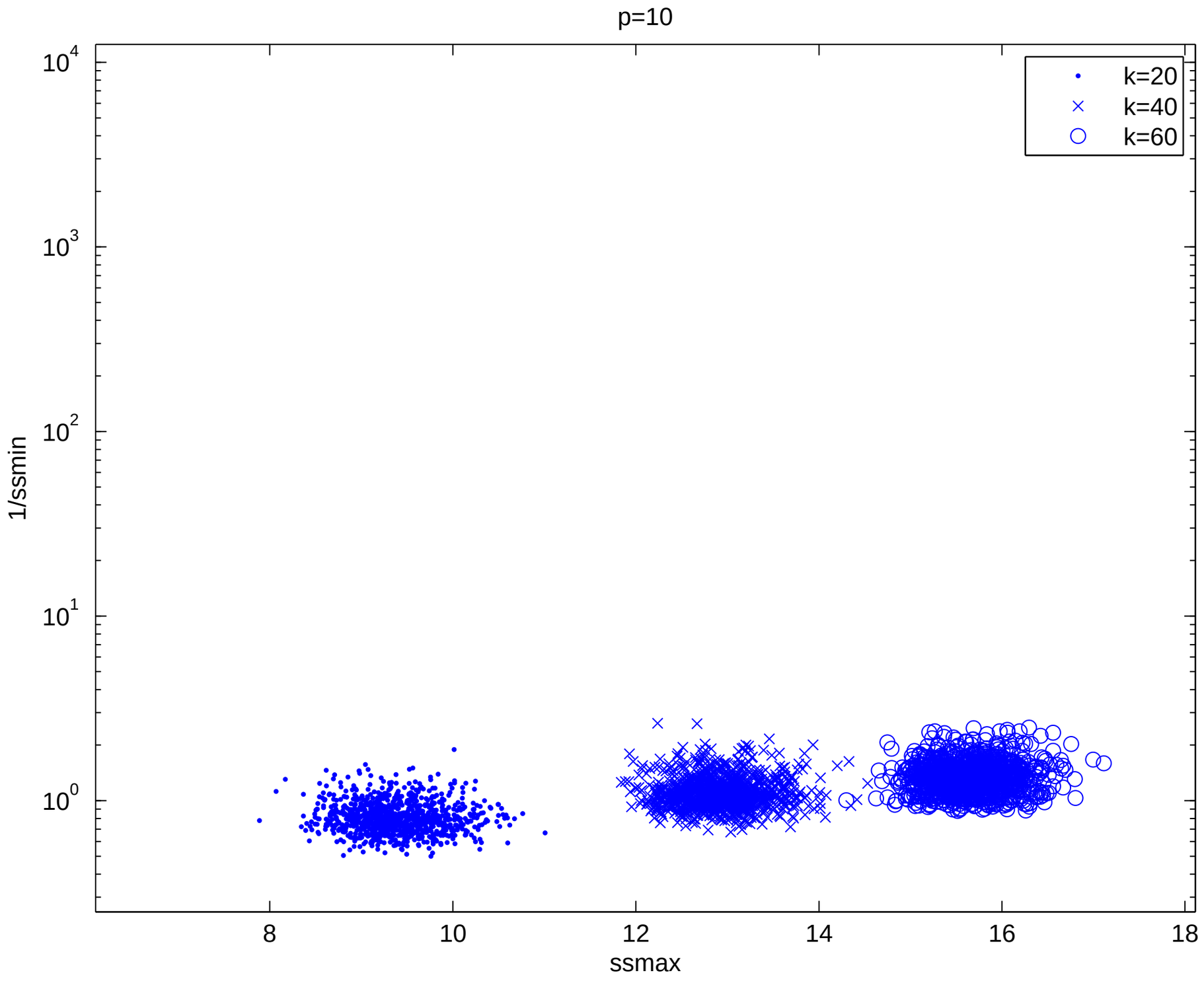
$1/\sigma_{\min}$ is plotted against $\sigma_{\max}$.

Scatter plot showing distribution of $1/\sigma_{\min}$ for $k \times (k+p)$ Gaussian matrices. $p = 5$



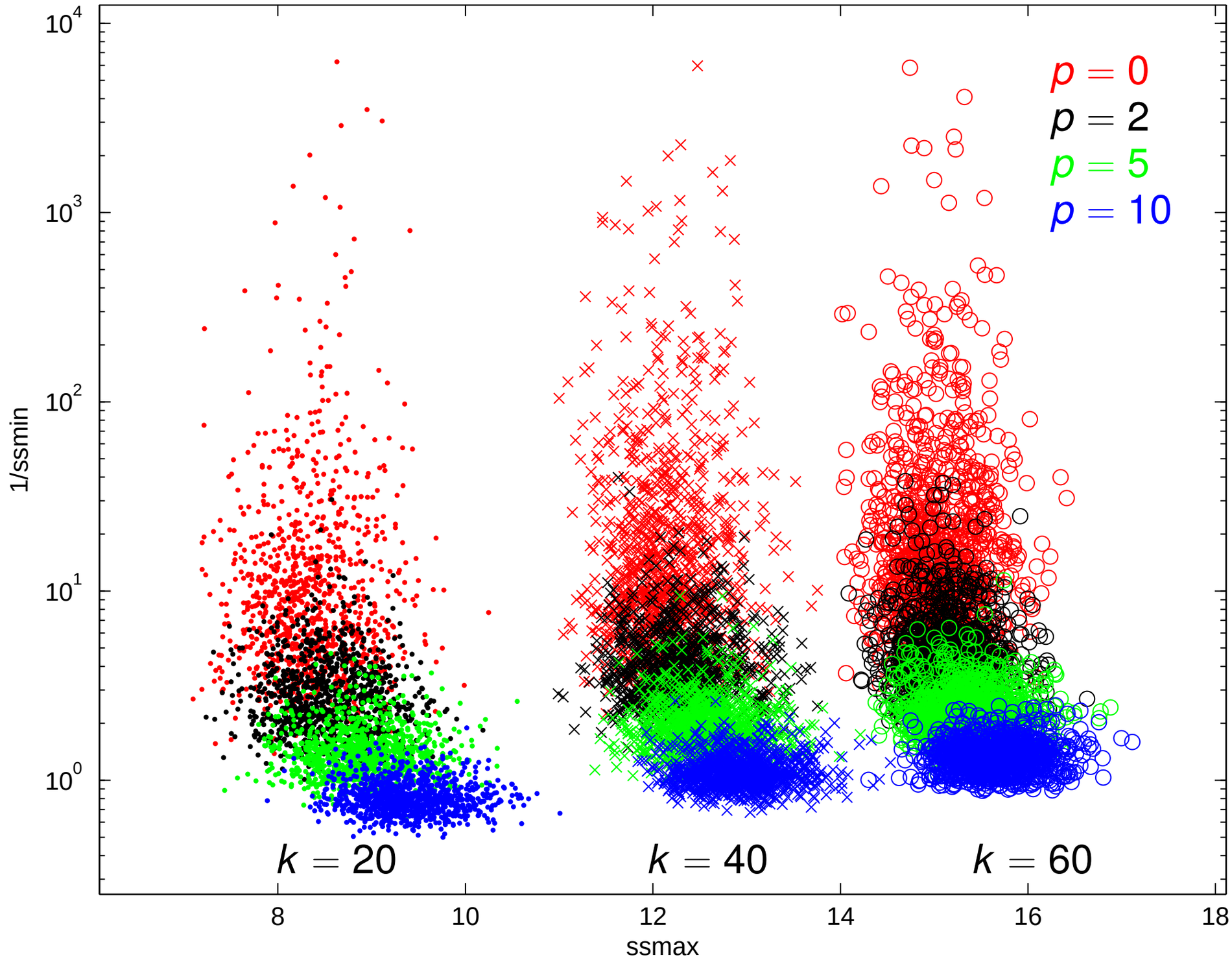
$1/\sigma_{\min}$ is plotted against $\sigma_{\max}$.

Scatter plot showing distribution of $1/\sigma_{\min}$ for $k \times (k+p)$ Gaussian matrices. $p = 10$



$1/\sigma_{\min}$ *is plotted against* $\sigma_{\max}$.

Scatter plot showing distribution of $k \times (k + p)$ Gaussian matrices.



$1/\sigma_{\min}$ *is plotted against* $\sigma_{\max}$.

Simplistic proof that a rectangular Gaussian matrix is well-conditioned:

Let **G** denote a $k \times \ell$ Gaussian matrix where $k < \ell$.
Let "g" denote a generic $\mathcal{N}(0,1)$ variable and "r_j^2" denote a generic χ_j^2 variable. Then

$$\mathbf{G} \sim \begin{bmatrix} g & g & g & g & g & g & \cdots \\ g & g & g & g & g & g & \cdots \\ g & g & g & g & g & g & \cdots \\ g & g & g & g & g & g & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \end{bmatrix} \sim \begin{bmatrix} r_\ell & 0 & 0 & 0 & 0 & 0 & \cdots \\ g & g & g & g & g & g & \cdots \\ g & g & g & g & g & g & \cdots \\ g & g & g & g & g & g & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \end{bmatrix}$$

$$\sim \begin{bmatrix} r_\ell & 0 & 0 & 0 & 0 & 0 & \cdots \\ r_{k-1} & g & g & g & g & g & \cdots \\ 0 & g & g & g & g & g & \cdots \\ 0 & g & g & g & g & g & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \end{bmatrix} \sim \begin{bmatrix} r_\ell & 0 & 0 & 0 & 0 & \cdots \\ r_{k-1} & r_{\ell-1} & 0 & 0 & 0 & \cdots \\ 0 & g & g & g & g & \cdots \\ 0 & g & g & g & g & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix}$$

$$\sim \begin{bmatrix} r_\ell & 0 & 0 & 0 & 0 & \cdots \\ r_{k-1} & r_{\ell-1} & 0 & 0 & 0 & \cdots \\ 0 & r_{k-2} & g & g & g & \cdots \\ 0 & 0 & g & g & g & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix} \sim \cdots \sim \begin{bmatrix} r_\ell & 0 & 0 & 0 & 0 & \cdots \\ r_{k-1} & r_{\ell-1} & 0 & 0 & 0 & \cdots \\ 0 & r_{k-2} & r_{\ell-2} & 0 & 0 & \cdots \\ 0 & 0 & r_{k-3} & r_{\ell-3} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix}$$

Gershgorin's circle theorem will now show that **G** is well-conditioned if, e.g., $\ell = 2k$.

More sophisticated methods are required to get to $\ell = k + 2$.

Some results on Gaussian matrices. *Adapted from HMT 2009/2011; Gordon (1985,1988) for Proposition 1; Chen & Dongarra (2005) for Propositions 2 and 4; Bogdanov (1998) for Proposition 3.*

Proposition 1: Let $\mathbf{G}$ be a Gaussian matrix. Then

$$\begin{aligned}\left(\mathbb{E}\left[\|\mathbf{SGT}\|_{\mathrm{F}}^2\right]\right)^{1/2} &\leq \|\mathbf{S}\|_{\mathrm{F}}\,\|\mathbf{T}\|_{\mathrm{F}} \\ \mathbb{E}\left[\|\mathbf{SGT}\|\right] &\leq \|\mathbf{S}\|\,\|\mathbf{T}\|_{\mathrm{F}} + \|\mathbf{S}\|_{\mathrm{F}}\,\|\mathbf{T}\|\end{aligned}$$

Proposition 2: Let $\mathbf{G}$ be a Gaussian matrix of size $k \times k+p$ where $p \geq 2$. Then

$$\begin{aligned}\left(\mathbb{E}\left[\|\mathbf{G}^{\dagger}\|_{\mathrm{F}}^2\right]\right)^{1/2} &\leq \sqrt{\frac{k}{p-1}} \\ \mathbb{E}\left[\|\mathbf{G}^{\dagger}\|\right] &\leq \frac{e\sqrt{k+p}}{p}.\end{aligned}$$

Proposition 3: Suppose h is Lipschitz $|h(\mathbf{X}) - h(\mathbf{Y})| \leq L\|\mathbf{X}-\mathbf{Y}\|_{\mathrm{F}}$ and $\mathbf{G}$ is Gaussian. Then

$$\mathbb{P}\left[h(\mathbf{G}) > \mathbb{E}[h(\mathbf{G})] + L\,u\right] \leq e^{-u^2/2}.$$

Proposition 4: Suppose $\mathbf{G}$ is Gaussian of size $k \times k+p$ with $p \geq 4$. Then for $t \geq 1$:

$$\begin{aligned}\mathbb{P}\left[\|\mathbf{G}^{\dagger}\|_{\mathrm{F}} \geq \sqrt{\frac{3k}{p+1}}t\right] &\leq t^{-p} \\ \mathbb{P}\left[\|\mathbf{G}^{\dagger}\| \geq \frac{e\sqrt{k+p}}{p+1}t\right] &\leq t^{-(p+1)}\end{aligned}$$

Recall: $\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|^2 \leq \|\mathbf{D}_2\|^2 + \|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\|^2$, where $\mathbf{R}_1$ and $\mathbf{R}_2$ are Gaussian and $\mathbf{R}_1$ is $k \times k + p$.

Theorem: $$\mathbb{E}\big[\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|\big] \leq \left(1 + \sqrt{\frac{k}{p-1}}\right)\sigma_{k+1} + \frac{e\sqrt{k+p}}{p}\left(\sum_{j=k+1}^{\min(m,n)} \sigma_j^2\right)^{1/2}.$$

Proof: First observe that

$$\mathbb{E}\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\| = \mathbb{E}\big(\|\mathbf{D}_2\|^2 + \|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\|^2\big)^{1/2} \leq \|\mathbf{D}_2\| + \mathbb{E}\|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\|.$$

Condition on $\mathbf{R}_1$ and use Proposition 1:

$$\begin{aligned}\mathbb{E}\|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\| &\leq \mathbb{E}\big[\|\mathbf{D}_2\|\,\|\mathbf{R}_1^\dagger\|_{\mathrm{F}} + \|\mathbf{D}_2\|_{\mathrm{F}}\,\|\mathbf{R}_1^\dagger\|\big] \\ &\leq \{\text{Hölder}\} \leq \|\mathbf{D}_2\|\left(\mathbb{E}\|\mathbf{R}_1^\dagger\|_{\mathrm{F}}^2\right)^{1/2} + \|\mathbf{D}_2\|_{\mathrm{F}}\,\mathbb{E}\|\mathbf{R}_1^\dagger\|.\end{aligned}$$

Proposition 2 now provides bounds for $\mathbb{E}\|\mathbf{R}_1^\dagger\|_{\mathrm{F}}^2$ and $\mathbb{E}\|\mathbf{R}_1^\dagger\|$ and we get

$$\mathbb{E}\|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\| \leq \sqrt{\frac{k}{p-1}}\|\mathbf{D}_2\| + \frac{e\sqrt{k+p}}{p}\|\mathbf{D}_2\|_{\mathrm{F}} = \sqrt{\frac{k}{p-1}}\sigma_{k+1} + \frac{e\sqrt{k+p}}{p}\left(\sum_{j>k}\sigma_j^2\right)^{1/2}.$$

Some results on Gaussian matrices. *Adapted from HMT2009/2011; Gordon (1985,1988) for Proposition 1; Chen & Dongarra (2005) for Propositions 2 and 4; Bogdanov (1998) for Proposition 3.*

Proposition 1: Let $\mathbf{G}$ be a Gaussian matrix. Then

$$\begin{aligned}\left(\mathbb{E}\left[\|\mathbf{SGT}\|_{\mathrm{F}}^2\right]\right)^{1/2} &\leq \|\mathbf{S}\|_{\mathrm{F}}\,\|\mathbf{T}\|_{\mathrm{F}} \\ \mathbb{E}\left[\|\mathbf{SGT}\|\right] &\leq \|\mathbf{S}\|\,\|\mathbf{T}\|_{\mathrm{F}} + \|\mathbf{S}\|_{\mathrm{F}}\,\|\mathbf{T}\|\end{aligned}$$

Proposition 2: Let $\mathbf{G}$ be a Gaussian matrix of size $k \times k+p$ where $p \geq 2$. Then

$$\begin{aligned}\left(\mathbb{E}\left[\|\mathbf{G}^\dagger\|_{\mathrm{F}}^2\right]\right)^{1/2} &\leq \sqrt{\frac{k}{p-1}} \\ \mathbb{E}\left[\|\mathbf{G}^\dagger\|\right] &\leq \frac{e\sqrt{k+p}}{p}.\end{aligned}$$

Proposition 3: Suppose h is Lipschitz $|h(\mathbf{X}) - h(\mathbf{Y})| \leq L\|\mathbf{X}-\mathbf{Y}\|_{\mathrm{F}}$ and $\mathbf{G}$ is Gaussian. Then

$$\mathbb{P}\left[h(\mathbf{G}) > \mathbb{E}[h(\mathbf{G})] + L\,u\right] \leq e^{-u^2/2}.$$

Proposition 4: Suppose $\mathbf{G}$ is Gaussian of size $k \times k+p$ with $p \geq 4$. Then for $t \geq 1$:

$$\begin{aligned}\mathbb{P}\left[\|\mathbf{G}^\dagger\|_{\mathrm{F}} \geq \sqrt{\frac{3k}{p+1}}t\right] &\leq t^{-p} \\ \mathbb{P}\left[\|\mathbf{G}^\dagger\| \geq \frac{e\sqrt{k+p}}{p+1}t\right] &\leq t^{-(p+1)}\end{aligned}$$

Recall: $\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|^2 \leq \|\mathbf{D}_2\|^2 + \|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\|^2$, where $\mathbf{R}_1$ and $\mathbf{R}_2$ are Gaussian and $\mathbf{R}_1$ is $k \times k+p$.

Theorem: With probability at least $1 - 2\,t^{-p} - e^{-u^2/2}$ it holds that

$$\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\| \leq \left(1 + t\sqrt{\frac{3k}{p+1}} + u\,t\,\frac{e\sqrt{k+p}}{p+1}\right)\,\sigma_{k+1} + \frac{t\,e\sqrt{k+p}}{p+1}\left(\sum_{j>k}\sigma_j^2\right)^{1/2}.$$

Proof: Set $E_t = \left\{\|\mathbf{R}_1\| \leq \frac{e\sqrt{k+p}}{p+1}t \quad \text{and} \quad \|\mathbf{R}_1^\dagger\|_{\mathrm{F}} \leq \sqrt{\frac{3k}{p+1}}\,t\right\}$. By Proposition 4: $\mathbb{P}(E_t^{\mathrm{c}}) \leq 2\,t^{-p}$.

Set $h(\mathbf{X}) = \|\mathbf{D}_2\mathbf{X}\mathbf{R}_1^\dagger\|$. A direct calculation shows

$$|h(\mathbf{X}) - h(\mathbf{Y})| \leq \|\mathbf{D}_2\|\,\|\mathbf{R}_1^\dagger\|\,\|\mathbf{X} - \mathbf{y}\|_{\mathrm{F}}.$$

Hold $\mathbf{R}_1$ fixed and take the expectation on $\mathbf{R}_2$. Then Proposition 1 applies and so

$$\mathbb{E}\big[h(\mathbf{R}_2) \;\big|\; \mathbf{R}_1\big] \leq \|\mathbf{D}_2\|\,\|\mathbf{R}_1^\dagger\|_{\mathrm{F}} + \|\mathbf{D}_2\|_{\mathrm{F}}\,\|\mathbf{R}_1^\dagger\|.$$

Now use Proposition 3 (concentration of measure)

$$\mathbb{P}\Big[\underbrace{\|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\|}_{=h(\mathbf{R}_2)} > \underbrace{\|\mathbf{D}_2\|\,\|\mathbf{R}_1^\dagger\|_{\mathrm{F}} + \|\mathbf{D}_2\|_{\mathrm{F}}\,\|\mathbf{R}_1^\dagger\|}_{=\mathbb{E}[h(\mathbf{R}_2)]} + \underbrace{\|\mathbf{D}_2\|\,\|\mathbf{R}_1^\dagger\|}_{=L}\,u \;\Big|\; E_t\Big] < e^{-u^2/2}.$$

When E_t holds true, we have bounds on the "badness" of $\mathbf{R}_1^\dagger$:

$$\mathbb{P}\left[\|\mathbf{D}_2\mathbf{R}_2\mathbf{R}_1^\dagger\| > \|\mathbf{D}_2\|\sqrt{\frac{3k}{p+1}}t + \|\mathbf{D}_2\|_{\mathrm{F}}\,\frac{e\sqrt{k+p}}{p+1}t + \|\mathbf{D}_2\|\,\frac{e\sqrt{k+p}}{p+1}\,ut \;\middle|\; E_t\right] < e^{-u^2/2}.$$

The theorem is obtained by using $\mathbb{P}(E_t^{\mathrm{c}}) \leq 2\,t^{-p}$ to remove the conditioning of E_t.

Power method for improving accuracy:

The error depends on how quickly the singular values decay. Recall that

$$\mathbb{E}\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\| \leq \left(1 + \sqrt{\frac{k}{p-1}}\right)\sigma_{k+1} + \frac{e\sqrt{k+p}}{p}\left(\sum_{j=k+1}^{\min(m,n)} \sigma_j^2\right)^{1/2}.$$

The faster the singular values decay — the stronger the relative weight of the dominant modes in the samples.

Idea: The matrix $(\mathbf{A}\mathbf{A}^*)^q\,\mathbf{A}$ has the same left singular vectors as $\mathbf{A}$, and singular values

$$\sigma_j((\mathbf{A}\mathbf{A}^*)^q\,\mathbf{A}) = (\sigma_j(\mathbf{A}))^{2q+1}.$$

Much faster decay — so let us use the sample matrix

$$\mathbf{Y} = (\mathbf{A}\mathbf{A}^*)^q\,\mathbf{A}\mathbf{R}$$

instead of

$$\mathbf{Y} = \mathbf{A}\mathbf{R}.$$

References: Paper by Rokhlin, Szlam, Tygert (2008). Suggestions by Ming Gu. Also similar to “block power method,” “block Lanczos,” “subspace iteration.”

Input: An $m \times n$ matrix **A**, a target rank ℓ, and a small integer q.	
Output: Rank-ℓ factors **U**, **D**, and **V** in an approximate SVD $\mathbf{A} \approx \mathbf{UDV}^*$.	
(1) Draw an $n \times \ell$ random matrix **R**.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^* \mathbf{A}$.
(2) Form the $m \times \ell$ sample matrix $\mathbf{Y} = (\mathbf{AA}^*)^q \mathbf{AR}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.
(3) Compute an ON matrix **Q** s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

- Detailed (and, we believe, close to sharp) error bounds have been proven.
  For instance, with $\mathbf{A}^{\text{computed}} = \mathbf{UDV}^*$, the expectation of the error satisfies:

$$\text{(1)} \qquad \mathbb{E}\left[\|\mathbf{A} - \mathbf{A}^{\text{computed}}\|\right] \leq \left(1 + 4\sqrt{\frac{2\ \min(m,n)}{k-1}}\right)^{1/(2q+1)} \sigma_{k+1}(\mathbf{A}).$$

  Reference: Halko, Martinsson, Tropp (2011).

- The improved accuracy from the modified scheme comes at a cost;
  $2q + 1$ passes over the matrix are required instead of 1.
  However, q can often be chosen quite small in practice, $q = 2$ or $q = 3$, say.

- The bound (1) assumes exact arithmetic.
  To handle round-off errors, variations of subspace iterations can be used.
  These are entirely numerically stable and achieve the same error bound.

Matrix approximation by sampling

Matrix approximation by sampling

To simplify slightly, there are two paradigms for how to use randomization to approximate matrices:

Randomized embeddings

(What we have discussed so far.)

Randomized sampling

(What we will discuss next.)

Matrix approximation by sampling

To simplify slightly, there are two paradigms for how to use randomization to approximate matrices:

Randomized embeddings (What we have discussed so far.)	**Randomized sampling** (What we will discuss next.)
Often faster than classical deterministic methods.	Sometimes *far* faster than classical deterministic methods. Faster than matrix-vector multiplication, even.
Highly reliable and robust.	Can fail.
High accuracy is attainable.	Typically low accuracy.
Best for scientific computing.	Enables solution of large scale problems in "big data" where no other methods work.

Matrix approximation by sampling

Suppose that $\mathbf{A} = \sum_{t=1}^{T} \mathbf{A}_t$ where each $\mathbf{A}_t$ is “simple” in some sense.

Matrix approximation by sampling

Suppose that $\mathbf{A} = \sum_{t=1}^{T} \mathbf{A}_t$ where each $\mathbf{A}_t$ is "simple" in some sense.

Example: Sparse matrix written as a sum over its nonzero entries

$$\underbrace{\begin{bmatrix} 5 & -2 & 0 \\ 0 & 0 & -3 \\ 1 & 0 & 0 \end{bmatrix}}_{=\mathbf{A}} = \underbrace{\begin{bmatrix} 5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=\mathbf{A}_1} + \underbrace{\begin{bmatrix} 0 & -2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=\mathbf{A}_2} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -3 \\ 0 & 0 & 0 \end{bmatrix}}_{=\mathbf{A}_3} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{=\mathbf{A}_4}$$

Example: Each $\mathbf{A}_i$ could be a column of the matrix

$$\underbrace{\begin{bmatrix} 5 & -2 & 7 \\ 1 & 3 & -3 \\ 1 & -1 & 1 \end{bmatrix}}_{=\mathbf{A}} = \underbrace{\begin{bmatrix} 5 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{=\mathbf{A}_1} + \underbrace{\begin{bmatrix} 0 & -2 & 0 \\ 0 & 3 & 0 \\ 0 & -1 & 0 \end{bmatrix}}_{=\mathbf{A}_2} + \underbrace{\begin{bmatrix} 0 & 0 & 7 \\ 0 & 0 & -3 \\ 0 & 0 & 1 \end{bmatrix}}_{=\mathbf{A}_3}.$$

Example: Matrix-matrix multiplication broken up as a sum of rank-1 matrices:

$$\mathbf{A} = \mathbf{BC} = \sum_{t} \mathbf{B}(:,t)\mathbf{C}(t,:).$$

Matrix approximation by sampling

Suppose that $\mathbf{A} = \sum_{t=1}^{T} \mathbf{A}_t$ where each $\mathbf{A}_t$ is "simple" in some sense.

Let $\{p_t\}_{t=1}^{T}$ be a probability distribution on the index vector $\{1, 2, \ldots, T\}$.
Draw an index $t \in \{1, 2, \ldots, T\}$ according to the probability distribution given, and set

$$\mathbf{X} = \frac{1}{p_t}\mathbf{A}_t.$$

Then from the definition of the expectation, we have

$$\mathbb{E}[\mathbf{X}] = \sum_{t=1}^{T} p_t \times \frac{1}{p_t}\mathbf{A}_t = \sum_{t=1}^{T} \mathbf{A}_t = \mathbf{A},$$

so $\mathbf{X}$ is an unbiased estimate of $\mathbf{A}$.

Clearly, a single draw is not a good approximation — unrepresentative, *large variance*.

Instead, draw several samples and average:

$$\bar{\mathbf{X}} = \frac{1}{k}\sum_{t=1}^{k} \mathbf{X}_t,$$

where $\mathbf{X}_t$ are independent samples from the same distribution.

As k grows, the variance will decrease, as usual. Various Bernstein inequalities apply.

Matrix approximation by sampling

As an illustration of the theory, we cite a matrix-Bernstein result from J. Tropp (2015):

Theorem: Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. Construct a probability distribution for $\mathbf{X} \in \mathbb{R}^{m \times n}$ that satisfies

$$\mathbb{E}[\mathbf{X}] = \mathbf{A} \quad \text{and} \quad \|\mathbf{X}\| \leq R.$$

Define the per-sample second-moment: $v(\mathbf{X}) := \max\{\|\mathbb{E}[\mathbf{X}\mathbf{X}^*]\|, \|\mathbb{E}[\mathbf{X}^*\mathbf{X}]\|\}$.

Form the matrix sampling estimator: $\bar{\mathbf{X}}_k = \frac{1}{k}\sum_{t=1}^{k} \mathbf{X}_i$ where $\mathbf{X}_t \sim \mathbf{X}$ are iid.

Then $\mathbb{E}\|\bar{\mathbf{X}}_k - \mathbf{A}\| \leq \sqrt{\frac{2v(\mathbf{X})\log(m+n)}{k}} + \frac{2R\log(m+n)}{3k}$.

Furthermore, for all $s \geq 0$: $\mathbb{P}[\|\bar{\mathbf{X}}_k - \mathbf{A}\| \geq s] \leq (m+n)\exp\left(\frac{-ks^2/2}{v(\mathbf{X}) + 2Rs/3}\right)$.

Suppose that we want $\mathbb{E}\|\mathbf{A} - \bar{\mathbf{X}}\| \leq 2\epsilon$. The theorem says to pick

$$k \geq \max\left\{\frac{2v(\mathbf{X})\log(m+n)}{\epsilon^2}, \frac{2R\log(m+n)}{3\epsilon}\right\}$$

In other words, the number k of samples should be proportional to both $v(\mathbf{X})$ and to the upper bound R.

The scaling $k \sim \frac{1}{\epsilon^2}$ is discouraging, and unavoidable.

Matrix approximation by sampling: Matrix matrix multiplication

Given two matrices **B** and **C**, consider the task of evaluating

$$\begin{array}{ccc} \mathbf{A} & = & \mathbf{B} \quad \mathbf{C}. \\ m \times n & & m \times T \;\; T \times n \end{array}$$

Sampling approach:

1. Fix a probability distribution $\{p_t\}_{t=1}^{T}$ on the index vector $\{1, 2, \ldots, T\}$.
2. Draw a subset of k indices $J = \{t_1, t_2, \ldots, t_k\} \subseteq \{1, 2, \ldots, T\}$.
3. Use $\bar{\mathbf{A}} = \sum_{i=1}^{k} \frac{1}{p_{t_i}} \mathbf{B}(:, t_i)\mathbf{C}(t_i, :)$ to approximate **A**.

You get an unbiased estimator regardless of the probability distribution. But the computational profile depends critically how which one you choose. Common choices:

Uniform distribution: Very fast. Not very reliable or accurate.

Sample according to column/row norms: Cost is $O(mnk)$, which is much better than $O(mnT)$ when $k \ll T$. Better outcomes than uniform, but still not particularly good.

In either case, you need $k \sim \frac{1}{\epsilon^2}$ to attain precision ϵ.

Matrix approximation by sampling: Low rank approximation.

Given an $m \times n$ matrix $\mathbf{A}$, we seek a rank-k matrix $\bar{\mathbf{A}}$ such that $\|\mathbf{A} - \bar{\mathbf{A}}\|$ is small.

Sampling approach:

1. Draw vectors J and I holding k samples from the column and row indices, resp.
2. Form matrices $\mathbf{C}$ and $\mathbf{R}$ consisting of the corresponding columns and rows

$$\mathbf{C} = \mathbf{A}(:, J), \qquad \text{and} \qquad \mathbf{R} = \mathbf{A}(I, :).$$

3. Use as your approximation

$$\begin{array}{ccccc} \bar{\mathbf{A}} & = & \mathbf{C} & \mathbf{U} & \mathbf{R}, \\ m \times n & & m \times k & k \times k & k \times n \end{array}$$

   where $\mathbf{U}$ is computed from information in $\mathbf{A}(I, J)$. (It should be an approximation to the optimal choice $\mathbf{U} = \mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$.)

The computational profile depends crucially on the probability distribution that is used.

Uniform probabilities: Can be *very* cheap. But in general not reliable.

Probabilities from "leverage scores": Optimal distributions can be computed using the information in the top left and right singular vectors of $\mathbf{A}$. Then quite strong theorems can be proven on the quality of the approximation. Problem: Computing the probability distribution is as expensive as computing a partial SVD.

Connection between sampling and random embeddings that "mix" variables

Task: Find a rank k approximation to a given $m \times n$ matrix $\mathbf{A}$.

Sampling approach: Draw a subset of k columns $\mathbf{Y} = \mathbf{A}(:, J)$ where J is drawn at random. Let our approximation to the matrix be

$$\mathbf{A}_k = \mathbf{Y}\mathbf{Y}^{\dagger}\mathbf{A}.$$

As we have seen, this in general does not work very well. But it does work well for the class of matrices for which uniform sampling is optimal.

Connection between sampling and random embeddings that "mix" variables

Task: Find a rank k approximation to a given $m \times n$ matrix $\mathbf{A}$.

Sampling approach: Draw a subset of k columns $\mathbf{Y} = \mathbf{A}(:, J)$ where J is drawn at random. Let our approximation to the matrix be

$$\mathbf{A}_k = \mathbf{Y}\mathbf{Y}^{\dagger}\mathbf{A}.$$

As we have seen, this in general does not work very well. But it does work well for the class of matrices for which uniform sampling is optimal. *We can turn* **A** *into such a matrix!*

Connection between sampling and random embeddings that "mix" variables

Task: Find a rank k approximation to a given $m \times n$ matrix $\mathbf{A}$.

Sampling approach: Draw a subset of k columns $\mathbf{Y} = \mathbf{A}(:, J)$ where J is drawn at random. Let our approximation to the matrix be

$$\mathbf{A}_k = \mathbf{Y}\mathbf{Y}^{\dagger}\mathbf{A}.$$

As we have seen, this in general does not work very well. But it does work well for the class of matrices for which uniform sampling is optimal. *We can turn* $\mathbf{A}$ *into such a matrix!* Let $\mathbf{U}$ be a matrix drawn from a uniform distribution on the set of $n \times n$ unitary matrices (the "Haar distribution"). Then form

$$\tilde{\mathbf{A}} = \mathbf{A}\mathbf{U}.$$

Now each column of $\tilde{\mathbf{A}}$ has exactly the same distribution! We may as well pick $J = 1 : k$, and can then pick a great sample through

$$\mathbf{Y} = \tilde{\mathbf{A}}(:, J) = \mathbf{A}\mathbf{U}(:, J).$$

The $n \times k$ "slice" $\mathbf{U}(:, J)$ is in a sense an optimal random embedding.

Fact: Using a Gaussian matrix is mathematically equivalent to using $\mathbf{U}(:, J)$.

Question: What other choices of random projection might mimic the action of $\mathbf{U}(:, J)$?

Structured random embeddings

Task: Find a rank k approximation to a given $m \times n$ matrix $\mathbf{A}$.

Approach: Find an $n \times k$ random embedding $\mathbf{R}$, set $\mathbf{Y} = \mathbf{AR}$, and then form $\mathbf{A}_k = \mathbf{YY}^\dagger\mathbf{A}$.

Choices of random embeddings:

- *Gaussian (or slice of Haar matrix):* Optimal. Leads to $O(mnk)$ overall cost.

- *Subsampled randomized Fourier transform (SRFT):* Indistinguishable from Gaussian in practice. Leads to $O(mn\log(k))$ overall cost. Adversarial counter examples can be built, so supporting theory is weak.

- *Chains of Givens rotations:* Similar profile to an SRFT.

- *Sparse random projections:* Need at least two nonzero entries per row. Works surprisingly well.

- *Additive random projections:* You can use a map with only ± 1 entries.

Key points on matrix approximation by sampling

- These techniques provide a path forwards for problems where traditional techniques are simply unaffordable.

  Kernel matrices in data analysis form a prime target. These are dense matrices, and you just cannot form the entire matrix. *Talk on Thu.*

- Popular topic for theory papers.

- When techniques based on randomized embeddings that systematically mix all coordinates *are* affordable, they perform far better. Higher accuracy, and less variability in the outcome.

Future and ongoing work:

- *Accelerate full factorizations of matrices.*

  Techniques for computing approximations to *all* singular values.

- *Engineering considerations.*

  In practice, the question of how the matrix is "presented" is important.

  Is it stored in RAM? Out of core (on a hard drive)? On a distributed memory system?

- *[High risk/high reward] Accelerate linear solvers for "general" systems* $\mathbf{Ax} = \mathbf{b}$.

  The goal is methods with complexity $O(N^\gamma)$ for $\gamma < 3$. Crucially, we seek methods that retain stability, and have high practical efficiency for realistic problem sizes. (Cf. Strassen — $O(N^{2.81})$, Coppersmith-Winograd $O(N^{2.38})$, etc.) *Talk on Thu.*

- *Use randomised embeddings to accelerate non-linear algebraic tasks.*

  Faster nearest neighbor search, faster clustering algorithms, etc. The idea is to use randomised projections for *sketching* to develop a rough map of a large data set. Then use high-accuracy deterministic methods for the actual computation.

Future and ongoing work:

- *Accelerate full factorizations of matrices.*

  Techniques for computing approximations to *all* singular values.

- *Engineering considerations.*

  In practice, the question of how the matrix is "presented" is important.

  Is it stored in RAM? Out of core (on a hard drive)? On a distributed memory system?

- *[High risk/high reward] Accelerate linear solvers for "general" systems* $\mathbf{Ax} = \mathbf{b}$.

  The goal is methods with complexity $O(N^\gamma)$ for $\gamma < 3$. Crucially, we seek methods that retain stability, and have high practical efficiency for realistic problem sizes. (Cf. Strassen — $O(N^{2.81})$, Coppersmith-Winograd $O(N^{2.38})$, etc.) *Talk on Thu.*

- *Use randomised embeddings to accelerate non-linear algebraic tasks.*

  Faster nearest neighbor search, faster clustering algorithms, etc. The idea is to use randomised projections for *sketching* to develop a rough map of a large data set. Then use high-accuracy deterministic methods for the actual computation.


Great potential for new discoveries in linear algebra!

Final remarks:

- For large scale SVD/PCA of dense matrices, these algorithms are highly recommended; they compare favorably to existing methods in almost every regard.

- The approximation error is a random variable, but its distribution is tightly concentrated. Rigorous error bounds that are satisfied with probability $1 - \eta$ where η is a user set "failure probability" (*e.g.* $\eta = 10^{-10}$ or 10^{-20}).

- This talk did not mention *error estimators*, but they are important.
  Can operate independently of the algorithm for improved robustness.
  Typically cheap and easy to implement. Used to determine the actual rank.

- The theory can be hard (at least for me), but *experimentation is easy!*
  Concentration of measure makes the algorithms behave as if deterministic.

Papers (see also http://users.oden.utexas.edu/∼pgm/main_publications.html):

- P.G. Martinsson, J. Tropp, "Randomized Numerical Linear Algebra: Foundations & Algorithms." *Acta Numerica*, 2020. *Available now as arxiv:2002.01387*
- N. Halko, P.G. Martinsson, J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions." *SIAM Review*, 2011.
- P.G. Martinsson, "Randomized Methods for Matrix Computations." In the 2018 book *The Mathematics of Data,* published by AMS. See also arxiv.org #1607.01649.
- E. Liberty, F. Woolfe, P.G. Martinsson, V. Rokhlin, and M. Tygert, "Randomized algorithms for the low-rank approximation of matrices". *PNAS*, **104**(51), 2007.

Tutorials, summer schools, etc:

- 2009: NIPS tutorial lecture, Vancouver, 2009. Online video available.
- 2014: CBMS summer school at Dartmouth College. 10 lectures on YouTube.
- 2016: Park City Math Institute (IAS): *The Mathematics of Data.*

Software packages:

- Column pivoted QR: https://github.com/flame/hqrrp (much faster than LAPACK!)
- Randomized UTV: https://github.com/flame/randutv
- RSVDPACK: https://github.com/sergeyvoronin (expansions are in progress)
- ID: http://tygert.com/software.html

96

Fast Direct Solvers for Elliptic PDEs

PER-GUNNAR MARTINSSON
The University of Texas at Austin

CBMS-NSF
REGIONAL CONFERENCE SERIES
IN APPLIED MATHEMATICS

SPONSORED BY
CONFERENCE BOARD OF
THE MATHEMATICAL SCIENCES

SUPPORTED BY
NATIONAL SCIENCE
FOUNDATION