CBMS Conference on Fast Direct Solvers

Dartmouth College

June 23 – June 27, 2014

Lecture 5: Randomized methods for low-rank approximation

Gunnar Martinsson

The University of Colorado at Boulder

Research support by:

Goal: Given an $m \times n$ matrix $\mathbf{A}$, we seek to compute a rank-k approximation, with $k \ll n$,

$$\begin{array}{ccccccc} \mathbf{A} & \approx & \mathbf{U} & \mathbf{D} & \mathbf{V}^* & = & \sum_{j=1}^{k} \sigma_j \, \mathbf{u}_j \, \mathbf{v}_j^*, \\ m \times n & & m \times k & k \times k & k \times n & & \end{array}$$

where

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq 0$ are the (approximate) singular values of $\mathbf{A}$
$\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ are orthonormal, the (approximate) *left singular vectors* of $\mathbf{A}$, and
$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ are orthonormal, the (approximate) *right singular vectors* of $\mathbf{A}$.

The methods presented are capable of solving other closely related problems:

- Interpolative decompositions, $\mathbf{A} \approx \mathbf{U}\,\mathbf{A}^{(\mathrm{row})}$, where $\mathbf{A}^{(\mathrm{row})}$ consists of k rows of $\mathbf{A}$.
- Partial LU-factorization, $\mathbf{A} \approx \mathbf{L}^{(k)}\,\mathbf{U}^{(k)}$.
- Eigenvalue decomposition $\mathbf{A} \approx \mathbf{V}\,\mathbf{D}\,\mathbf{V}^*$ (when $\mathbf{A}$ is Hermitian).
- *etc.*

The randomized methods are particularly powerful when m and n are very large, the matrix $\mathbf{A}$ may be stored on a distributed system, or “out-of-core.”

Excellent algorithms for computing SVDs exist, but many of them are not well suited for an emerging computational environment where *communication* is the bottleneck. Complications include:

- *Multi-processor computing.*
  CPU speed is growing slowly, but processors get cheaper all the time.

- *Communication speeds improve only slowly.*
  Communication between different levels in memory hierarchy, latency in hard drives, inter-processor communication, *etc.*

- *The size of data sets is growing very rapidly.*
  The cost of slow memory is falling rapidly, and information is gathered at an ever faster pace — automatic DNA sequencing, sensor networks, *etc.*

  - From a numerical linear algebra perspective, an additional problem resulting from increasing matrix sizes is that noise in the data, and propagation of rounding errors, become increasingly problematic.

The power of randomization in the modern context has been observed before:

C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala (2000)

A. Frieze, R. Kannan, and S. Vempala (1999, 2004)

D. Achlioptas and F. McSherry (2001)

P. Drineas, R. Kannan, M. W. Mahoney, and S. Muthukrishnan (2006a, 2006b, 2006c, 2006d, *etc*)

S. Har-Peled (2006)

A. Deshpande and S. Vempala (2006)

S. Friedland, M. Kaveh, A. Niknejad, and H. Zare (2006)

T. Sarlós (2006a, 2006b, 2006c)

K. Clarkson, D. Woodruff (2009)

For details, see review (Halko/Martinsson/Tropp 2009)

Goal (restated):

Given an $m \times n$ matrix $\mathbf{A}$ we seek to compute a rank-k approximation

$$\begin{array}{ccccc} \mathbf{A} & \approx & \mathbf{U} & \mathbf{D} & \mathbf{V}^* & = \sum_{j=1}^{k} \sigma_j \, \mathbf{u}_j \, \mathbf{v}_j^*, \\ m \times n & & m \times k & k \times k & k \times n & \end{array}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices holding the left and right (approximate) *singular vectors* of $\mathbf{A}$, respectively, and where $\mathbf{D}$ is a diagonal matrix holding the (approximate) *singular values* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq 0$.

Recall:

Once you have the SVD, “any” other standard factorization is easily obtained.

Goal: Given an $m \times n$ matrix $\mathbf{A}$, compute an approximate rank-k SVD $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

Goal: Given an $m \times n$ matrix $\mathbf{A}$, compute an approximate rank-k SVD $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

Algorithm:

1.
2.
3.

Find an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\mathbf{A} \approx \mathbf{Q}\,\mathbf{Q}^*\,\mathbf{A}$.
(I.e., the columns of $\mathbf{Q}$ form an ON-basis for the range of $\mathbf{A}$.)

4. Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.
5. Compute the SVD of the small matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\,\mathbf{D}\,\mathbf{V}^*$.
6. Form the matrix $\mathbf{U} = \mathbf{Q}\,\hat{\mathbf{U}}$.

Goal: Given an $m \times n$ matrix $\mathbf{A}$, compute an approximate rank-k SVD $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

Algorithm:

1.
2.
3.

Find an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\mathbf{A} \approx \mathbf{Q}\,\mathbf{Q}^*\,\mathbf{A}$.
(I.e., the columns of $\mathbf{Q}$ form an ON-basis for the range of $\mathbf{A}$.)

4. Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.
5. Compute the SVD of the small matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\,\mathbf{D}\,\mathbf{V}^*$.
6. Form the matrix $\mathbf{U} = \mathbf{Q}\,\hat{\mathbf{U}}$.

Note: Steps 4 – 6 are exact; the error in the method is all in $\mathbf{Q}$:

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

Note: The classical *Golub-Businger algorithm* follows this pattern. It finds $\mathbf{Q}$ in Step 3 via direct orthogonalization of the columns of $\mathbf{A}$ via, e.g., Gram-Schmidt.

Range finding problem: Given an $m \times n$ matrix $\mathbf{A}$ and an integer $k < \min(m,n)$, find an orthonormal $m \times k$ matrix $\mathbf{Q}$ such that $\mathbf{A} \approx \mathbf{Q}\mathbf{Q}^*\mathbf{A}$.

Solving the primitive problem via randomized sampling — intuition:

1. Draw random vectors $\omega_1,\,\omega_2,\,\ldots,\,\omega_k \in \mathbb{R}^n$.

   (We will discuss the choice of distribution later — think Gaussian for now.)

2. Form "sample" vectors $\mathbf{y}_1 = \mathbf{A}\,\omega_1,\, \mathbf{y}_2 = \mathbf{A}\,\omega_2,\,\ldots,\,\mathbf{y}_k = \mathbf{A}\,\omega_k \in \mathbb{R}^m$.

3. Form orthonormal vectors $\mathbf{q}_1,\,\mathbf{q}_2,\,\ldots,\,\mathbf{q}_k \in \mathbb{R}^m$ such that

$$\mathrm{Span}(\mathbf{q}_1,\,\mathbf{q}_2,\,\ldots,\,\mathbf{q}_k) = \mathrm{Span}(\mathbf{y}_1,\,\mathbf{y}_2,\,\ldots,\,\mathbf{y}_k).$$

   For instance, Gram-Schmidt can be used — pivoting is rarely required.

If $\mathbf{A}$ has *exact* rank k, then $\mathrm{Span}\{\mathbf{q}_j\}_{j=1}^k = \mathrm{Ran}(\mathbf{A})$ with probability 1.

Range finding problem: Given an $m \times n$ matrix $\mathbf{A}$ and an integer $k < \min(m, n)$, find an orthonormal $m \times k$ matrix $\mathbf{Q}$ such that $\mathbf{A} \approx \mathbf{Q}\mathbf{Q}^{*}\mathbf{A}$.

Solving the primitive problem via randomized sampling — intuition:

1. Draw random vectors $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_k \in \mathbb{R}^n$.
   (We will discuss the choice of distribution later — think Gaussian for now.)

2. Form "sample" vectors $\mathbf{y}_1 = \mathbf{A}\,\boldsymbol{\omega}_1,\ \mathbf{y}_2 = \mathbf{A}\,\boldsymbol{\omega}_2,\ \ldots,\ \mathbf{y}_k = \mathbf{A}\,\boldsymbol{\omega}_k \in \mathbb{R}^m$.

3. Form orthonormal vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k \in \mathbb{R}^m$ such that
   $$\operatorname{Span}(\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k) = \operatorname{Span}(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k).$$
   For instance, Gram-Schmidt can be used — pivoting is rarely required.

If $\mathbf{A}$ has *exact* rank k, then $\operatorname{Span}\{\mathbf{q}_j\}_{j=1}^{k} = \operatorname{Ran}(\mathbf{A})$ with probability 1.

What is perhaps surprising is that even in the general case, $\{\mathbf{q}_j\}_{j=1}^{k}$ often does almost as good of a job as the theoretically optimal vectors (which happen to be the k leading left singular vectors).

Range finding problem: Given an $m \times n$ matrix $\mathbf{A}$ and an integer $k < \min(m, n)$, find an orthonormal $m \times k$ matrix $\mathbf{Q}$ such that $\mathbf{A} \approx \mathbf{Q}\mathbf{Q}^*\mathbf{A}$.

Solving the primitive problem via randomized sampling — intuition:

1. Draw a random matrix $\mathbf{G} \in \mathbb{R}^{n \times k}$.
   (We will discuss the choice of distribution later — think Gaussian for now.)
2. Form a “sample” matrix $\mathbf{Y} = \mathbf{A}\,\mathbf{G} \in \mathbb{R}^{m \times k}$.
3. Form an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{m \times k}$ such that $\mathbf{Y} = \mathbf{Q}\mathbf{R}$.
   For instance, Gram-Schmidt can be used — pivoting is rarely required.

If $\mathbf{A}$ has *exact* rank k, then $\mathbf{A} = \mathbf{Q}\mathbf{Q}^*\mathbf{A}$ with probability 1.

Goal: Given an $m \times n$ matrix $\mathbf{A}$, compute an approximate rank-k SVD $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

Algorithm:

1. Draw an $n \times k$ Gaussian random matrix $\mathbf{G}$.
2. Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{A}\,\mathbf{G}$.
3. Form an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\mathbf{Y} = \mathbf{Q}\,\mathbf{R}$.
4. Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.
5. Compute the SVD of the small matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\,\mathbf{D}\,\mathbf{V}^*$.
6. Form the matrix $\mathbf{U} = \mathbf{Q}\,\hat{\mathbf{U}}$.

Goal: Given an $m \times n$ matrix $\mathbf{A}$, compute an approximate rank-k SVD $\mathbf{A} \approx \mathbf{UDV}^*$.

Algorithm:

1. Draw an $n \times k$ Gaussian random matrix $\mathbf{G}$. `G = randn(n,k)`
2. Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{A}\,\mathbf{G}$. `Y = A * G`
3. Form an $m \times k$ orthonormal matrix $\mathbf{Q}$ such that $\mathbf{Y} = \mathbf{Q}\,\mathbf{R}$. `[Q, R] = qr(Y)`
4. Form the $k \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$. `B = Q' * A`
5. Compute the SVD of the small matrix $\mathbf{B}$: $\mathbf{B} = \hat{\mathbf{U}}\,\mathbf{D}\,\mathbf{V}^*$. `[Uhat, Sigma, V] = svd(B,0)`
6. Form the matrix $\mathbf{U} = \mathbf{Q}\,\hat{\mathbf{U}}$. `U = Q * Uhat`

The randomized algorithm for computing an *interpolative decomposition* is even easier!

Goal: Given an $m \times n$ matrix $\mathbf{A}$, compute an ID $\mathbf{A} \approx \mathbf{U}\,\mathbf{A}(\tilde{I},:)$.

Algorithm:

1. Draw an $n \times k$ Gaussian random matrix $\mathbf{G}$. `G = randn(n,k)`
2. Form the $m \times k$ sample matrix $\mathbf{Y} = \mathbf{A}\,\mathbf{G}$. `Y = A * G`
3. Compute an ID of $\mathbf{Y}$: $\mathbf{Y} = \mathbf{U}\,\mathbf{Y}(\tilde{I},:)$. `[U, J] = id_decomp(Y)`

Note that the ID constructed in this way will necessarily work for **A**!

Single pass algorithms:

A is symmetric:	**A is not symmetric:**
Generate a random matrix $\mathbf{G}$.	Generate random matrices $\mathbf{G}$ and $\mathbf{H}$.
Compute a sample matrix $\mathbf{Y}$.	Compute sample matrices $\mathbf{Y} = \mathbf{A}\,\mathbf{G}$ and $\mathbf{Z} = \mathbf{A}^{*}\,\mathbf{H}$.
Find an ON matrix $\mathbf{Q}$ such that $\mathbf{Y} = \mathbf{Q}\,\mathbf{Q}^{*}\,\mathbf{Y}$.	Find ON matrices $\mathbf{Q}$ and $\mathbf{W}$ such that $\mathbf{Y} = \mathbf{Q}\,\mathbf{Q}^{*}\,\mathbf{Y}$ and $\mathbf{Z} = \mathbf{W}\,\mathbf{W}^{*}\,\mathbf{Z}$.
Solve for $\mathbf{T}$ the linear system $\mathbf{Q}^{*}\,\mathbf{Y} = \mathbf{T}\,(\mathbf{Q}^{*}\,\mathbf{G})$.	Solve for $\mathbf{T}$ the linear systems $\mathbf{Q}^{*}\,\mathbf{Y} = \mathbf{T}\,(\mathbf{W}^{*}\,\mathbf{G})$ and $\mathbf{W}^{*}\,\mathbf{Z} = \mathbf{T}^{*}\,(\mathbf{Q}^{*}\,\mathbf{H})$.
Factor $\mathbf{T}$ so that $\mathbf{T} = \hat{\mathbf{U}}\,\tilde{}\,\hat{\mathbf{U}}^{*}$.	Factor $\mathbf{T}$ so that $\mathbf{T} = \hat{\mathbf{U}}\,\mathbf{D}\,\hat{\mathbf{V}}^{*}$.
Form $\mathbf{U} = \mathbf{Q}\,\hat{\mathbf{U}}$.	Form $\mathbf{U} = \mathbf{Q}\,\hat{\mathbf{U}}$ and $\mathbf{V} = \mathbf{W}\,\hat{\mathbf{V}}$.
Output: $\mathbf{A} \approx \mathbf{U}\tilde{}\mathbf{U}^{*}$	Output: $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^{*}$

Note: With $\mathbf{B}$ as on the previous slide we have $\mathbf{T} \approx \mathbf{B}\,\mathbf{Q}$ (sym. case) and $\mathbf{T} \approx \mathbf{B}\,\mathbf{W}$ (nonsym. case).

References: *Woolfe, Liberty, Rokhlin, and Tygert (2008), Clarkson and Woodruff (2009), Halko, Martinsson and Tropp (2009).*

Question: What if the rank is not known in advance?

Answer: It is possible to incorporate error estimators.

- Almost free in terms of flop count.
- Use of an error estimator may increase the pass count.
- Even though they are barely mentioned in this presentation, error estimators are very important! In addition to allowing adaptive rank determination, they greatly improve on the robustness and reliability of randomized methods.

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 $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

(1) Draw an $n \times k$ random matrix $\mathbf{G}$.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.
(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$.
(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question:

How much does the computation above cost?

Answer:

Cost of steps 2 and 4: Application of $\mathbf{A}$ and $\mathbf{A}^*$ to k vectors.

Cost of steps 3,5,6: Dense operations on matrices of sizes $m \times k$ and $k \times n$.

We will consider three proto-typical environments.

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 $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

(1) Draw an $n \times k$ random matrix $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Cost of steps 2 and 4: Application of $\mathbf{A}$ and $\mathbf{A}^*$ to k vectors.

Cost of steps 3,5,6: Dense operations on matrices of sizes $m \times k$ and $k \times n$.

Case 1: $\mathbf{A}$ is presented as an array of real numbers in RAM.

Cost is dominated by the $2\,m\,n\,k$ flops required for steps 2 and 4.

The $O(m\,n\,k)$ flop count is the same as that of standard methods such as Golub-Businger. However, the algorithm above requires no random access to the matrix $\mathbf{A}$ — the data is retrieved *in two passes.* (One pass in the modified version.)

Remark 1: Improved data access leads to a moderate gain in execution time.

Remark 2: Using a special structured random sampling matrix (for instance, a "subsampled random Fourier transform"), substantial gain in execution time, and asymptotic complexity of $O(m\,n\,\log(k))$ can be achieved.

To attain complexity $O(mn\log(k))$, the scheme should be slightly modified so that randomized sampling is used to determine both the column space and the row space:

Given an $m \times n$ matrix $\mathbf{A}$, find $\mathbf{U}$, $\mathbf{V}$, $\mathbf{D}$ of rank k such that $\mathbf{A} \approx \mathbf{U}\,\mathbf{D}\,\mathbf{V}^{*}$.

1. Choose an over-sampling parameter p and set $\ell = k + p$. (Say $p = k$ so $\ell = 2k$.)

2. Generate SRFT's $\mathbf{G}$ and $\mathbf{H}$ of sizes $n \times \ell$, and $m \times \ell$.

3. Form the sample matrices $\mathbf{Y} = \mathbf{A}\,\mathbf{G}$ and $\mathbf{Z} = \mathbf{A}^{*}\,\mathbf{H}$.

4. Find ON matrices $\mathbf{Q}$ and $\mathbf{W}$ such that $\mathbf{Y} = \mathbf{Q}\,\mathbf{Q}^{*}\,\mathbf{Y}$ and $\mathbf{Z} = \mathbf{W}\,\mathbf{W}^{*}\,\mathbf{Z}$.

5. Solve for the $k \times k$ matrix $\mathbf{T}$ the systems $\mathbf{Q}^{*}\,\mathbf{Y} = \mathbf{T}\,(\mathbf{W}^{*}\,\mathbf{G})$ and $\mathbf{W}^{*}\,\mathbf{Z} = \mathbf{T}^{*}\,(\mathbf{Q}^{*}\,\mathbf{H})$.

6. Compute the SVD of the small matrix $\mathbf{T} = \hat{\mathbf{U}}\,\mathbf{D}\,\hat{\mathbf{V}}^{*}$ (and truncate if desired).

7. Form $\mathbf{U} = \mathbf{Q}\,\hat{\mathbf{U}}$ and $\mathbf{V} = \mathbf{W}\,\hat{\mathbf{V}}$.

Observation 1: Forming $\mathbf{A}\,\mathbf{G}$ and $\mathbf{A}^{*}\,\mathbf{H}$ in Step 2 has cost $O(m\,n\,\log(k))$ since $\ell \sim k$.

Observation 2: All other steps cost at most $O((m+n)\,k^{2})$.

Practical speed of $O(mnk\log(k))$ complexity randomized

Consider the task of computing a rank-k SVD of a matrix $\mathbf{A}$ of size $n \times n$.

$t^{(\mathrm{direct})}$ Time for classical (Golub-Businger) method — $O(k\,n^2)$

$t^{(\mathrm{srft})}$ Time for randomized method with an SRFT — $O(\log(k)\,n^2)$

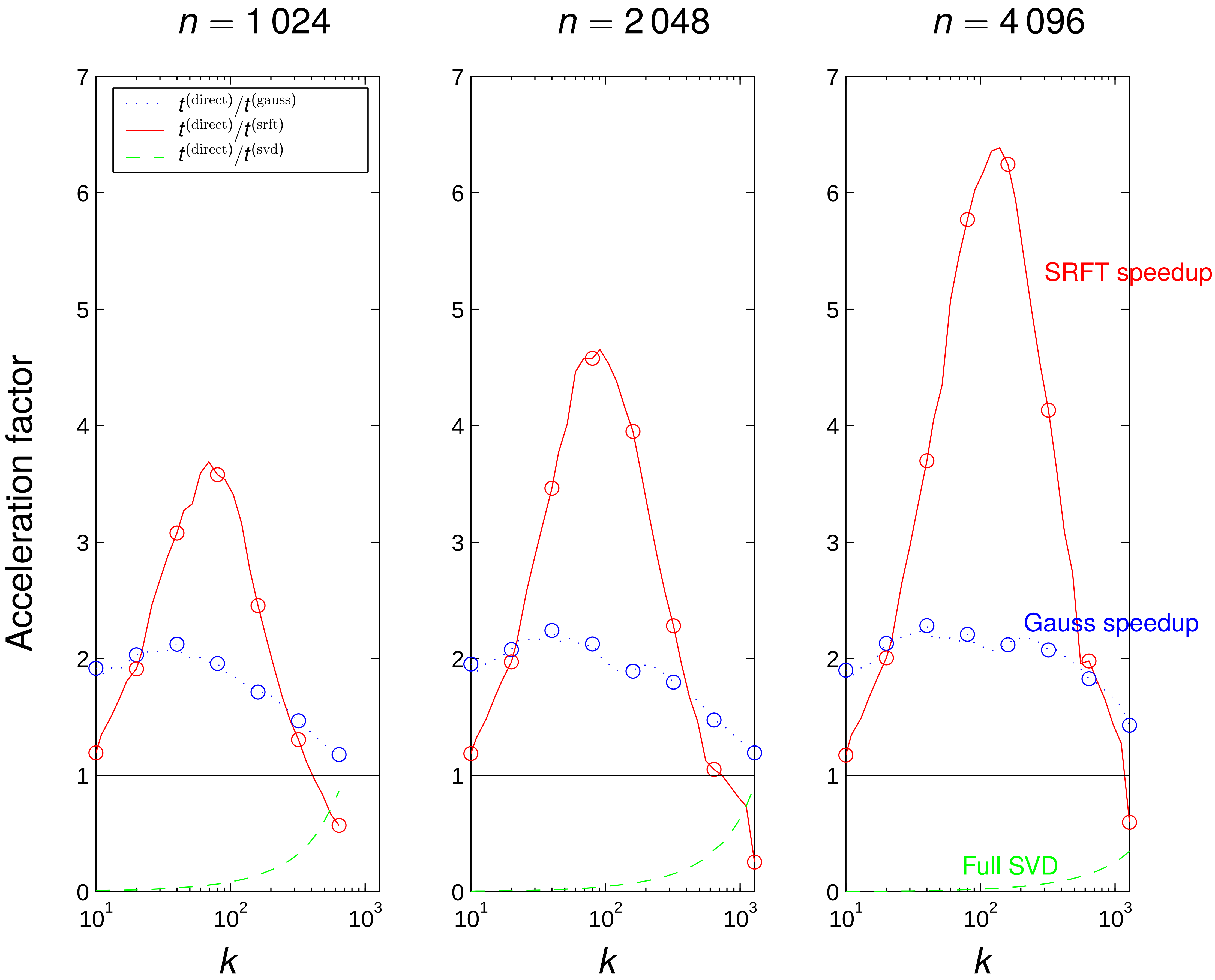
$t^{(\mathrm{gauss})}$ Time for randomized method with a Gaussian matrix — $O(k\,n^2)$

$t^{(\mathrm{svd})}$ Time for a full SVD — $O(n^3)$

We will show the

acceleration factors: $\dfrac{t^{(\mathrm{direct})}}{t^{(\mathrm{srft})}}$ $\dfrac{t^{(\mathrm{direct})}}{t^{(\mathrm{gauss})}}$ $\dfrac{t^{(\mathrm{direct})}}{t^{(\mathrm{svd})}}$

for different values of n and k.



Observe: Large speedups (up to a factor 6!) for moderate size matrices.

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 $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{AG}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Cost of steps 2 and 4: Application of $\mathbf{A}$ and $\mathbf{A}^*$ to k vectors.

Cost of steps 3,5,6: Dense operations on matrices of sizes $m \times k$ and $k \times n$.

Case 2: $\mathbf{A}$ is presented as an array of real numbers in slow memory — on “disk”.

In this case, standard methods such as Golub-Businger become prohibitively slow due to the random access requirements on the data.

However, the method described above works just fine.

Limitation 1: Matrices of size $m \times k$ and $k \times n$ must fit in RAM.

Limitation 2: For matrices whose singular values decay slowly (as is typical in the data-analysis environment), the method above is typically not accurate enough.
We will shortly discuss the accuracy of the method in detail, and describe a much more accurate variation.

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 $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{AG}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Cost of steps 2 and 4: Application of $\mathbf{A}$ and $\mathbf{A}^*$ to k vectors.

Cost of steps 3,5,6: Dense operations on matrices of sizes $m \times k$ and $k \times n$.

Case 3: $\mathbf{A}$ and $\mathbf{A}^*$ admit fast matrix-vector multiplication.

In this case, the “standard” method is some variation of Krylov methods such as Lanczos (or Arnoldi for non-symmetric matrices) whose cost $T_{\rm Krylov}$ satisfy:

$$T_{\rm Krylov} \sim k\,T_{\text{mat-vec-mult}} + O(k^2\,(m+n)).$$

The asymptotic cost of the randomized scheme is the same;
its advantage is again in how the data is accessed — the k matrix-vector multiplies can be executed in parallel.

The method above is in important environments less accurate than Arnoldi,
but this can be fixed while compromising only slightly on the pass-efficiency.

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 $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{AG}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

How accurate is the proposed method?

Let $\mathbf{A}_k^{\mathrm{computed}}$ denote the output of the method above,

$$\mathbf{A}_k^{\mathrm{computed}} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^*.$$

We are interested in the error

$$e_k = ||\mathbf{A} - \mathbf{A}_k^{\mathrm{computed}}||.$$

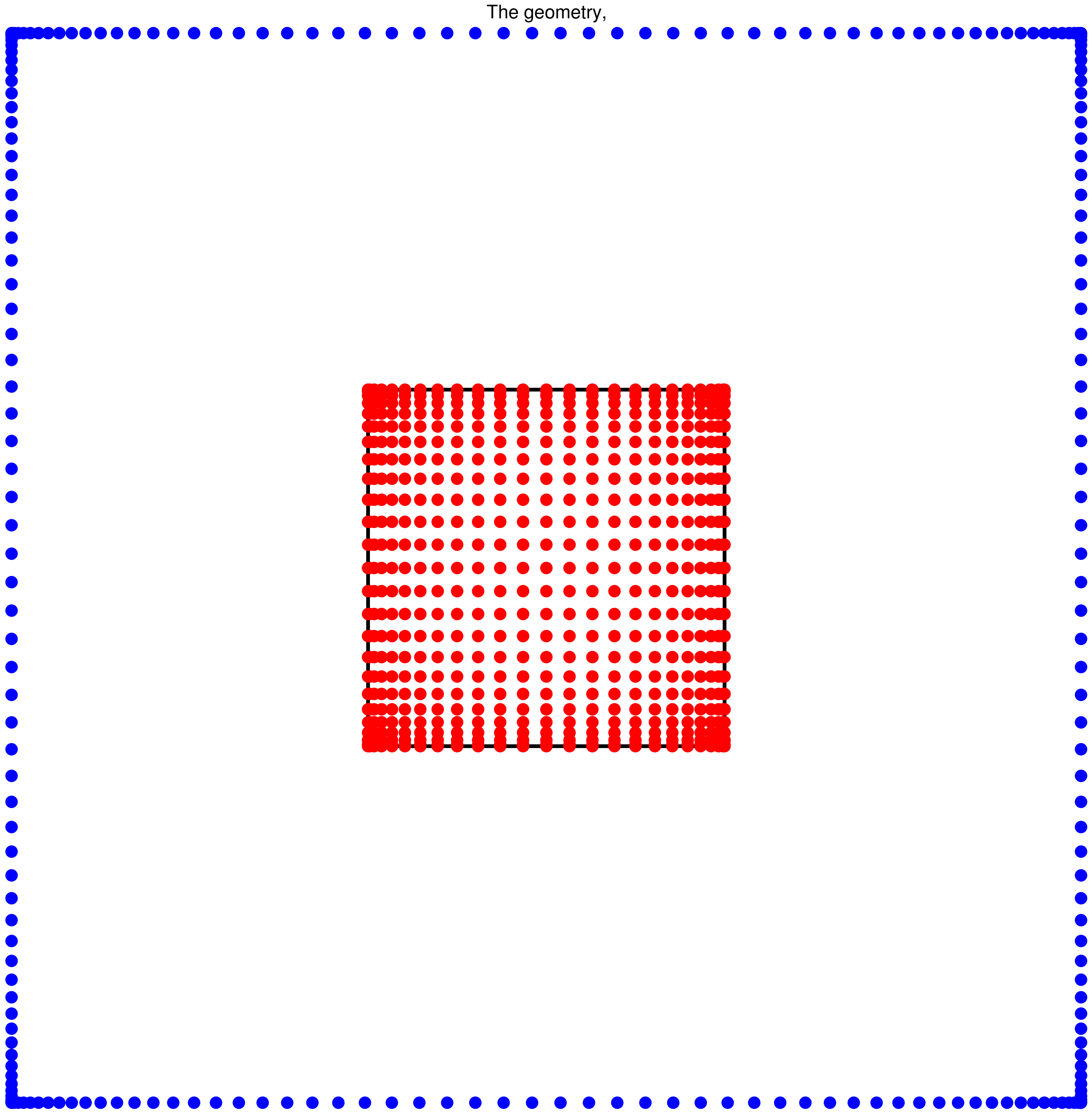
The error e_k should be compared to the theoretically minimal error

$$\sigma_{k+1} = \min\{||\mathbf{A} - \mathbf{A}_k|| \,:\, \mathbf{A}_k \text{ has rank } k\},$$

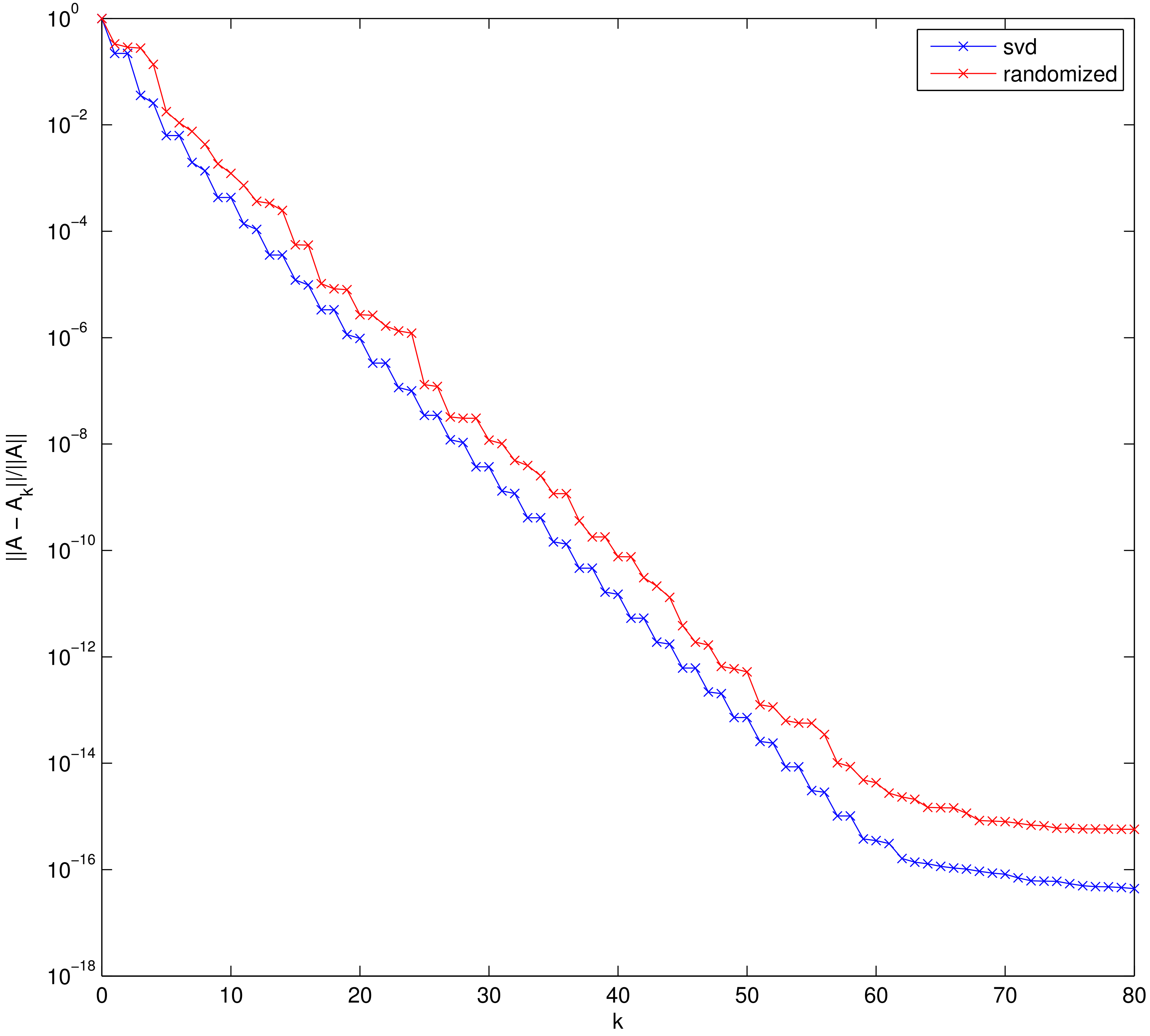
where σ_k is the (exact) k'th singular value of $\mathbf{A}$.

(We criminally switched notation here — σ_j used to be the j'th computed *approximate* singular value of $\mathbf{A}$ — now it is the exact singular value.)

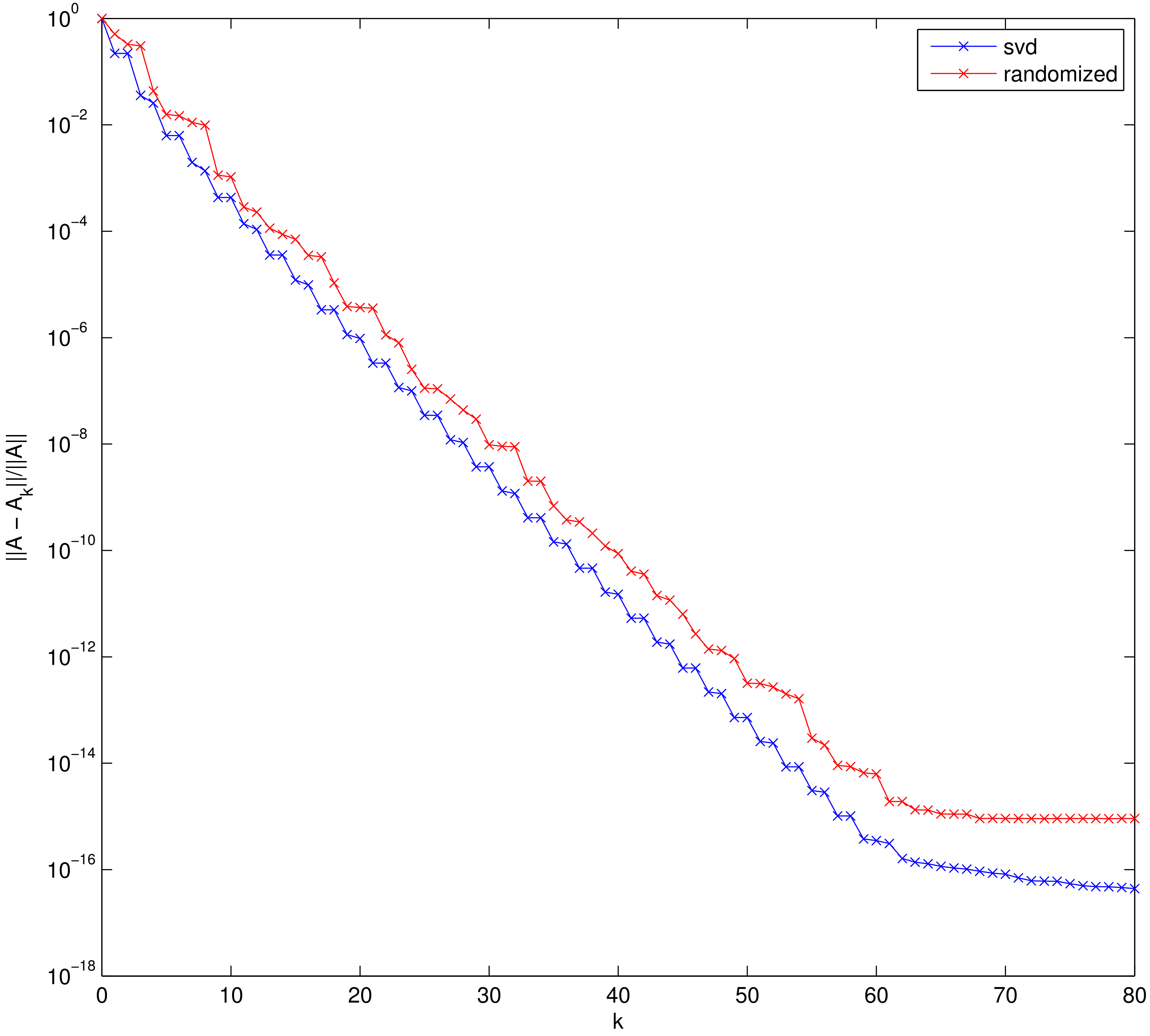
Example 1: Laplace potential evaluation map: $\mathbf{A}(i,j) = \sqrt{w_i\,v_j}\,\log|\boldsymbol{x}_i - \boldsymbol{y}_j|$.



Example 1: Laplace potential evaluation map: $\mathbf{A}(i,j) = \sqrt{w_i\, v_j}\, \log|\boldsymbol{x}_i - \boldsymbol{y}_j|$.

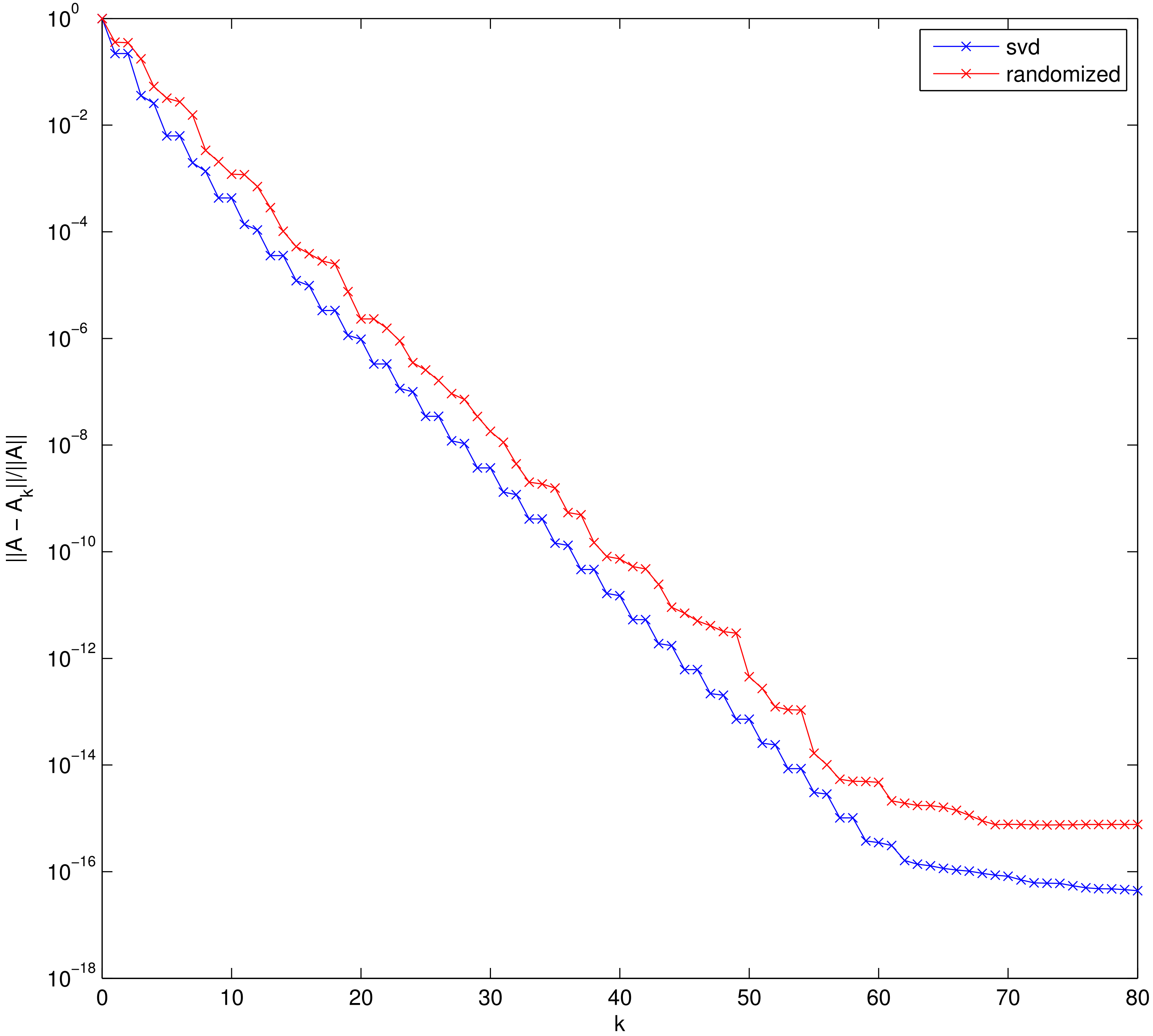


Example 1: Laplace potential evaluation map: $\mathbf{A}(i,j) = \sqrt{w_i\, v_j}\, \log|\boldsymbol{x}_i - \boldsymbol{y}_j|$.



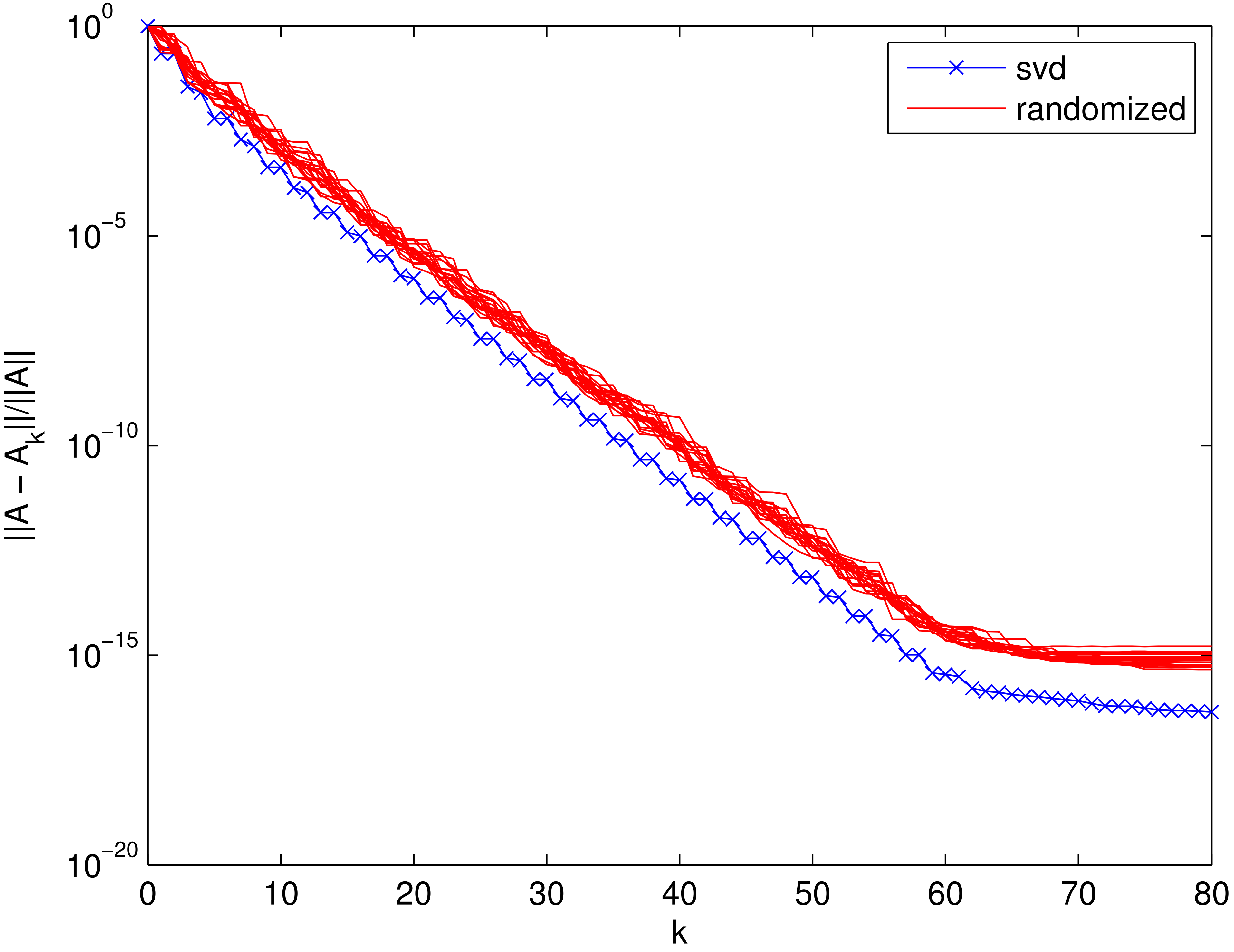
Another instantiation...

Example 1: Laplace potential evaluation map: $\mathbf{A}(i,j) = \sqrt{w_i\, v_j}\, \log|\boldsymbol{x}_i - \boldsymbol{y}_j|$.



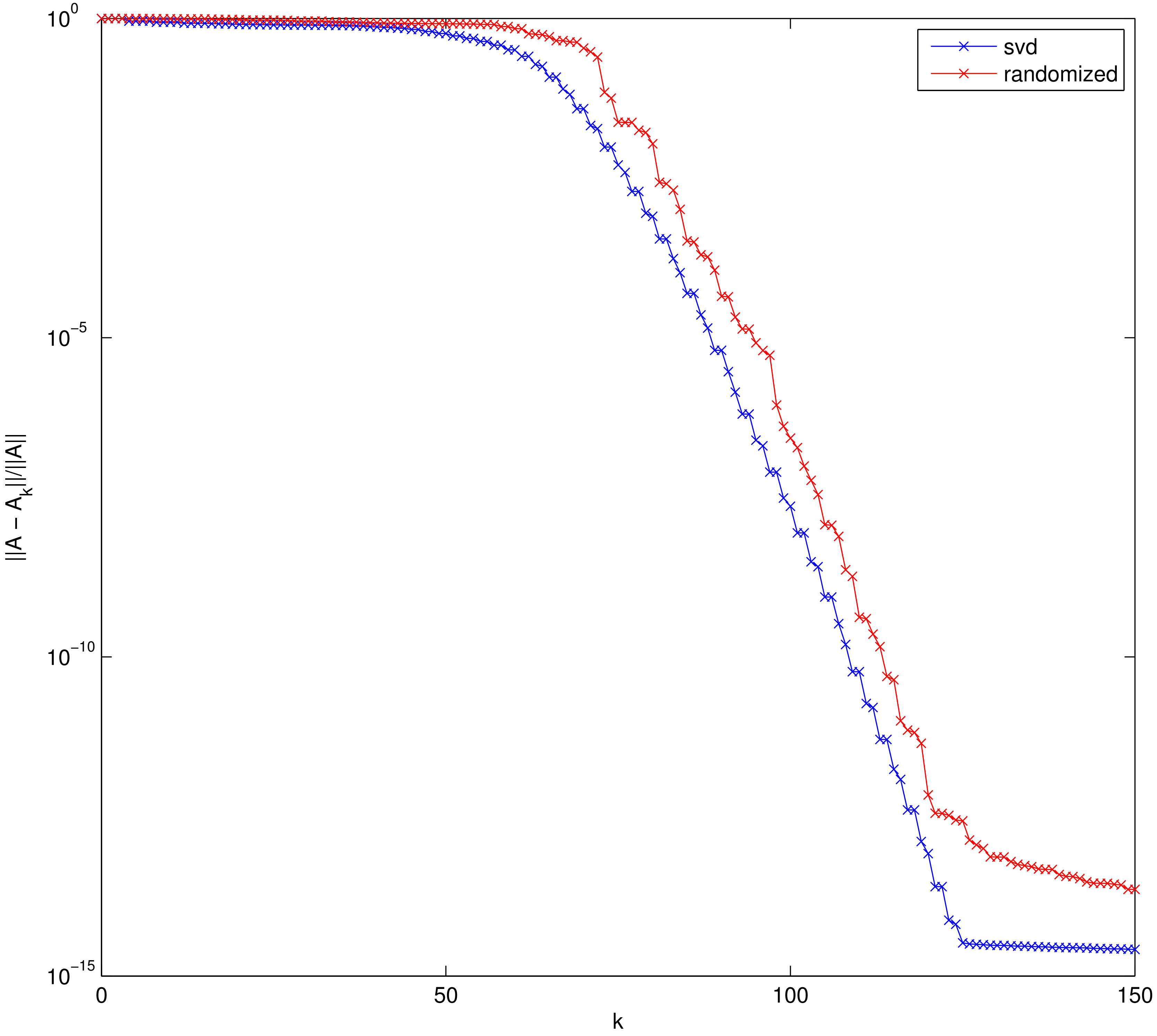
Yet another instantiation...

Example 1: Laplace potential evaluation map: $\mathbf{A}(i,j) = \sqrt{w_i\, v_j}\, \log|\boldsymbol{x}_i - \boldsymbol{y}_j|$.

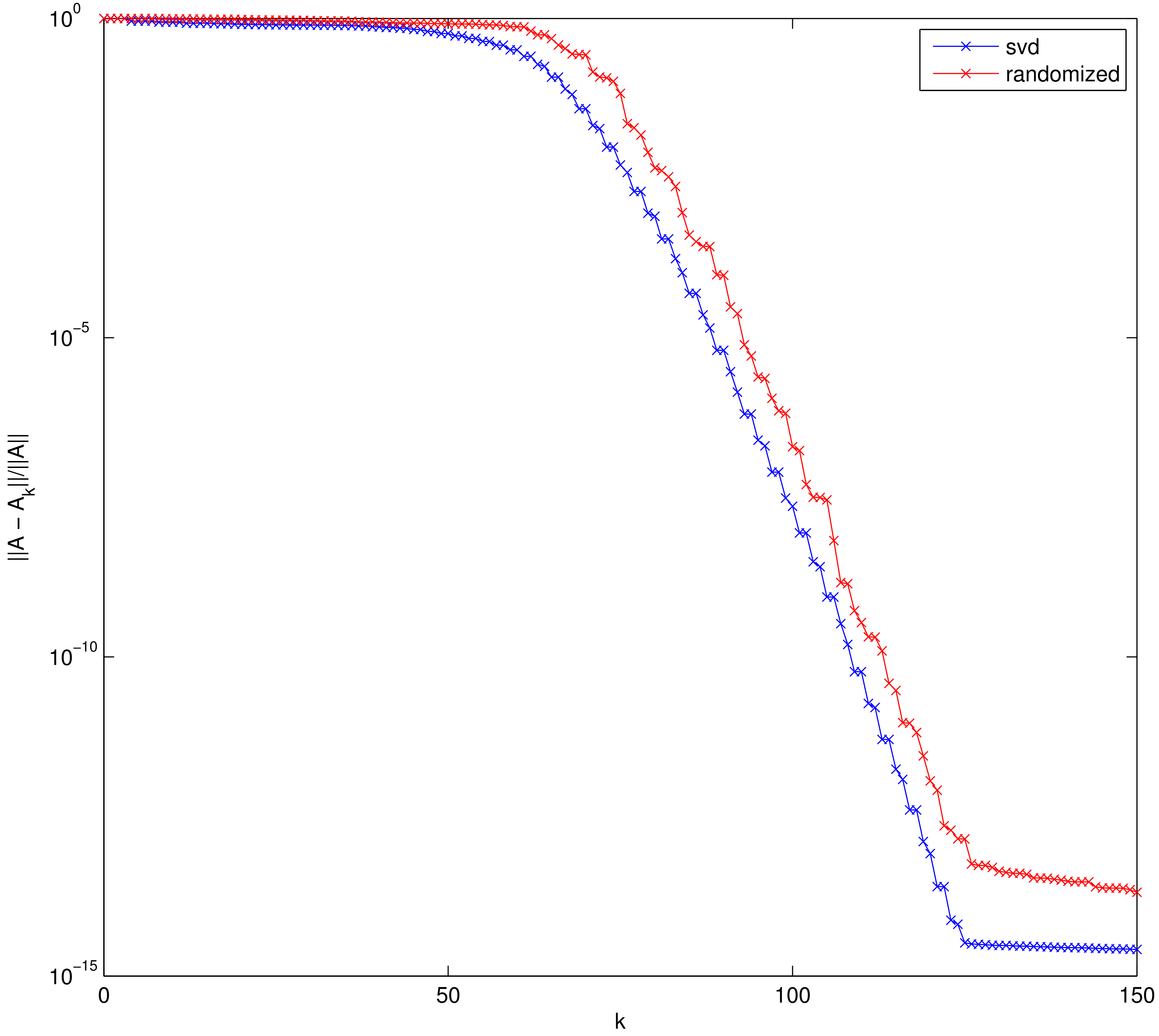


20 different instantiations.

Example 2: Helmholtz potential map (side=8λ): $\mathbf{A}(i,j) = \sqrt{w_i\,v_j}\,H_0(\kappa|\boldsymbol{x}_i - \boldsymbol{y}_j|)$.

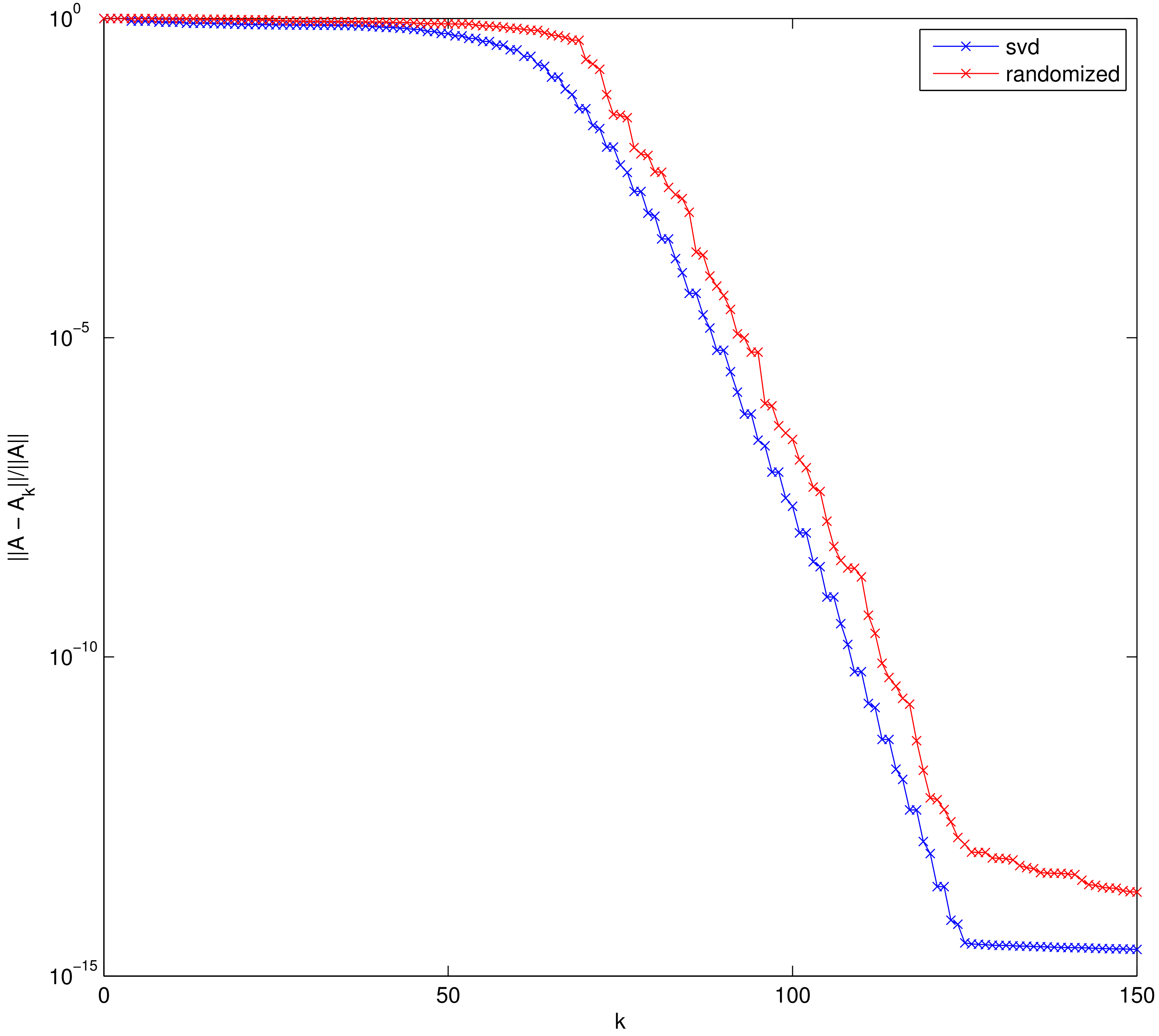


Example 2: Helmholtz potential map (side=8λ): $\mathbf{A}(i,j) = \sqrt{w_i\, v_j}\, H_0(\kappa|\boldsymbol{x}_i - \boldsymbol{y}_j|)$.



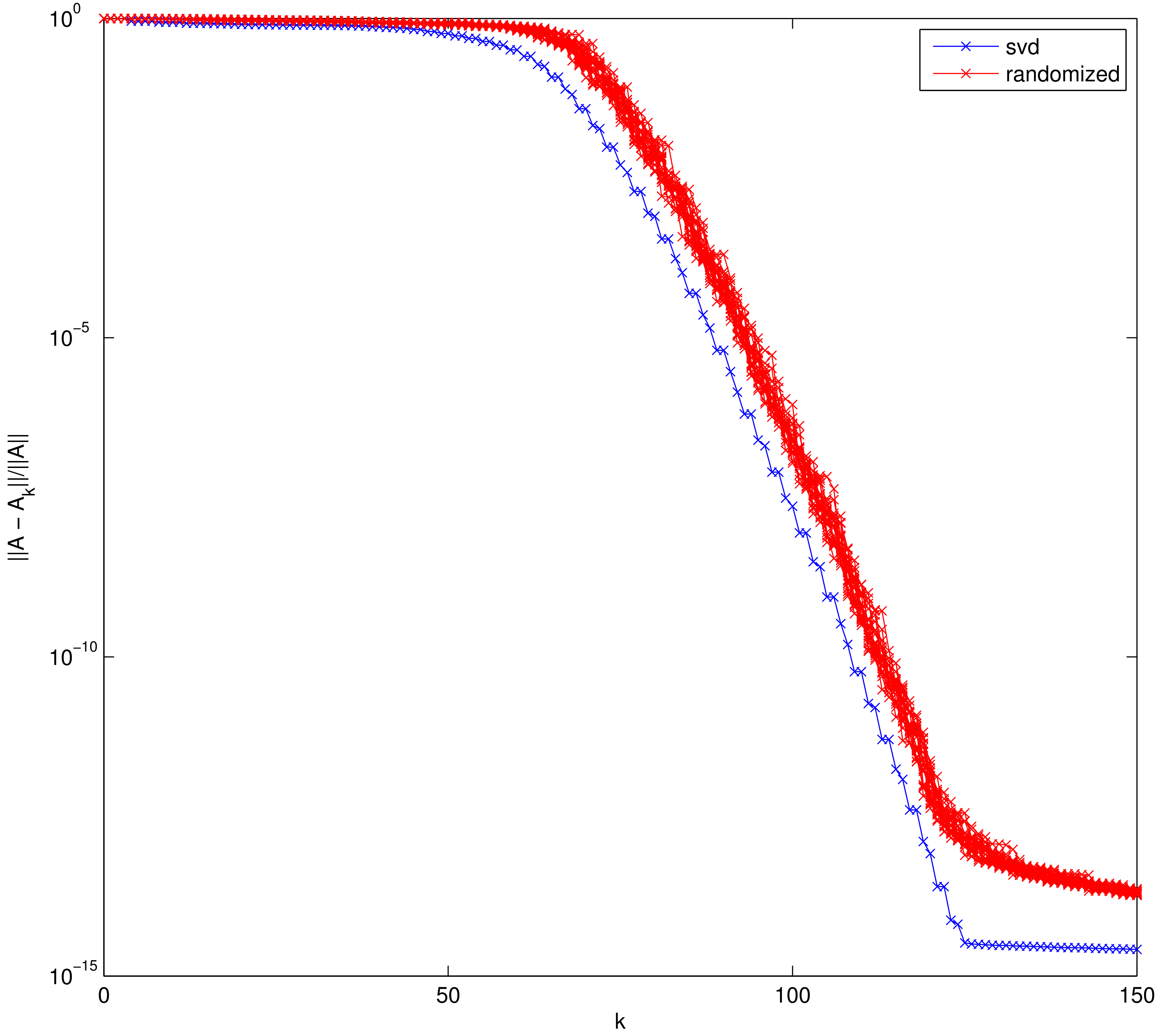
Another instantiation...

Example 2: Helmholtz potential map (side=8λ): $\mathbf{A}(i,j) = \sqrt{w_i\,v_j}\,H_0(\kappa|\boldsymbol{x}_i - \boldsymbol{y}_j|)$.



Yet another instantiation...

Example 2: Helmholtz potential map (side=8λ): $\mathbf{A}(i,j) = \sqrt{w_i\, v_j}\, H_0(\kappa|\boldsymbol{x}_i - \boldsymbol{y}_j|)$.



20 different instantiations.

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 $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

(1) Draw an $n \times k$ random matrix $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question: What is the error $e_k = ||\mathbf{A} - \mathbf{U}\mathbf{D}\mathbf{V}^*||$? (Recall $e_k = ||\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}||$.)

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 $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

(1) Draw an $n \times k$ random matrix $\mathbf{G}$.	(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.
(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G}$.	(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$.
(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^*\mathbf{Y}$.	(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question: What is the error $e_k = ||\mathbf{A} - \mathbf{U}\mathbf{D}\mathbf{V}^*||$? (Recall $e_k = ||\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}||$.)

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 $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{AG}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

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

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 $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{AG}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{QQ}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{DV}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

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

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 $\mathbf{A}$ and a target rank k.

Output: Rank-k factors $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ in an approximate SVD $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

(1) Draw an $n \times k$ random matrix $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Question: What is the error $e_k = ||\mathbf{A} - \mathbf{U}\mathbf{D}\mathbf{V}^*||$? (Recall $e_k = ||\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}||$.)

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.

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

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 $\mathbf{A}$, a target rank k, *and an over-sampling parameter p (say $p = 5$).*

Output: Rank-$(k+p)$ factors $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ in an approximate SVD $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

(1) Draw an $n \times (k+p)$ random matrix $\mathbf{G}$.

(2) Form the $n \times (k+p)$ sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$.

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

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

If $p \geq 2$, then

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

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

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

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

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

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,

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

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

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

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 $3\,p^{-p}$.

Let us look at the error bound a little closer:

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

Case 1 — the singular values decay rapidly: If (σ_j) decays sufficiently rapidly that $\left(\sum_{j>k}\sigma_j^2\right)^{1/2} \approx \sigma_{k+}$, then we are fine — a minimal amount of over-sampling (say $p = 5$ or $p = k$) drives the error down close to the theoretically minimal value.

Let us look at the error bound a little closer:

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

Case 1 — the singular values decay rapidly: If (σ_j) decays sufficiently rapidly that $\left(\sum_{j>k}\sigma_j^2\right)^{1/2} \approx \sigma_{k+}$, then we are fine — a minimal amount of over-sampling (say $p = 5$ or $p = k$) drives the error down close to the theoretically minimal value.

☞ *This is precisely our situation when building direct solvers!*

Let us look at the error bound a little closer:

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

Case 1 — the singular values decay rapidly: If (σ_j) decays sufficiently rapidly that $\left(\sum_{j>k}\sigma_j^2\right)^{1/2} \approx \sigma_{k+}$, then we are fine — a minimal amount of over-sampling (say $p = 5$ or $p = k$) drives the error down close to the theoretically minimal value.

☞ *This is precisely our situation when building direct solvers!*

Case 2 — the singular values do not decay rapidly: In the worst case, we have

$$\left(\sum_{j>k}\sigma_j^2\right)^{1/2} \sim \sqrt{n-k}\,\sigma_{k+1}.$$

If n is large, and σ_{k+1}/σ_1 is not that small, we could lose all accuracy.

Power method for improving accuracy:

The error depends on how quickly the singular values decay.

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 its singular values are

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

Much faster decay — so let us use the sample matrix

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

instead of

$$\mathbf{Y} = \mathbf{A}\,\mathbf{G}.$$

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

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 $\mathbf{A} \approx \mathbf{U}\mathbf{D}\mathbf{V}^*$.

(1) Draw an $n \times k$ random matrix $\mathbf{G}$.

(2) Form the $n \times k$ sample matrix $\mathbf{Y} = (\mathbf{A}\mathbf{A}^*)^q\mathbf{A}\mathbf{G}$.

(3) Compute an ON matrix $\mathbf{Q}$ s.t. $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^*\mathbf{Y}$.

(4) Form the small matrix $\mathbf{B} = \mathbf{Q}^*\,\mathbf{A}$.

(5) Factor the small matrix $\mathbf{B} = \hat{\mathbf{U}}\mathbf{D}\mathbf{V}^*$.

(6) Form $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.

Theorem: $\mathbb{E}||\mathbf{A} - \mathbf{A}_{k+p}^{\rm computed}||$ converges exponentially fast to the optimal value of σ_{k+1} as q increases.

The theorem assumes exact arithmetic — in real life some complications arise.

These can be handled by careful implementation.

The modified scheme obviously comes at a substantial 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.

Proofs of probabilistic error bounds

Let us first recall the two theorems we flashed earlier.

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

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

If $p \geq 2$, then

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

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

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

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

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

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,

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

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

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

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

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

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,

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

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

Proofs — Overview:

Let $\mathbf{A}$ 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. Set $\ell = k + p$.

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

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

1. Make no assumption on $\mathbf{G}$. Construct a deterministic bound of the form

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

2. Assume that $\mathbf{G}$ is drawn from a normal 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 $\mathbf{G}$ is drawn from a normal Gaussian distribution.

   Take expectations of the deterministic bound conditioned on "bad behavior" in $\mathbf{G}$ to get

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

   that hold with probability at least $\cdots$.

Part 1 (out of 3) — deterministic bound:

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. Set $\ell = k + p$.

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

Partition the SVD of $\mathbf{A}$ as follows:

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

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

$$\mathbf{G}_1 = \mathbf{V}_1^* \mathbf{G} \quad \text{and} \quad \mathbf{G}_2 = \mathbf{V}_2^* \mathbf{G}.$$

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

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

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

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

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

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

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

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

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

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

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. Set $\ell = k + p$.

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

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

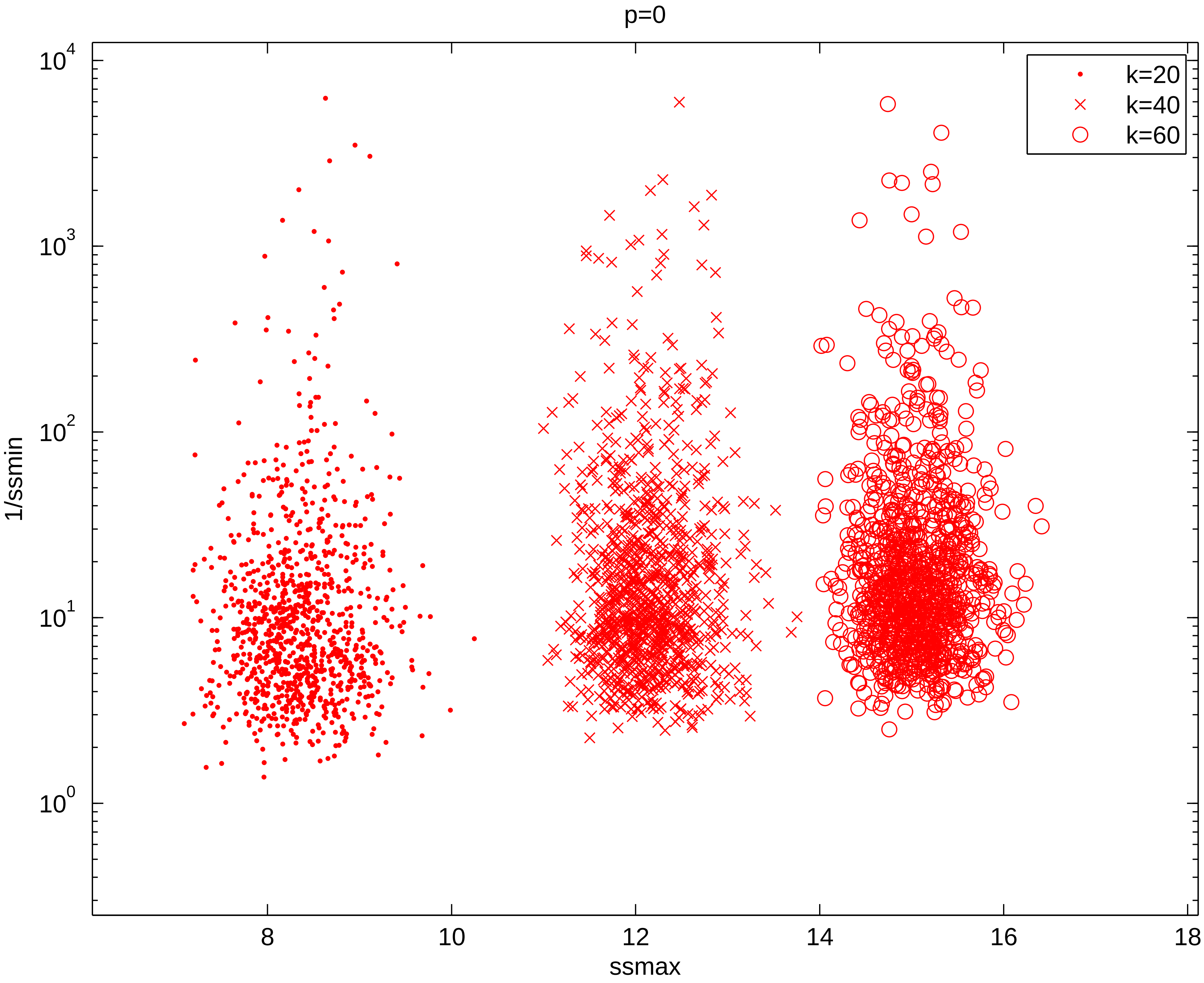
Assumption: $\mathbf{G}$ is drawn from a normal Gaussian distribution.

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

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

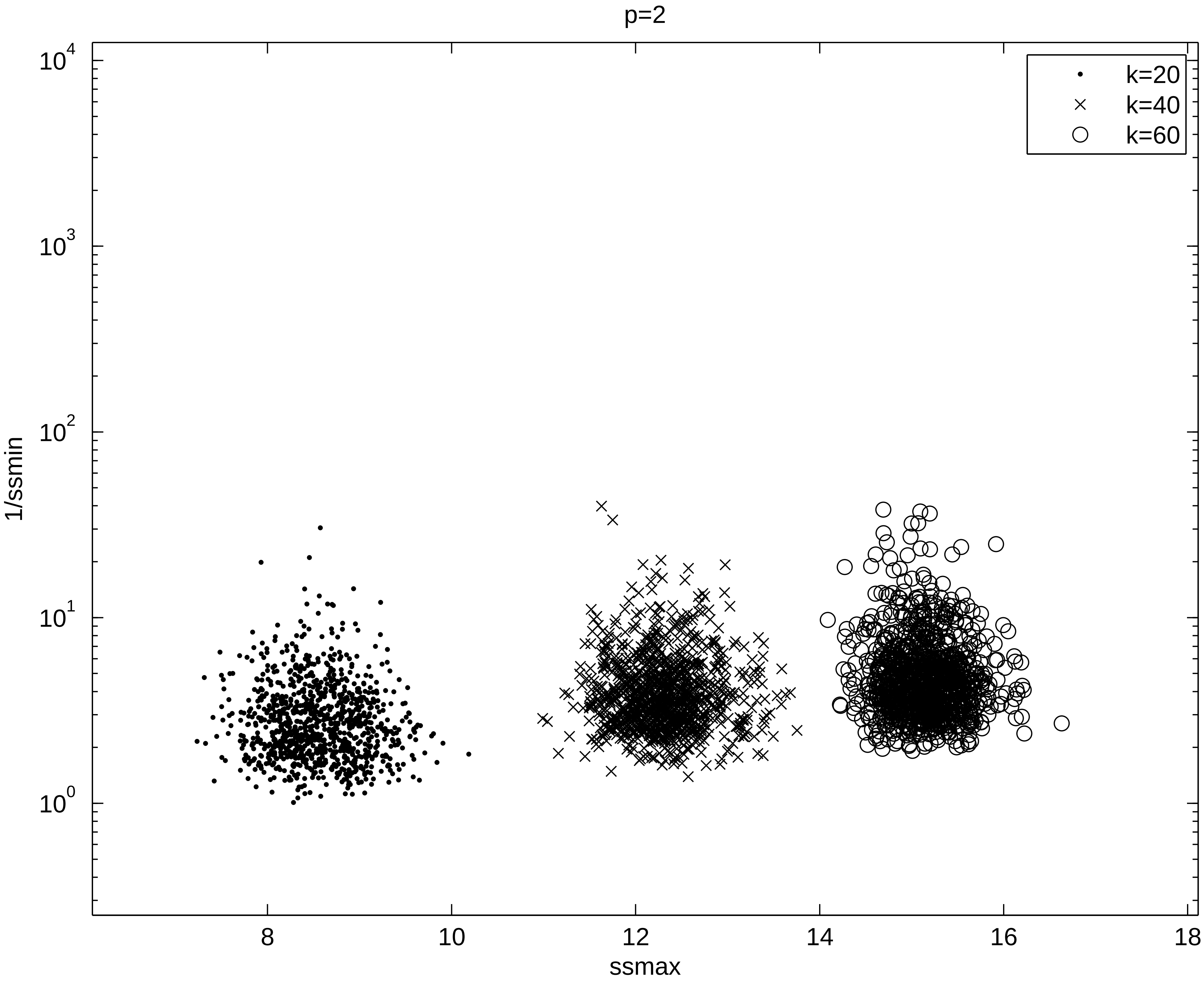
If $p = 0$, then $||\mathbf{G}_1^\dagger||$ is typically large, and is very unstable.

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



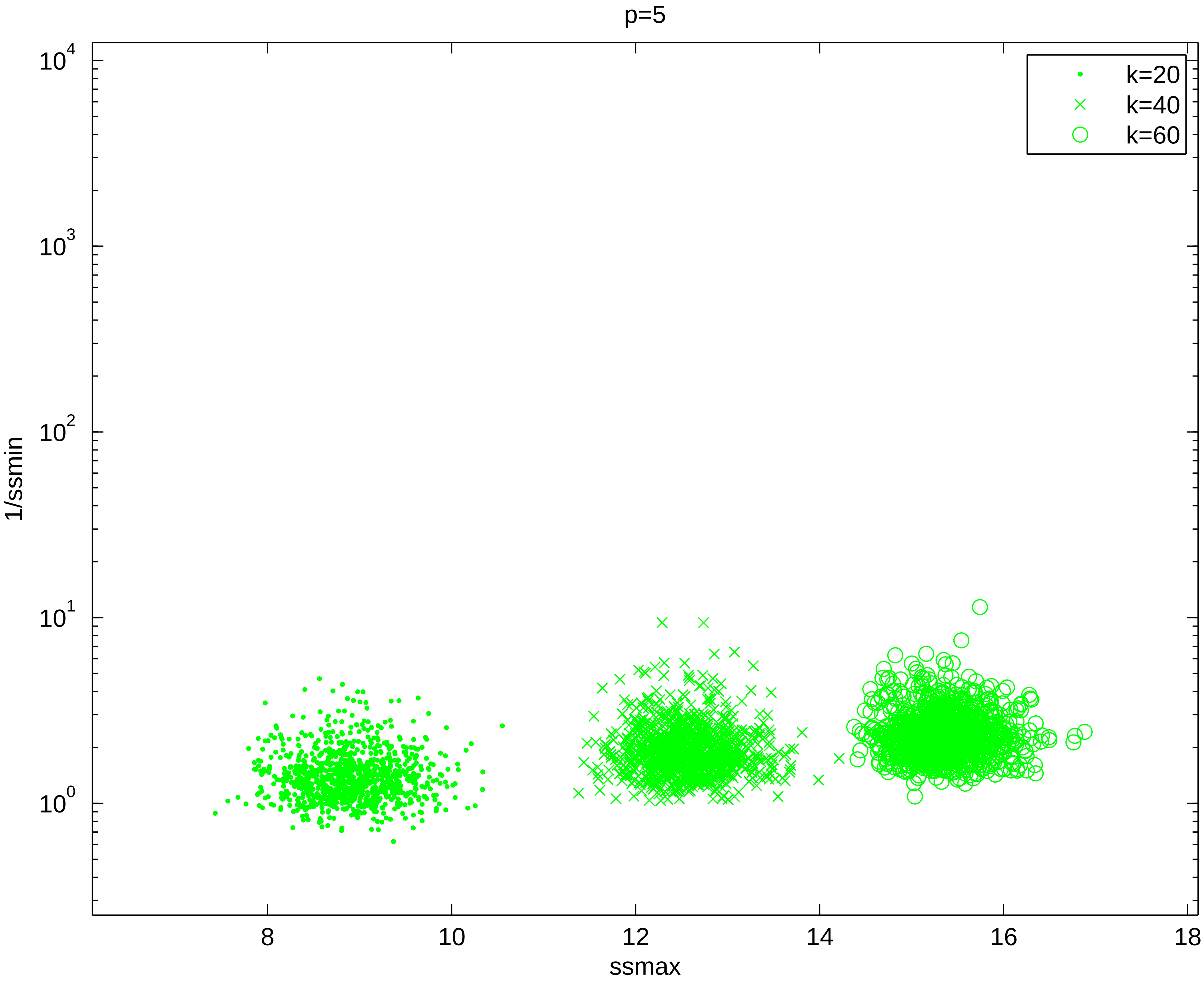
$1/\sigma_{\min}$ is plotted against $\sigma_{\max}$.

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



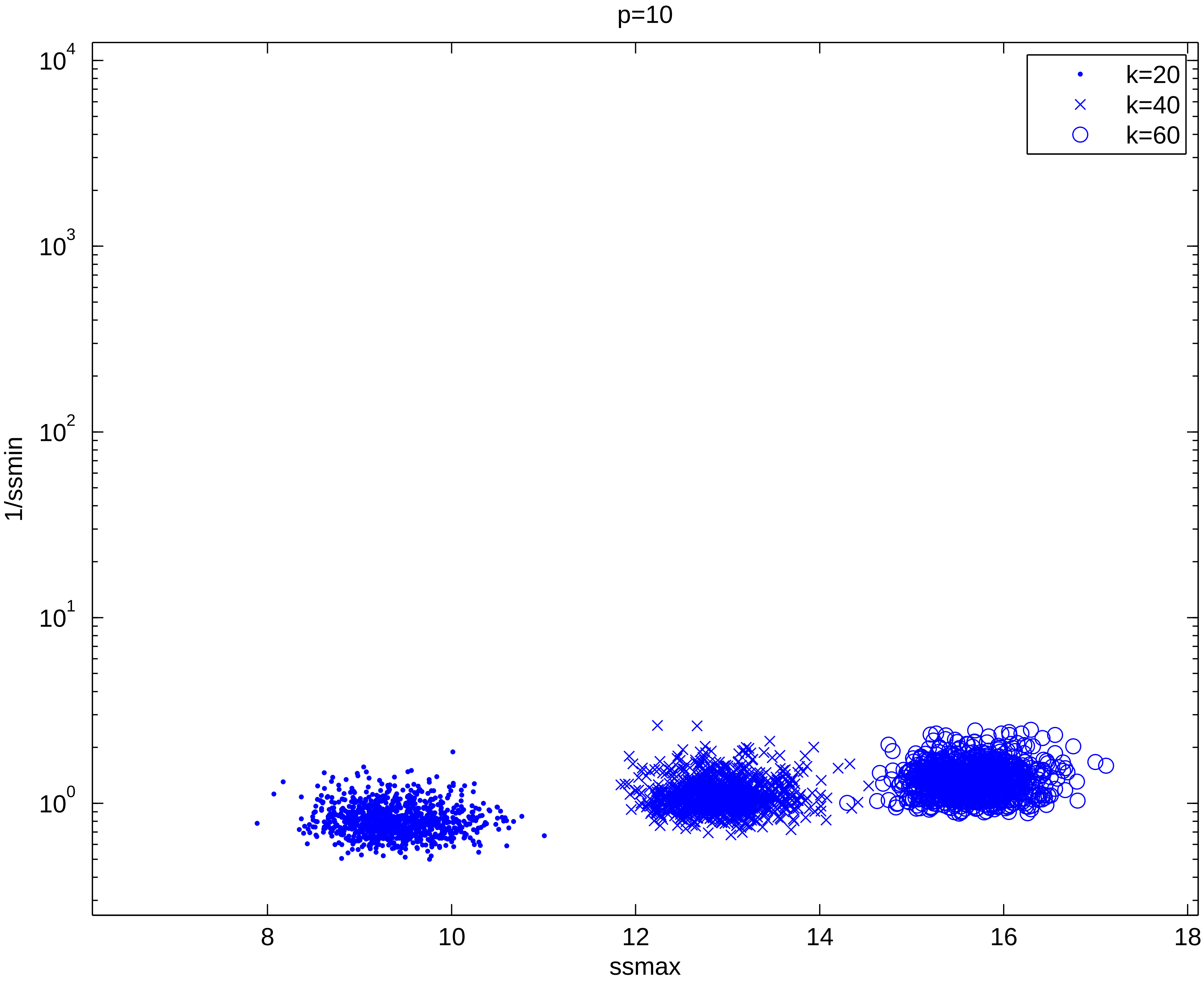
$1/\sigma_{\min}$ *is plotted against* $\sigma_{\max}$.

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



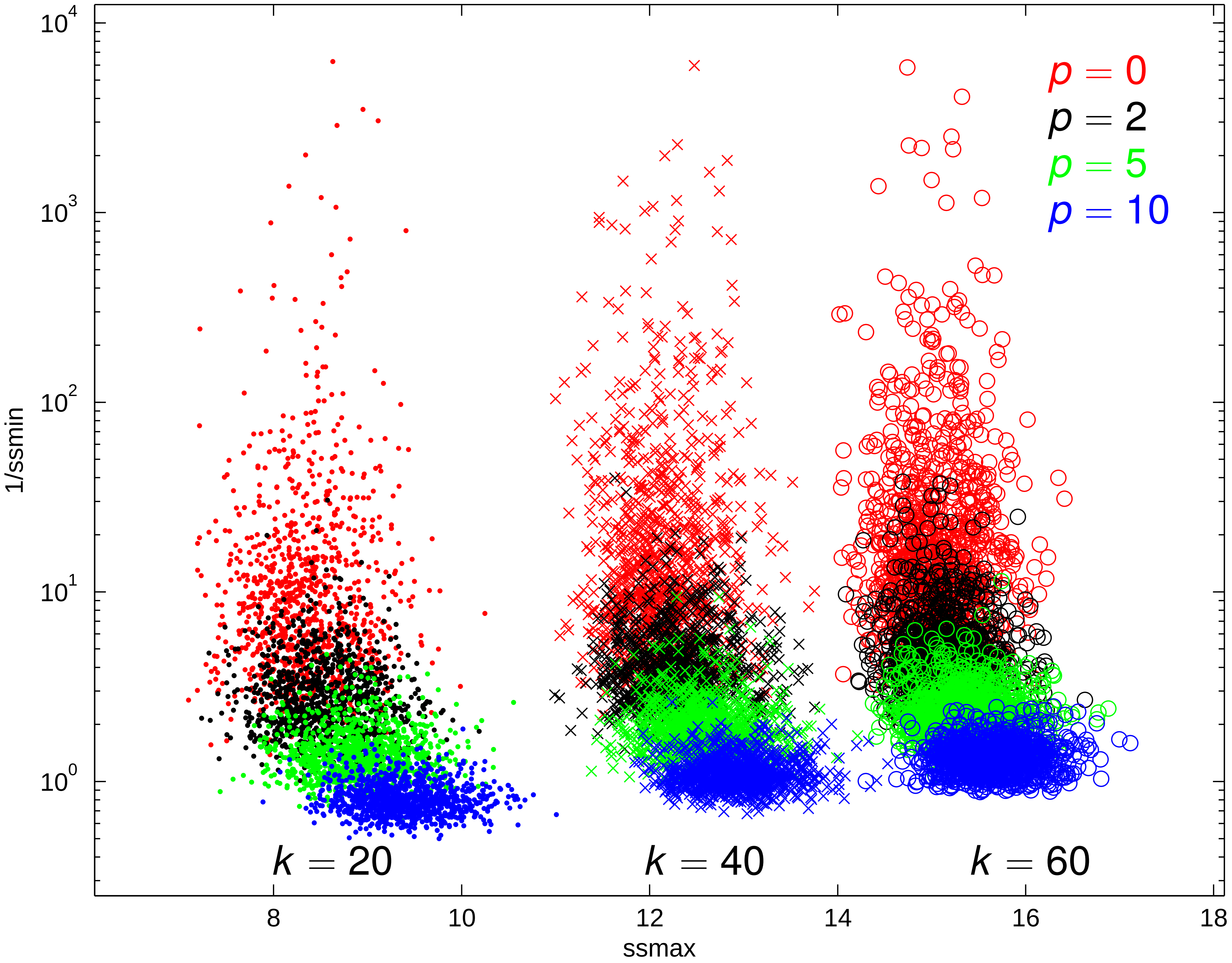
$1/\sigma_{\min}$ *is plotted against* $\sigma_{\max}$.

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.



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

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

Let $\mathbf{G}$ 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

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

Gershgorin's circle theorem will now show that $\mathbf{G}$ 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

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

$$(2) \qquad \mathbb{E}\left[||\mathbf{SGT}||\right] \leq ||\mathbf{S}||\,||\mathbf{T}||_{\mathrm{F}} + ||\mathbf{S}||_{\mathrm{F}}\,||\mathbf{T}||$$

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

$$(3) \qquad \left(\mathbb{E}\left[||\mathbf{G}^{\dagger}||_{\mathrm{F}}^{2}\right]\right)^{1/2} \leq \sqrt{\frac{k}{p-1}}$$

$$(4) \qquad \mathbb{E}\left[||\mathbf{G}^{\dagger}||\right] \leq \frac{e\sqrt{k+p}}{p}.$$

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

$$(5) \qquad \mathbb{P}\left[h(\mathbf{G}) > \mathbb{E}[h(\mathbf{G})] + L\,u\right] \leq e^{-u^2/2}.$$

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

$$(6) \qquad \mathbb{P}\left[||\mathbf{G}^{\dagger}||_{\mathrm{F}} \geq \sqrt{\frac{3k}{p+1}}t\right] \leq t^{-p}$$

$$(7) \qquad \mathbb{P}\left[||\mathbf{G}^{\dagger}|| \geq \frac{e\sqrt{k+p}}{p+1}t\right] \leq t^{-(p+1)}$$

Recall: $||\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}||^2 \leq ||\mathbf{D}_2||^2 + ||\mathbf{D}_2\mathbf{G}_2\mathbf{G}_1^\dagger||^2$, where $\mathbf{G}_1$ and $\mathbf{G}_2$ are Gaussian and $\mathbf{G}_1$ 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

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

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

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

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

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

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

$$\left(\mathbb{E}\left[||\mathbf{SGT}||_{\mathrm{F}}^{2}\right]\right)^{1/2} \leq ||\mathbf{S}||_{\mathrm{F}}\,||\mathbf{T}||_{\mathrm{F}} \tag{8}$$

$$\mathbb{E}\left[||\mathbf{SGT}||\right] \leq ||\mathbf{S}||\,||\mathbf{T}||_{\mathrm{F}} + ||\mathbf{S}||_{\mathrm{F}}\,||\mathbf{T}|| \tag{9}$$

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

$$\left(\mathbb{E}\left[||\mathbf{G}^{\dagger}||_{\mathrm{F}}^{2}\right]\right)^{1/2} \leq \sqrt{\frac{k}{p-1}} \tag{10}$$

$$\mathbb{E}\left[||\mathbf{G}^{\dagger}||\right] \leq \frac{e\sqrt{k+p}}{p}. \tag{11}$$

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

$$\mathbb{P}\left[h(\mathbf{G}) > \mathbb{E}[h(\mathbf{G})] + L\,u\right] \leq e^{-u^2/2}. \tag{12}$$

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

$$\mathbb{P}\left[||\mathbf{G}^{\dagger}||_{\mathrm{F}} \geq \sqrt{\frac{3k}{p+1}}t\right] \leq t^{-p} \tag{13}$$

$$\mathbb{P}\left[||\mathbf{G}^{\dagger}|| \geq \frac{e\sqrt{k+p}}{p+1}t\right] \leq t^{-(p+1)} \tag{14}$$

Recall: $||\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}||^2 \leq ||\mathbf{D}_2||^2 + ||\mathbf{D}_2\mathbf{G}_2\mathbf{G}_1^\dagger||^2$, where $\mathbf{G}_1$ and $\mathbf{G}_2$ are Gaussian and $\mathbf{G}_1$ 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 $E_t = \left\{||\mathbf{G}_1|| \leq \frac{e\sqrt{k+p}}{p+1}t \quad \text{and} \quad ||\mathbf{G}_1^\dagger||_{\mathrm{F}} \leq \sqrt{\frac{3k}{p+1}}\,t\right\}$. By Proposition 4: $\mathbb{P}(E_t^{\mathrm{c}}) \leq 2\,t^{-p}$.

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

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

Hold Ω_1 fixed and take the expectation on Ω_2. Then Proposition 1 applies and so

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

Now use Proposition 3 (concentration of measure)

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

When E_t holds true, we have bounds on the “badness” of $\Omega_1^\dagger$:

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

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

Bibliography:

- Original paper: P.G. Martinsson, V. Rokhlin, and M. Tygert, “A randomized algorithm for the decomposition of matrices”. *Applied and Computational Harmonic Analysis*, **30**(1), pp. 47–68, 2011. (This is the published version of our *2006* tech report YALEU/DCS/TR-1361 . . .)
- Acceleration from $O(mnk)$ to $O(mn\log(k))$: F. Woolfe, E. Liberty, V. Rokhlin, M. Tygert “A fast randomized algorithm for the approximation of matrices” *Applied and Computational Harmonic Analysis*, **25**, pp. 335–366, 2008.
- Out-of-core version: N. Halko, P.G. Martinsson, Y. Shkolnisky, M. Tygert, “An Algorithm for the Principal Component Analysis of large Data Sets”. *SIAM J. on Scientific Computation*, **33**(5), pp. 2580–2594, 2011.
- Brief survey: E. Liberty, F. Woolfe, P.G. Martinsson, V. Rokhlin, and M. Tygert, “Randomized algorithms for the low-rank approximation of matrices”. *Proc. of the National Academy of Sciences*, **104**(51), 2007.
- Long survey: N. Halko, P.G. Martinsson, J. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions.” *SIAM Review*, **53**(2), pp. 217–288, 2011.
- A third survey: M. Mahoney, “Randomized algorithms for matrices and data” Foundations and Trends in Machine Learning, **3**(2), 2011.

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. Free software can be downloaded $\rightarrow$ google *Mark Tygert*.

- The approximation error is a random variable, but its distribution is very narrowly concentrated. Any preset accuracy can be met to within probability $1 - \eta$ where η is a user set "failure probability" (*e.g.* $\eta = 10^{-10}$ or 10^{-20}).

- This lecture mentioned *error estimators* only briefly, 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, but *experimentation is easy!* Concentration of measure makes the algorithms behave as if deterministic.

- To find out more:
  - A tutorial long version of this talk is available on the *NIPS 2009* website.
  - Review: *Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions* N. Halko, P.G. Martinsson, J. Tropp — SIAM Review, **53**(2), 2011.