Randomized Low-Rank Preconditioners

Benjamin Ong and Allan Struthers

Michigan Technological University
onbw@mtu.edu

June 16, 2022

Michigan Tech
Create the Future

Setting

Goal: Accelerate solution to

$$Ax = b,$$

where

- Scenario 1: $A \in \mathbb{R}^{m \times n}$ is big, hard to access.
- Scenario 2: A contains sensitive data. Data gatekeeper only willing to give weighted averages of columns/rows.

Michigan Tech
Create the Future

Our approach: Randomization

Iteratively utilize two-sided samples of the matrix

$$U_k^\top A V_k$$

For simplicity, let $A \in \mathbb{R}^{m \times m}$, $\{U_k . V_k\} \in \mathbb{R}^{m \times s}$, $s \ll m$

- $\{U_k, V_k\}$: sampling matrices

- two-sided matrix sample, $U_k^\top A V_k$

Michigan Tech
Create the Future

Today's story:

Iteratively utilize two-sided samples of the matrix

$$U_k^\top A V_k$$

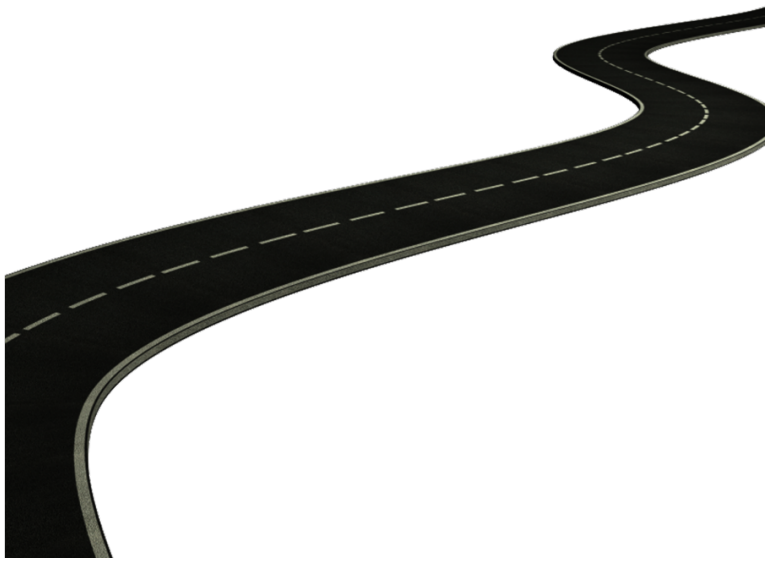
to find $(A)_r$, where

$$(A)_r := \arg\min_B \{ \|A - B\|_2 + \delta \mid \mathrm{rank}(B) \leq r \}$$

where r is the (unknown) numerical rank of A.

Michigan Tech
Create the Future

Road-map



1. Our previous work:
   Approximating A
   using two-sided samples
2. Modifications to find $(A)_r$
   using two-sided samples
3. Numerical evidence

Michigan Tech
Create the Future

Previous work: Approximating A

Want: $\{B_k\} \to A$ in the sense $\|B_k - A\|_F \to 0$ as k increases.

Given B_k & two-sided samples $\{U_k^\top A\, V_k\}$.

Minimum change formulation:

$$B_{k+1} = \arg\min_B \frac{1}{2}\|B - B_k\|_F^2 \text{ subject to } U_k^\top B\, V_k = U_k^\top A\, V_k$$

This gives a self correcting update,

$$B_{k+1} = B_k + P_{U_k}(A - B_k)P_{V_k}^\top$$

where the projectors are defined as

$$P_Q = Q(Q^\top Q)^{-1}Q^\top$$

Michigan Tech
Create the Future

Previous work: Approximating A

Want: $\{B_k\} \to A$ in the sense $\|B_k - A\|_F \to 0$ as k increases.

Given B_k & two-sided samples $\{U_k^\top A\, V_k\}$.

Minimum change formulation:

$$B_{k+1} = \arg\min_B \frac{1}{2}\|B - B_k\|_F^2 \text{ subject to } U_k^\top B\, V_k = U_k^\top A\, V_k$$

This gives a self correcting update,

$$B_{k+1} = B_k + P_{U_k}(A - B_k)P_{V_k}^\top$$

where the projectors are defined as

$$P_Q = Q(Q^\top Q)^{-1}Q^\top$$

Michigan Tech
Create the Future

Previous work: Approximating A

Want: $\{B_k\} \to A$ in the sense $\|B_k - A\|_F \to 0$ as k increases.

Given B_k & two-sided samples $\{U_k^\top A\, V_k\}$.

Minimum change formulation:

$$B_{k+1} = \arg\min_B \frac{1}{2}\|B - B_k\|_F^2 \text{ subject to } U_k^\top B\, V_k = U_k^\top A\, V_k$$

This gives a self correcting update,

$$B_{k+1} = B_k + P_{U_k}(A - B_k)P_{V_k}^\top$$

where the projectors are defined as

$$P_Q = Q(Q^\top Q)^{-1}Q^\top$$

Previous work: Approximating A (Algorithm)

Require: $B_0 \in \mathbb{R}^{m \times n}$, $s \in \mathbb{N}$.

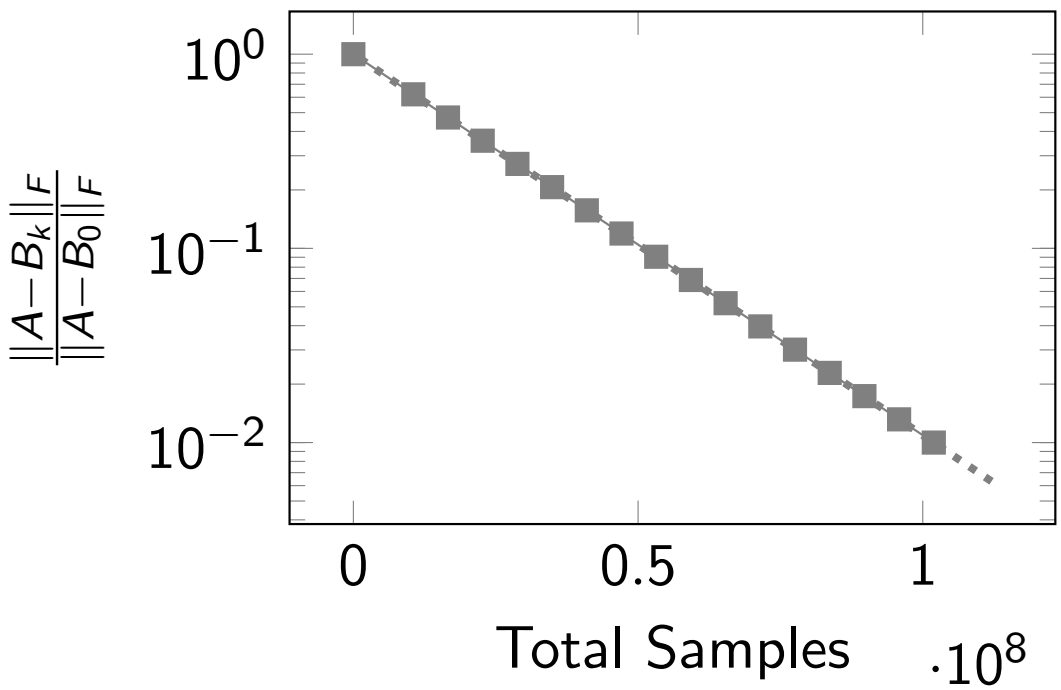
1: **repeat** $\{k = 0, 1, \ldots\}$
2: Generate: $U_k \sim N(0,1)^{m \times s}$ and $V_k \sim N(0,1)^{n \times s}$
3: Compute residual: $\Lambda_k = U_k^\top A V_k - U_k^\top B_k V_k \in \mathbb{R}^{s \times s}$
4: Update: $B_{k+1} = B_k + U_k (U_k^\top U_k)^{-1} \Lambda_k (V_k^\top V_k)^{-1} V_k^\top$
5: **until** convergence
6: **return** B_{k+1}

Michigan Tech
Create the Future

Previous Work: doi:10.1007/978-3-030-95470-3_17

If $\{U_k, V_k\}$ sampled from rotationally invariant distributions, have provable convergence rates (in expectation) realized in numerical experiments.

Approximation of **NASA4704** (Suite Sparse), $n = 4700, s = 69$



Motivation

But . . .

- Iterates $\{B_k\}$ can be hard (expensive) to store
  Solution: only keep low-rank iterates

Want:

- supporting theoretical convergence estimates
- incrementally update a low-rank factorization efficiently
- rank revealing

Michigan Tech
Create the Future

Keeping Low-Rank Iterates

Require: $B_0 \in \mathbb{R}^{m\times n}$, $s \in \mathbb{N}$.

1: **repeat** $\{k = 0, 1, \ldots\}$
2: Generate: $U_k \sim N(0,1)^{m\times s}$ and $V_k \sim N(0,1)^{n\times s}$
3: Compute residual: $\Lambda_k = U_k^\top A\, V_k - U_k^\top B_k V_k \in \mathbb{R}^{s\times s}$
4: Update: $\hat{B}_{k+1} = B_k + U_k(U_k^\top U_k)^{-1}\Lambda_k(V_k^\top V_k)^{-1}V_k^\top$
5: $B_{k+1} = \text{LowRankApprox}[\hat{B}_{k+1}]$
6: **until** convergence
7: **return** B_{k+1}

- terribly inefficient algorithm: improve shortly.
- if desired rank unspecified (or unknown), choose

$$\frac{\|B_{k+1}\|_F^2}{\|\hat{B}_{k+1}\|_F^2} \leq 1 - \gamma$$

Michigan Tech
Create the Future

Keeping Low-Rank Iterates

Require: $B_0 \in \mathbb{R}^{m\times n}$, $s \in \mathbb{N}$.

1: **repeat** $\{k = 0, 1, \ldots\}$
2: Generate: $U_k \sim N(0,1)^{m\times s}$ and $V_k \sim N(0,1)^{n\times s}$
3: Compute residual: $\Lambda_k = U_k^\top A\, V_k - U_k^\top B_k V_k \in \mathbb{R}^{s\times s}$
4: Update: $\hat{B}_{k+1} = B_k + U_k(U_k^\top U_k)^{-1}\Lambda_k(V_k^\top V_k)^{-1}V_k^\top$
5: $B_{k+1} = \text{LowRankApprox}[\hat{B}_{k+1}]$
6: **until** convergence
7: **return** B_{k+1}

- terribly inefficient algorithm: improve shortly.
- if desired rank unspecified (or unknown), choose

$$\frac{\|B_{k+1}\|_F^2}{\|\hat{B}_{k+1}\|_F^2} \leq 1 - \gamma$$

Michigan Tech
Create the Future

Why?

Choose this simplistic implementation (first) because we anticipated theoretical estimates possible:

(Penalized) Minimum Change:

$$B_{k+1} = \arg\min_{B} \frac{1}{2}\|B - B_k\|_F^2 + \lambda\|B\|_*$$

again, subject to $U_k^\top B\, V_k = U_k^\top A\, V_k$

$\|\cdot\|_*$: nuclear norm (commonly used proxy for matrix rank)

Need $\lambda = \sigma_r/\sigma_1$ to get a rank-r cutoff.

Michigan Tech
Create the Future

Why?

Choose this simplistic implementation (first) because we anticipated theoretical estimates possible:

(Penalized) Minimum Change:

$$B_{k+1} = \arg\min_{B} \frac{1}{2}\|B - B_k\|_F^2 + \lambda\|B\|_*$$

again, subject to $U_k^\top B\, V_k = U_k^\top A\, V_k$

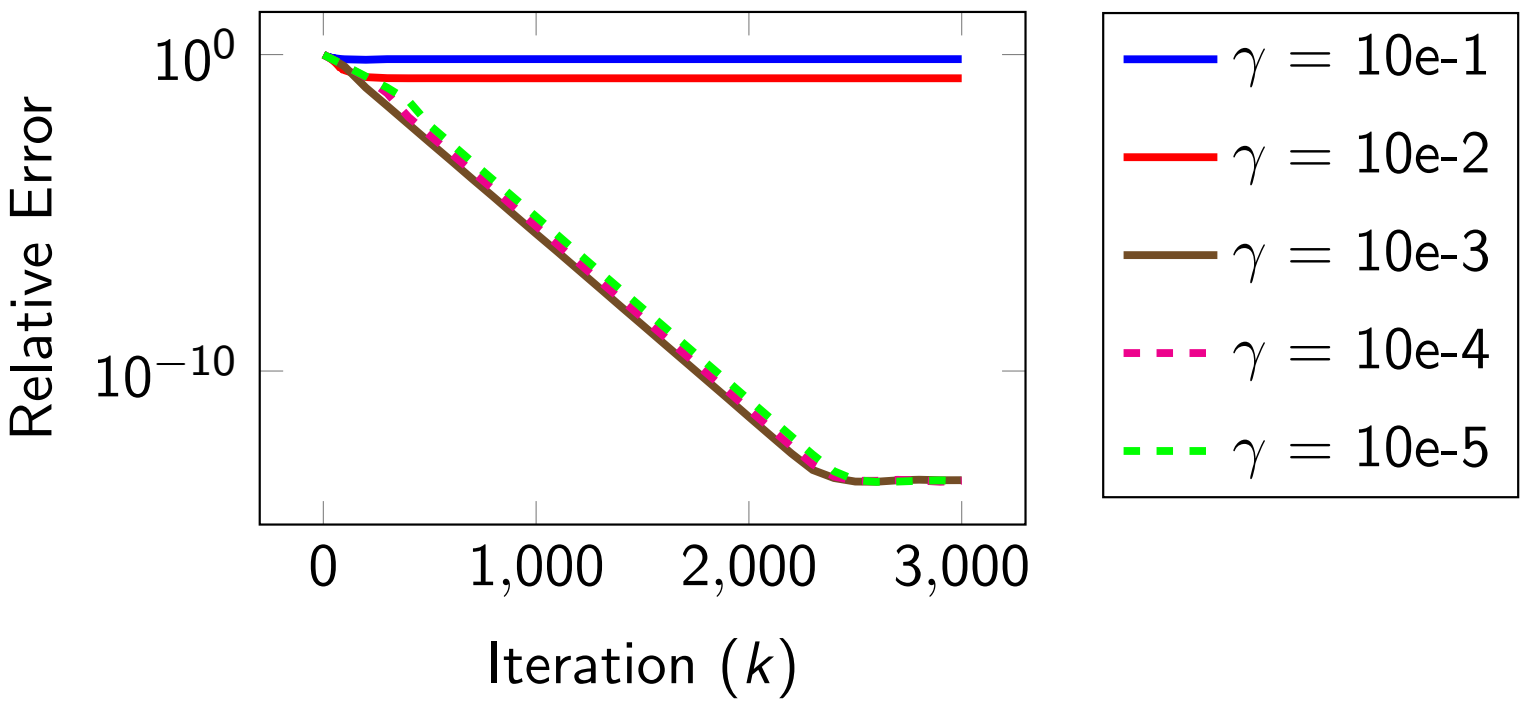
$\|\cdot\|_*$: nuclear norm (commonly used proxy for matrix rank)

Need $\lambda = \sigma_r/\sigma_1$ to get a rank-r cutoff.

Demonstration

$$A = \sum_{i=1}^{30} \alpha_i x_i y_i^\top, \{x_i, y_i\} \text{ unit vectors of size } 256 \times 1,$$

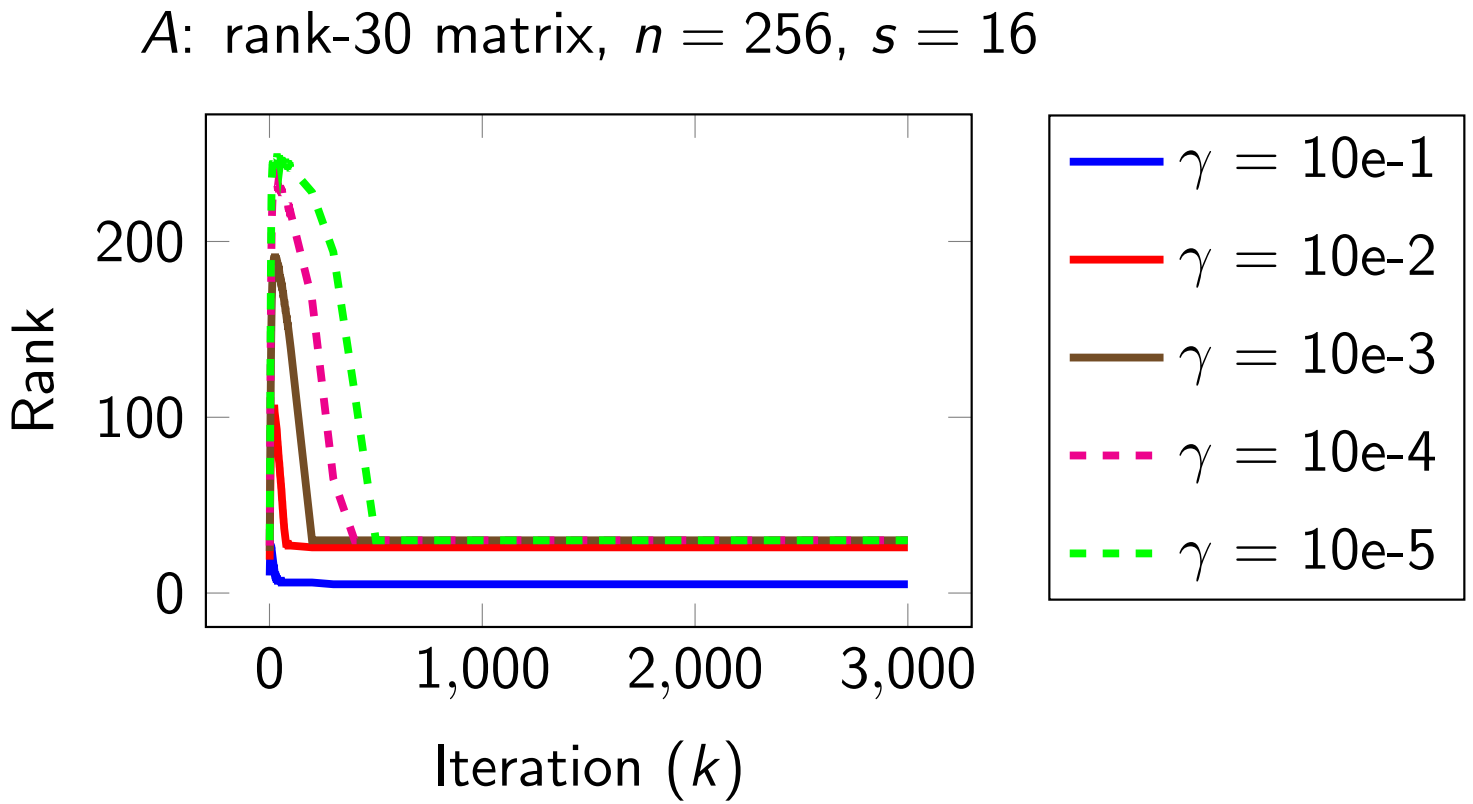
A: rank-30 matrix, $n = 256$, $s = 16$



Michigan Tech
Create the Future

Demonstration

But only sorta works . . .



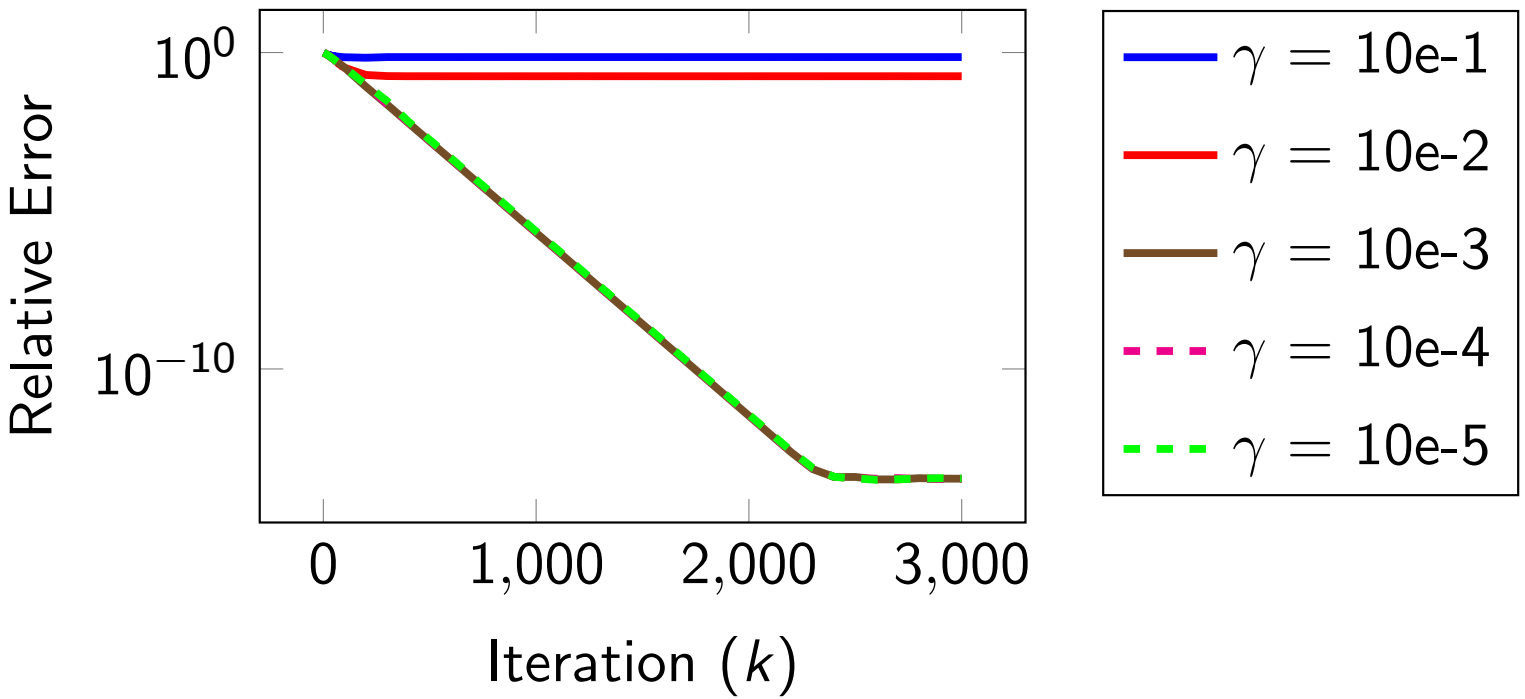
Intermediate iterates may not resolve low-rank structure.

MichiganTech
Create the Future

Demonstration: band-aid

Band-aid: restrict maximum rank of iterates that can be kept.

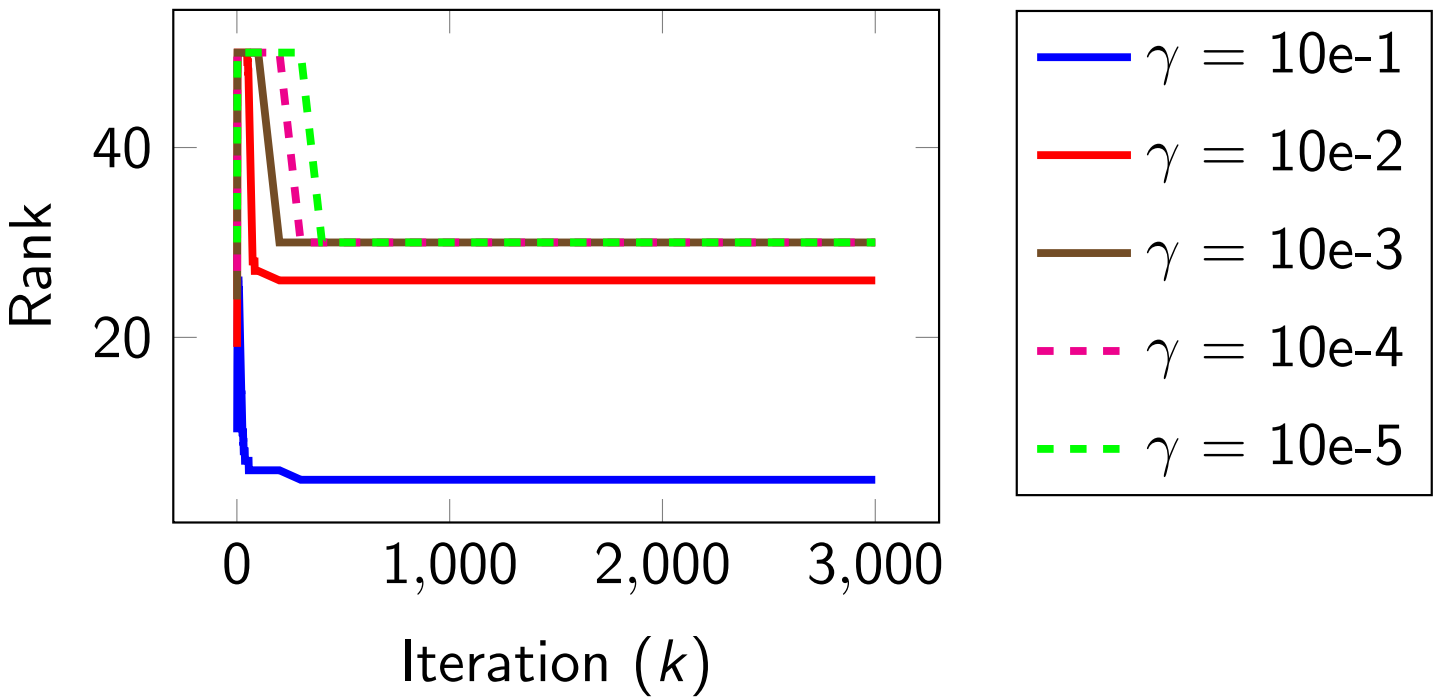
A: rank-30 matrix, $n = 256$, $s = 16$, $R = 50$



Demonstration

Better . . .

A: rank-30 matrix, $n = 256$, $s = 16$, $R = 50$



Instead . . .

- storing/working with SVD factors $U_k\, S_k\, V_k^\top$
- incrementally update (cheaply) each iterate with two-sided sample

Details:

- Augment singular spaces with orthogonal directions to generate expanded orthogonal sampling spaces

$$\mathbb{U}_{k+1} = (U_k \mid \mathcal{U}_k), \qquad \mathbb{V}_{k+1} = (V_k \mid \mathcal{V}_k)$$

New sampled data:

$$\mathbb{U}_{k+1}^\top A\, \mathbb{V}_{k+1} = \begin{bmatrix} U_k^\top A\, V_k & U_k^\top A\, \mathcal{V}_k \\ \mathcal{U}_k^\top A\, V_k & \mathcal{U}_k^\top A\, \mathcal{V}_k \end{bmatrix}$$

Michigan Tech
Create the Future

Instead . . .

- storing/working with SVD factors $U_k\, S_k\, V_k^\top$
- incrementally update (cheaply) each iterate with two-sided sample

Details:

- Augment singular spaces with orthogonal directions to generate expanded orthogonal sampling spaces

$$\mathbb{U}_{k+1} = (U_k \mid \mathcal{U}_k), \qquad \mathbb{V}_{k+1} = (V_k \mid \mathcal{V}_k)$$

New sampled data:

$$\mathbb{U}_{k+1}^\top A\, \mathbb{V}_{k+1} = \begin{bmatrix} U_k^\top A\, V_k & U_k^\top A\, \mathcal{V}_k \\ \mathcal{U}_k^\top A\, V_k & \mathcal{U}_k^\top A\, \mathcal{V}_k \end{bmatrix}$$

Working in Reduced Subspace

- Always working now in the "Reduced" subspace
- (Penalized) Minimum change formulation (in reduced space)

$$\min \frac{\|\mathbb{U}_{k+1}^{\top} A \mathbb{V}_{k+1}^{\top} - B\|_F^2}{\|\mathbb{U}_{k+1}^{\top} A \mathbb{V}_{k+1}\|_F^2} + 2\lambda \frac{\|B\|_*}{\|\mathbb{U}_{k+1}^{\top} A \mathbb{V}_{k+1}\|_?}$$

where the minimization argument B now is in the small-small space and computable.

Michigan Tech
Create the Future

Expanded Orthogonal Space

- Generate expanded orthogonal sampling spaces

$$\mathbb{U}_{k+1} = (U_k \mid \mathcal{U}_k), \qquad \mathbb{V}_{k+1} = (V_k \mid \mathcal{V}_k)$$

using an explicit block Householder rotation
doi://10.1016/S0893-9659(99)00028-2

- Data in the (expanded) orthogonal space is an arrowhead matrix

$$\mathbb{U}_{k+1}^\top A \mathbb{V}_{k+1} = \begin{bmatrix} \Sigma_k & U_k^\top A \mathcal{V}_k \\ \mathcal{U}_k^\top A V_k & \mathcal{U}_k^\top A \mathcal{V}_k \end{bmatrix}$$

SVD of this arrowhead matrix can be computed as a rank-s update to the factorization $U_k \sigma_k V_k^\top$.

Michigan Tech
Create the Future

Expanded Orthogonal Space

- Generate expanded orthogonal sampling spaces

$$\mathbb{U}_{k+1} = (U_k \mid \mathcal{U}_k), \qquad \mathbb{V}_{k+1} = (V_k \mid \mathcal{V}_k)$$

using an explicit block Householder rotation
doi://10.1016/S0893-9659(99)00028-2

- Data in the (expanded) orthogonal space is an arrowhead matrix

$$\mathbb{U}_{k+1}^\top A \mathbb{V}_{k+1} = \begin{bmatrix} \Sigma_k & U_k^\top A \mathcal{V}_k \\ \mathcal{U}_k^\top A V_k & \mathcal{U}_k^\top A \mathcal{V}_k \end{bmatrix}$$

SVD of this arrowhead matrix can be computed as a rank-s update to the factorization $U_k \sigma_k V_k^\top$.

Michigan Tech
Create the Future

Expanded Orthogonal Space

- Generate expanded orthogonal sampling spaces

$$\mathbb{U}_{k+1} = (U_k \mid \mathcal{U}_k), \qquad \mathbb{V}_{k+1} = (V_k \mid \mathcal{V}_k)$$

  using an explicit block Householder rotation
  doi://10.1016/S0893-9659(99)00028-2

- Data in the (expanded) orthogonal space is an arrowhead matrix

$$\mathbb{U}_{k+1}^\top A \, \mathbb{V}_{k+1} = \begin{bmatrix} \Sigma_k & U_k^\top A \, \mathcal{V}_k \\ \mathcal{U}_k^\top A \, V_k & \mathcal{U}_k^\top A \, \mathcal{V}_k \end{bmatrix}$$

  SVD of this arrowhead matrix can be computed as a rank-s update to the factorization $U_k \sigma_k V_k^\top$.

Michigan Tech
Create the Future

Efficient Low-Rank Update Algorithm

1: Specify λ and $\Sigma_0, U_0, V_0, s, \beta_0$
2: **repeat** $\{k = 0, 1, \ldots\}$
3: $\quad \mathcal{U}_k = \text{RandomOrthogonallyAugment}[U_k, s]$
4: $\quad \mathcal{V}_k = \text{RandomOrthogonallyAugment}[V_k, s]$
5: $\quad$ Assemble $\Lambda_k = \mathbb{U}_{k+1}^\top A \mathbb{V}_{k+1} = \begin{bmatrix} \Sigma_k & U_k^\top A \mathcal{V}_k \\ \mathcal{U}_k^\top A V_k & \mathcal{U}_k^\top A \mathcal{V}_k \end{bmatrix}$
6: $\quad$ Compute SVD factorization, $\Lambda_k = R_k S_k T^\top$
7: $\quad$ Set $\beta_{k+1} = \max(\beta_k, ||\Lambda_k||_F)$. (ensures monotonicity)
8: $\quad$ Solution to Penalized Minimization Problem:

$$R_{k+1} \leftarrow \text{Threshold}[R_k, \lambda \beta_{k+1}]$$
$$\Sigma_{k+1} \leftarrow \text{Threshold}[S_k, \lambda \beta_{k+1}]$$
$$T_{k+1} \leftarrow \text{Threshold}[T_k, \lambda \beta_{k+1}]$$

9: $\quad$ Embed in expanded space:

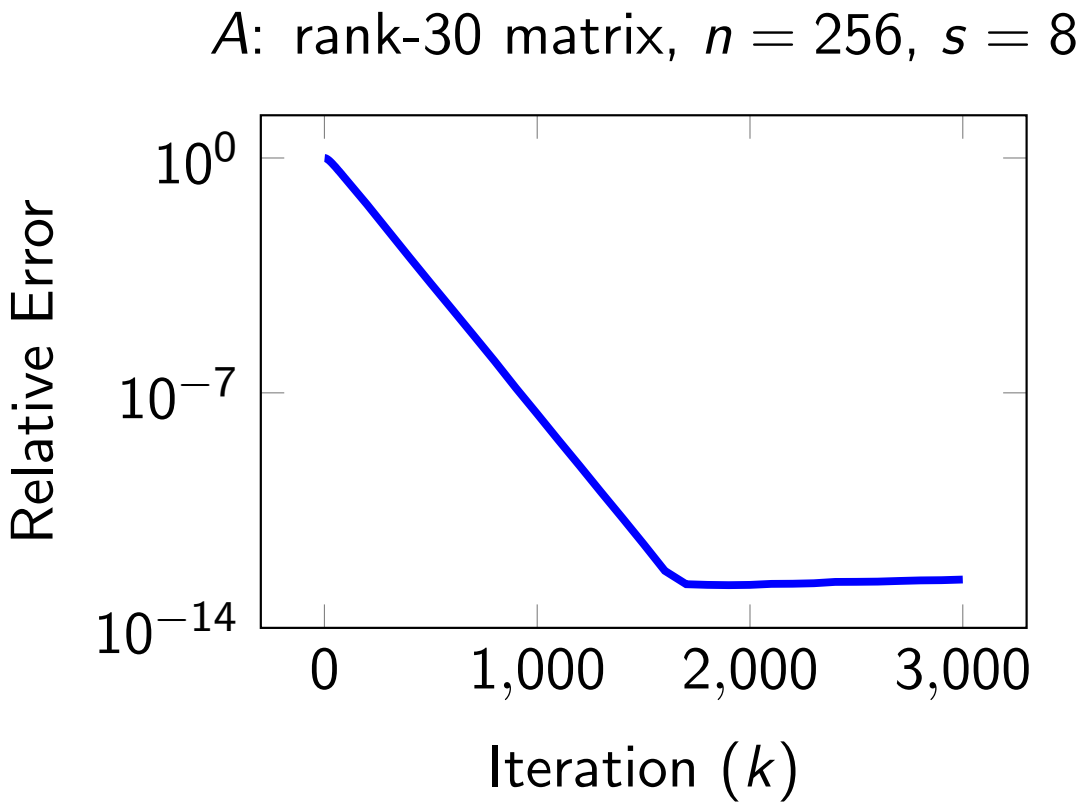
$$U_{k+1} = \mathbb{U}_{k+1} R_{k+1}$$
$$V_{k+1} = \mathbb{V}_{k+1} T_{k+1}$$

10: **until** convergence

MichiganTech
Create the Future

Numerical Evidence

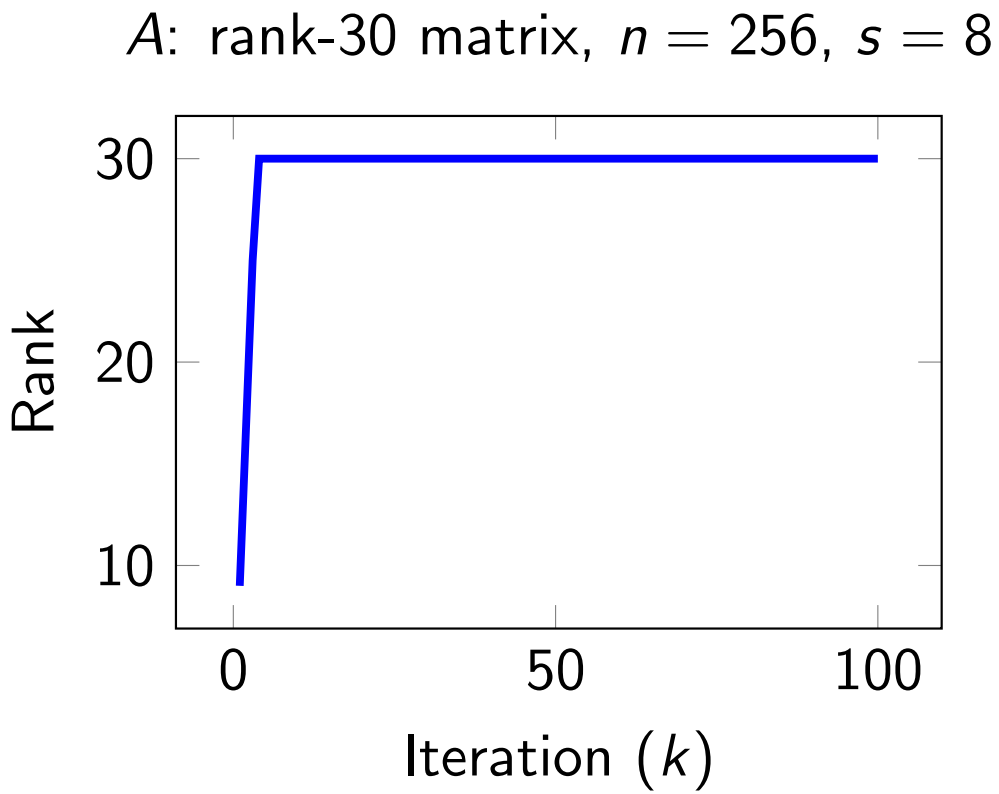
Converges!



Michigan Tech
Create the Future

Numerical Evidence

Reveals rank! no extra storage needed!



Michigan Tech
Create the Future

Conclusion & Future Work

Conclusions:

- Practical algorithm iteratively embeds two-sided sample of A to generate $(A)_r$
- If tuned correctly, reveals matrix rank.

Future Work:

- Can prove convergence (in expectation) if β is monotonically decreasing.
- Currently working on rate of convergence estimates
- Complicated because no longer sampling from rotationally invariant subspace

Michigan Tech
Create the Future