

Matrices and Numerical Stability

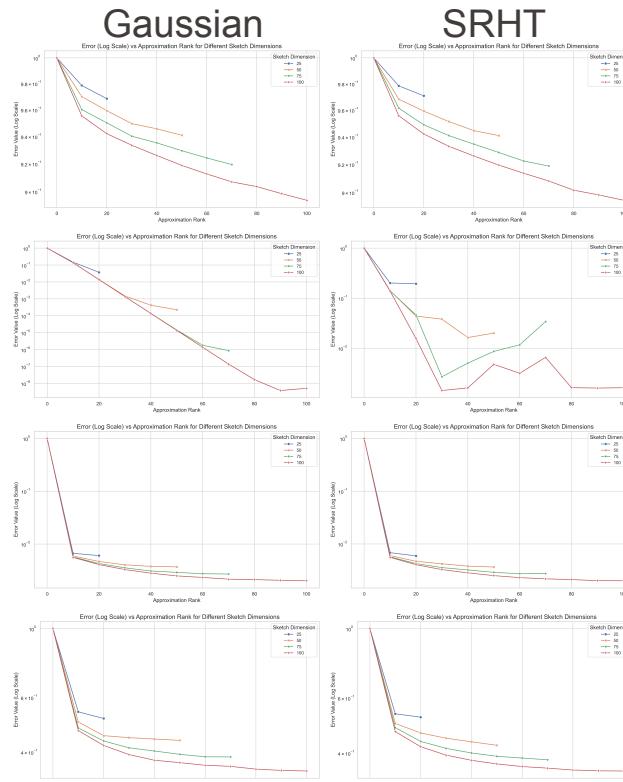
Matrices Summary

A1: Polynomial decay matrix with $R = 5$ and slow decay $p = 0.5$

A2: Exponential decay matrix with $R = 5$ and slow decay $q = 0.1$

A3: MNIST 780 dataset processed with an RBF kernel, $c = 100$

A4: MNIST 780 dataset processed with an RBF kernel, $c = 10$



HPC Report
Runchi Tan

Sequential and Parallel Nyström

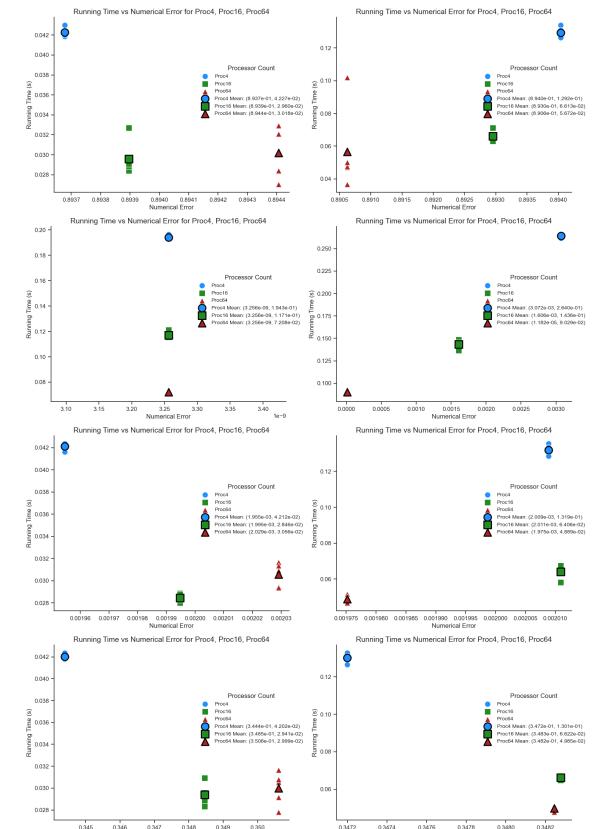
Algorithm

Algorithm 2 Distributed Low-Rank Approximation

- 1: **Step 1: MPI Setup and matrix partitioning**
- 2: **Initialize MPI:** Each processor obtains a unique rank from 0 to size – 1 and is arranged in a 2D grid.
- 3: **Each processor** (i, j) loads a local sub-block \mathbf{A}_{ij} of the global matrix \mathbf{A} .
- 4: **Step 2: Seed broadcasting and local generation**
 - 5: **On root processor:**
 - 6: Generate a base seed for reproducibility.
 - 7: Spawn distinct seeds for each block of Ω (plus one more if using block SRHT for the subsampling indices).
 - 8: **Broadcast** all seeds so that each processor P_{ij} can generate Ω_i and Ω_j .
 - 9: **Local Generation:**
 - 10: Each processor locally generates its sketching matrices (Gaussian) or elements to apply the sketching (block SRHT).
- 11: **Step 3: Compute C and B**
 - 12: **If block SRHT:** Apply sign flips, Hadamard transforms, and subsampling on the rows to compute \mathbf{C}_{ij} and \mathbf{B}_{ij} .
 - 13: **If Gaussian:** Compute \mathbf{C}_{ij} and \mathbf{B}_{ij} using matrix multiplication.
- 14: **Step 4: Reduction**
 - 15: Reduce \mathbf{C}_{ij} to processor P_{i1} (to form \mathbf{C}_i).
 - 16: Reduce \mathbf{B}_{ij} to processor P_{i1} .
 - 17: **Only first column of processors:**
 - 18: Allreduce \mathbf{B}_i to form matrix \mathbf{B} .
- 19: **Step 5: Create Z (only first column)**
 - 20: **If well-conditioned:** Use Cholesky decomposition to compute $\mathbf{B} = \mathbf{L}\mathbf{L}^\top$ and $\mathbf{Z}_i = \mathbf{C}_i\mathbf{L}^{-\top}$.
 - 21: **Otherwise:** Use eigenvalue decomposition to compute $\mathbf{B} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$ and $\mathbf{Z}_i = \mathbf{C}_i\mathbf{X}\mathbf{\Lambda}^{-0.5}$.
- 22: **Step 6: Distributed QR via TSQR (only first column)**
 - 23: Factorize \mathbf{Z}_i using TSQR to obtain \mathbf{Q}_i and \mathbf{R} .
- 24: **Step 7: SVD on R and Low-Rank Reconstruction (only first column)**
 - 25: Perform SVD on \mathbf{R} to compute $\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.
 - 26: Truncate the factors to the desired rank to obtain \mathbf{U}_k and $\mathbf{\Sigma}_k$.
- 27: **Step 8:** Compute $\hat{\mathbf{U}}_{k,i} = \mathbf{Q}_i \mathbf{U}_k$.
- 28: **Allgather** $\hat{\mathbf{U}}_{k,i}$ to form $\hat{\mathbf{U}}_k$.
- 29: **Step 9:** Assemble the local approximation:

$$[\![\mathbf{A}_{\text{Nyström}}]\!]_{k,i} = \hat{\mathbf{U}}_{k,i} \mathbf{\Sigma}_k^2 \hat{\mathbf{U}}_k^\top.$$

Runtime vs Error



Sequential and Parallel Performance

