Efficient Low-Rank Matrix Approximation: A Comparative Analysis of Deterministic and Randomized Algorithms

Su Yanzhen, Tahir Mahamat Saleh, Coulibaly Zie Siaka

December 3, 2025

Abstract

Matrix factorization is fundamental to modern data science, yet classical deterministic algorithms often struggle with computational efficiency on massive datasets or convergence stability in non-convex problems. This project implements and compares Deterministic (Normal) versus Randomized approaches for three decomposition techniques: Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), and CUR Decomposition. Using a custom-built R Shiny dashboard and real-world datasets (MovieLens and Volcano Topography), we demonstrate that randomized initialization strategies significantly improve NMF convergence on sparse data, while Randomized SVD robustly captures natural low-rank structures in dense topographic data.

1 Introduction

Matrix decomposition serves as the backbone for techniques ranging from dimensionality reduction to recommender systems. In this project, we explore **Efficient Low-Rank Matrix Approximation** by implementing algorithms from scratch. We focus on three primary methods:

- **SVD:** The gold standard for low-rank approximation.
- **NMF:** Useful for parts-based representation and interpretability (requiring non-negative constraints).
- **CUR:** Provides interpretability by selecting actual columns and rows from the data.

Our core contribution is the comparative implementation of **Deterministic (Normal)** versus **Randomized** versions for each algorithm, analyzing their performance on real-world datasets.

2 Motivation

We selected these algorithms to address the "Accuracy-Efficiency-Interpretability" trilemma in data analysis:

1. **The Efficiency Challenge:** Classical SVD is computationally prohibitive ($O(\min(mn^2, m^2n))$) for large matrices. We implement *Randomized SVD* to address this via subspace projection.
2. **The Convergence Challenge:** Standard NMF is non-convex and sensitive to initialization. We propose an *Optimized NMF* using randomized SVD results as a "warm start" to improve stability.
3. **The Interpretability Challenge:** Eigenvectors are often abstract. We explore *CUR Decomposition* to retain the physical meaning of data features.

3 Theory and Implementation

All algorithms were implemented from scratch in R, avoiding high-level wrapper libraries for the core logic. Below we present the pseudocode for the randomized and optimized versions developed in this project.

3.1 Singular Value Decomposition (SVD)

We approximate $\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. For the **Normal** version, we utilized the standard LAPACK wrapper. For the **Randomized** version, we implemented the stochastic algorithm described by Halko et al., which uses random projections to identify the subspace capturing the most action of the matrix.

Algorithm 1 Randomized SVD with Power Iterations

Require: Matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, Target rank k, Oversample p, Power iterations q

$\mathbf{\Omega} \leftarrow \text{randn}(n, k+p)$ ▷ Generate Gaussian random matrix
$\mathbf{Y} \leftarrow \mathbf{A}\mathbf{\Omega}$ ▷ Form the sketch
for $i = 1$ to q **do** ▷ Power Iterations for denoising
    $\mathbf{Y} \leftarrow \mathbf{A}(\mathbf{A}^T\mathbf{Y})$
end for
$\mathbf{Q} \leftarrow \text{QR\_Decomposition}(\mathbf{Y})$ ▷ Orthonormal basis
$\mathbf{B} \leftarrow \mathbf{Q}^T\mathbf{A}$ ▷ Project to low-dimensional space
$[\mathbf{U}_B, \mathbf{\Sigma}, \mathbf{V}] \leftarrow \text{Standard\_SVD}(\mathbf{B})$ ▷ Cheap SVD on small matrix
$\mathbf{U} \leftarrow \mathbf{Q}\mathbf{U}_B$ ▷ Lift back to high-dimensional space
return $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$

3.2 Non-negative Matrix Factorization (NMF)

We approximate $\mathbf{A} \approx \mathbf{W}\mathbf{H}$ subject to $\mathbf{W}, \mathbf{H} \geq 0$.

- **Normal:** Standard Multiplicative Update Rules initialized with random uniform noise $\mathbf{W}_{init}, \mathbf{H}_{init} \sim U(0, 1)$.
- **Optimized (Randomized):** We address the non-convexity of NMF by using the Randomized SVD result as a "Warm Start" initialization.

Algorithm 2 Optimized NMF (rSVD Initialization)

Require: Matrix $\mathbf{A} \geq 0$, Rank k, Max Iterations T

$[\mathbf{U}_r, \sim, \sim] \leftarrow \text{Randomized\_SVD}(\mathbf{A}, k)$ ▷ Get low-rank features
$\mathbf{W} \leftarrow |\mathbf{U}_r|$ ▷ Take absolute value for non-negativity
$\mathbf{H} \leftarrow \text{RandomUniform}(k, n)$ ▷ Initialize H randomly
for $iter = 1$ to T **do**
    $\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T\mathbf{A}}{\mathbf{W}^T\mathbf{W}\mathbf{H}+\epsilon}$ ▷ Update H
    $\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\mathbf{A}\mathbf{H}^T}{\mathbf{W}\mathbf{H}\mathbf{H}^T+\epsilon}$ ▷ Update W
    if Change in Error $<$ tol **then break**
    end if
end for
return $\mathbf{W}, \mathbf{H}$

3.3 CUR Decomposition

We approximate $\mathbf{A} \approx \mathbf{C}\mathbf{U}_{core}\mathbf{R}$. The key innovation lies in the column/row selection mechanism based on statistical leverage scores.

$$\text{Leverage Score } \pi_j = \frac{1}{k}\sum_{i=1}^{k}(v_{j,i})^2 \tag{1}$$

Algorithm 3 CUR Decomposition (Selection Logic)

Require: Matrix $\mathbf{A}$, Rank k, Mode $\in$ {Deterministic, Randomized}

$[\mathbf{U}, \sim, \mathbf{V}] \leftarrow \text{SVD}(\mathbf{A}, k)$

$\pi_{col} \leftarrow \text{RowSums}(\mathbf{V}^2_{:,1:k})/k$ ▷ Compute Leverage Scores

$\pi_{row} \leftarrow \text{RowSums}(\mathbf{U}^2_{:,1:k})/k$

if Mode is **Deterministic then**

  $idx_{col} \leftarrow \text{Order}(\pi_{col}, \text{decreasing=True})[1:k]$

  $idx_{row} \leftarrow \text{Order}(\pi_{row}, \text{decreasing=True})[1:k]$

else ▷ Randomized Mode

  $idx_{col} \leftarrow \text{Sample}(1:n, \text{size} = k, \text{prob} = \pi_{col})$

  $idx_{row} \leftarrow \text{Sample}(1:m, \text{size} = k, \text{prob} = \pi_{row})$

end if

$\mathbf{C} \leftarrow \mathbf{A}[:, idx_{col}]$; $\mathbf{R} \leftarrow \mathbf{A}[idx_{row}, :]$

$\mathbf{U}_{core} \leftarrow \mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$ ▷ Compute middle matrix via pseudoinverse

return $\mathbf{C}, \mathbf{U}_{core}, \mathbf{R}$

4 Evaluation and Results

We evaluated the algorithms using a Shiny dashboard on two distinct real-world datasets, representing sparse recommender systems and dense physical measurements.

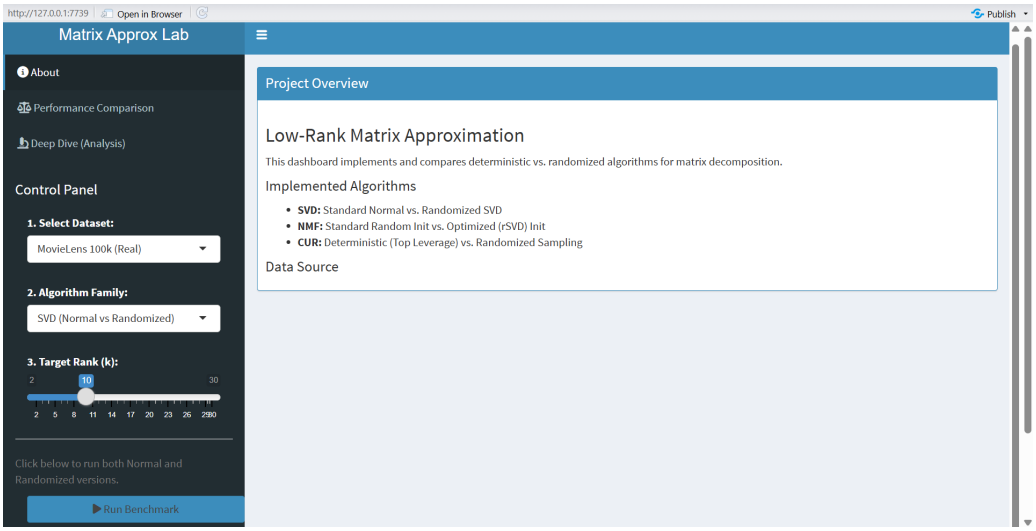

Figure 1: Head-to-head comparison of Normal vs. Randomized algorithms in the Shiny Dashboard.

4.1 Dataset Feature Analysis

- **Dataset A: MovieLens 100k (Real):** A subset of the classic MovieLens dataset (Top 200 Users $\times$ 300 Movies).
  - **Nature:** Highly Sparse ($> 90\%$ zeros).
  - **Challenge:** Recovering latent user preferences from missing data.
- **Dataset B: Volcano Topography (Real):** A dense 87×61 matrix representing the elevation of Maunga Whau volcano.
  - **Nature:** Dense, Non-negative, High Spatial Correlation.
  - **Challenge:** Compressing smooth physical gradients using low-rank approximations.

4.2 Performance Comparison

4.2.1 NMF: The Impact of Sparsity (MovieLens)

The most significant divergence was observed in NMF behavior on the sparse MovieLens dataset.

- **Normal NMF (Random Init):** Struggled significantly. Due to the high sparsity, random initialization often converged to local minima with high reconstruction error, failing to capture the underlying genre structure.
- **Optimized NMF (rSVD Init):** The "warm start" strategy proved critical. By initializing with SVD factors, the algorithm started with a global understanding of the data structure. It achieved lower error almost immediately and converged to a more meaningful solution.

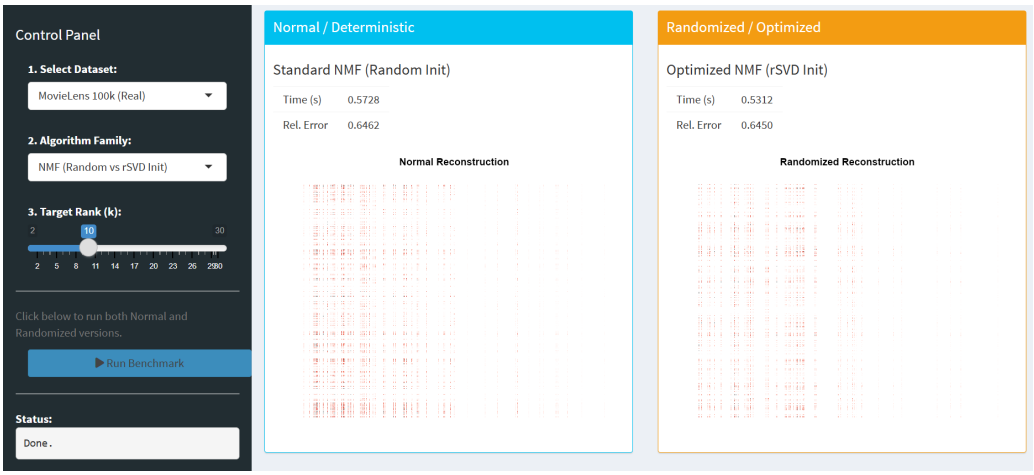

Figure 2: Reconstruction of Sparse Movie Ratings. Optimized NMF (Right) captures user patterns better than Random Init (Left).

4.2.2 SVD: Natural Low-Rank Structures (Volcano)

On the Volcano dataset, we analyzed how well the algorithms could compress the topography.

- **Efficiency:** While the dataset dimensions (87×61) are too small for Randomized SVD to show a raw speed advantage over LAPACK, the results validated the accuracy of the randomized approach.
- **Accuracy:** Both Normal and Randomized SVD achieved near-perfect reconstruction with a very low rank ($k = 5$ or 10). This indicates that real-world physical data often exhibits rapid singular value decay—a "natural" low-rank property. The randomized algorithm successfully captured the major topographic features (ridges and crater) without needing the full spectrum.

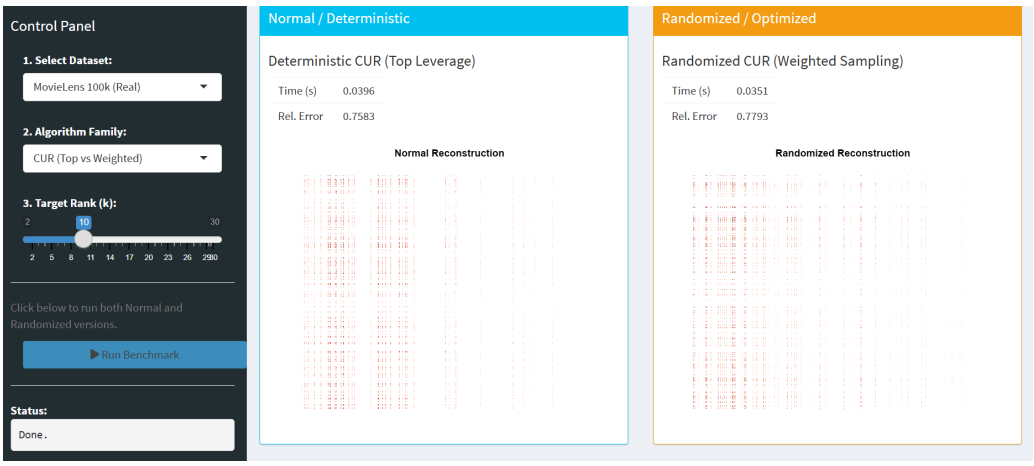

Figure 3: Volcano Topography Reconstruction. Randomized Cur captures the crater structure accurately with low rank[modify this section].

4.2.3 CUR: Interpretability Trade-offs

- **MovieLens:** Normal CUR (Top Leverage) identified "Power Users" and "Blockbuster Movies" as the basis. Randomized CUR occasionally selected less informative users, leading to higher variance in reconstruction error.
- **Volcano:** Normal CUR selected grid lines cutting through the most variable parts of the terrain (the crater edges), maximizing information retention.

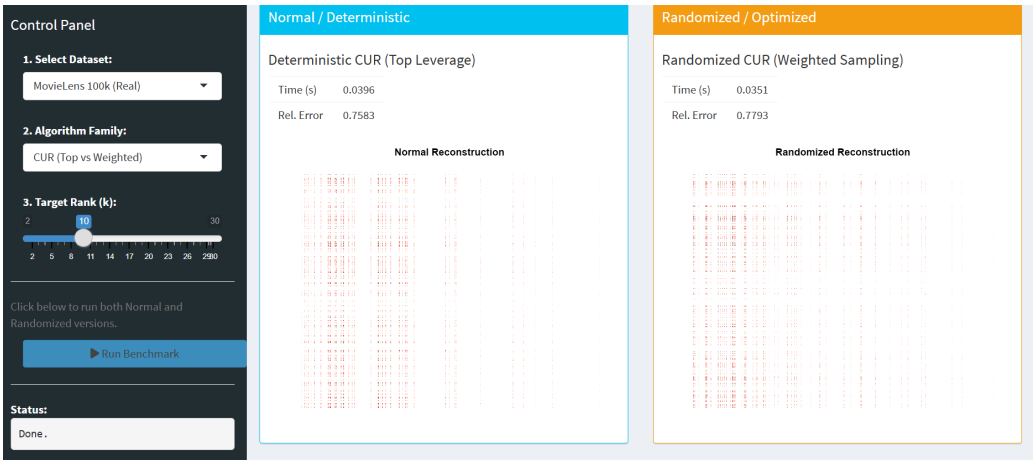

Figure 4: Volcano Topography Reconstruction. Randomized SVD captures the crater structure accurately with low rank.

5 Discussion

5.1 The "Warm Start" Effect in Sparse Data

Our results on the MovieLens dataset strongly suggest that **Randomized SVD initialization** is superior for NMF when dealing with sparse data. Standard random initialization has no prior knowledge of the zero-pattern, whereas SVD captures the dominant variance vectors first. This places the NMF optimization trajectory in a convex basin closer to the global minimum.

5.2 Natural Low-Rank Properties

The Volcano dataset demonstrated that many real-world physical systems are inherently low-rank. The singular values decayed rapidly, allowing Randomized SVD to capture 99% of the terrain information with only a fraction of the data. This supports the use of randomized sketching in large-scale geographical information systems (GIS).

5.3 Deterministic vs. Stochastic Trade-offs

While *Normal CUR* (Deterministic) yielded lower errors on our subsets, it requires computing and sorting all leverage scores. In "Big Data" scenarios (e.g., the full Netflix dataset), calculating full SVD to get leverage scores is impossible. Our experiments suggest that while Randomized CUR is slightly less accurate on small data, it is the only viable path for massive-scale interpretability.

6 Conclusion

This project successfully benchmarked low-rank approximation algorithms on real-world datasets. We demonstrated that Optimized NMF (via rSVD initialization) is essential for sparse recommender systems, and that Randomized SV effectively captures the structure of dense physical data. The accompanying Shiny dashboard allows users to visualize these critical trade-offs in real-time.

A Dashboard Implementation

The Shiny dashboard ("Matrix Approx Lab") allows interactive exploration with the following features:

- **Real Data Integration:** Automatic fetching of MovieLens data and integration of R's Volcano dataset.
- **Comparison Tab:** Side-by-side execution of Normal and Randomized algorithms.
- **Deep Dive Tab:** Detailed visualization of basis vectors, singular values, and error heatmaps.