

# CUR Decomposition and Its Applications

## A Comprehensive Overview

Kevin Smith

May 1, 2024

- **Overview of Matrix Factorizations:**

- Matrix factorizations are pivotal in numerical analysis, data science, and signal processing.
- Key types: LU, QR, SVD—each serves specific applications and offers different insights into matrix structures.

- **Introduction to CUR Decomposition:**

- CUR selectively uses actual columns and rows from the matrix to form low-rank approximations.
- Emphasizes interpretability and efficiency in large, sparse datasets.

- **Mathematical Definition of CUR:**

- For a given matrix  $A$ , CUR decomposition finds matrices  $C$ ,  $U$ , and  $R$  such that  $A \approx CUR$ .
- $C$  and  $R$  consist of selected columns and rows from  $A$ , while  $U$  is a smaller connecting matrix.

- **Importance in Data Science:**

- Provides an interpretable low-rank approximation useful in scenarios like recommender systems and principal component analysis where interpretability is as crucial as dimensionality reduction.

- **Definition and Calculation:**

- Leverage scores quantify the influence of specific rows or columns on the rank- $k$  approximation of  $A$ .
- Computed as the squared Euclidean norm of the rows of  $V$  in the SVD  $A = U\Sigma V^T$ .

- **Role in CUR Decomposition:**

- Guide the selection of columns/rows that best capture the underlying structure of  $A$ .
- High leverage scores correlate with high influence on the matrix's spectral properties.

- **Statistical Interpretation:**

- Leverage scores can be normalized to sum to one, forming a probability distribution over the columns or rows.
- This probabilistic interpretation is crucial for sampling methods in CUR decomposition, emphasizing columns or rows that have a disproportionately large impact on the data structure.

# Derivation of Leverage Scores

- **Definition:** Leverage scores indicate the importance of rows or columns in capturing the data structure.
- **Mathematical Basis:**
  - Consider matrix  $A$  of size  $m \times n$  with Singular Value Decomposition (SVD):  $A = U\Sigma V^T$ .
  - The leverage score of the  $i$ -th row of  $U$  (or  $i$ -th column of  $V^T$ ) is defined as:

$$l_i = \|U[i, :]\|^2 = U[i, :] \cdot U[i, :]^T$$

- This represents the squared norm of the  $i$ -th row of  $U$ , indicating its contribution to the rank- $k$  approximation.
- **Significance:**
  - High leverage scores identify rows/columns that have significant impact on the matrix's spectral properties.
  - Essential for selecting informative rows/columns in CUR decomposition.

# CUR Decomposition Algorithm

## Detailed Algorithm

- 1 Compute an approximate SVD of  $A$  to obtain  $V$ .
  - 2 Calculate leverage scores for all columns.
  - 3 Select columns and rows with the highest scores.
  - 4 Construct  $U$  to minimize  $\|A - CUR\|_F$ , typically using the Moore-Penrose pseudoinverse.
- **Computational Considerations:** Efficiency depends on the method for computing or approximating SVD and the sparsity of the matrix.

# Spectral Properties and CUR Decomposition

- **Impact on Eigenvalues:**

- CUR decomposition approximates the original matrix  $A$  by selecting a subset of its rows and columns.
- This selection can alter the spectral properties (eigenvalues) of  $A$ , especially if the selected columns/rows are not representative of the entire data.

- **Matrix Conditioning:**

- The conditioning of the matrix  $C$  and  $R$  in CUR can significantly affect the stability and accuracy of the decomposition.
- Poorly chosen subsets can lead to a high condition number, which increases the sensitivity to numerical errors.

- **Improving Stability:**

- Using regularization techniques or improved selection algorithms that consider both leverage scores and conditioning can enhance stability.

## Matrix C, U, R and CUR Approximation

Matrix  $C$  (Selected Columns based on Leverage Scores):

$$C = \begin{bmatrix} 2 & 1 & 5 \\ 4 & 5 & 1 \\ 3 & 2 & 6 \\ 5 & 6 & 2 \\ 3 & 5 & 4 \end{bmatrix}$$

Matrix  $U$  (Connection Matrix):

$$U = \begin{bmatrix} 0.5 & 0.35714286 & -0.57142857 \\ -0.16666667 & -0.30952381 & 0.42857143 \\ -0.16666667 & 0.11904762 & 0.14285714 \end{bmatrix}$$

Matrix  $R$  (Selected Rows based on Leverage Scores):

$$R = \begin{bmatrix} 5 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 1 & 6 & 4 \end{bmatrix}$$



# CUR Approximation of Matrix A

$$A \approx CUR = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \\ 2 & 3 & 4 & 5 & 6 \\ 6 & 5 & 4 & 3 & 2 \\ 5 & 3 & 1 & 6 & 4 \end{bmatrix}$$

**Leverage Scores Calculation:** The leverage score of row  $i$  is calculated using the squared Euclidean norm of the corresponding row in matrix  $V$  from the SVD of  $A$ .

$$\text{Leverage Score of Row } i = \|V[i, :]\|^2 = \sum_{j=1}^n V[i, j]^2$$

where  $n$  is the number of columns in  $V$ .

$$LVGScores : [0.2, 0.2, 0.2, 0.2, 0.2]$$

**Note:** Each column was equally likely to be selected due to uniform leverage scores.

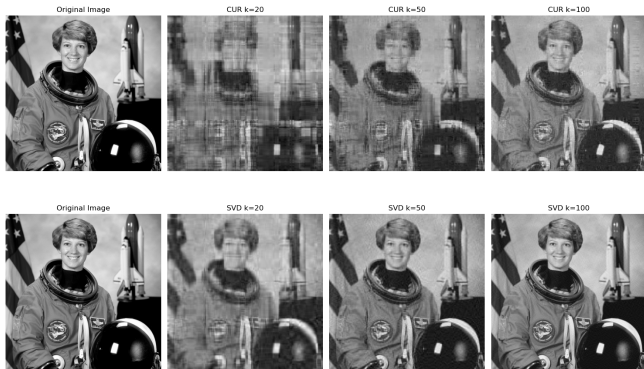
- **Error Analysis:**

- The error of CUR,  $\|A - CUR\|_F$ , depends on the quality of column and row selection.
- Theoretically, CUR aims to approach the optimal rank- $k$  SVD approximation error.

- **Stability and Robustness:**

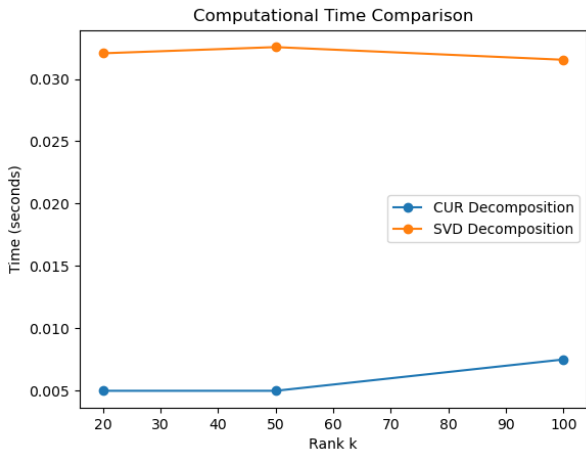
- Stability concerns arise from the conditioning of  $C$  and  $R$ .
- Techniques like regularization or enhanced selection criteria (beyond leverage scores) can mitigate numerical instabilities.

# Image Compression with CUR: Original vs. Compressed



**Figure:** Comparison of the original image with its CUR-compressed version

# Computational Time Comparison for CUR Compression



**Figure:** Graph showing the computational time comparison between CUR and SVD image compression

- **Use Cases:**

- CUR is instrumental in systems where the interpretability of decomposed matrices is crucial, such as detailed data analysis and feature selection processes in bioinformatics and text mining.
- Enhanced feature selection leads to more robust machine learning models by retaining only the most significant predictors.

- **Data Matrix Description:**

- Dimensions: 2000 genes (rows)  $\times$  14 expression level assays (columns).
- Content Variability: Includes genes with different types of transcriptional responses—noise, noisy sine pattern, and noisy exponential pattern.

- **CUR Approach:**

- Selection Based on Leverage Scores: High statistical leverage indicates significant influence on the data's structure.
- Interpretability and Insight: Enhances the ability to identify specific genes contributing to observed patterns, improving biological relevance.

1

---

<sup>1</sup>M. W. Mahoney and P. Drineas, "CUR Matrix Decompositions for Improved Data Analysis," 2009.

# Real-world Example: Genomic Data Analysis

- **Key Discoveries:**

- Focused Analysis on Influential Genes: Highlights biologically significant patterns, useful for experimental validation.
- Direct Link to Biological Processes: Facilitates understanding of gene regulation and responses, crucial for drug discovery.

- **Example Outcomes:**

- Enhanced Data Interpretation: Pinpoints specific gene expression patterns.
- Practical Applications: Vital for areas like drug discovery, where specific gene responses to treatments are studied.

2

---

<sup>2</sup>M. W. Mahoney and P. Drineas, "CUR Matrix Decompositions for Improved Data Analysis," 2009.

- **CUR Decomposition:**

- CUR decomposition involves selecting a subset of columns and rows from the original matrix, which can be computed efficiently using randomized algorithms such as leverage score sampling.
- The computational cost of CUR is often  $O(n \cdot r)$ , where  $n$  is the number of rows, and  $r$  is the desired rank, making it suitable for real-time processing and big data applications.

- **Comparison with Other Techniques:**

- **SVD (Singular Value Decomposition):** SVD typically has a computational complexity of  $O(n^2 \cdot m)$  for an  $n \times m$  matrix, making it computationally expensive, especially for large matrices.
- **PCA (Principal Component Analysis):** PCA's computational complexity is  $O(n^3)$  for computing eigenvalues and eigenvectors of the covariance matrix, which can be high for high-dimensional data.



# CUR vs. PCA: A Comparative Analysis

- **Objective Differences:**

- PCA seeks to maximize variance captured by projecting the data onto orthogonal principal components.
- CUR selects actual rows and columns, aiming to preserve the matrix structure and interpretability.

- **Performance in Sparse Data:**

- PCA can struggle with sparse data where the variance is not a straightforward indicator of data structure importance.
- CUR excels in sparse settings due to its direct selection of influential matrix elements.

- **Use Cases:**

- PCA is preferred in dense datasets and scenarios where dimensionality reduction is crucial.
- CUR is advantageous in applications requiring direct interpretation of the original features, such as text analysis and genomic data.

- **Unique Advantages:**

- **Interpretability:** CUR allows users to retain actual rows and columns from the original dataset, enhancing the interpretability of results, crucial in fields such as genomics and social sciences.
- **Efficiency:** Particularly effective for large sparse matrices, CUR can be more efficient than traditional SVD, reducing computational load and memory usage.

- **Potential Limitations:**

- **Dependency on Initial Selection:** The performance of CUR heavily depends on the choice of columns and rows, which can vary significantly with different selection methods.
- **Approximation Quality:** While CUR provides a useful approximation, it might not always achieve the same level of accuracy as the best rank-k approximation provided by SVD, particularly in tightly coupled datasets.

- **Improving Accuracy and Efficiency:**
  - **Advanced Algorithms:** Research into more sophisticated algorithms for selecting columns and rows could enhance both the accuracy and efficiency of CUR decompositions.
  - **Hybrid Approaches:** Combining CUR with other matrix factorizations or machine learning models to improve performance and stability.
- **Emerging Applications:**
  - **Deep Learning:** Exploring the use of CUR in optimizing neural network training by efficiently approximating weight matrices.
  - **Network Analysis:** Applying CUR to the study of network flow and connectivity patterns in large-scale networks, potentially improving the understanding of complex systems like internet traffic or social networks.
  - **Real-time Data Processing:** Utilizing CUR in real-time data systems, such as streaming data analysis and online learning environments, where quick and efficient data processing is crucial.