SVD Compression of the Red Sea Region Climate Data

Ahmad Sait
ahmed.sait@kaust.edu.sa
KAUST

Abstract

The efficient management and compression of large-scale scientific datasets is a critical challenge in data-intensive fields. This project focuses on compressing climate data from the Red Sea region, leveraging a Singular Value Decomposition (SVD)-based error-bounded lossy compression framework. The dataset, stored in TileDB, was represented as a three-dimensional matrix (time × spatial dimensions). The methodology involved decomposing the matrix into its most significant components, storing them as separate TileDB arrays, and incorporating correction information to ensure error bounds were maintained under varying thresholds (1e-1, 1e-2, 1e-3, 1e-4). This compression approach adheres to error-bounded criteria as outlined in the literature, including methods discussed in "A Survey on Error-Bounded Lossy Compression for Scientific Datasets." The "row-vector x time" approach with data de-averaging and quantization achieved a high compression ratio of 10.36, demonstrating the potential of SVD-based techniques for handling large scientific datasets with guaranteed accuracy.

ACM Reference Format:
Ahmad Sait. 2018. SVD Compression of the Red Sea Region Climate Data. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The rapid growth of scientific data has created challenges in storing, processing, and sharing large datasets, especially in fields like climate science. Climate datasets often consist of multi-dimensional arrays containing detailed information, such as temperature or pressure, across wide regions and time periods. Finding efficient ways to manage and compress these datasets is important to save storage space while keeping the data accurate enough for scientific research.

One of the main difficulties is balancing data compression with accuracy. Lossy compression, which reduces the size of data by simplifying it, can be effective if it follows error-bounded rules. These rules ensure that the difference between the original and compressed data stays within acceptable limits, so the compressed data remains useful. Singular Value Decomposition (SVD) is one method that can simplify large datasets by keeping only the most important parts of the data.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

This project looks at compressing climate data from the Red Sea region using an SVD-based error-bounded compression method. The data is stored in TileDB and is organized as a three-dimensional array (time × spatial dimensions). By breaking the data into smaller parts and applying corrections, this project aims to achieve high compression while keeping the data accurate.

2 Methodology

2.1 Dataset

The dataset used in this project represents climate data from the Red Sea region, specifically focusing on temperature values. It is structured as a three-dimensional matrix with dimensions representing time (T) and spatial coordinates (X, Y). Each cell in the matrix stores a temperature measurement in Kelvin, allowing detailed analysis of temporal and spatial variations.

The dataset includes 4,000 timesteps and spatial dimensions of 855 x 1,215, resulting in a total of over 4 billion data points. The total size of the dataset is approximately 16.5 GB, with each value stored as a 32-bit floating-point number (4 billion data points x 4 bytes). This high resolution ensures that the dataset captures fine-grained details of temperature patterns over a large geographical area, including the Arabian Peninsula, the Red Sea, and neighboring regions.

Given its size and complexity, the dataset presents significant challenges for storage and processing. High-dimensional data like this requires substantial storage resources, and transferring or sharing such datasets can be inefficient without compression. Additionally, scientific datasets demand a high level of accuracy, which means any compression method must ensure minimal distortion to preserve the usability of the data.

This project utilizes the dataset to demonstrate the effectiveness of a Singular Value Decomposition (SVD)-based compression approach. The dataset is stored in TileDB, a versatile database engine designed for multi-dimensional arrays. TileDB's structure allows efficient storage and retrieval of data, making it suitable for managing the large-scale dataset and facilitating the implementation of the compression algorithm.

2.2 Data Processing

The climate data matrix was processed in a sequential and systematic manner to prepare it for compression. The data matrix, originally structured as a three-dimensional array ($T \times X \times Y$) of size $855 \times 1215 \times 4000$, was accessed and processed one sub-matrix at a time. Each sub-matrix corresponded to a single row in the spatial dimension ($1 \times 1215 \times 4000$).

Processing started with the first row matrix at the bottom of the data matrix and progressed upwards to the top row. For each row matrix, the process of de-averaging was applied to normalize the data and remove large-scale trends. De-averaging involved

subtracting the average of the entire data matrix ($855 \times 1215 \times 4000$) from the corresponding row matrix ($1 \times 1215 \times 4000$).

2.3 Compression

The compression process relied on Singular Value Decomposition (SVD) to reduce the dimensionality of the dataset while preserving its key features. SVD was applied to each row matrix of the climate data sequentially. Each row matrix, corresponding to a single slice of the data ($1 \times 1215 \times 4000$), was decomposed into three matrices: U, S, and V.

The U matrix contains the left singular vectors, representing the main features of the data, while S is a diagonal matrix holding the singular values, which quantify the importance of these features. The V matrix contains the right singular vectors, capturing the relationships between variables. These three components together provide an efficient representation of the original data.

This process was repeated for all 855 rows of the dataset, ensuring that SVD was performed comprehensively across the entire three-dimensional data matrix ($855 \times 1215 \times 4000$). As each row was processed, the resulting U, S, and V matrices were stored in separate TileDB arrays. TileDB provided an efficient and structured way to manage these arrays, ensuring scalability and ease of access for subsequent steps in the workflow.

2.4 Reconstruction

Reconstructing the lossy compressed data matrix (D) involved reversing the compression process by recombining the decomposed components (S, U, and V) obtained during Singular Value Decomposition (SVD). Each row matrix of the dataset was reconstructed by multiplying these three matrices together, restoring an approximation of the original data. The reconstruction process for a given row matrix followed the equation:

$$D = USV^T \tag{1}$$

where U and V are the orthogonal matrices from SVD, and S is the diagonal matrix containing the singular values. This multiplication was performed sequentially for each of the 855 rows in the dataset, ensuring the entire matrix was reconstructed accurately within the bounds of the error constraints.

To restore the original data distribution, the average value of the entire dataset, which had been subtracted during the de-averaging step in data processing, was added back to the reconstructed matrix. This de-centering step ensured that the final output (D) closely approximated the original dataset while maintaining the global data trends.

2.5 Correction Matrix

To further refine the approximation of the original dataset and meet the specified error-bounded constraints, a correction matrix (E) was computed. The correction matrix accounted for discrepancies between the original data matrix (D) and the reconstructed data matrix (D) generated from the SVD components (S, U, and V). This step ensured that all data points adhered to the error threshold (ϵ).

The correction matrix was computed element-by-element by evaluating the absolute error between the original and reconstructed datasets. For any element where the error exceeded the threshold (ϵ), the computed error value was recorded in the correction matrix. Conversely, if the error was below or equal to (ϵ), the corresponding element in the correction matrix was set to zero. This selective population of the correction matrix resulted in a highly sparse structure, as only a small fraction of the elements required correction.

2.5.1 Quantization. To optimize storage further, the non-zero elements of the correction matrix were quantized. This process reduced the bit size of each element from 32-bit floating-point (float32) representation to 16-bit unsigned integer (uint16). Quantization not only decreased the storage size of the correction matrix by half but also ensured efficient handling without significantly affecting accuracy.

The sparse nature of the correction matrix made it ideal for storage in a TileDB Sparse array rather than a Dense array. TileDB's sparse array format efficiently managed the non-zero elements, avoiding the storage overhead associated with the majority of zero-valued elements.

2.6 Compression Ratio

The compression ratio (*rho*) is a key metric used to evaluate the effectiveness of the compression method. It quantifies the reduction in storage achieved by the compression process while maintaining the data within the specified error bounds. The compression ratio was calculated by comparing the size of the original data matrix (D) to the combined size of the compressed components.

The sizes of the SVD component matrices and the sparse correction matrix components were included in the formula to compute the compression ratio:

$$\rho = \frac{\text{sizeof}(D)}{\text{sizeof}(U) + \text{sizeof}(S) + \text{sizeof}(V) + \text{sizeof}(E)} \tag{2}$$

Here, sizeof represents the total storage size of each matrix, including any optimizations such as quantization for the correction matrix. By summing the sizes of all compressed components and dividing the original matrix size by this sum, the compression ratio provided a clear measure of how much the dataset was compressed.

This step ensured that the data was centered around zero, reducing biases caused by global averages. De-averaging also helped enhance the representation of local variations within the dataset, which is crucial for effectively applying the compression algorithm. By processing the data row-by-row, memory usage was minimized, making the approach suitable for handling such a large dataset efficiently. The processed data was then stored in TileDB arrays, maintaining the integrity of the original structure while preparing it for the subsequent steps in the compression pipeline.

3 Experiments

To evaluate the effectiveness of the proposed compression framework, three main experiments were conducted, each targeting a specific aspect of the methodology. These experiments aimed to test different preprocessing and optimization strategies and their impact on the compression ratio. Subexperiments within each main experiment varied the rank of the SVD components to identify the

optimal rank for achieving the highest compression ratio while maintaining error bounds.

3.1 Main Experiment 1: Standard SVD Compression

The first experiment applied SVD directly to each row matrix of the dataset without any prior data de-averaging (normalization) and without quantizing the correction matrix. The goal was to establish a baseline performance for the compression framework. This setup represented a straightforward application of SVD, capturing the raw effectiveness of the method without additional optimizations. Subexperiments under this main experiment adjusted the rank of the SVD components to determine how the rank influenced the compression ratio and the quality of the reconstructed data.

3.2 Main Experiment 2: SVD with Data De-Averaging

The second experiment incorporated data de-averaging as part of the preprocessing step before applying SVD. In this case, the average value of the original dataset was subtracted from each row matrix, centering the data distribution. This normalization step was intended to reduce bias and enhance the efficiency of the SVD representation. Subexperiments in this category again involved varying the rank of the SVD components to find the optimal balance between compression and reconstruction accuracy.

3.3 Main Experiment 3: SVD with Data De-Averaging and Correction Matrix Quantization

The third and final experiment combined both data de-averaging and the quantization of the correction matrix. Quantization reduced the size of non-zero correction values from 32-bit floating-point (float32) to 16-bit unsigned integers (uint16), further decreasing the storage requirements. This experiment represented the most optimized version of the compression framework. Subexperiments within this category explored different ranks for the SVD components to assess how the combination of de-averaging and quantization influenced the overall compression ratio.

3.4 Subexperiments: Varying SVD Ranks

Within each main experiment, subexperiments were conducted by tweaking the rank of the SVD components. The rank determines the number of singular values and corresponding vectors retained during decomposition. Higher ranks preserve more data fidelity but increase storage size, while lower ranks improve compression at the cost of accuracy. By systematically varying the rank, the subexperiments sought to identify the configuration that yielded the highest compression ratio while ensuring the reconstructed data remained within the specified error bounds.

4 Results

The experimental results demonstrated the impact of different preprocessing and optimization strategies on the compression ratio. All these experiments were on the threshold value of 1e-2. Among the three main experiments conducted, the approach that combined data normalization (de-averaging) and quantization of the correction matrix produced the best results. This method achieved a compression ratio of 10.36, which was the highest among all configurations tested. The results highlight the combined effectiveness of data normalization and correction matrix quantization in enhancing compression efficiency. Data normalization reduced global biases, allowing the SVD components to capture the data structure more effectively. Quantization further optimized storage by reducing the size of the sparse correction matrix.

Table 1: Results of the Experiments

Method	Rank	Compression Ratio
SVD	10	2.45
SVD + DeAvg	10	3.79
SVD + DeAvg	30	6.37
SVD + DeAvg	50	8.30
SVD + DeAvg	70	8.92
SVD + DeAvg	80	8.80
SVD + DeAvg	120	7.24
SVD + DeAvg + Quant. (uint16)	30	9.00
SVD + DeAvg + Quant. (uint16)	50	**10.36**
SVD + DeAvg + Quant. (uint16)	70	10.08

5 Weakness with Current Approach

While the proposed compression framework proved effective for this specific dataset, it has potential weaknesses that may limit its applicability to other datasets or configurations. A primary concern arises from the row-matrix-based approach used to calculate the SVD. This method assumes that data points along a row matrix exhibit patterns or repetitions across the time dimension. For the climate dataset in question, this assumption holds reasonably well, as temperature patterns tend to show temporal consistency. However, this assumption may not generalize to other datasets or scenarios where data behavior is fundamentally different.

In cases where the repetitions or correlations exist along other dimensions, such as across spatial coordinates or combinations of time and space, the row-matrix-based approach could lead to suboptimal compression. The method's effectiveness is inherently tied to the structure and patterns of the dataset. If the data does not align with the assumptions made during SVD computation, the resulting compression ratio and reconstruction accuracy may be adversely affected.

6 Future Work

The current approach does not explore higher-dimensional decompositions that could capture patterns across multiple dimensions simultaneously. Such methods, while computationally more intensive, might yield better compression for datasets with complex interdependencies across dimensions. An example of such method is the Higher-Order Singular Value Decomposition (HOSVD). Unlike standard SVD, which works on two-dimensional matrices, HOSVD operates on multi-dimensional arrays (or tensors) and captures patterns across all dimensions simultaneously. Since it leverages

long-range correlation in the dataset across different dimensions (such as time dimension and different fields), it can achieve an extremely high ratio. However, it is very expensive because of its intrinsic iterative steps in error control [1].

By utilizing HOSVD, error-bounded lossy compression could be generalized to a wider variety of datasets where the structure and interdependencies differ. Unlike the current approach, which compresses the data by processing individual row matrices ($1 \times 1215 \times 4000$), HOSVD would operate directly on the entire 3D dataset ($855 \times 1215 \times 4000$). This multi-dimensional decomposition could efficiently capture relationships along all axes, improving compression performance while maintaining accuracy.

7 Conclusion

This project successfully implemented an SVD-based error-bounded lossy compression framework to address the challenge of managing large-scale scientific datasets. By applying Singular Value Decomposition to climate data from the Red Sea region, the framework achieved a compression ratio of 10.36 while maintaining data accuracy within specified error bounds. The methodology combined techniques such as data de-averaging and quantization of the correction matrix to enhance compression efficiency and reduce storage requirements.

The results demonstrated the effectiveness of the proposed approach in optimizing storage for high-dimensional datasets while preserving critical scientific information. However, the row-matrix-based method assumes specific patterns in the data that may not generalize to other datasets with different structures or interdependencies. This limitation suggests the need for more adaptable methods, such as Higher-Order Singular Value Decomposition (HOSVD), which can capture correlations across all dimensions of multi-dimensional data.

Future work should focus on extending the framework to incorporate higher-dimensional decomposition techniques like HOSVD, which could generalize the compression strategy to diverse datasets and improve performance for cases with complex interdependencies. Additionally, exploring methods to reduce the computational overhead of HOSVD will be crucial for making it practical for large-scale applications.