Singular Value Decomposition and Dynamic Mode Decomposition: Background, Origin, and Applications

Marcin Sobotka
with valuable Mentorship from Professor Anthony Bloch
and Joshua Pickard

July 9, 2024

Abstract

Matrix decompositions, such as Singular Value Decomposition (SVD) and Dynamic Mode Decomposition (DMD), are fundamental tools in computational mathematics and data analysis. This paper provides an overview of their mathematical structure, geometry, and applications in fields such as principal component analysis, image compression, noise reduction, and fluid dynamics.

1 Introduction

Matrix decompositions are essential in computational mathematics, providing solutions to problems such as linear least squares and eigenvalue computations. This paper explores two key decompositions: Singular Value Decomposition (SVD) and Dynamic Mode Decomposition (DMD).

2 Importance of Matrix Decompositions

In computational mathematics, we often have complex matrices to work with. A classical example from linear algebra is the QR factorization of a matrix

A. QR expresses a matrix A as the product of an orthogonal matrix Q and an upper triangular matrix R.

This factorization is often helpful in solving linear least squares problems and eigenvalue computations.

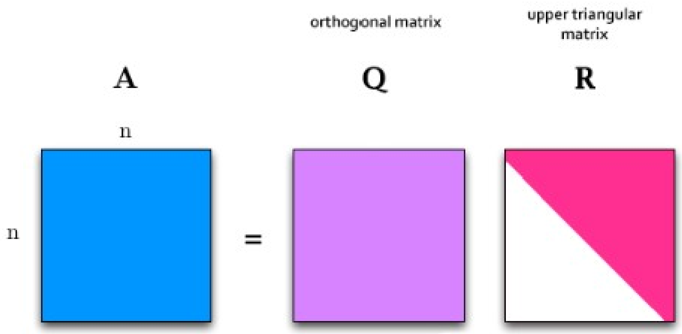


Figure 1: Matrix structure in QR factorization.

3 Singular Value Decomposition (SVD)

3.1 Definition and Components

Definition: For any matrix $A \in \mathbb{R}^{m \times n}$, there exist orthogonal matrices U and V, as well as diagonal matrix $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{\min(m,n)})$ such that:

$$A = U \Sigma V^T$$

Component matrices:

- $U \in \mathbb{R}^{m \times m}$ consists of the left singular vectors.
- $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix containing non-negative values (the singular values).
- $V \in \mathbb{R}^{n \times n}$ consists of the right singular vectors.

3.2 Geometric Interpretation

The SVD can be used to understand the transformation of a unit sphere in $\mathbb{R}^n$ into an ellipsoid. Here, the lengths of the semi-axes of the ellipsoid represent the singular values of matrix A.

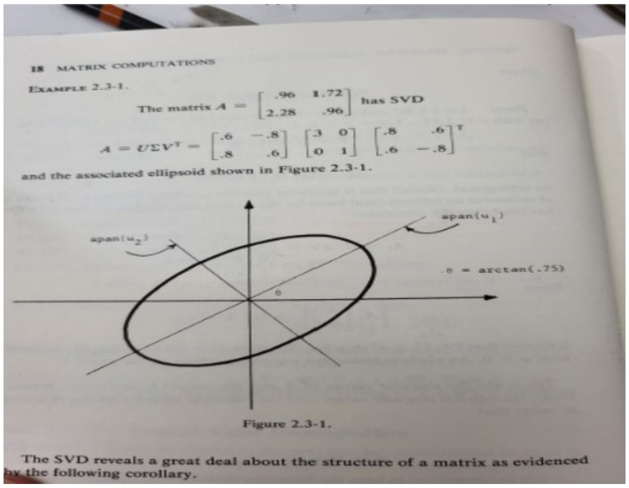
With SVD, we act on vectors some unit sphere in $\mathbb{R}^n$ (i.e., all vectors x have norm=1).

Transformation Process:

- First, we apply V for a rotation.
- Next, we scale along the principal axes by the singular values in Σ.
- Finally, we apply U for another rotation.

Hence, unit sphere transformed by A becomes an ellipsoid by the following:

$$y = U\Sigma V^T x = Ax$$



18 MATRIX COMPUTATIONS

EXAMPLE 2.3-1.

The matrix $A = \begin{bmatrix} .96 & 1.72 \\ 2.28 & .96 \end{bmatrix}$ has SVD

$$A = U\Sigma V^T = \begin{bmatrix} .6 & -.8 \\ .8 & .6 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} .8 & .6 \\ .6 & -.8 \end{bmatrix}^T$$

and the associated ellipsoid shown in Figure 2.3-1.

Figure 2.3-1.

The SVD reveals a great deal about the structure of a matrix as evidenced by the following corollary.

Figure 2: Geometric interpretation of SVD transforming a unit sphere into an ellipsoid.

3.3 Applications of SVD

- **Principal Component Analysis (PCA)**: Reducing dimensionality by projecting data onto principal components.

- **Image Compression**: Retaining significant singular values approximates the original image with reduced storage.
- **Noise Reduction**: Smaller singular values associated with noise can be filtered out.

4 Analysis of a Randomly Generated 10×10 Matrix

In this section, we analyze a randomly generated 10×10 matrix X.

4.1 Matrix X and Singular Value Decomposition

The matrix X is provided below:

$$X = \begin{bmatrix} 0.15818 & 0.81836 & 0.24962 & 0.06544 & 0.90386 & 0.11633 & 0.91511 & 0.08482 & 0.32417 & 0.31303 \\ 0.06215 & 0.93817 & 0.38706 & 0.61035 & 0.45223 & 0.64622 & 0.90097 & 0.29813 & 0.66904 & 0.07644 \\ 0.70184 & 0.00033 & 0.42104 & 0.70155 & 0.07069 & 0.10841 & 0.21424 & 0.91713 & 0.29629 & 0.79142 \\ 0.08648 & 0.64039 & 0.64008 & 0.11162 & 0.24128 & 0.98350 & 0.54706 & 0.47052 & 0.92995 & 0.36538 \\ 0.61679 & 0.00736 & 0.78755 & 0.09582 & 0.73187 & 0.24834 & 0.78471 & 0.26947 & 0.28196 & 0.58510 \\ 0.17377 & 0.10642 & 0.26999 & 0.59783 & 0.04049 & 0.60636 & 0.19444 & 0.76297 & 0.16888 & 0.18334 \\ 0.65140 & 0.10679 & 0.84398 & 0.81223 & 0.42452 & 0.81670 & 0.74689 & 0.77217 & 0.74517 & 0.07691 \\ 0.49870 & 0.36711 & 0.74047 & 0.81458 & 0.54022 & 0.83005 & 0.47556 & 0.02130 & 0.47713 & 0.15366 \\ 0.28451 & 0.23961 & 0.82610 & 0.08944 & 0.95383 & 0.48904 & 0.58326 & 0.87999 & 0.65344 & 0.82688 \\ 0.83056 & 0.34614 & 0.18219 & 0.73128 & 0.20891 & 0.76073 & 0.26055 & 0.79817 & 0.96658 & 0.30096 \end{bmatrix}$$

4.2 Singular Values of X

The singular value decomposition (SVD) of X yields the following singular values:

$$\{4.9312, 1.7089, 1.4394, 0.9715, 0.8457, 0.6195, 0.4715, 0.3952, 0.2257, 0.0506\}.$$

These values indicate the importance of each singular vector in representing the structure of X.

4.3 Thresholding for Retained Singular Values

To analyze some matrix A we will learn from X through DMD, we use several thresholds for retaining singular values. For each threshold, we calculate the learned matrix$A$ and observe its corresponding structure. For example:

Threshold = 0.1: Only the largest singular value (4.9312) is retained. Threshold = 0.5: Two singular values (4.9312, 1.7089) are retained. Threshold = 0.9: Six singular values are retained.

The learned A matrices reflect an approximation of matrix X based on the retained singular values.

4.4 Example Learned A Matrix for Threshold 0.5

For threshold 0.5 above, the retained singular values are:

$$\{4.9312, 1.7089\}.$$

The learned A matrix is:

$$A = \begin{bmatrix} 0.9623 & 0.2173 \\ 0.0224 & -0.5593 \end{bmatrix}.$$

This new matrix A learned through DMD reduces computational complexity while retaining essential structural properties of X.

This example demonstrates how randomly generated matrices can be analyzed using SVD, with thresholding enabling the extraction of lower-dimensional approximations make computations efficient all the while maintaining the dynamics of the original matrix A

5 Dynamic Mode Decomposition (DMD)

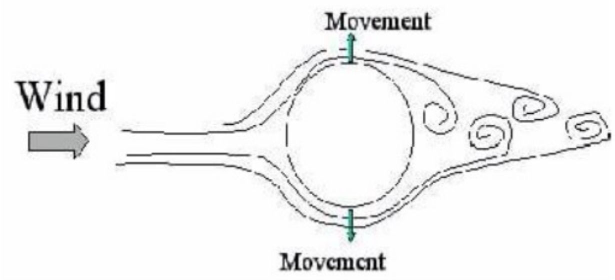
Definition: For any matrix $X \in \mathbb{R}^{m \times n}$ that represents n snapshots of a system's state at different points in time, DMD decomposes the system into modes that capture its dynamics. There exists a matrix $A \in \mathbb{R}^{m \times m}$ which linearly approximates the evolution of these n snapshots.

DMD can be used to describe a fluid state (such as a velocity or vorticity field) as a superposition of many computed basis vectors, known as *modes*.

5.1 Background on DMD

The number of modes required to capture the main behavior of a flow is typically much smaller than the system state dimension. For instance, a system with a state dimension on the order of 10^6 might only need around 10 modes to represent the main dynamics.

A common example of DMD application is in observing airflow within a cylinder, assuming laminar flow as a reference. The flow of air creates vortices or swirling air patterns. Over time, these patterns develop into regularly observed structures that shed and form off the cylinder in a process known as *vortex shedding*. Vortex shedding is valuable for understanding the strength and behavior of fluid forces.



Vortex shedding visualized through DMD, illustrating air flow patterns around an object (e.g. cylinder).

6 DMD Process and Experiment Design

6.1 Defining the System

Consider a system represented by matrix A of rank 5 with randomly chosen initial state vector $\mathbf{x}_0$. $\mathbf{x}_0$ is selected at random from a normal distribution, as so will ensure a close approximation to the happenings in nature. We generate snapshots $\mathbf{x}(t)$ for $t = 0$ to $t = 200$ by iterating:

$$\mathbf{x}(t+1) = A\mathbf{x}(t)$$

This general formula captures the system dynamics when using DMD.

6.2 DMD Process

1. **Collect the snapshots**: Collect the n snapshots of the system state and form data matrices X, X':
$$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1}], \quad X' = [\mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n]$$
2. **Perform Singular Value Decomposition (SVD)**: Apply SVD on X:
$$X = U\Sigma V^T$$

3. **Choose the number of singular values**: Select the number r of singular values based on an energy threshold.

4. **Reduce system state**: Reduce the system state by finding a rank for some $\tilde{A}$ such that $\text{rank}(\tilde{A})$ is much less than the dimension of the system. For example, there can be 20,000 genes we observe, which leads to a matrix X with 4 million entries. The learned matrix from DMD can be $20,000 \times 20,000$, but it might only need around 8 time points, resulting in an 8×8 matrix instead.

   $$\tilde{A} = U^T A U = U^T X' V \Sigma^{-1}$$

   Here, U and V remain orthogonal and maintain unitary matrix properties.

5. **Eigenvalue decomposition**: Compute the eigenvalues and eigenvectors of matrix A as:

   $$AW = W\Lambda$$

   where Λ contains the eigenvalues, and the columns of W are the eigenvectors.

6. **Compute DMD Modes**: The DMD modes Φ are given by:

   $$\Phi = X' V \Sigma^{-1} W$$

7 Noise Sensitivity Analysis of DMD

We consider a simple 5×5 matrix A with entries defined as:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We generate a data matrix X from A under the influence of small noise, $\eta = 0.01$, and apply Dynamic Mode Decomposition (DMD) to approximate A.

7.1 Key Observations

1. Retained Rank: For a threshold = 0.9, only the largest singular value is retained ($r = 1$).
2. Learned Matrix A_{DMD}: The matrix A_{DMD}, derived from DMD, deviates significantly from A due to noise:

$$A_{\text{DMD}} = \begin{bmatrix} 1.0220 & 0.0230 & 0.0004 & 0.0000 & 0.0000 \\ 0.0230 & 0.0005 & 0.0000 & 0.0000 & 0.0000 \\ 0.0004 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}.$$

3. Frobenius Norm Error: The Frobenius norm difference between A and A_{DMD} is what tells us our norm error between original A and A learned through DMD, calculated as:

$$\|A - A_{\text{DMD}}\|_F = 2.8204.$$

This shows that DMD is sensitive to noise: The learned matrix A_{DMD} becomes rank deficient, reflecting the dominant singular mode while losing finer details of A. Even small levels of noise ($\eta = 0.01$) can significantly perturb the reconstruction of A, with a large Frobenius norm error (> 2.8). So, noisy systems of low rank have an effect on the robustness of DMD-based approximations.

Currently Studying and Learning

8 Extending DMD to Large Datasets of Gene Expression Data

In this section, we extend the application of Dynamic Mode Decomposition (DMD) to now analyze larger-scale datasets, such as gene expression profiles.

The observed data matrix X can reach dimensions of approximately:

$$X \in \mathbb{R}^{20{,}000 \times 20{,}000}$$

representing gene expression across thousands of genes over multiple time points.

Challenge:

Directly handling a $20,000 \times 20,000$ matrix poses computational difficulties and inefficiencies.

Solution:

DMD reduces this large matrix to a much smaller representation of a reduced model state while marvelously capturing the essential dynamics of the original system. The resulting reduced matrix $\tilde{A}$ has significantly lower dimensions, in this case:

$$\tilde{A} \in \mathbb{R}^{8\times 8}$$

This reduction represents the dominant modes of the system, efficiently preserving the key information while discarding noise and less representative dynamics.

Interpretation:

Even though X is initially high-dimensional, the system's behavior is governed by the resulting much smaller set of 8 dominant modes learned through DMD. This reflects the intrinsic low-rank nature of the data, allowing DMD to construct an accurate reduced-order model.

Mathematical Representation:

$$\tilde{A} = U^T A U = U^T X' V \Sigma^{-1}$$

where:

- U, V: Left and right singular vectors from SVD.
- Σ: Diagonal matrix of singular values.
- X': Time-shifted data matrix.

Gene Dataset Reformatting

The raw experimental data were reformatted and represent data matrix X for DMD analysis:

$$X = \begin{pmatrix}
2.0648 & 1.7041 & 2.6391 & \dots & 2.0884 & 2.5323 & 1.3546 \\
2.9231 & 11.446 & 2.4907 & \dots & 2.2925 & 6.6267 & 12.231 \\
3.2734 & 3.1920 & 3.7709 & \dots & 2.9807 & 3.1222 & 2.8154 \\
1.1567 & 1.1925 & 1.3668 & \dots & 1.2409 & 0.9862 & 0.9392 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
1.0616 & 0.9433 & 0.7816 & \dots & 0.9319 & 0.8561 & 0.8635 \\
0.7637 & 1.1972 & 0.7680 & \dots & 0.7813 & 0.7522 & 0.9880 \\
0.3608 & 0.4576 & 0.4929 & \dots & 0.4372 & 0.3857 & 0.3950 \\
9.1759 & 8.4802 & 6.8843 & \dots & 7.0422 & 7.6655 & 8.6989 \\
0.9331 & 0.8282 & 0.7968 & \dots & 0.8761 & 0.9297 & 0.7818 \\
0.5267 & 0.6562 & 0.6184 & \dots & 0.5887 & 0.6090 & 0.6556
\end{pmatrix}$$

Figure 3: reformatted data into matrix X

- **Input Format**: Gene names, time points (or hours), and TPM (transcripts per million) values.
- **Reformatted Matrix**: Rows represent genes, columns represent time points, and entries are TPM values.
- **Replicates**: For consistency, our X matrix is created using only one replicate at a time considering the raw data has the same experiment done 2 or 3 times.

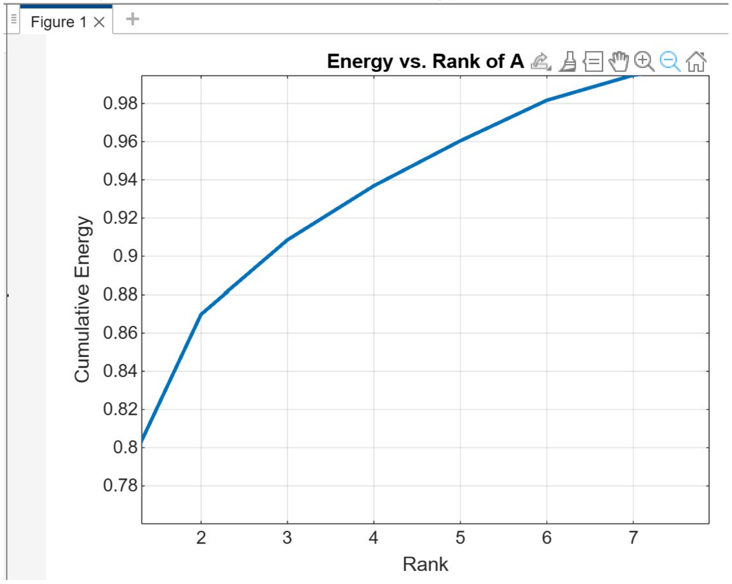
Thresholding and Dimensionality Reduction

Thresholds were applied to retain singular values resulting from SVD that preserve cumulative energy from the system while ignoring noise. The plot below shows the cumulative energy versus rank of A learned through DMD, namely $\tilde{A}$:

Plot Description

- **X-axis**: Rank of $\tilde{A}$.
- **Y-axis**: Cumulative energy of singular values.

Figure 4: Cumulative Energy vs. Rank of $\tilde{A}$



- **Interpretation**: Thresholds like 0.8, 0.9, and 0.95 determine how much energy and which singular values are preserved in the reduced dynamics. Effectively, we see how many singular values are needed to encapsulate the majority of the energy of the system, thus we also see them as some of our principal components.

The following table summarizes the ranks of $\tilde{A}$ obtained under various threshold values:

Threshold (Thresh)	**Rank of A**
0.1	1
0.2 to 0.7	1
0.8	2
0.9	3
0.95	5
0.97	5
0.99	6
0.999	7

Table 1: Thresholds and corresponding ranks of A.

Interpretation

- Thresholds such as 0.2 to 0.7 retain only the dominant singular value, resulting in a rank 1 degenrate matrix.
- Higher thresholds like 0.8 and above retain more singular values, capturing finer details of the dynamics.
- As the threshold approaches 0.999, more energy is preserved, increasing the rank to 7.

As in the example of lower rank where A is known initially, we must be keen to distinguish noise, be it experimental, through measurement, or noise in nature. In most cases, we begin without knowing some matrix A a data matrix X is constructed through formatting our dataset.

9 Summary and Conclusion

Dynamic Mode Decomposition (DMD) applies large-scale data reduction to data matrices, such as those of gene expression. It does so by retaining the dominant modes through Singular Value Decomposition (SVD). while also identifying the principal components - an effective interpretation of a separation of our data from noise. Furthermore, $\tilde{A}$ is learned through DMD, and we benefit greatly from the dynamics represented by the reduced model state of our large original data set. It is an effective and representative low-rank approximation as a whole.

10 Acknowledgements

This REU is largely inspired and made possible by Professor Anthony Bloch. Many meetings and conversations with him served to be very insightful, and I learned a lot. I am very grateful for his support and for his encouragement.

I am also thankful for the help and support of Joshua Pickard. I gained a lot of insight and tips that made me realize and gain many perspectives along the way.

11 References

- Risto Hinno QR decomposition graphic: https://ristohinno.medium.com/qr-decomposition-903e8c61eaab
- Vortex shedding graphic: https://www.mecaenterprises.com/stack-vibration-dampers/vortex-shedding/
- Golub, G. H., and C. F. Van Loan, Matrix Computations, 4th ed., Johns Hopkins University Press, Baltimore, 2013.
- Rowley, C. W., Mezic, I., Bagheri, S., Schlatter, P., Henningson, D. S. (2009). Spectral analysis of nonlinear flows. Journal of Fluid Mechanics, 641, 115-127.
- Pickard, J., Stansbury, C., Surana, A., Muir, L., Bloch, A., Rajapakse, I. (2024). Biomarker Selection for Adaptive Systems. arXiv preprint arXiv:2405.09809