Data-Driven Reduced Order Modelling

Dynamical Mode Decomposition

Veerendra Harshal Budhi

CATERPILLAR

July 4, 2021

Aim

- To model the time dynamics of an evolving system using only data generated.

Aim

- To model the time dynamics of an evolving system using only data generated.
- To reduce the order at which the computation is done.

Aim

- To model the time dynamics of an evolving system using only data generated.
- To reduce the order at which the computation is done.
- To find the Natural Mode Shapes of the system (The Eigenvalue Problem).

Inputs

The n-vector constituting of all possible measurements that would cover the complexity of the system under observation. We will refer to this as $\vec{x_n}$. Depending upon the system, n can be large to very large. The crux of reduced order modelling lies in being able to condense the above n-vector space to a more compact vector space $\vec{x_r}$ without incurring significant information loss.

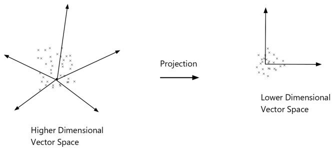


Figure 1: Projecting between Vector Spaces

Outputs

Most time dynamic evolution can for a limited amount of time be linearised. Having such an Evolution matrix leads to a linear transformation being able to convey the complexity of a time step evolution under the hood.

$$\vec{x_n}(t + \Delta t) = A_n \vec{x_n}(t) \qquad (1)$$

Now that we have the Linear model, we can thus work with the Eigenvalue problem that gives us our natural modes and frequencies. Hence, the output will be Eigen-Decomposition of A_n.

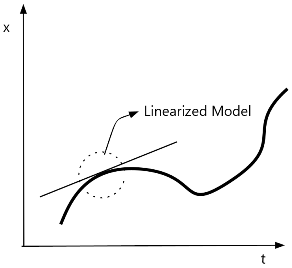


Figure 2: Linearization

As the eigenvalue problem is solved, we will obtain the frequencies and modes of the problem. Studying the modes will also give us insights into the system. Moreover, as we have lost useless info, we shall have a more clearer info on the reduced system than the whole one.

$$\vec{x_n}^{(k)} = \Phi \Lambda^k \vec{b} \tag{2}$$

This can also help with predicting measurements.

Problem with n

The number of dimensions "n" is the dimensionality of the vector space that we are currently working in. Oftentimes, the amount of dimensions is more than the actual pertinent ones (referred to commonly as the rank of the system). Also, as the experiment becomes more sophisticated, the stream of measurements also becomes huge and A_n size grows as the square of n thereby slowing down things. Hence, the need to work in a smaller more compact dimensional subspace.

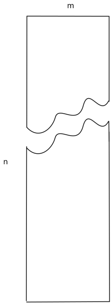


Figure 3: Dataset

Existence of Linear Matrix

- There will, in general, not be a matrix that can linearly do the job. Hence, we settle for approximations which might lead to a plethora of possible solutions near the true value.
- As with most of machine learning strategies, we choose to tackle the problem of uniqueness or lack thereof using optimisation strategies.
- The Least Squares Approach is widely used for it's simplicity and the advantage of being a maximum likelihood estimate.

1 Introduction

2 Problem Statement

3 Difficulties

4 Theory

Principal Components Based Dimensionality Reduction

5 Procedure

6 References

SVD

In linear algebra, the Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices. It has some interesting algebraic properties and conveys important geometrical and theoretical insights about linear transformations.

$$A = U\Sigma V^* \tag{3}$$

The diagonal matrix Σ contains the Weights of the Vectors given in U. Taking a compact svd shall arrange the vectors in descending order. Vectors in U which have insignificant weights in Σ will not be of much use. Only those vectors really of weight span a subspace containing the n-vectors.

Dimensionality Reduction

As we obtain the highest performing vectors by Singular Value Decomposition, we can thus get rid of the low performers and truncate to r modes that contain a good portion of the total information. The r vector space now will be made of the linear combination of the highest performing r n-vectors in U. By definition, U is unitary thereby leading to orthogonality in the vectors.

$$U = [u_1, u_2, u_3, ..., u_r] \tag{4}$$

$$\vec{x_n} = c_1 u_1 + c_2 u_2 + ... + c_r u_r \tag{5}$$

$$\vec{x_n} = U\vec{x_r} \tag{6}$$

Flowchart

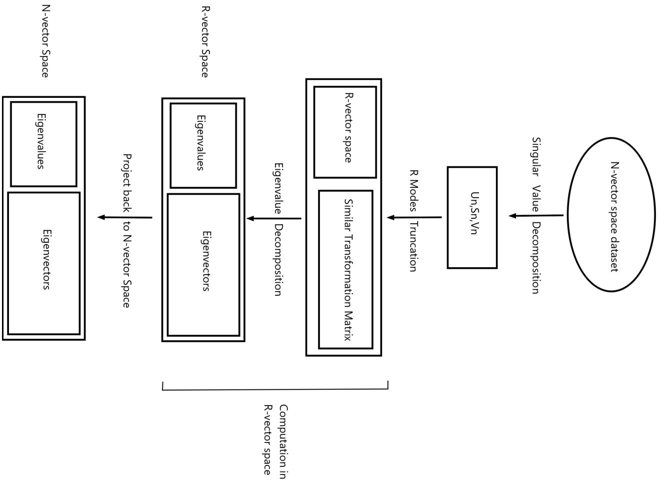
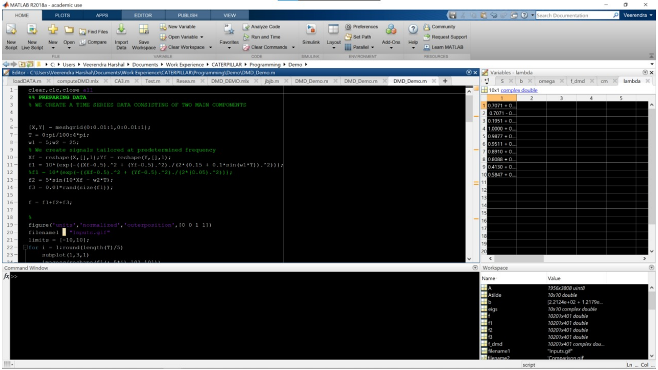


Figure 4: Flowchart Devising the Procedure

Algorithm

```
% Data Preparation : N-vector Space
X1 = f(:,1:end-1);X2 = f(:,2:end);

% Singular Value Decomposition
[U,S,V] = svd(X1,'econ');

% Truncation to r modes
r = 5;
U = U(:,1:r);S = S(1:r,1:r);V = V(:,1:r);

% Finding Reduced Transformation Matrix
Ar = transpose(U)*X2*V*inv(S);

% EigenValue Decomposition - R vector Space
[W,eigs] = eig(Ar);

% Re-project into N-vector Space
Phi = X2*V*inv(S)*W;
```

Demonstration



Demonstration

References

1. Brunton, S. and Kutz, J., 2020. Data-driven science and engineering. Cambridge: Cambridge University Press.
2. Kutz, N., Brunton, S., Brunton, B. and Proctor, J., n.d. Dynamic mode decomposition.
3. Schmid, P., 2010. Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics, 656, pp.5-28.
4. Uzga-Rebrovs, O. and Kulesova, G., 2020. Initial Dataset Dimension Reduction Using Principal Component Analysis. Information Technology and Management Science, 23, pp.41-44.
5. Kutz, N., Brunton, S., Brunton, B. and Proctor, J., 2021. DMD Theory | Dynamic Mode Decomposition. [online] Dmdbook.com. Available at: <http://dmdbook.com/> [Accessed 4 July 2021].

Thanks!