

WILFRID LAURIER UNIVERSITY

MA 680 - SEMINAR IN MATHEMATICAL
MODELLING

DEPARTMENT OF MATHEMATICS

Dynamic Mode Decomposition

Authors

Austin Sammon

Timothy Ponopal

Wayne Shen

Instructor

Dr. Cristina Stoica

Tuesday December 20, 2022



Abstract

Data-driven modelling and control of complex systems represent a quickly advancing field with significant potential that can be applied to a wide array of industrial and academic areas. Dynamic mode decomposition (DMD) is an equation-free and data-driven technique that takes advantage of dimensional reduction and provides a spatiotemporal decomposition of data into a set of dynamic modes. This paper provides a discussion of the theoretical foundation behind DMD including singular value decomposition as well as its connection to the Koopman operator. Additionally, examples of dynamic mode decomposition's application to epidemiology, trading markets, and weather systems are provided, accompanied by an examination of the useful information that can be derived from the output.

Contents

1	Introduction	3
1.1	Background	3
1.2	Compact Singular Value Decomposition	5
2	Mathematical Foundation	9
2.1	Koopman Spectral Theory	15
2.1.1	Application in DMD	16
2.2	APPLICATION TO PDEs	24
3	Interpretation	26
3.1	Interpretation of Eigenvectors	26
3.2	Interpretation of Eigenvalues	27
4	Application	30
4.1	Modeling infectious disease spread	30
4.1.1	DMD Output	32
4.2	Developing Financial Trading Strategies	36

4.2.1	DMD Output	38
4.3	Modelling Weather Systems	42
4.4	Fluid Dynamics	56
5	Conclusion	62
6	Algorithm	65
7	Bibliography	70

1. Introduction

In modern society, data can be collected from anywhere at any time. Scientists and mathematicians have been finding ways to analyze and discover a connecting model between each data point. However, with the limitation of computational machines increasing gradually as time proceeded, we are able to take advantage of their computational ability to calculate high-dimensional datasets at different time intervals. Dynamic mode decomposition is an equation-free and complete data-driven mechanism that utilizes the essence of dimensional reduction from a high dimensional measurement into the size of the number of measurements taken.

1.1 Background

The idea of dynamic mode decomposition was initially introduced by Peter J. Schmid, professor of mechanical engineering at King Abdullah University of Science and Technology, in 2008 in the fluid dynamics society to monitor and extract dynamic features in fluid data. Clarence W.



Figure 1.1: Peter J. Schmid

Rowley later connected dynamic mode decomposition to the non-linear dynamical system, known as Koopman's Theory, in 2009, and DMD has been modified into analyzing non-linear systems.

Due to its simplicity in application, dynamic mode decomposition has been applied to a wide range of categories apart from fluid dynamics, such as disease modelling, plasma, neuroscience, financial trading, etc. In short, DMD can be applied and analyze the system when there exists a collection of data, and predict the future time frame interval.

$$X = \begin{bmatrix} | & | & & | \\ x_0 & x_1 & \dots & x_{m-1} \\ | & | & & | \end{bmatrix}$$

$$X' = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_m \\ | & | & & | \end{bmatrix}$$

1.2 Compact Singular Value Decomposition

Dynamic mode decomposition is based on singular value decomposition. Singular value decomposition (SVD) is one of the most important decomposition techniques that splits a system into a set of linearly independent components. This is essential for the computation of dimensional reduction that can be utilized in the data-driven mechanism of DMD. In the view of linear algebra, SVD can separate any matrices into a set of low-ranked linearly independent sub-matrices. That is, for $r = \min(m, n)$, any $X_{n \times m}$ matrices can be separated into

$$X_{n \times m} = U_{n \times r} \Sigma_{r \times r} V_{r \times m}^T \quad (1.1)$$

Where

- $\Sigma_{r \times r}$ is the matrix of the eigenvalues of $\begin{cases} X^T X & \text{if } n \leq m \\ XX^T & \text{if } m \leq n \end{cases}$
- $V_{m \times r}$ is the normalized eigenvectors of $\begin{cases} X^T X & \text{if } n \leq m \\ XX^T & \text{if } m \leq n \end{cases}$

- $U_{n \times r}$ is the multiplication of the inverse of eigenvectors, X , and the normalized eigenvectors

Example:

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Let $Y = X^T X$,

$$Y = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

The eigenvalues, λ , and eigenvectors, v , of Y are

- $\lambda_1 = 1$ with $v_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ has the unit vector $\hat{v}_1 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$
- $\lambda_2 = 3$ with $v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ has the unit vector $\hat{v}_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Normalize the eigenvectors, we have the matrix

$$V = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

The square root of non-zero eigenvalues

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \end{bmatrix}$$

$$u_1 = \frac{1}{1} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$$

$$u_2 = \frac{1}{\sqrt{3}} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \end{bmatrix}$$

Now combine the u_i 's into U matrix,

$$U = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \end{bmatrix}$$

Now X can be represented by

$$X = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = U\Sigma V^T$$

2. Mathematical Foundation

Given a dynamical system of the form

$$\frac{dx}{dt} = f(x, t)$$

We seek to find a solution to the associated system, where B is a matrix and x is a vector.

$$\frac{dx}{dt} = Bx$$

We will relate the above to the Dynamical Mode Decomposition. Given a photo which is a matrix of pixels in RGB format, we first have to turn the matrix into a vector. We do this with the following map.

$$\begin{bmatrix} x_{11} & x_{21} & \dots & \dots & x_{1n} \\ x_{12} & x_{22} & \dots & \dots & x_{2n} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & \dots & x_{mn} \end{bmatrix}_{m \times n} \longrightarrow (x_{11}, \dots, x_{1n}, x_{21}, \dots, x_{mn})^T$$

After relabelling the indices we get that $(x_{11}, \dots, x_{1n}, x_{21}, \dots, x_{mn})^T$ becomes $\vec{x}_1 = (x_1, \dots, x_k)^T$ where $k = mn$.

Next, we will arrange the vectors of snapshots of photos into two matrices of dimension nm .

$$X = \begin{bmatrix} | & | & & | \\ \vec{x}_0 & \vec{x}_1 & \dots & \vec{x}_{j-1} \\ | & | & & | \end{bmatrix}_{k \times j}, j \leq k \quad (2.1)$$

$$X' = \begin{bmatrix} | & | & & | \\ \vec{x}_1 & \vec{x}_2 & \dots & \vec{x}_j \\ | & | & & | \end{bmatrix}_{k \times j}, j \leq k \quad (2.2)$$

Where $\vec{x}_0, \dots, \vec{x}_j$ are the vectors of pixels of the photo frames. We seek to find a matrix A that solves the following equation,

$$X' = AX \quad (2.3)$$

The matrix A satisfies the following map between the vectors of pixels of the photo frames.

$$\begin{array}{c}
 X = \begin{bmatrix} | & | & & | \\ x_0^T & x_1^T & \dots & x_{j-1}^T \\ | & | & & | \end{bmatrix}_{k \times j} \\
 \\
 X' = \begin{bmatrix} | & | & & | \\ x_1^T & x_2^T & \dots & x_j^T \\ | & | & & | \end{bmatrix}_{k \times j}
 \end{array}$$

Figure 2.1: X to X'

Where

$$\vec{x}_0 \longrightarrow \vec{x}_1, \dots, \vec{x}_{j-1} \longrightarrow \vec{x}_j \quad (2.4)$$

After performing the algorithm for the computation we would like to minimize the error $\| X' - AX \|_F$ where F is the Frobenius norm is given by

$$\| X \|_F = \sqrt{\sum_{j=1}^n \sum_{i=1}^m X_{ji}^2} \quad (2.5)$$

We can solve for A via a Moore-Penrose pseudo-inverse. We can rearrange $X' = AX$ to solve for A using the Moore-Penrose Pseudo Inverse, to give $A = X'X^+$ where X^+ is the pseudo-inverse of X . X is a $k \times j$ ma-

trix and X' is $k \times j$ matrix. Hence the X^\dagger is a $j \times k$ yielding A to be a $k \times k$ square matrix. Examples of the Moore-Penrose pseudo-inverse will be given below.

$$X^\dagger = \begin{cases} X^T(XX^T)^{-1} & \text{if } X \text{ is an onto linear map of independent rows} \\ (XX^T)^{-1}X^T & \text{if } X \text{ is a 1-1 linear map of independent columns} \end{cases} \quad (2.6)$$

For X and X' , we perform the Compact Singular Value Decomposition upon X . The compact form of SVD is given by there exist matrices U, Σ, V of type $k \times r, r \times r, r \times k$ respectively, such that $X = U\Sigma V^T$ and $UU^T = VV^T = I$, where $r = \min(k, j)$. That is,

$$\begin{aligned} X &= U_{k \times k} \Sigma_{k \times j} V_{k \times j}^T \text{ for regular SVD} \\ X &= U_{k \times r} \Sigma_{r \times r} V_{r \times k} \text{ for compact numerical SVD} \end{aligned} \quad (2.7)$$

The eigenvalues of V are from AA^T , the eigenvalues of U are from AA^T . Solving For A Using the SVD of X yields

$$A = X'_{k \times j} (U_{k \times r} \Sigma_{r \times r} V_{r \times j}^T)^{-1} = X'_{k \times j} V_{j \times r} \Sigma_{r \times r}^{-1} U_{r \times k}^T \quad (2.8)$$

One way of doing SVD is by using the Polar Decomposition of a square matrix. Given a square $k \times k$ matrix A , A can be written as $A = R_1 S$, where R_1 is an orthogonal matrix and S is a symmetric matrix. A symmetric

matrix can be diagonalized hence

$$\begin{aligned}
A &= R_1 S \\
&= R_1 (R_2 D R_2^T) \\
&= (R_1 R_2) D R_2^T
\end{aligned} \tag{2.9}$$

Where

$$\begin{cases} U = (R_1 R_2) \\ \Sigma = D \\ V = R_2 \end{cases} \tag{2.10}$$

Hence we get an SVD of A from a polar Decomposition. It is more efficient to project our Matrix A onto the $r \times r$ subspace of the $k \times k$ subspace so the calculations are less taxing. We do this by rearranging our matrix $A = X' V \Sigma^{-1} U^T$ and then orthogonally projecting onto the $r \times r$ subspace.

$$\begin{aligned}
A &= X' V \Sigma^{-1} U^T \\
\implies AU &= X' V \Sigma^{-1} \\
\implies U^T AU &= U^T X' V \Sigma^{-1}
\end{aligned} \tag{2.11}$$

The $U^T AU$ is the orthogonal projection of A onto the $r \times r$ subspace. label $U^T AU = \tilde{A}$. We check the Dimensions of $U^T AU$. The matrix $U^T AU$ is given by matrices U^T, A, U of type $r \times k, k \times k, k \times r$ respectively, hence \tilde{A} is of type $r \times r$. The matrix \tilde{A} defines a lower rank linear model of the form

$\tilde{A}\tilde{x}_l = \tilde{x}_{l+1}$. Where the vectors \tilde{x}_l are of type $r \times 1$. We can reconstruct the the original data by using the U matrix given by $x_l = U\tilde{x}_{l+1}$.

We now compute the eigenvalues and eigenvectors of \tilde{A} , which will give an eigenvector basis. Let W be the eigenvector matrix associated to the eigenvectors of \tilde{A} . We can then reconstruct the eigendecomposition of A from W . Where we are given that $\tilde{A}W = W\Lambda$ where Λ is the matrix of eigenvalues. Hence

$$\begin{aligned}
U^T AU &= W\Lambda W^T \\
\implies A &= UW\Lambda W^T U^T \\
\implies A &= (UW)\Lambda(UW)^T \\
\implies AU &= X'V\Sigma^{-1} = UW\Lambda W^T \\
\implies AUW &= X'V\Sigma^{-1}W := \Phi \\
\implies A_{k \times k}U_{k \times r}W_{r \times r} &:= \Phi_{k \times r}
\end{aligned} \tag{2.12}$$

Notice that $A_{k \times k}U_{k \times r}W_{r \times r} := \Phi_{k \times r}$ is the alternate form of the SVD. Also note that if $\Phi_1 = UW$ then Φ_1 will converge to Φ if X and X' have the same column space. $\Phi_1 = UW$ are known as the projected DMD modes. The modes in $X'V\Sigma^{-1}W := \Phi$ are termed exact modes.

2.1 Koopman Spectral Theory

Notice that $A_{k \times k} U_{k \times r} W_{r \times r} := \Phi_{k \times r}$ and that if $\Phi_1 = UW$ then Φ_1 will converge to Φ if X and X' have the same column space. $\Phi_1 = UW$ are note as the projected DMD modes. The modes in $X' V \Sigma^{-1} W := \Phi$ are termed exact modes.

Given a system $\frac{dx}{dt} = f(x, t)$ and the associated system $\frac{dx}{dt} = Bx$, One solves this system by using the exponential matrix, that is $x(t) = c_1 v_1 e^{\lambda_1 t} + \dots + c_n v_n e^{\lambda_n t}$ where for $i = 1..n$, c_i are constants, v_i are eigenvectors and λ_i are the eigenvalues. Similarly for our system, $x(t) = \sum_{k=1}^r \phi_k \exp(\omega_k t) b_k$ which in compact matrix notation becomes $x(t) = \Phi \exp(\Omega t) \mathbf{b}_k$, where $\Omega = \text{diag}(\omega)$ is a diagonal matrix with eigenvalues $w_k = \ln(\lambda_k) / \Delta t$ where we use the $\ln(\lambda_k)$ to smooth out large eigenvalues. The eigenvectors of Φ are given by ϕ_k and b_k can be solved by evaluating $x(0) = x_1$ and $\mathbf{b} = \Phi^\dagger x_1$, where Φ^\dagger is the Moore Penrose Pseudo Inverse. $x(t) = \sum_{k=1}^r \phi_k \exp(\omega_k t) b_k$ is interpreted as a least square fit in the sense of the two norms given by $\|x_{k+1} - Bx_k\|_2$ and iterating $\frac{dx}{dt} = Bx$ gives $x_{k+1} = Bx_k$.

If a low dimension representation exists then as iterations are done the singular values in the matrix Σ will appear and the $**$ below will converge to zero where $\sum_{k=1}^r \sigma_k^2 = \text{tr}(A^T A)$.

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \sigma_r \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \quad (2.13)$$

Hence this is why it is considered a dimension-reducing algorithm.

2.1.1 Application in DMD

For $g : \mathbb{C}^n \rightarrow \mathbb{C}$, g is a function that takes our snapshot frame and spits out a number, we think of this function as a measurement. We will use this in conjunction with a flow map. Define a flow map by

$$F_t(x_0) := x(t + t_0) = x(t_0) + \int_{t_0}^{t+t_0} f(x, \tau) d\tau \quad (2.14)$$

This comes from integrating the equation $\frac{dx}{dt} = f(x, t)$. We can now iterate the flow map to get

$$x_{k+1} = F_t(x_k) = x(t_0) + \int_{t_0}^{t+t_0} f(x_k, \tau) d\tau \quad (2.15)$$

We now introduce the Koopman operator K . The Koopman operator

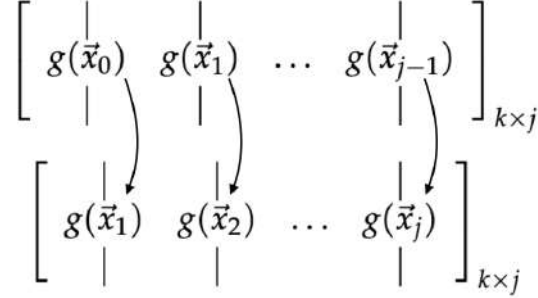


Figure 2.2: Koopman Advanced Measurements

acts on functions by a composition map defined by

$$Kg = g \circ f \quad (2.16)$$

We apply this to our flow map to get

$$\begin{aligned} Kg(x_k) &= g \circ F_t(x_k) \\ &= g(x_{k+1}) \end{aligned} \quad (2.17)$$

Hence we see that the Koopman operator advances measurements from $g(x_k)$ to $g(x_{k+1})$. We have an analogous setup to that Of the DMD below since the Koopman operator advances measurements

Koopman looked at Hamiltonian flows that were measure preserving, hence unitary in nature.

The Koopman operator lives in the space known as the Hilbert space. The Hilbert space is equipped with a norm called the L_2 norm, given by

$$\|f\|_{L_2} = \int |f|^2 < \infty.$$

One can look at the eigenvalues of the Koopman operator analogous to looking at the eigenvalues of matrices. With regards to Matrices and eigenvalues $Ax = lx$ for an eigenvalue l and an eigenvector x . Let ϕ be an eigenfunction of the Koopman operator then $K\phi = \lambda\phi$, where λ is the associated eigenvalue. We give an example of this.

$$\text{let } g(x) = (\log(x))^c$$

$$\text{let } f(x) = x^p$$

$$\begin{aligned} Kg(x) &= (g \circ f)(x) \\ &= (\log(x^p))^c \\ &= (p \log(x))^c \\ &= p^c (\log(x))^c \\ \lambda &= p^c \end{aligned} \tag{2.18}$$

Instead of g being a single function that the Koopman operator acts on, we can consider a vector of measurement functions given by $(g_1(\vec{x}_i), \dots, g_k(\vec{x}_i))^T$.

We can write

$$\vec{g}(x) = \sum_{k=1}^{\infty} \phi_k v_k \tag{2.19}$$

Where we use the L_2 norm to find the v_k given by

$$v_k = \begin{bmatrix} \langle \phi_k, g_1 \rangle_{L_2} \\ \langle \phi_k, g_2 \rangle_{L_2} \\ \vdots \\ \vdots \\ \vdots \\ \langle \phi_k, g_k \rangle_{L_2} \end{bmatrix} \quad (2.20)$$

Instead of having a finite vector basis as we would have in the regular DMD, we now have an infinite dimensional space spanned by the eigenvectors $\{\phi_k\}$. The Koopman operator can act on the vector-valued function $\vec{g}(x)$ in a linear way.

$$\begin{aligned} K\vec{g}(x) &= K \sum_{k=1}^{\infty} \phi_k v_k \\ &= \sum_{k=1}^{\infty} K \phi_k v_k \\ &= \sum_{k=1}^{\infty} \lambda_k \phi_k v_k \end{aligned} \quad (2.21)$$

Where we have expanded $\vec{g}(x)$ using the eigenfunction basis. If g is chosen well it can simplify calculations, we give a heuristic view of this via figure (2.3).

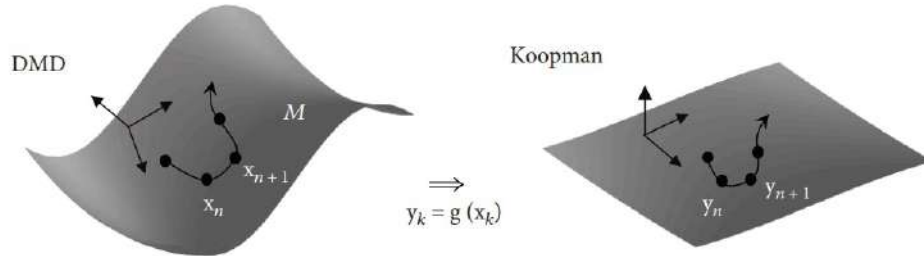


Figure 2.3: Visual representation of the Koopman operator restricted to a finite-dimensional invariant subspace spanned by Eigen functions

If g is chosen well it can take a flow on a nonlinear manifold to a more linearized version in which calculations can become more manageable. Given the above construction, we can now perform DMD in Hilbert space analogous to DMD in a vector space with the euclidean norm.

$$\begin{aligned}
Y &= \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \vec{g}([x_0]^T) & \vec{g}([x_1]^T) & \dots & \dots & \vec{g}([x_{j-1}]^T) \\ | & | & \dots & \dots & | \\ | & | & \dots & \dots & | \end{bmatrix} \\
Y' &= \begin{bmatrix} | & | & \dots & \dots & | \\ \downarrow & \downarrow & \dots & \dots & \downarrow \\ \vec{g}([x_1]^T) & \vec{g}([x_2]^T) & \dots & \dots & \vec{g}([x_j]^T) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}
\end{aligned}$$

We now perform SVD on Y and follow the algorithm previously defined in the DMD.

If $\phi_k \in \text{span}(g_1(\vec{x}_i), \dots, g_p(\vec{x}_i))^T$ then $\phi_k = a_1 g_1 + \dots + a_p g_p$ and p is not necessarily equal to k , the a_i are constants in the field

$$\begin{aligned}
K\phi_k &= \lambda_k \phi_k \\
&= \lambda_k (a_1 g_1 + \dots + a_p g_p) \\
&= \lambda_k a_1 g_1 + \dots + \lambda_k a_p g_p
\end{aligned} \tag{2.22}$$

Hence K becomes a linear operator. For if $[a_1, \dots, a_p]^T \in \text{Range}(Y)$ then $a = [a_1, \dots, a_p]^T$ is a left eigenvector of $A_Y = Y'Y^\dagger$ and $\vec{a}^T A_Y = \lambda_k \vec{a}^T$

We now apply the DMD to get

$$\begin{aligned}\Phi_Y &:= Y' V \Sigma^{-1} W \\ y(t) &= \Phi_Y \exp(\Omega t) \mathbf{b} \\ \mathbf{b} &= \Phi_Y^\dagger y_1\end{aligned}\tag{2.23}$$

We now transform back from measured observables to the DMD state space, the vector space is given by

$$y_k(x) = \vec{g}(x_k) \implies x_k = \vec{g}^{-1}(y_k)\tag{2.24}$$

The inversion of \vec{g} can be difficult if not chosen well.

The projected low dimensional solution in Hilbert space of the Koopman DMD is analogous to the DMD on a finite-dimensional vector space from the relation $\phi_k \in \text{span} (g_1(\vec{x}_i), \dots, g_p(\vec{x}_i))$. It is best to make sure we pick out the dominant(leading) eigenfunctions based on the (leading)eigenvalue. The relation between DMD and the Koopman DMD is analogous to a Fourier transform which takes a system from state space to phase space. Here the state space is the finite-dimensional vector space in DMD and the Phase space is in the Hilbert Space. One can modify the Koopman operator and apply to that of Partial Differential Equation (PDE). The Koopman DMD and DMD approximation produce a least squares fit the data of the dynamical system which approximates the flow map of the system. When the Koopman operator is chosen well it can potentially linearize the space,

hence making the system potentially easier to solve. Dynamics can change when parameters change in the PDE, the dynamics can undergo bifurcation. When the Bifurcation happens the Koopman operators will have to be altered in conjunction with the bifurcation. We will see in the upcoming data section that DMD will be applied, DMD has a hard time seeing subtle in-variances which exist in the system, such as helix style rotations or what is considered a U(1) gauge and say laminar flow. Invariances undermine the ability to compute lower dimensional systems since one has to mod out the specific invariance. To know these invariances one needs to know the dynamics of the system. This is analogous to if one has a group action on a manifold, the manifold structure is recovered by modding out the group action. The DMD algorithm can be regarded as an application of a flow on the manifold of fixed rank matrices which is then projected onto a lower dimensional manifold. The vector field of the flow of the original system is projected onto the tangent space of the lower dimensional manifold. Here the geometry of fixed rank matrices can be studied.

The predictive power of DMD is not expansive and needs to be improved. One way of improving the predictive power is to add an existing known control into the system, by which the control is governed. The system would be in the form of $X' = AX + CY$ where C is a matrix, possibly of functions, which steers the system. Y is a set of new snapshots similar to X though is considered a control snapshot. One would then get $(X' - CY)V\Sigma^{-1}U^T = A$.

2.2 APPLICATION TO PDEs

We look at Burger's equation

$$u_t + u_x u - \varepsilon u_{xx} = 0, \varepsilon > 0 \quad (2.25)$$

The existence of the diffusion term is known to regularize the PDE. We transform the PDE using

$$u = -2\varepsilon v_x / v \quad (2.26)$$

the PDE transforms as

$$v_t = -\varepsilon v_{xx} \quad (2.27)$$

We then use the Fourier Transform on the above PDE. the Fourier Transform is given by

$$\widehat{v}(k, t) = \int_{-\infty}^{\infty} v(x, t) e^{-ikx} dx \quad (2.28)$$

We use the property $F(\frac{d}{dx}(f(x))) = ik\widehat{f}(k)$ and $F(\frac{d^n}{(dx)^n}(f(x))) = (ik)^n \widehat{f}(k)$ and apply this to $v_t = -\varepsilon v_{xx}$

$$\begin{aligned} \widehat{v}_t &= -\varepsilon k^2 \widehat{v} \\ \widehat{v} &= \widehat{v}_0 \exp(-\varepsilon k^2 t) \end{aligned} \quad (2.29)$$

Using $u = -2\varepsilon v_x / v$ and $\widehat{v} = \widehat{v}_0 \exp(-\varepsilon k^2 t)$ one can construct a Koopman Operator via composition of the Fourier transform with $u = -2\varepsilon v_x / v$

Hence one gets

$$v(x, t) = \exp \left(-\frac{\int_{-\infty}^x u(\alpha, t) d\alpha}{2\epsilon} \right) \quad (2.30)$$

We verify this works by checking $u = -2\epsilon v_x / v$

$$v_x(x, t) = \frac{d}{dx} \left(-\frac{\int_{-\infty}^x u(\alpha, t) d\alpha}{2\epsilon} \right) \exp \left(-\frac{\int_{-\infty}^x u(\alpha, t) d\alpha}{2\epsilon} \right) \quad (2.31)$$

Using the fundamental theorem of calculus on $\frac{\int_{-\infty}^x u(\alpha, t) d\alpha}{2\epsilon}$ we get

$$\begin{aligned} v_x(x, t) &= \frac{u(x, t)}{-2\epsilon} v(x, t) \\ u(x, t) &= (-2\epsilon) \frac{v_x(x, t)}{v(x, t)} \end{aligned} \quad (2.32)$$

And this verifies our function works. We define our Koopman Operator g by

$$g(u) = \hat{v} \quad (2.33)$$

3. Interpretation

DMD decomposes complex flows into their fundamental spectral components which correspond to spatial and temporal properties that characterize the behaviour of the flow. Thus in order to extract useful information from the DMD analysis an understanding of the computed eigenvalues and eigenvectors of the best fit DMD operator A is essential. Detailed interpretation of the eigenvalues and eigenvectors of an equation (3.1) will be provided in this section.

$$x(t) = \Phi \exp(\Omega t) \mathbf{b} \quad (3.1)$$

3.1 Interpretation of Eigenvectors

The eigenvectors Φ of the dynamic mode decomposition also referred to as the DMD modes represent the spatial properties of the system or data field being analyzed. Each entry of a vector-valued mode corresponds to a spatial location of the data field and can be used to identify both local and

global features such as mixing, symmetry, and long-time behaviour. Each eigenvector is scaled by an amplitude \mathbf{b} which is also determined during the analysis by DMD and is used to determine each mode's influence. The more dominant the mode the more it is represented in the decomposition of the data. This helps to allow for the selection of the relevant modes to be made.

3.2 Interpretation of Eigenvalues

The eigenvalues λ of a mode allow us to determine the temporal development or dynamical behaviour of said mode. These eigenvalues can be plotted on a complex-valued unitary circle which represents stable or stationary flow structures, their position on the circle presents information regarding frequency and growth rate. The eigenvalues found from the analysis with DMD are discrete eigenvalues, for better interpretation they can be converted into continuous time eigenvalues. The transformation from discrete to continuous time eigenvalues is done using equation (3.2) where the Δt is the time between snapshots.

$$\Omega = \frac{\ln(\lambda)}{\Delta t} \quad (3.2)$$

Eigenvalues can be sorted into three categories based on their location in the complex unitary circle. When considering discrete eigenvalues, if the eigenvalue corresponds to a point within the unit circle, which will de-

cay and vanish over time. If the point can be found on the unit circle then it will oscillate over time. Finally, if the point appears outside the unit circle that means it will grow and diverge over time. After the transformation to continuous time eigenvalues, we see that decaying modes occur when $\text{Re}(\Omega) < 0$, oscillating modes occur when $\text{Re}(\Omega) = 0$, and growing modes occur when $\text{Re}(\Omega) > 0$. A summary of the temporal behaviour of discrete and continuous time eigenvalues is presented in Table (3.1).

Table 3.1: Temporal Behaviour Summary

Discrete Eigenvalue	Continuous time Eigenvalue	Temporal Evolution
Inside the unit circle	$\text{Re}(\Omega) < 0$	Decaying
On the unit circle	$\text{Re}(\Omega) = 0$	Oscillating
Outside the unit circle	$\text{Re}(\Omega) > 0$	Growing

To show just how beneficial it is to transform the eigenvalues from discrete time to continuous time eigenvalues an example data set was analyzed using DMD. The original and transformed eigenvalues were plotted beside each other for comparison in figure (3.1). It is clearly much easier to differentiate between the different types of temporal behaviour using the continuous-time eigenvalues.

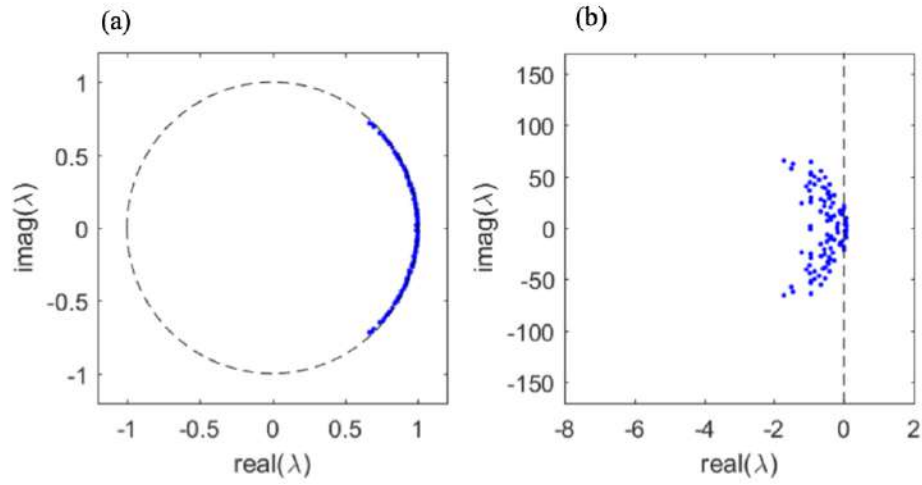


Figure 3.1: Spectrum Plot Comparison of (a) discrete versus (b) continuous time eigenvalues

A visual representation of each type of temporal evolution is depicted in figure (3.2). The red line represents a decaying mode, the green line represents a growing mode and the blue line represents an oscillating mode.

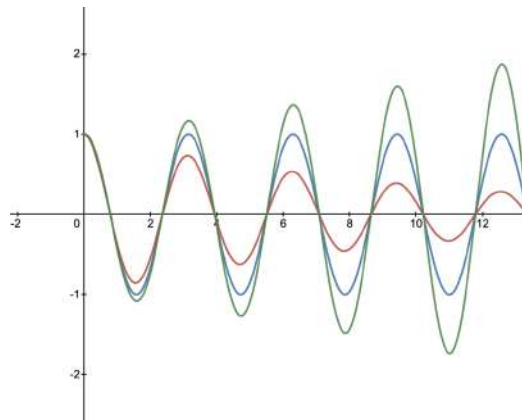


Figure 3.2: Time Evolution of Each Type of Mode

4. Application

The DMD technique can be applied to a multitude of areas, the two we looked at were its application to modelling the spread of infectious disease, and to the development of financial trading strategies.

4.1 Modeling infectious disease spread

Outbreaks of infectious diseases are the cause of death for millions of people worldwide each year. Stopping the spread of infectious diseases is one of the most important objectives of global health. Unfortunately due to its complexity and a lack of a single set of physics-based equations governing it, modelling its spread can be challenging. One such solution to this problem is dynamic mode decomposition, as the equation-free nature of DMD makes it easier to analyze infectious disease data.

An example of DMD applied to infectious disease data is described in [3], it was applied to Google's Flu Trends tool, which used aggregated Google search data and historical flu data to construct a method for de-

termining the state of flu activity. The collection of data consists of a two-dimensional array with the location axis showing state, city, and health and human services region, while the time axis shows seven-day advancements. A visualization of the flu trend data for each of the 50 states in America is shown in figure (4.1). Note that the y-axis is the number of standard deviations from the mean and 0 represents the historic baseline level of influenza activity for its corresponding region.

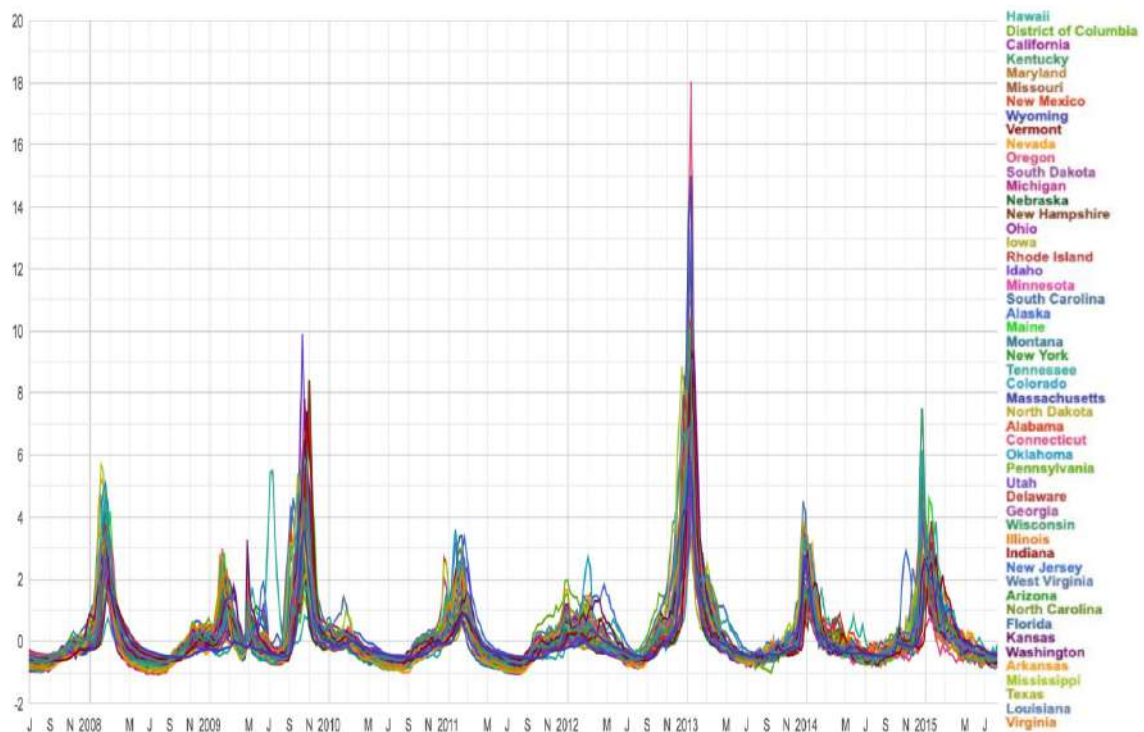


Figure 4.1: Google Flu Trends Data for United States [4]

Vectors can be constructed from the data collected at different spatial locations in order to represent the evolution of state snapshots over time.

Spatial and temporal patterns extracted from the dataset are represented by the collection of eigenvalues and their respective dynamic modes of decomposition.

The dynamic modes or eigenvectors give two critical insights into the disease system being modelled. The first is the magnitude of the mode, which allows us to quantify the spatial location's participation in that mode. The second is the angle between the real and imaginary components of the mode, which indicates the spatial location's phase of oscillation compared to the other locations in the analysis.

The eigenvalues are used as described above to determine the temporal nature of each dynamic mode as time progresses. Because it can be challenging to interpret the oscillatory frequency for a given Δt in the complex plane this paper converted from discrete to continuous time eigenvalues and then to a continuous oscillatory frequency via equation (4.1). Note that the Δt was 7 days in this case.

$$frequency_j = \frac{imag(\ln(\lambda_j)/\Delta t)}{2\pi} \quad (4.1)$$

4.1.1 DMD Output

The discrete eigenvalues determined from the dynamic mode decomposition are displayed in the following figure (4.2). The eigenvalue distribution on the unit circle shows that a large number of eigenvalues fall well inside the unit circle. This means their corresponding dynamic modes are

fast decaying and thus do not help to explain the long-term evolution of the disease system.

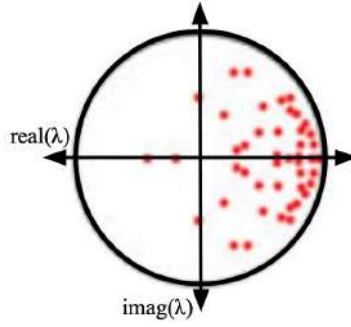


Figure 4.2: Discrete Eigenvalue Spectrum

A quantification of the level of dominance of each mode can be made by combining the decay rates of these eigenvalues with the overall magnitude of their corresponding modes and the number of iterative steps in the following relation (4.2). The mode selection plot Figure (4.3) presents a comparison of the dominance of each mode and is used to select the most relevant mode of the long-term behaviour.

$$power = \lambda_j^p \parallel \phi_j \parallel \quad (4.2)$$

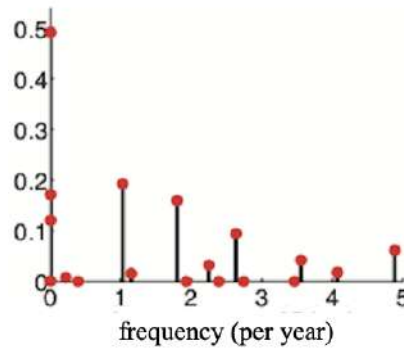


Figure 4.3: Mode Selection Plot

From the mode selection plot, it is evident that the relevant mode that should be chosen is the one associated with the once-per-year frequency as it is the most dominant. Figure (4.4) is an illustration of how the phase space of the selected dynamic mode can be plotted on a map of the United States, where each element of the dynamic mode represents a location on the map. The phase was plotted on a scale of 0 to 1 with that value indicating the time of year. Due to the fact that the phase value exists on a circle, the values near 0 and 1 are actually close in phase, which makes intuitive sense when thinking about how December is connected to January.

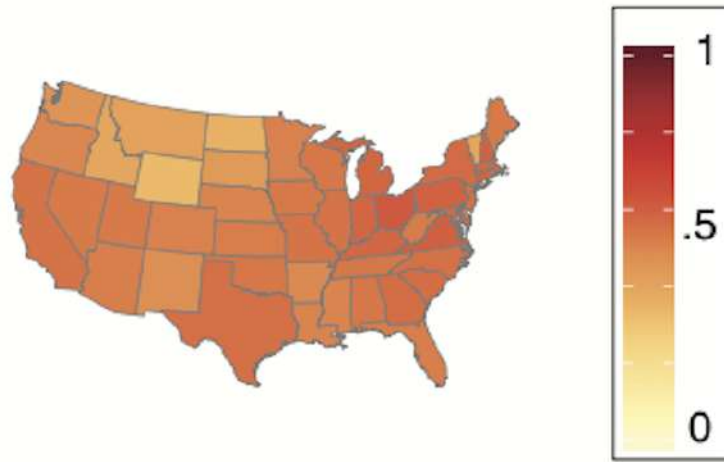


Figure 4.4: Yearly Phase Space Plotted on US Map

This phase plot can provide many key insights of high value to public health organizations. For example, it can aid in predicting the best timing to launch the fight against the flu and vaccination campaigns in specific regions. This can be done by observing the change of phase when moving around the map which represents the peak time of flu and the spread of the disease. Furthermore, DMD can be used to identify places with epidemiological connections, which are areas not necessarily geographically connected but share a common phase or time of peak flu activity. This is becoming increasingly relevant as long-distance travel is continuously becoming more commonplace.

4.2 Developing Financial Trading Strategies

At any given time, millions of companies, individuals, institutions, and even governments are trading on financial markets. Regardless of whether the trade is placed by a professional, a novice or an amateur, they all carry some degree of risk, no matter what instrument is traded. In order to be successful in trading and making money, one must balance potential profit against risk. One of the most effective ways to maximize profit is through the use of trading strategies that make use of prior market data. Modern financial investment strategies are becoming increasingly more reliant on algorithmic trading schemes. DMD can be applied to market data to construct financial trading strategies, taking advantage of the growth and decay behaviour of the decomposition. The DMD decomposes stock portfolio data into low-rank features that behave with prescribed time dynamics. Using the least-square fit linear dynamical system, it is possible to predict short-term future states of the system, allowing one to make automated purchases, sales, or hold decisions on investments.

In the paper, [5] DMD was applied to a dataset containing stock price quotes over a period between 2005 and 2015 provided by Yahoo! finance.yahoo.com. Using the closing prices of the stocks to be analyzed, a data matrix was constructed and organized chronologically with a one-day difference in time (Δt) between data extractions. A diagram illustrating the data matrix being used is shown in figure 1, where each stock can

be identified by its stock ticker symbol assigned by the NASDAQ.

$$\mathbf{X} = \begin{bmatrix} \dots & \vdots & \dots \\ \dots & \text{'JPM'} & \dots \\ \dots & \text{'BOA'} & \dots \\ \dots & \text{'AMZN'} & \dots \\ \dots & \text{'CVX'} & \dots \\ \dots & \vdots & \dots \end{bmatrix} \quad \begin{array}{l} \downarrow \text{companies} \\ \longrightarrow \\ \text{daily stock prices} \end{array}$$

Figure 4.5: Matrix Representation of Stock Data Provided by Yahoo!

To be able to determine the best combination for predicting the future market the approximate solution for a future time given by $x(t) = \Phi \exp(\Omega t) \mathbf{b}$ is instead parameterized by two variables $x(m, l)$. The variable m represents the number of past days of market snapshot data taken, and l represents the number of days in the future predicted, allowing us to control the sampling window and the prediction time frame. Their ranges were set at $1 \leq m \leq 25$ and $1 \leq l \leq 10$, the range for l was chosen as predicting too far into the future decreases the accuracy of the algorithm. The range for m was chosen such that the prior data was sufficiently large enough to capture current trends, but not so long that too much emphasis is placed on the data from the distant past.

The combinations leading to the optimal predictions are referred to as hot spots, to be considered a hot spot the combination must have a predic-

tion accuracy of 53% or more, and the average prediction score of this combination and its 8 adjacent cells is at least 53%. Based on empirical observations of the DMD algorithm performance over a 2-year training period from 2005 to 2007, 0.53 was chosen as the optimal prediction threshold.

4.2.1 DMD Output

Figure (4.6) shows the singular value decomposition for the data matrix X , which is comprised of the portfolio data for 18 companies in the transport and home construction sectors over an 18-day sampling window. Note k represents the number of measurement times and $\sigma_k / \sum \sigma_j$ represents the k^{th} diagonal element of the diagonal matrix Σ found from SVD divided by the sum of the eigenvalues from j to k . The decay of singular values of X can clearly be seen in the image. Since the singular values of sigma decrease sharply to zero we can say that there is a low-rank structure to the data and only 4 dominant modes in the data can be seen. These represent the modes with the maximal variance and are indicated by the colour change. The first mode represents the average price across the sampling window of the 18 companies, and the other 3 modes represent the weightings across the 18 companies. This reveals that the data may be appropriately represented as rank $r = 4$ allowing for a low-rank truncation to be performed.

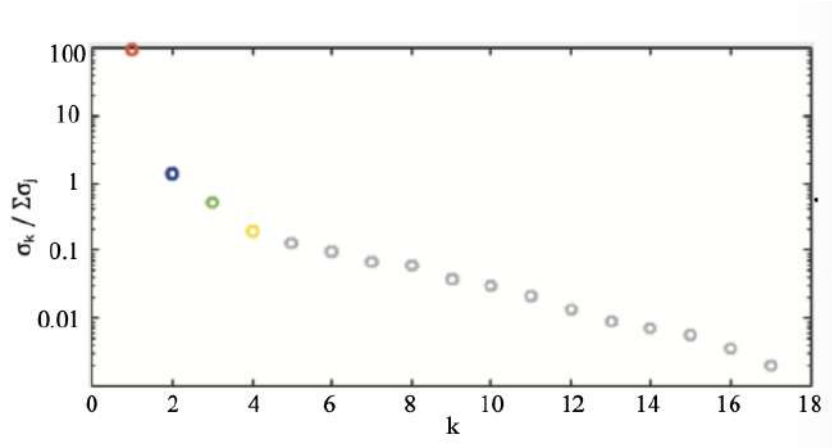


Figure 4.6: Singular Values of Data Matrix X on Log Scale

The accompanying spectrum plot of continuous time eigenvalues is shown in (4.7). Observe that the eigenvalues appear to cluster along the imaginary axis and many of the eigenvalues are only slightly decaying or growing. It can be seen that in this case three of the four dominant modes correspond to growth modes. The DMD scheme takes advantage of the largest growth modes to determine the best investment opportunities.

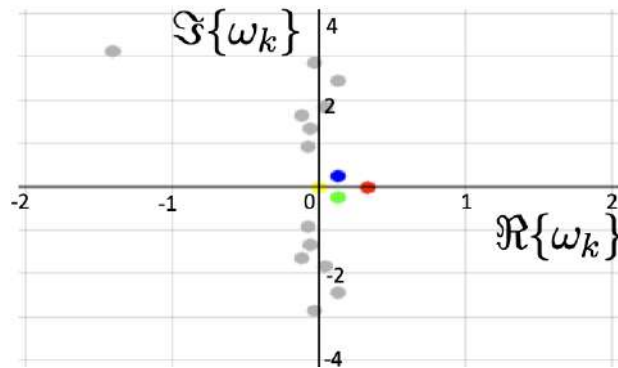


Figure 4.7: Continuous Time Eigenvalues of Data Matrix X

The following figure (4.8) shows the success rates achieved using different values for m and l for the DMD algorithm when back-testing, for three portfolios each with 9 companies in either the transport, home construction, or retail sectors over the remaining 8 year period. The optimal values of m and l were determined as explained prior during the 2-year training period and then used to execute trades from 2007 to 2015. The hot spots determined by the dynamic mode decomposition indicated by a star showed a success rate of 53% or greater over the last 8 years. In other words, 53% of all the trades executed using that combination was correct and profited. So the best-predicted value for the transport sector occurs when a sampling window of 8 days and a prediction time frame of 4 days are used. The home construction and retail sectors (11,5) and (8,5) provide similar results.

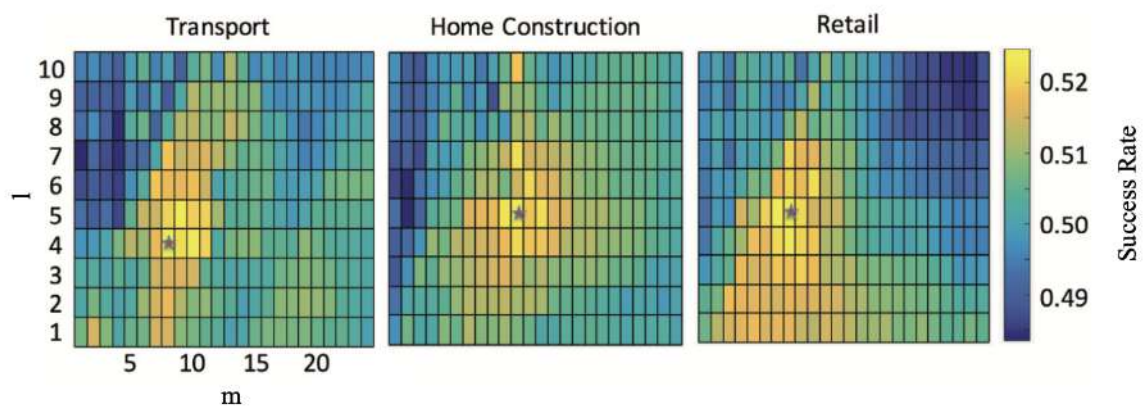


Figure 4.8: Success Rates of the DMD Algorithm for Transport, Home Construction, and Retail Sectors When Back Testing

To assess the DMD trading strategy the potential amount of money earned from the DMD algorithm was computed by theoretically entering a position on every company in the algorithm each day based on the predictions of the algorithm. It is important to note that three assumptions are made, the initial investment capital was \$1 million, transaction costs are \$8 for each position, and all money is invested evenly across all companies in a given portfolio. The returns for each portfolio over the 8-year period are indicated below (4.9) and are compared to simply holding stock in the S&P500 over the same period. It can be seen that while the transport sector greatly performs the other 2 sectors analyzed with DMD all 3 generate profits well above holding stock in the S&P500 demonstrating the effectiveness of DMD in trading strategies.

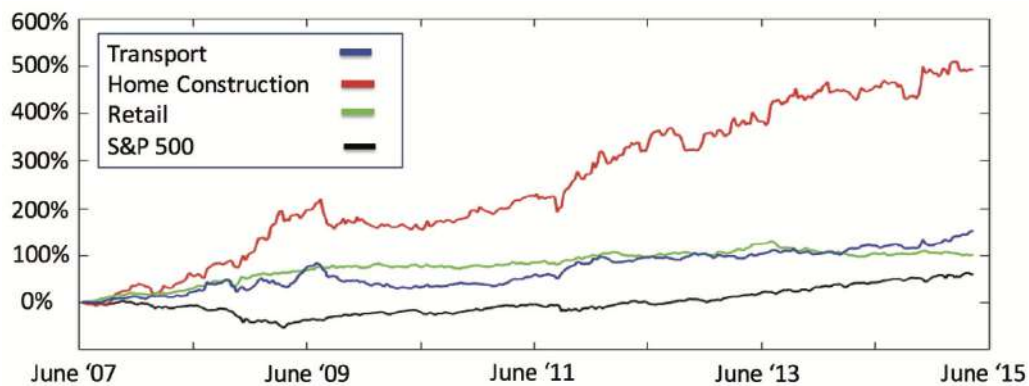


Figure 4.9: Trading Success of the DMD Algorithm for the 3 Sectors Compared to Stock in the S&P500

4.3 Modelling Weather Systems

We now apply DMD to weather systems, but we first include a primer on weather systems and supercells.

Supercell thunderstorms require certain ingredients for them to form. An unstable Air mass is dictated by warm moist air with warmer air below colder air, A trigger to start the storms such as a mass of cold air colliding with the unstable air, shear within the atmosphere and the sun radiating the earth heating the ground creating an unstable environment. Shear is governed by large changes in wind velocity at different heights above the earth. Figure (4.10) below shows a weather system.

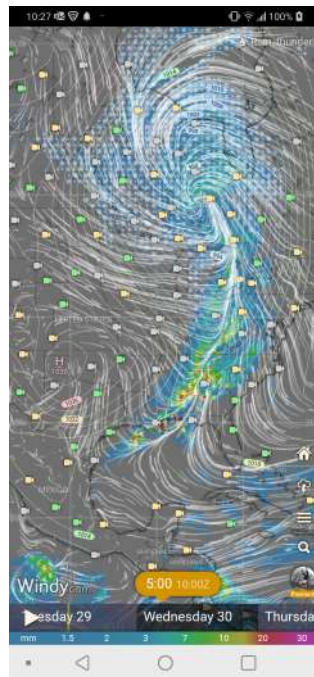


Figure 4.10: Example Weather system

The lines are areas of constant pressure. The white lines are winds. The kinks in the pressure lines are caused by a cold air mass, as temperature and pressure are related the kinks are caused by sudden changes in temperature and we denote this a frontal boundary. Ahead of the frontal boundary supercells develop which is seen further south in states given by red radar image. Figure (4.11) below shows how weather systems resemble a dynamical system.

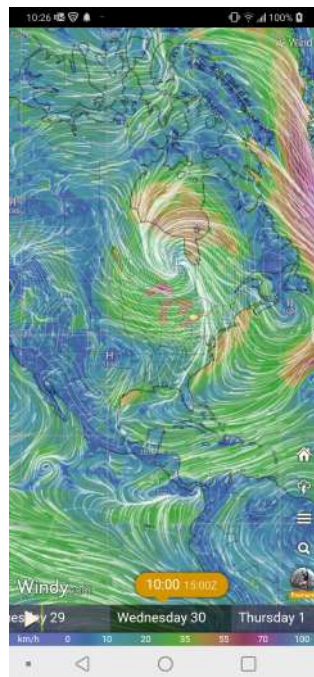


Figure 4.11: Weather system Resembling a Dynamical System

Low pressure is governed by rising air that rotates counterclockwise, High pressure is governed by sinking air rotating clockwise. Hence, where the central low can be thought of as a sink and where the central high can

be considered a source. The white lines show where winds converge in both Figure (4.10) and Figure (4.11). The converging winds can be thought of as an attractor where the winds converge along the attractor, an area of stability in the dynamical sense. Though the frontal boundary is thermodynamically unstable.

Shear is the main ingredient in rotating updrafts in a supercell. An idealized updraft of a supercell can be thought of U(1) gauge which resembles a helix. Figure (4.12) below shows the shear.

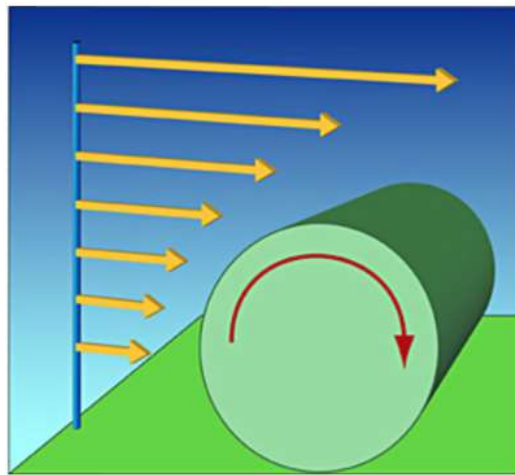


Figure 4.12: Visualization of Wind Shear

In Figure (4.12) the wind speed at higher levels vs lower levels in the atmosphere cause the winds to roll based on fluid dynamics and vorticity.

In figures (4.13) through figure (4.16) below we see how the horizontal vorticity gets tilted into a vertical vorticity which is caused by a frontal boundary as a lifting mechanism for the updraft and by the sun radiating

earth heating parcels of air which rise as an initial updraft.



Figure 4.13: Tornado Formation

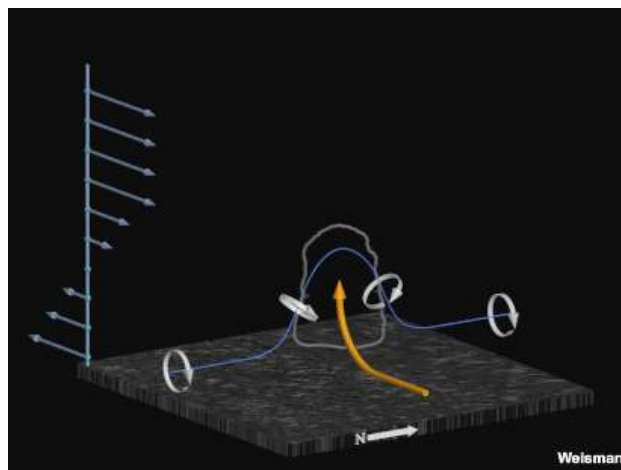


Figure 4.14: Transition from Horizontal to Vertical Vorticity (I)

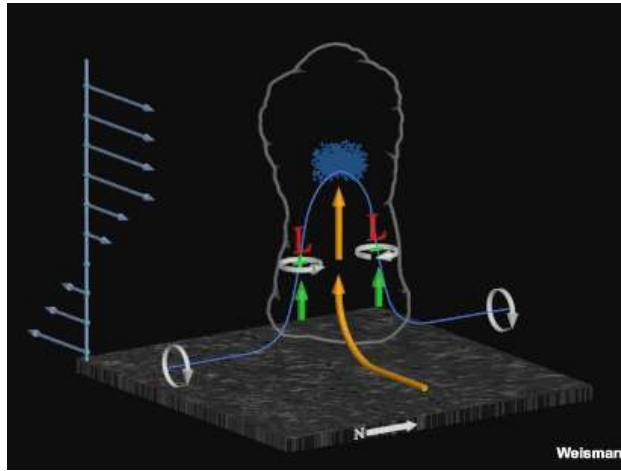


Figure 4.15: Transition from Horizontal to Vertical Vorticity (II)

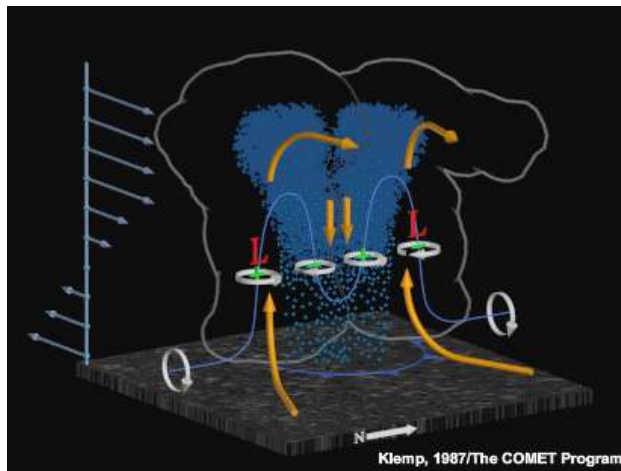


Figure 4.16: Transition from Horizontal to Vertical Vorticity (III)

In Figures (4.14) through (4.16) as a supercell matures it can create down drafts, which deform the vertical updraft into a sine-like wave which then can make new supercells that either attach to the previous ones or split off to form new ones. Below in figure (4.17) is another depiction of an

idealized horizontal vorticity which through an updraft and lifting mechanism gets tilted vertically.



Figure 4.17: Real World Example of Vorticity Transition

Now with enough background information on weather systems and supercells, we apply DMD to recorded videos of supercells to see how accurate a lower dimensional representation is of the supercell. We hope it will capture the dynamics of the system and the rotating updrafts and the structure.

Figures (4.18) and (4.19) below are supercell thunder linked by many supercells embedded within, multiple rotating updrafts. Specifically in this photo, two storms are colliding creating a rapid updraft scenario which can be seen in the shelf cloud. A shelf cloud is a very low cloud that sweeps across the above ground and is governed by multiple strong up-

drafts.

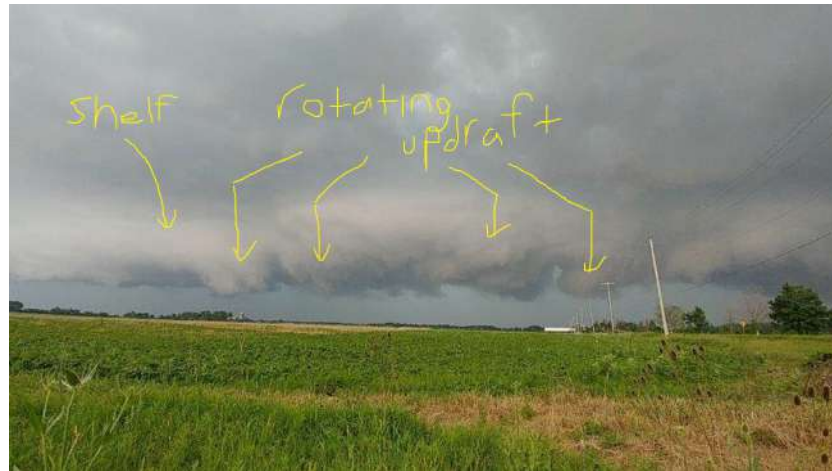


Figure 4.18: Collision of Two Storms (I)

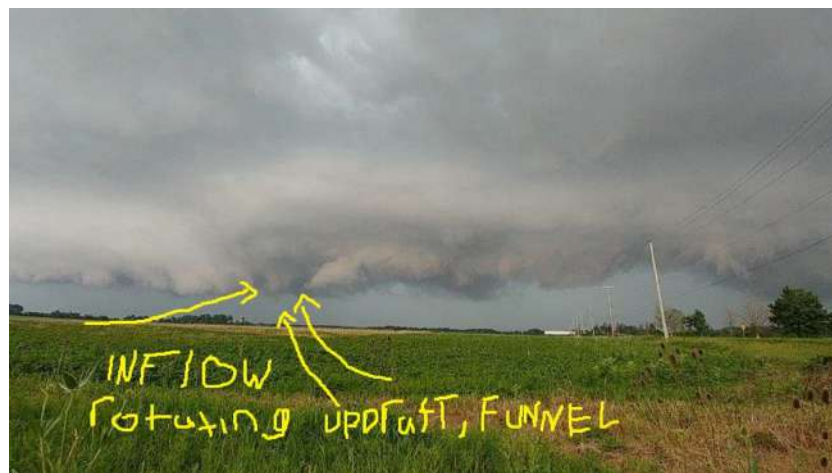


Figure 4.19: Collision of Two Storms (II)

After performing DMD on 73 frames, which has a rank of 73, the output has dimension 14 given by 14 eigenvalues. We here let the code determine the optimal dimension. the eigenvalues are in Figure (4.20) below.

The eigenvalues are plotted on the unit circle.

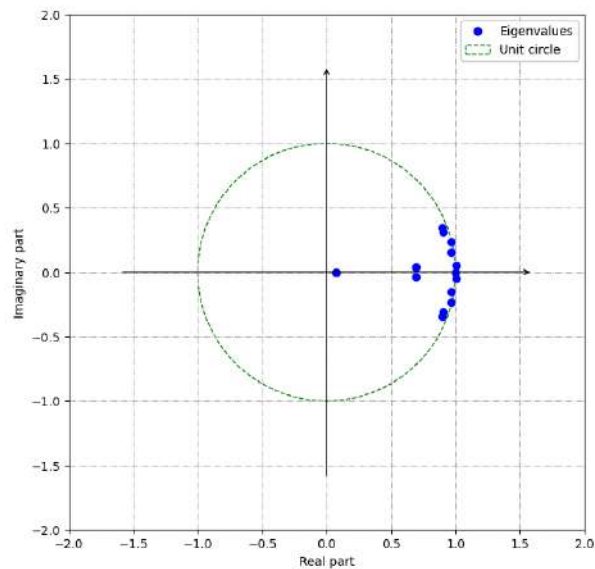


Figure 4.20: Eigenvalues of Storm Collision

In Figure (4.21) and (4.22) below we have the output of the DMD of dimension 14. The resulting supercell has been smoothed out, in which has left the shelf cloud retained though has lost much of the structure of the rotating updrafts. As we noted before, one of the challenges of DMD is DMD not picking up these subtle small invariances. We would expect the more eigenvalues given the better DMD will be at retaining the structure of the supercell. Notice most of the eigenvalues are positive representing an unstable system. What would happen if we have negative eigenvalues mostly? If we have eigenvalues outside the unit circle the DMD mode is

growing and is better at predicting when this happens.



Figure 4.21: Frame Produced From DMD With Dimension 14 (I)



Figure 4.22: Frame Produced From DMD With Dimension 14 (II)

In Figures (4.23) and (4.24) below we choose a rank of 60, 60 frames, to see how much structure of the supercell is retained.



Figure 4.23: Frame Produced from DMD With Rank 60 (I)



Figure 4.24: Frame Produced from DMD With Rank 60 (II)

In Figure (4.25) below the eigenvalues are depicted which are now more symmetrical, being distributed along the unit circle, as the symmetry of the eigenvalues increases so does the retaining of the structure of the supercell. Here one can clearly see the rotating updrafts vs the one

with a rank of 14. One requires great amounts of frames to retain the dynamics of the system, the more data the better. Though we are interested in low-dimensional representations which keep the same structure. More data is computation intensive and we are looking for a lower dimensional algorithm that retains the structure. This is one of the challenges of DMD, lower dimensional representation losing the subtle invariances. If a low dimensional representation of the dynamical system exists then the DMD algorithm would pick it up and retain the same structure as the original. Taking the contra-positive of the statement, Not a representation of the dynamical system implies a low dimensional representation of the system does not exist, hence loss of the structure. Though given the challenges of DMD the above statement applies when the DMD algorithm can identify invariances in the system along with other dynamics. As the rank increases to be close to the number of frames given, the eigenvalues distribute more symmetrically around the circle, hence a neutrally stable system. When we try to predict the future the DMD will produce a periodic extrapolation which is not very useful.

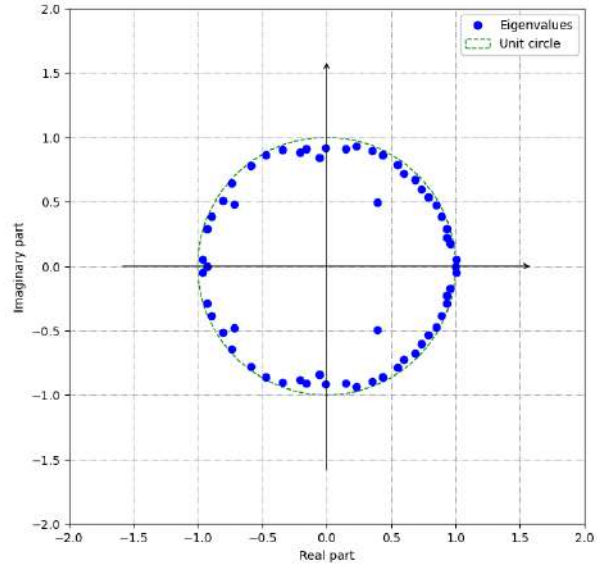


Figure 4.25: Updated Eigenvalues of Storm Collision

Next, we try to predict the future of the supercell by running DMD on the movie plus one full length of the movie time into the future. In Figures (4.26) through (4.29) an image is presented at the final frame, and images after the final frame.



Figure 4.26: Future Frame Produced by DMD (I)



Figure 4.27: Future Frame Produced by DMD (II)



Figure 4.28: Future Frame Produced by DMD (III)



Figure 4.29: Future Frame Produced by DMD (IV)

We see in Figures (4.26) through (4.29), The motion of the main super-cell has stopped though the DMD is still working within the boundaries given.

4.4 Fluid Dynamics

Then, we apply DMD to water flowing over rocks to simulate turbulence and see how it predicts future scenarios. Given 13 frames of data, we apply DMD specified with a rank of 13. The eigenvalues are depicted below in Figure (4.30)

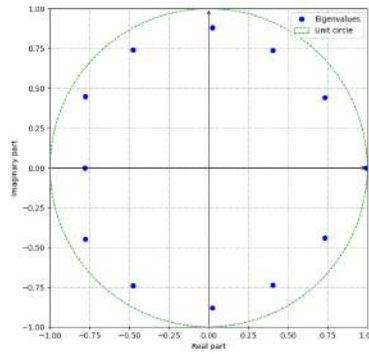


Figure 4.30: Eigenvalues of Water Flow

Notice this time we have eigenvalues that are symmetrical due to the full rank we specified. The DMD mode is decaying and oscillatory. Below is Figures (4.31) through (4.34) where Figures (4.30) and (4.34) are actual frames and in between those is the interpolation between the two frames produced by DMD.



Figure 4.31: Initial Frame of Water Flow From Recording



Figure 4.32: Frame Produced by DMD (I)



Figure 4.33: Frame Produced by DMD (II)



Figure 4.34: Final Frame of Water Flow From Recording

Next, we list Figures (4.35) through (4.38) where these figures occur after the last photo has stopped and it is predicting future outcomes, up to double the length of the time of the original video. We see here now the DMD is predicting outcomes of the turbulence and is not picking up the

almost laminar flow before the turbulence. The frames do not pick up the almost laminar flow and images appear to shift around vs flow. We recognized that DMD is not picking up some invariances, being the laminar flow. Given that laminar flow in the system has a consistent pixel intensity between frames it is not surprising that it is not picking up this invariance. Given the symmetrical nature of the eigenvalues, the prediction is periodic and decaying. This is not useful for future predictions.



Figure 4.35: Future Frame Produced by DMD (I)



Figure 4.36: Future Frame Produced by DMD (II)



Figure 4.37: Future Frame Produced by DMD (III)



Figure 4.38: Future Frame Produced by DMD (IV)

5. Conclusion

Dynamic mode decomposition (DMD) is a data-analyzation method that applies to a broad range of areas. It organized the data into two large matrices that share the same time interval but with different time stamps, then attempts to generate a mapping from one matrix to the next. Since the matrix can have an enormous amount of elements, DMD utilized the idea of dimensional reduction from singular value decomposition (SVD) for manageable calculations. From the sub-matrices decomposed in SVD, DMD reduces the dimension of the mapping by reformulating and reconstructing the data through its mode. Then from the DMD modes, we can predict the future data at a specific time frame.

This technique can be applied to a host of fields, two of which we studied were how to model infectious disease spread and how to develop financial trading strategies using it. It is clear from the examples above that the algorithm yielded high value information in both cases. In the case of the epidemiology example DMD automatically recognizes the importance of the yearly cycle, providing insight into how the dynamic pattern of the

associated mode spreads across a spatial domain. In developing a financial trading strategy DMD was able to produce a strategy that was able to generate profits well above holding stock in the S&P500.

However, as with everything else DMD has its limitations, one such issue with data-driven, equation-free methods like DMD is that they are limited by the quality and quantity of the data. Another of DMD's limitations is in predicting the future, as seen in the weather systems and fluid dynamics examples. Given the reduced dimension matrices, it can be difficult for computers to calculate as regenerating the data after the reconstruction process, the computer would have to regenerate the same amount of elements back by the end of the process. For the example of frames, although we can reduce the dimension to the number of frames, which is significantly smaller than the number of pixels per frame, we regenerate the same amount of pixels back to each time frame. Additionally, DMD does not capture invariances well. Rotations, translations and other types of low dimensional outputs pick out dominant movement yet miss the subtle invariances. A chaotic system that evolves over time will change and depends on these invariances, hence the system will change from the DMD output over time.

To improve DMD, we can introduce a controlling matrix to the system that acts on the matrix calculation. In the stance of snapshots, we can use a control to steer the snapshots by way of a set of snapshots and a controlling matrix that acts on the usual snapshots and matrix we have.

Additionally, introducing analytic solutions of integral systems that approximate the system for a certain time or studying the geometry of fixed rank matrices in order to establish a control to steer the system. It has been conjectured that Maxwell's equations could be investigated to approximate the governing equations of fluid dynamics. Here we are studying the manifold of fixed rank matrices, we could embed this in a higher dimension and project it onto another manifold of known symmetry, analyze the system, and pull it back to the original manifold to try and establish a control to steer system for a certain period of time. Given there is no analytical solution for the Navier-Stokes equations, multiple manifolds will have to be used with a given symmetry to approximate the Navier-Stokes equations, as an evolving system could change the type of symmetry it has as time progresses.

Another improvement is to have a relation between pixel density and governing equations to be able to incorporate noticing symmetries via DMD in the system. An example is the laminar flow of water, DMD would see that as stationary since the laminar flow does not look like its moving.

To conclude, although Dynamic Model Decomposition is applicable given a set of data, it is computationally intensive. However, some strategies involve the use of tensor products of matrices to reduce computational time.

6. Algorithm

```
!pip install scipy
!pip install matplotlib
import os

import matplotlib.pyplot as plt
import warnings
import scipy
warnings.filterwarnings('ignore')

!pip install pydmd
!pip install pykoopman

from pydmd import DMD
from matplotlib import animation
from IPython.display import HTML
from imageio import imread
```

```

import numpy as np
from os import listdir
import pykoopman as pk
from pydmd import DMDc
from pydmd import DMD, SubspaceDMD
from pydmd import HODMD
from google.colab import files
uploaded = files.upload()
os.chdir('/content/june29/')

frames = os.listdir('/content/june29/')
vectorized_frames = []

for frame in frames:
    vectorized_picture =
    np.array(imread("/content/june29/" + frame).flatten())
    # reshaped_vectorized_picture =
    vectorized_picture.reshape(1, -1).T
    vectorized_frames.append(vectorized_picture)

vectorized_frames = np.array(vectorized_frames)

print(vectorized_frames)

```

```

dmd = DMD(svd_rank=35)
dmd.fit(vectorized_frames.T/255)
dmdc = DMD(svd_rank=35, tlsq_rank=20)
#dmd3 = HODMD(svd_rank=12, d=1)
#dmd3.fit(vectorized_frames.T/255)
#dmd3.plot_eigs()
#dmdc = DMDc(svd_rank=35, tlsq_rank=10)
#dmdc.fit(vectorized_frames.T/255, np.random.rand(34, 34))
dmdc.fit(vectorized_frames.T/255)
for eig in dmd.eigs:
    print('Eigenvalue_{:}_distance_from_unit_circle_{:}'.format(eig,
        np.abs(np.sqrt(eig.imag**2+eig.real**2) - 1)))
for eig in dmdc.eigs:
    print('Eigenvalue_{:}_distance_from_unit_circle_{:}'.format(eig,
        np.abs(np.sqrt(eig.imag**2+eig.real**2) - 1)))
#for eig in dmd3.eigs:
#    print('Eigenvalue {}: distance from unit circle {}'.format(eig,
#        np.abs(np.sqrt(eig.imag**2+eig.real**2) - 1)))

dmd.plot_eigs(show_axes=True, show_unit_circle=True)
dmdc.plot_eigs(show_axes=True, show_unit_circle=True)
#dmd3.plot_eigs(show_axes=True, show_unit_circle=True)

```

```

dmd.dmd_time[ 'dt' ] *= .10
dmd.dmd_time[ 'tend' ] *=2
dmdc.dmd_time[ 'dt' ] *= .10
dmdc.dmd_time[ 'tend' ] *=2
# dmd3.dmd_time[ 'dt' ] *= .10
# dmd3.dmd_time[ 'tend' ] *=1

fig = plt.figure()

out = [[plt.imshow(state.reshape(450, 800, 3),
interpolation='nearest')]
for state in np.abs(dmd.reconstructed_data.T)]

ani = animation.ArtistAnimation(fig,out,
interval=500*dmd.dmd_time[ 'dt' ],blit=False)

ani.save('interp.mp4')

plt.show()

out1 = [[plt.imshow(state.reshape(450, 800, 3),

```

```

interpolation='nearest')]
for state in np.abs(dmdc.reconstructed_data.T)]
ani1 = animation.ArtistAnimation(fig,out,
interval=500*dmdc.dmd_time['dt'],blit=False)
ani1.save('interp1.mp4')
plt.show()

# out2 = [[plt.imshow(state.reshape(450, 800, 3), interpolation=
'nearest')] for state in
np.abs(dmd3.reconstructed_data)]
# ani2 = animation.ArtistAnimation(fig,out2,
interval=3000*dmd3.dmd_time['dt'],blit=False)
# ani2.save('interp2.mp4')
# plt.show()

#DMD.plot_snapshots_2D(, index_snap=None)
#####

```

7. Bibliography

- [1] Muniraju A, (2018) *Analysis of Dynamic Mode Decomposition*, Theses and Dissertations, 1879.
- [2] Kutz JN, Brunton SL, Brunton BW, et al. (2016) *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, Philadelphia, PA, Society for Industrial and Applied Mathematics.
- [3] Proctor JL, Eckhoff PA, (2015) *Discovering dynamic patterns from infectious disease data using dynamic mode decomposition*. International Health 7: 139–145.
- [4] Google flu trends estimates https://www.google.com/publicdata/explore?ds=z3bsqef7ki44ac_#!ctype=l&strail=false&bcs=d&nselm=h&met_y=flu_index&scale_y=lin&ind_y=false&rdim=region&idim=region:US-AL:US-AK:US-AZ:US-AR:US-CA:US-CO:US-CT:US-DE:US-DC:US-FL:US-GA:US-HI:US-ID:US-IL:US-IN:US-IA:US-KS:US-KY:US-LA:US-ME:US-MD:US-MA:US-MI:US-MN:US-MS:US-MO:US-MT:US-NE:

US-NV:US-NH:US-NJ:US-NM:US-NY:US-NC:US-ND:US-OH:US-OK:
US-WY:US-WI:US-WV:US-WA:US-VA:US-VT:US-UT:US-TX:US-TN:
US-OR:US-PA:US-RI:US-SC:US-SD&ifdim=region:country:
US&tstart=1183003200000&tend=1438833600000&hl=
en_US&dl=en_US&ind=false (accessed Nov 1, 2022).

- [5] Mann J, Kutz JN, (2016) *Dynamic mode decomposition for financial trading strategies*. Quantitative Finance 16: 1643–1655.