Data-Driven Dynamical Systems

EE736: Introduction to Stochastic Optimization

Rathour Param Jitendrakumar
190070049
Department of Electrical Engineering
Indian Institue of Technology Bombay
Guide: Prof. Vivek Borkar

Acknowledgements

This report was developed as a part of the course *EE736: Introduction to Stochastic Optimization* to summarise the concept of Dynamic Mode Decomposition and Sparse Identification of Nonlinear Dynamics. I thank my parents for always motivating me. I would also like to express gratitude to my guide: Prof. Vivek Borkar for introducing me to this fascinating topic. Most of the content here is inspired by [18, 15, 4, 7], many statements and equations are directly taken from there and my explanations are based on my understanding from the book and papers. I thought of including mathematical details of Koopman Theory but decided against it in the end to avoid too much mathematics. And, lastly thanks to you, reader, I have worked hard in making this report, in the hope that this work helps you in some way.

Contents

Part I

Introduction

Dynamical systems provide a mathematical framework to describe the world around us, they model the rich interactions between ever evolving quantities. Dynamical systems involves the analysis, understanding and prediction of the behaviour of systems. The system could be a differential equation or an iterative map that describes the evolution of the state of a system. The study can be viewed as the realisation of hundreds of years of mathematical modelling, originating with the classical mechanics of Newton and Leibniz.

Currently, analytical and first-principles models are giving way to data-driven approaches in Modern dynamical systems. This paradigm shift in the study of dynamical systems is result of the convergence of big data and machine learning. Many critical data-driven problems, such as predicting climate change, predicting and suppressing the spread of disease, understanding cognition from neural recordings, or controlling turbulence for energy-efficient power production and transportation, are primed to take advantage of progress in the data-driven discovery of dynamics.

Now, an operator-theoretic perspective based on the evolution of measurements of the system, adds on to the classical geometric and statistical perspectives on dynamical systems. The increasing availability of data from complex systems is highly beneficial for Koopman operator theory [15]. Moreover, with Koopman theory, one can identify intrinsic coordinate systems to represent nonlinear dynamics in a linear framework. This has the potential to revolutionize our ability to predict and control these systems.

This report presents a modern data-driven outlook on dynamical systems. Data-driven dynamical systems has been a rapidly growing field. Hence, our primary focus is on a blend of established and emerging methods for thee same. We will also look at the key challenges of identifying dynamics from data and uncovering data-driven representations that make nonlinear systems amenable to linear ones.

1 Dynamical Systems

Consider dynamical systems of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t; \boldsymbol{\beta}) \tag{1.1}$$

where $\mathbf{x}$ is the state of the system and $\mathbf{f}$ is a vector field possibly depending on the state $\mathbf{x}$, time t, and a set of parameters $\boldsymbol{\beta}$. A even simpler case without time dependence (an autonomous system) or parameters

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)) \tag{1.2}$$

Formally, $\mathbf{x}(t) \in \mathbf{M}$ is an n-dimensional state that lives on a smooth manifold $\mathbf{M}$, and $\mathbf{f}$ is an element of the tangent bundle $\mathbf{TM}$ of $\mathbf{M}$ so that $\mathbf{f}(\mathbf{x}(t)) \in \mathbf{T}_{\mathbf{x}(t)}\mathbf{M}$. Nevertheless, let's consider the simpler case where $\mathbf{x}$ is a vector, $\mathbf{M} = \mathbb{R}^n$, and $\mathbf{f}$ is a Lipschitz continuous function, guaranteeing the existence and uniqueness of solutions to 1.2.

1.1 Discrete-Time Systems

Also known as a map, given by

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k) \tag{1.3}$$

The discrete-time dynamics are more general than the continuous-time formulation in 1.2, encompassing discontinuous and hybrid systems as well.

Discrete-time dynamics may be generated from continuous-time dynamics, where $\mathbf{x}_k$ is obtained by sampling the trajectory in 1.2 in time, so that $\mathbf{x}_k = \mathbf{x}(k\Delta t)$. The discrete-time propagator $\mathbf{F}_{\Delta t}$ is now parameterized by the time-step Δt. For an arbitrary time t, the flow map $\mathbf{F}_t$ is defined as

$$\mathbf{F}_t(\mathbf{x}(t_0)) = \mathbf{x}(t_0) + \int_{t_0}^{t_0+t} \mathbf{f}(\mathbf{x}(\tau))\,\mathrm{d}\tau \tag{1.4}$$

2 Linear Dynamics and Spectral Decomposition

Linear dynamics are of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \mathbf{A}\mathbf{x}, \tag{2.1}$$

and their closed-form solutions are given by,

$$\mathbf{x}(t_0 + t) = e^{\mathbf{A}t}\mathbf{x}(t_0) \tag{2.2}$$

Unsurprisingly, the dynamics are entirely characterized by the eigenvalues and eigenvectors of the matrix $\mathbf{A}$, called the spectral decomposition (a.k.a. eigendecomposition) of $\mathbf{A}$

$$\mathbf{AT} = \mathbf{T\Lambda} \tag{2.3}$$

where the matrix $\mathbf{\Lambda}$ consists of Jordan blocks, its diagonal entries are eigenvalues λ_j . If this $\mathbf{\Lambda}$ is diagonal, then the columns of $\boldsymbol{T}$, say ξ_j are the corrensponding eigenvalues. Now, by applying the transformation $\mathbf{z} = \mathbf{T}^{-1}\mathbf{x}$, the system gets decoupled

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z} = \mathbf{\Lambda}\mathbf{z}, \tag{2.4}$$

2.1 Challenges in Modern Dynamical Systems

Nonlinearity Working with linear systems is simple and desirable. Sadly, no closed form solution or simple linear transformation like 2.4 exist for nonlinear systems, in general. A way is to use geometry of subspaces of local linearizations around fixed points, periodic orbits and such attractors. But in such a locally linear model, predicting global phenomena remains difficult.

Unknown dynamics In many fields such as neuroscience, epidemiology, and ecology which tackle complex realistic systems, there is a basic lack of physical laws governing these systems. Even if the dynamics are known like in turbulence, protein folding, with higher dimensions uncovering dominant behviour is tough.

2.2 Possible Solutions

Operator-theoretic representations Using Koopman operator, we can represent nonlinear dynamical systems in terms of infinite-dimensional linear operators

Data-driven regression and machine learning is becoming a critical tool to discover discover dynamical systems from data and it forms the basis of DMD, SINDy.

Part II

Dynamic Mode Decomposition (DMD)

It has been observed that many dynamical systems exhibit low-dimensional behavior, despite their dynamics possibly being infinite dimensional. Fluid flows is one such an example. To analyse such systems, we use modal decomposition techniques. One such technique, Dynamic Mode Decomposition (DMD) provides a modal decomposition where each mode consists of spatially correlated structures that have the same linear behaviour in time. Each mode is associated with a characteristic value $\lambda = a + ib$, where a is growth or decay rate and b is frequency of oscillation. DMD also provides a model for the evolution of each mode with time. It utilizes the computationally efficient singular value decomposition (SVD) to calculate these modes. DMD is fundamentally a data-driven algorithm and does not require the knowledge of governing equations. It is thus equally applicable to experimental and numerical data. It has many variants which are applicable to existing system identification and modal extraction techniques.

1 The DMD Algorithm

The DMD algorithm seeks a best fit linear matrix operator $\mathbf{A}$ that approximately advances the state of a system, $\mathbf{x} \in \mathbb{R}^n$, forward in time according to the linear dynamical system

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \tag{1.1}$$

where $\mathbf{x}_k = \mathbf{x}(k\Delta t)$, and Δt denotes a fixed time step that is small enough to determine the highest frequencies in the dynamics. This matrix $\mathbf{A}$ is an approximation of the Koopman operator $\mathcal{K}$ restricted to a measurement subspace spanned by direct measurements of the state $\mathbf{x}$ [15].

Note. *The DMD algorithm is fundamentally based on the assumption that there are dominant low-dimensional patterns even in high-dimensional data, such as fluid flow fields.*

1.1 Exact DMD

The matrix $\mathbf{A}$ is approximated from a collection of snapshot pairs of the system, $\{(\mathbf{x}(t_k), \mathbf{x}(t'_k)\}_{k=1}^{m}$, where $t'_k = t_k + \Delta t$. A snapshot is typically a measurement of the full state of the system, such as velocity of fluid field sampled at a large number of spatially discretized locations, reshaped into a huge column vector. The original formulation of Schmid [2] required data from a single trajectory with uniform sampling in time, so that $t_k = k\Delta t$. But Tu [4] after making slight changes to the algorithm, demonstrated an *Exact* DMD which works for irregularly spaced data and concatenated data from multiple different time series. Thus, in exact DMD, the times t_k need not be sequential or evenly spaced, but for each snapshot $\mathbf{x}(t_k)$ there is a corresponding snapshot $\mathbf{x}(t'_k)$ one time step Δt in the future. These snapshots are arranged into two data matrices, $\mathbf{X}$ and $\mathbf{X}'$:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_1) & \mathbf{x}(t_2) & \cdots & \mathbf{x}(t_m) \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t'_1) & \mathbf{x}(t'_2) & \cdots & \mathbf{x}(t'_m) \\ | & | & & | \end{bmatrix}. \tag{1.2}$$

Equation (1.1) may be written in terms of these data matrices as

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X}. \tag{1.3}$$

The best fit matrix $\mathbf{A}$ establishes a linear dynamical system that approximately advances snapshot measurements forward in time. This can also be formulated as an optimization problem as shown below

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F = \mathbf{X}'\mathbf{X}^\dagger \tag{1.4}$$

where $\|\cdot\|_F$ is the Frobenius norm and † denotes the pseudo-inverse. The pseudo-inverse can be computed using the SVD of $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ as $\mathbf{X}^\dagger = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^*$, where the matrices $\mathbf{U} \in \mathbb{C}^{n\times n}$ and $\mathbf{V}^{m\times m}$ are unitary, so that $\mathbf{U}^*\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^*\mathbf{V} = \mathbf{I}$, where * denotes complex-conjugate transpose. The columns of $\mathbf{U}$ are known as POD modes, as $\mathbf{A}$ is an approximate representation of the Koopman operator restricted to a finite-dimensional subspace of linear measurements. Now, the eigenvectors $\mathbf{\Phi}$ and eigenvalues $\mathbf{\Lambda}$ of $\mathbf{A}$ satisfy:

$$\mathbf{A\Phi} = \mathbf{\Phi\Lambda}. \tag{1.5}$$

However, the matrix $\mathbf{A}$ has n^2 elements, so for high-dimensional data it is intractable to represent, let alone compute the corresponding eigendecomposition. Instead, the DMD algorithm seeks the leading spectral decomposition (i.e., eigenvalues and eigenvectors) of $\mathbf{A}$ without ever explicitly constructing it. The data matrices $\mathbf{X}$ and $\mathbf{X}'$ typically have far more rows than columns, i.e. $m \ll n$, so that $\mathbf{A}$ will have at most m nonzero eigenvalues and non-trivial eigenvectors. Usually, the effective rank of these data matrices, and hence the matrix $\mathbf{A}$, may be even lower, given by $r < m$. Hence, instead of computing $\mathbf{A}$ in (1.4), we may project $\mathbf{A}$ onto the first r POD modes in $\mathbf{U}_r$ and approximate the pseudo-inverse using the rank-r SVD approximation $\mathbf{X} \approx \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^*$:

$$\tilde{\mathbf{A}} = \mathbf{U}_r^*\mathbf{A}\mathbf{U}_r = \mathbf{U}_r^*\mathbf{X}'\mathbf{X}^\dagger\mathbf{U}_r = \mathbf{U}_r^*\mathbf{X}'\mathbf{V}_r\mathbf{\Sigma}_r^{-1}\mathbf{U}_r^*\mathbf{U}_r \tag{1.6}$$

$$= \mathbf{U}_r^*\mathbf{X}'\mathbf{V}_r\mathbf{\Sigma}_r^{-1}. \tag{1.7}$$

Now, the leading spectral decomposition of $\mathbf{A}$ is approximated from the spectral decomposition of $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W\Lambda}. \tag{1.8}$$

The diagonal matrix $\mathbf{\Lambda}$ contains the *DMD eigenvalues* corresponding to eigenvalues of the high-dimensional matrix $\mathbf{A}$. The columns of $\mathbf{W}$ are eigenvectors of $\tilde{\mathbf{A}}$. They provide a coordinate transformation that diagonalizes the matrix. These columns may be thought of as linear combinations of POD mode amplitudes that behave linearly with a single temporal pattern given by the corresponding eigenvalue λ.

The eigenvectors of $\mathbf{A}$ are the *DMD modes* $\mathbf{\Phi}$, and they are reconstructed using the eigenvectors $\mathbf{W}$ of the reduced system and the time-shifted data matrix $\mathbf{X}'$:

$$\mathbf{\Phi} = \mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{W}. \tag{1.9}$$

Tu [4] proved that these DMD modes are eigenvectors of the full $\mathbf{A}$ matrix under certain conditions one of which is linear consistency.

Definition 1 (Linear consistency). *Two $n \times m$ matrices X and Y are* linearly consistent *if, whenever $Xc = 0$, then $Yc = 0$ as well.*

Thus X and Y are linearly consistent if and only if the nullspace of Y contains the nullspace of X.

Theorem 1.1. Define $\mathbf{A} = \mathbf{X}'\mathbf{X}^\dagger$, then $\mathbf{X}' = \mathbf{AX}$ if and only if $\mathbf{X}$ and $\mathbf{X}'$ are linearly consistent.

A demonstration of the algorithm for fluid flow is shown at 1.1.

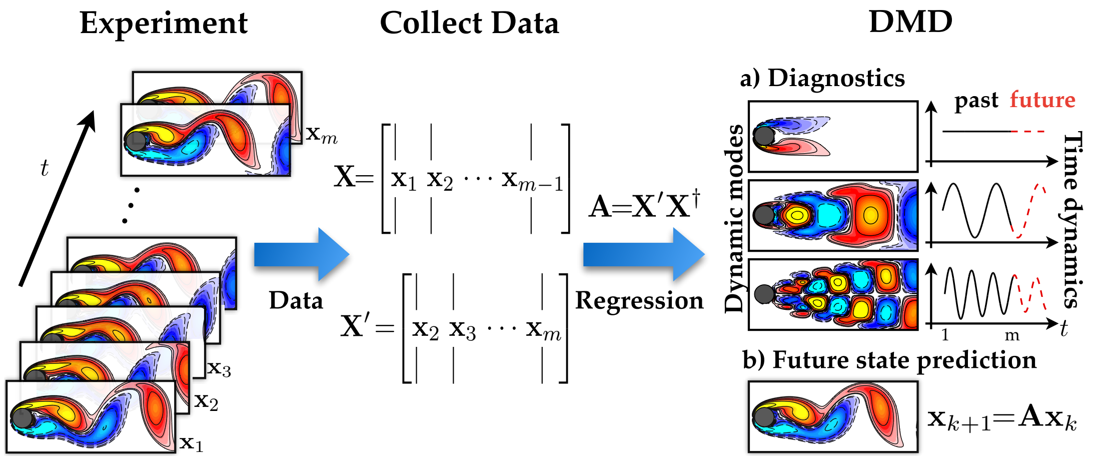

Experiment
Collect Data
DMD
a) Diagnostics
past
future
Dynamic modes
Time dynamics
t
$\mathbf{x}_m$
$\mathbf{x}_3$
$\mathbf{x}_2$
$\mathbf{x}_1$
Data
$\mathbf{X}=[\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_{m-1}]$
$\mathbf{X}'=[\mathbf{x}_2\ \mathbf{x}_3\ \cdots\ \mathbf{x}_m]$
$\mathbf{A}=\mathbf{X}'\mathbf{X}^\dagger$
Regression
1
m
t
b) Future state prediction
$\mathbf{x}_{k+1}=\mathbf{A}\mathbf{x}_k$

Figure 1.1: From [9], Overview of DMD demonstrated on the fluid flow past a circular cylinder

1.2 Original DMD

The matrices $\mathbf{X}$ and $\mathbf{X}'$ are created from sequential snapshots, evenly spaced in time [2],

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{m+1} \\ | & | & & | \end{bmatrix}. \tag{1.10}$$

The columns of $\mathbf{X}$ belong to a Krylov subspace generated by $\mathbf{A}$ and $\mathbf{x}_1$:

$$\mathbf{X} \approx \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{A}\mathbf{x}_1 & \cdots & \mathbf{A}^{m-1}\mathbf{x}_1 \\ | & | & & | \end{bmatrix}. \tag{1.11}$$

Thus, DMD is related to Arnoldi iteration to find the dominant eigenvalues and eigenvectors of $\mathbf{A}$.

Matrices $\mathbf{X}$ and $\mathbf{X}'$ are related through a finite-dimensional representation of the *shift* operator called $\mathbf{S}$.

$$\mathbf{X}' = \mathbf{X}\mathbf{S}, \tag{1.12}$$

Now, $\mathbf{S}$ acts on columns of $\mathbf{X}$, as opposed to $\mathbf{A}$, which acts on rows of $\mathbf{X}$. The shift matrix $\mathbf{S}$ has the form of a companion matrix and is given by

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & a_1 \\ 1 & 0 & 0 & \cdots & 0 & a_2 \\ 0 & 1 & 0 & \cdots & 0 & a_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & a_m \end{bmatrix}. \tag{1.13}$$

The first $m-1$ columns of $\mathbf{X}'$ are obtained by shifting the last $m-1$ columns of $\mathbf{X}$, and the last column is obtained as a best-fit combination of the m columns of $\mathbf{X}$ that minimizes the residual. as the shift matrix advances snapshots forward in time it can be viewed as a matrix representation of the Koopman operator. The $m \times m$ matrix $\mathbf{S}$ has the same non-zero eigenvalues as $\mathbf{A}$, so that $\mathbf{S}$ may be used to obtain dynamic modes and eigenvalues. But, computations with $\mathbf{S}$ are not as numerically stable as the DMD algorithm.

2 Spectral Decomposition and DMD Expansion

DMD has the ability to expand the system state in terms of a data-driven spectral decomposition

$$\mathbf{x}_k = \sum_{j=1}^{r} \boldsymbol{\phi}_j \lambda_j^{k-1} b_j = \boldsymbol{\Phi}\boldsymbol{\Lambda}^{k-1}\mathbf{b} \quad = \begin{bmatrix} | & & | \\ \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_r \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_r \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_r \end{bmatrix} \tag{2.1}$$

where b_j is the mode amplitude, $\boldsymbol{\phi}_j$ are DMD modes (eigenvectors of the $\mathbf{A}$ matrix) and λ_j are DMD eigenvalues (eigenvalues of the $\mathbf{A}$ matrix). The DMD expansion may be written equivalently as

$$\mathbf{x}_k = \begin{bmatrix} | & & | \\ \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_r \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_r \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_r \end{bmatrix} \tag{2.2}$$

which makes it possible to express the data matrix $\mathbf{X}$ as

$$\mathbf{X} = \begin{bmatrix} | & & | \\ \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_r \\ | & & | \end{bmatrix} \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_r \end{bmatrix} \begin{bmatrix} \lambda_1 & \cdots & \lambda_1^{m-1} \\ \vdots & \ddots & \vdots \\ \lambda_r & \cdots & \lambda_r^{m-1} \end{bmatrix}. \tag{2.3}$$

Now, $\mathbf{b}$ is calculated using the first snapshot to determine the mixture of DMD mode amplitudes

$$\mathbf{b} = \mathbf{\Phi}^{\dagger}\mathbf{x}_1, \tag{2.4}$$

The spectral expansion in (2.2) is converted to continuous time by continuous eigenvalues $\omega = \log(\lambda)/\Delta t$:

$$\mathbf{x}(t) = \sum_{j=1}^{r} \boldsymbol{\phi}_j e^{\omega_j t} b_j = \mathbf{\Phi}\exp(\mathbf{\Omega}t)\mathbf{b}, \tag{2.5}$$

where $\mathbf{\Omega}$ is a diagonal matrix containing the continuous-time eigenvalues ω_j.

Thus, the data matrix $\mathbf{X}$ may be represented as

$$\mathbf{X} \approx \begin{bmatrix} | & & | \\ \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_r \\ | & & | \end{bmatrix} \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_r \end{bmatrix} \begin{bmatrix} e^{\omega_1 t_1} & \cdots & e^{\omega_1 t_m} \\ \vdots & \ddots & \vdots \\ e^{\omega_r t_1} & \cdots & e^{\omega_r t_m} \end{bmatrix} = \mathbf{\Phi}\mathrm{diag}(\mathbf{b})\mathbf{f}T(\boldsymbol{\omega}). \tag{2.6}$$

3 Alternative Optimizations to De-Noise and Robustify DMD

The effects of sensor noise and stochastic disturbances must be taken care of when characterising experimental data with DMD as the DMD algorithm is purely data-driven, Dawson [8] proposed to remove the systematic bias of the DMD algorithm, to compute both forward and backward in time and average the equivalent matrices. Thus the two following approximations are considered

$$\mathbf{X}' \approx \mathbf{A}_1\mathbf{X} \qquad \text{and} \qquad \mathbf{X} \approx \mathbf{A}_2\mathbf{X}' \tag{3.1}$$

where $\mathbf{A}_2^{-1} \approx \mathbf{A}_1$ for noise-free data. Thus the matrix $\mathbf{A}_2$ is the backward time-step or simply the inverse mapping the snapshots from time t_{k+1} to t_k. Now averaging the forward and backward time matrices, removes the systematic bias from the measurement noise

$$\mathbf{A} = \frac{1}{2}\left(\mathbf{A}_1 + \mathbf{A}_2^{-1}\right) \tag{3.2}$$

where the optimisation (1.4) can be used to compute both the forward and backward mapping $\mathbf{A}_1$ and $\mathbf{A}_2$. Now the final optimisation problem can be formulated as

$$\mathbf{A} = \underset{\mathbf{A}}{\mathrm{argmin}}\,\frac{1}{2}\left(\|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F + \|\mathbf{X} - \mathbf{A}^{-1}\mathbf{X}'\|_F\right), \tag{3.3}$$

which is non-convex and nonlinear due to the inverse $\mathbf{A}^{-1}$. Azencot [12] proposes an improved framework

$$\mathbf{A} = \underset{\mathbf{A}_1,\mathbf{A}_2}{\mathrm{argmin}}\,\frac{1}{2}\left(\|\mathbf{X}' - \mathbf{A}_1\mathbf{X}\|_F + \|\mathbf{X} - \mathbf{A}_2\mathbf{X}'\|_F\right) \;\text{ s.t. }\; \mathbf{A}_1\mathbf{A}_2 = \mathbf{I},\; \mathbf{A}_2\mathbf{A}_1 = \mathbf{I}, \tag{3.4}$$

to circumvent some of the optimisation difficulties in (3.3). More recently, Askham and Kutz [11] introduced the *optimised DMD* algorithm, which significantly mitigates the bias due to noise by utilising a variable projection method for nonlinear least squares to compute the Dynamic Mode Decomposition for unevenly timed samples. The optimised DMD algorithm solves the exponential fitting problem directly:

$$\underset{\boldsymbol{\omega},\mathbf{\Phi_b}}{\mathrm{argmin}}\,\|\mathbf{X} - \mathbf{\Phi_b}\mathbf{T}(\boldsymbol{\omega})\|_F. \tag{3.5}$$

Due to its ability to optimally suppress bias and handle snapshots collected at arbitrary times, this has been shown to provide a superior decomposition. But, the disadvantage of optimised DMD is tits nonlinear optimisation problem. However, the optimised DMD algorithm can be stabilised by using statistical bagging methods, and the *boosted optimised DMD* (BOP-DMD) method can not only improve the performance of the decomposition but also produce UQ metrics for the DMD eigenvalues and DMD eigenmodes [17]. This is a nearly optimal linear model for predicting dynamics.

4 Examples

4.1 Stochastic dynamics

Consider a system with stochastic dynamics

$$z_{k+1} = \lambda z_k + n_k, \qquad z_k \in \mathbb{R} \tag{4.1}$$

Let n_k be white noise with variance $\sigma^2 = 10$ and $\lambda = 0.5$ be decay rate. Figure 4.1 (left) shows a typical trajectory for an initial condition $z_0 = 0$. By applying DMD, we estimate a decay rate $\tilde{\lambda} = 0.55$, despite the fact that the nominal (noiseless) trajectory is simply given by $z_k = 0$. A global, linear analysis of the trajectory 4.1 (left) would identify a stationary process ($\tilde{\lambda} = 0$).

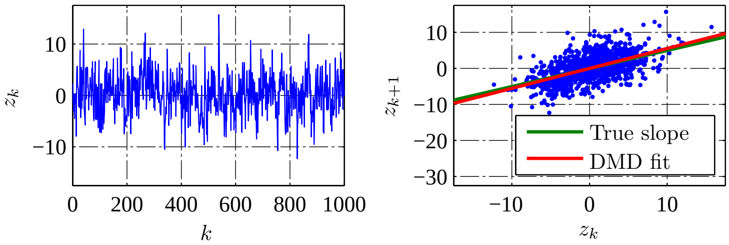


Figure 4.1: From [4], (Left) Typical trajectory of the noisy one-dimensional system (4.1). (Right) Scatter plot showing the correlation of z_{k+1} and z_k. The relationship between future and past values of z is identified by DMD even though the dataset is rank-deficient.

As for rank-deficient data, least-squares fit is used by DMD to approximate the dynamics relating X and Y. So it is no wonder that we can accurately estimate λ from this time series.

4.2 Standing waves

DMD is often used to analyze oscillatory behaviour, irrespective of the linearity of underlying dynamics as each DMD mode has a corresponding DMD eigenvalue and thus a corresponding growth rate and frequency. Consider data describing a standing wave:

$$z_k = \cos(k\theta)q, \qquad k = 0, \ldots, m, \tag{4.2}$$

where q is a fixed vector in $\mathbb{R}^n$. For instance, such data can arise from the linear system

$$\begin{aligned} u_{k+1} &= (\cos\theta)u_k - (\sin\theta)v_k \\ v_{k+1} &= (\sin\theta)u_k + (\cos\theta)v_k \end{aligned} \qquad (u_0, v_0) = (q, 0), \tag{4.3}$$

where $(u_k, v_k) \in \mathbb{R}^{2n}$. By measuring only the state u_k, we observe the standing wave (4.2). Such behavior can also arise in nonlinear systems, for instance by measuring only one component of a multi-dimensional limit cycle.

Suppose we compute DMD modes and eigenvalues from data satisfying (4.2). By construction, a single vector q will span the columns of the data matrix X. As such, the SVD of X will generate a matrix U with a single column, and the matrix $\tilde{A}$ will be 1×1. Hence, there will be exactly one DMD eigenvalue λ. λ is real since z is real-valued, meaning it captures only exponential growth/decay, and no oscillations. This is even though the original data are known to oscillate with a fixed frequency. So, this conclusion is incorrect.

It turns out that, the data is not linearly consistent in this example. To see this, let x and y be vectors with components $x_k = \cos(k\theta)$ and $y_k = \cos((k+1)\theta)$ (for $k = 0, \ldots, m-1$), then the data matrices become

$$X = qx^T, \qquad Y = qy^T.$$

Now, unless θ is a multiple of π, X and Y are not linearly consistent.

For instance, the vector $a = (-\cos\theta, 1, 0, \ldots, 0)$ is in $\mathcal{N}(X)$, since $x^Ta = 0$. However, $y^*a = -\cos^2\theta + \cos 2\theta = \sin^2\theta$, so $a \notin \mathcal{N}(Y)$ unless $\theta = j\pi$. If $\theta = \pi$, then the columns of X simply alternate sign, and in this case, DMD yields the (correct) eigenvalue of -1. As such, even though the data in this example arise from the linear system (4.3), by Theorem 1.1 there is no A such that $Y = AX$. So, DMD fails to capture the correct dynamics.

5 Extensions

A significant advantage of DMD is its simple framing in terms of linear regression. It does not require knowledge of governing equations. For this reason, DMD has been rapidly extended to include several techniques and has been widely applied beyond the filed it originated from, fluid dynamics. Some leading algorithmic extensions are as follows

Combining multiple trajectories Experimental data is typically noisy. Hence, while phase averaging or filtering can be done to eliminate undesirable noise prior to DMD analysis, as the noise may remove features of the true dynamics. These effects of noise can be averaged out by combining multiple trajectories in a single DMD computation.

Including inputs and control Often, the goal of obtaining reduced-order models is to manipulate the behaviour of the system by designing effective controllers. Similarly, in many systems, such as the climate, there are external forcing variables that make it challenging to identify the underlying unforced dynamics. Proctor [3] introduced the DMD with control (DMDc) algorithm to disambiguate the natural unforced dynamics and the effect of actuation This algorithm results in another linear regression problem as follows

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{5.1}$$

Compression and randomized linear algebra Randomized algorithms are designed to exploit DMD patterns to accelerate numerical linear algebra. In the randomized DMD algorithm [13] data is randomly projected into a lower-dimensional subspace where computations may be performed more efficiently. Patterns also facilitate more efficient measurement strategies based on principles of *sparsity* to reduce the number of measurements required in time and space. This can potentially enable high-resolution characterization of systems from under-resolved measurements [15].

Multiresolution DMD is often applied to complex, high-dimensional dynamical systems, such as fluid turbulence or epidemiological systems, exhibiting multiscale dynamics in space and time. Many multiscale systems exhibit transient or intermittent phenomena, such as the El Niño observed in global climate data. These transient dynamics are not captured accurately by DMD, which seeks spatiotemporal modes that are globally coherent across the entire time series of data. To address this challenge, the multi-resolution DMD (mrDMD) algorithm was introduced [5], which effectively decomposes the dynamics into different timescales, isolating transient and intermittent patterns.

6 Applications

Fluid dynamics DMD has been applied to a wide range of flow geometries (jets, cavity flow, wakes, channel flow, boundary layers, etc.), to study mixing, acoustics, and combustion, among other phenomena.

Epidemiology Epidemiological data often consists of high-dimensional spatiotemporal time series measurements, such as the number of infections in a given neighbourhood or city. Proctor and Eckhoff [6] showed that the DMD provides particularly interpretable decompositions for such systems. Here, modal frequencies often correspond to yearly or seasonal fluctuations and the phase of DMD modes gives insight into how disease fronts propagate spatially, potentially informing future intervention efforts.

Video processing DMD can also be used for segmentation in video processing, as a video is also a time series of high-dimensional snapshots (images) evolving in time. Separating foreground and background objects in video is common in surveillance applications. Increasing video resolutions keeps worsening the real-time separation problem.

7 Limitations

Traveling Waves DMD is based on the SVD of a data matrix $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\tilde{\mathbf{V}}$ whose columns are spatial measurements evolving in time. Here, the SVD is a space–time separation of variables into spatial modes, given by the columns of $\mathbf{U}$, and time dynamics, given by the columns of $\mathbf{V}$. As in POD, DMD thus has limitations for problems that exhibit travelling waves, where separation of variables is known to fail.

Transients Transients and intermittent phenomena characterize many systems of interest. The multi-resolution DMD has been proposed to identify these events. However, it is still necessary to formalize the window size to compute DMD and the choice of relevant timescales.

Strong Nonlinearity DMD can accurately identify an approximate linear model for linear, periodic, or quasi-periodic dynamics. However, DMD cannot capture a linear dynamical system model with essential nonlinear features, such as multiple fixed points, unstable periodic orbits, or chaos. For example, DMD trained on data from the chaotic Lorenz system will fail to yield a reasonable linear model, and the resulting DMD matrix will also not capture essential features of the linear portion of the Lorenz model. The Sparse Identification of Nonlinear Dynamics (SINDy) [7] identifies fully nonlinear dynamical systems models from data but it has its own limitations.

Part III

Sparse Identification of Nonlinear Dynamics (SINDy)

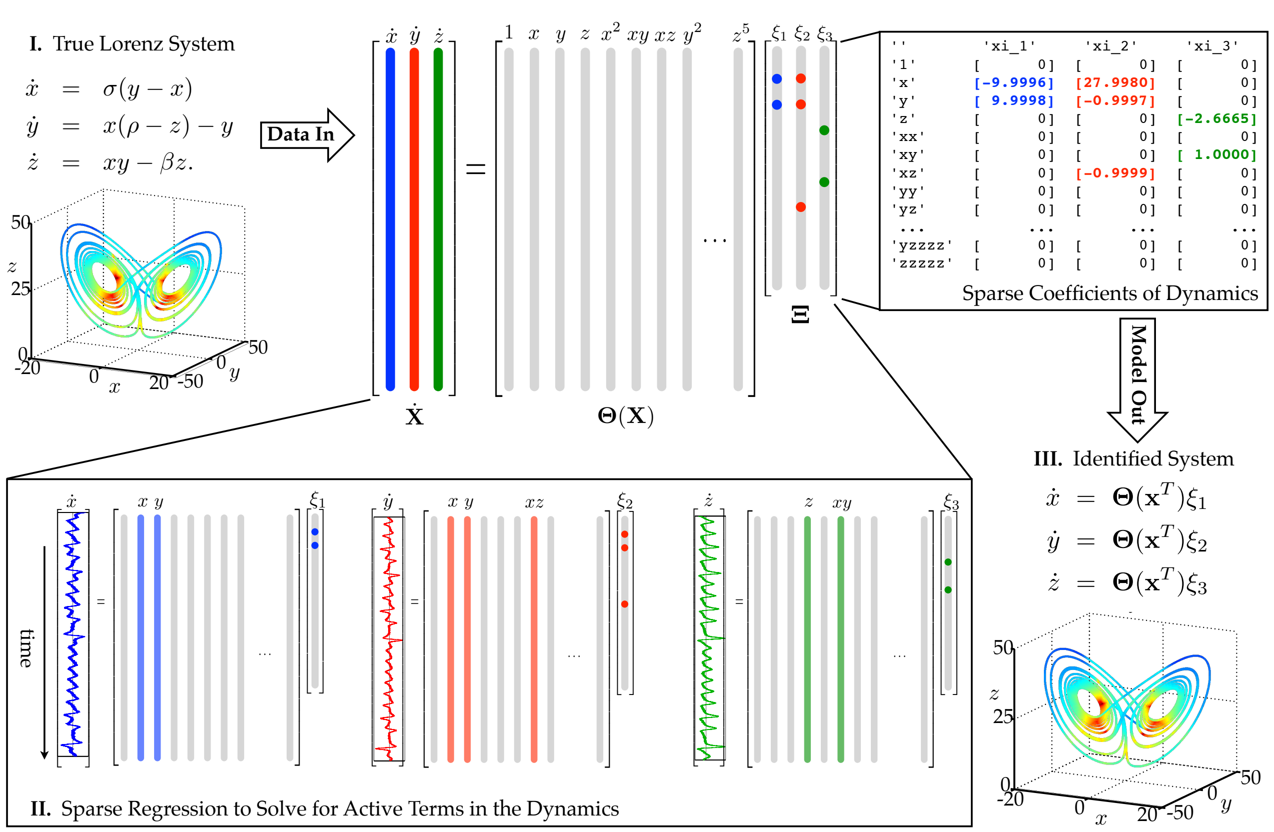
Discovering dynamical systems models from data is a central challenge in physics, with a rich history going back to the time when Kepler and Newton discovered the laws of planetary motion. The automated discovery of governing equations and dynamical systems is a new and exciting scientific paradigm with increasing computational power and vast data.

The sparse identification of nonlinear dynamics (SINDy) algorithm [7] bypasses the intractable combinatorial search through all possible model structures, and uses the fact that many dynamical systems have dynamics $\mathbf{f}$ with only a few active terms in the space of possible right-hand side functions as shown in 1.1.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \mathbf{f}(\mathbf{x}) \tag{0.1}$$

This has been possible due to recent advances in compressed sensing and sparse regression. It also allows the dynamics to vary concerning bifurcation parameters $\boldsymbol{\mu} \in \mathbb{R}^q$.

1 The Setup


I. True Lorenz System
$\dot{x} = \sigma(y - x)$
$\dot{y} = x(\rho - z) - y$
$\dot{z} = xy - \beta z.$
50
z
25
0
-20
0
x
20
-50
0
y
50
Data In
$\dot{x}$ $\dot{y}$ $\dot{z}$
=
1 x y z x^2 xy xz y^2 z^5
ξ_1 ξ_2 ξ_3
…
$\dot{\mathbf{X}}$
$\boldsymbol{\Theta}(\mathbf{X})$
$\boldsymbol{\Xi}$
Model Out
III. Identified System
$\dot{x} = \boldsymbol{\Theta}(\mathbf{x}^T)\xi_1$
$\dot{y} = \boldsymbol{\Theta}(\mathbf{x}^T)\xi_2$
$\dot{z} = \boldsymbol{\Theta}(\mathbf{x}^T)\xi_3$
50
z
25
0
-20
0
x
20
-50
0
y
50
time
$\dot{x}$ = x y … ξ_1
$\dot{y}$ = x y xz … ξ_2
$\dot{z}$ = z xy … ξ_3
II. Sparse Regression to Solve for Active Terms in the Dynamics

```
''          'xi_1'      'xi_2'      'xi_3'
'1'       [      0]  [      0]  [      0]
'x'       [-9.9996]  [27.9980]  [      0]
'y'       [ 9.9998]  [-0.9997]  [      0]
'z'       [      0]  [      0]  [-2.6665]
'xx'      [      0]  [      0]  [      0]
'xy'      [      0]  [      0]  [ 1.0000]
'xz'      [      0]  [-0.9999]  [      0]
'yy'      [      0]  [      0]  [      0]
'yz'      [      0]  [      0]  [      0]
 ...           ...        ...        ...
'yzzzz'   [      0]  [      0]  [      0]
'zzzzz'   [      0]  [      0]  [      0]
```

Sparse Coefficients of Dynamics

Figure 1.1: From [7], Schematic of the SINDy algorithm, demonstrated on the Lorenz equations

Collect a time-history of the state $\mathbf{x}(t)$ and its derivative $\dot{\mathbf{x}}(t)$ sampled at several instances in time $t_1, t_2, \cdots, t_m$ to determine the form of the function $\underline{f}$ from data. These data are then arranged into two large matrices:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \begin{array}{cc} \xrightarrow{\text{state}} & \\ \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} & \Bigg\downarrow \text{time} \end{array} \tag{1.1}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix}. \tag{1.2}$$

Next, we construct an augmented library $\mathbf{\Theta}(\mathbf{X})$ consisting of candidate nonlinear functions of the columns of $\mathbf{X}$. Say for example, $\mathbf{\Theta}(\mathbf{X})$ may consist of constant, polynomial and trigonometric terms

$$\mathbf{\Theta}(\mathbf{X}) = \begin{bmatrix} \vert & \vert & \vert & \vert & & \vert & \vert & \vert & \vert & \\ \mathbf{1} & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \sin(2\mathbf{X}) & \cos(2\mathbf{X}) & \cdots \\ \vert & \vert & \vert & \vert & & \vert & \vert & \vert & \vert & \end{bmatrix}. \tag{1.3}$$

Here, higher polynomials are denoted as $\mathbf{X}^{P_2}, \mathbf{X}^{P_3}$, etc. For example, $\mathbf{X}^{P_2}$ denotes the quadratic nonlinearities in the state variable $\mathbf{x}$, given by

$$\mathbf{X}^{P_2} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & x_2(t_1)x_3(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^2(t_2) & x_2(t_2)x_3(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & x_2(t_m)x_3(t_m) & \cdots & x_n^2(t_m) \end{bmatrix}. \tag{1.4}$$

Each column of $\mathbf{\Theta}(\mathbf{X})$ represents a candidate function for the right-hand side of (0.1). There is immense freedom of choice in constructing the entries in this matrix of nonlinearities. Due to sparsity, only a few of these nonlinearities are active in each row of $\mathbf{f}$, hence set up a sparse regression problem to determine the sparse vectors of coefficients $\mathbf{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \cdots & \boldsymbol{\xi}_n \end{bmatrix}$ that determine which nonlinearities are active, as in 1.1.

$$\dot{\mathbf{X}} = \mathbf{\Theta}(\mathbf{X})\mathbf{\Xi}. \tag{1.5}$$

Each column $\boldsymbol{\xi}_k$ of $\mathbf{\Xi}$ represents a sparse vector of coefficients determining which terms are active in the right-hand side for one of the row equations $\dot{\mathbf{x}}_k = \mathbf{f}_k(\mathbf{x})$ in (0.1). Once $\mathbf{\Xi}$ has been determined, a model of each row of the governing equations may be constructed as follows:

$$\dot{\mathbf{x}}_k = \mathbf{f}_k(\mathbf{x}) = \mathbf{\Theta}(\mathbf{x}^T)\boldsymbol{\xi}_k. \tag{1.6}$$

Note that $\mathbf{\Theta}(\mathbf{x}^T)$ is a vector of symbolic functions of elements of $\mathbf{x}$, as opposed to the data matrix $\mathbf{\Theta}(\mathbf{X})$. This results in the overall model

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \mathbf{\Xi}^T(\mathbf{\Theta}(\mathbf{x}^T))^T. \tag{1.7}$$

Now, solve for $\mathbf{\Xi}$ in (1.5) using sparse regression.

2 The Algorithm

There are a number of algorithms to determine sparse solutions $\boldsymbol{\Xi}$ to the regression problem in (1.5). Each column of (1.5) requires a distinct optimization problem to find the sparse vector of coefficients $\boldsymbol{\xi}_k$ for the k^{th} row equation.

The matrix $\boldsymbol{\Theta}(\mathbf{X})$ has dimensions $m \times p$, where p is the number of candidate nonlinear functions, and $m \gg p$ since there are more time samples of data than there are candidate nonlinear functions. Realistically, the data $\mathbf{X}$ and $\dot{\mathbf{X}}$ will be contaminated with noise so that (1.5) does not hold exactly. If $\mathbf{X}$ is relatively clean but the derivatives $\dot{\mathbf{X}}$ are noisy, the equation becomes

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi} + \eta\mathbf{Z}, \tag{2.1}$$

where η is the noise magnitude and $\mathbf{Z}$ is a matrix of independent identically distributed Gaussian entries with zero mean. This sparse solution to an overdetermined system with noise is required. An economical model will provide an accurate model fit in (1.5) with as few terms as possible in $\boldsymbol{\Xi}$. Such a model may be identified using a convex $\ell 1$-regularized sparse regression

$$\boldsymbol{\xi}_k = \underset{\boldsymbol{\xi}'_k}{\operatorname{argmin}} \|\dot{\mathbf{X}}_k - \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\xi}'_k\|_2 + \lambda\|\boldsymbol{\xi}'_k\|_1 \tag{2.2}$$

where λ is a sparsity-promoting knob and $\dot{\mathbf{X}}_k$ is the kth column of $\dot{\mathbf{X}}$. Sparse regression, such as LASSO [1] or the sequential thresholded least-squares (STLS) algorithm used in SINDy [7], improves the numerical robustness of this identification for noisy over-determined problems over earlier methods that used compressed sensing.

2.1 Discrete-time representation

This can also be implemented on discrete-time dynamical systems:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k). \tag{2.3}$$

There are a number of reasons to implement (2.3). Many systems are inherently discrete-time systems and it may be possible to recover specific integration schemes used to advance (**??**). The discrete-time formulation also foregoes the calculation of a derivative from noisy data. The data collection will now involve two matrices $\mathbf{X}_1^{m-1}$ and $\mathbf{X}_2^m$:

$$\mathbf{X}_1^{m-1} = \begin{bmatrix} \rule{2em}{0.4pt} & \mathbf{x}_1^T & \rule{2em}{0.4pt} \\ \rule{2em}{0.4pt} & \mathbf{x}_2^T & \rule{2em}{0.4pt} \\ & \vdots & \\ \rule{2em}{0.4pt} & \mathbf{x}_{m-1}^T & \rule{2em}{0.4pt} \end{bmatrix} \quad \text{and} \quad \mathbf{X}_2^m = \begin{bmatrix} \rule{2em}{0.4pt} & \mathbf{x}_2^T & \rule{2em}{0.4pt} \\ \rule{2em}{0.4pt} & \mathbf{x}_3^T & \rule{2em}{0.4pt} \\ & \vdots & \\ \rule{2em}{0.4pt} & \mathbf{x}_m^T & \rule{2em}{0.4pt} \end{bmatrix}. \tag{2.4}$$

The continuous-time sparse regression problem in (1.5) now becomes:

$$\mathbf{X}_2^m = \boldsymbol{\Theta}(\mathbf{X}_1^{m-1})\boldsymbol{\Xi} \tag{2.5}$$

and the function $\mathbf{f}$ is the same as in (1.7).

There is a striking resemblance to dynamic mode decomposition in the discrete setting for (2.3) and for linear dynamics. In particular, if $\boldsymbol{\Theta}(\mathbf{x}) = \mathbf{x}$, so that the dynamical system is linear, then (2.5) becomes

$$\mathbf{X}_2^m = \mathbf{X}_1^{m-1}\boldsymbol{\Xi} \quad \Longrightarrow \quad (\mathbf{X}_2^m)^T = \boldsymbol{\Xi}^T \left(\mathbf{X}_1^{m-1}\right)^T. \tag{2.6}$$

Which seeks a dynamic regression onto linear dynamics $\boldsymbol{\Xi}^T$ equivalent to the DMD. Here $\boldsymbol{\Xi}^T$ is $n \times n$ dimensional, which may be prohibitively large for a high-dimensional state $\mathbf{x}$. Thus, DMD identifies the dominant terms in the eigendecomposition of $\boldsymbol{\Xi}^T$.

2.2 External forcing, bifurcation parameters, and normal forms

Many real-world systems depend on parameters and bifurcations or dramatic changes, may occur when the parameter is varied. The algorithm above is readily extended to encompass these important parameterized systems, allowing for the discovery of normal forms associated with a bifurcation parameter $\boldsymbol{\mu}$. In first step, we append $\boldsymbol{\mu}$ to the dynamics.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}) \tag{2.7}$$

$$\dot{\boldsymbol{\mu}} = \mathbf{0}. \tag{2.8}$$

Now, it is possible to identify the right-hand side $\mathbf{f}(\mathbf{x}; \boldsymbol{\mu})$ as a sparse combination of functions of components in the bifurcation parameter $\boldsymbol{\mu}$ and $\mathbf{x}$. This idea is explained using two examples, the one-dimensional logistic map and the two-dimensional Hopf normal form.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}(t), t) \tag{2.9}$$

$$\dot{t} = 1. \tag{2.10}$$

Time-dependence, known external forcing or control $\mathbf{u}(t)$ may also be added to the dynamics of system. This generalization makes it possible to analyze systems that are externally forced or controlled. Such as climate and financial markets

3 Examples

3.1 Logistic Map: Bifurcations

The logistic map is a classical model that exhibits rapid bifurcations, leading to chaotic trajectories. The dynamics with stochastic forcing η_k and parameter μ are given by

$$x_{k+1} = \mu x_k(1 - x_k) + \eta_k. \tag{3.1}$$

Sampling the stochastic system at ten parameter values of μ, the algorithm correctly identifies the underlying parameterized dynamics, as shown in Fig. 3.1.

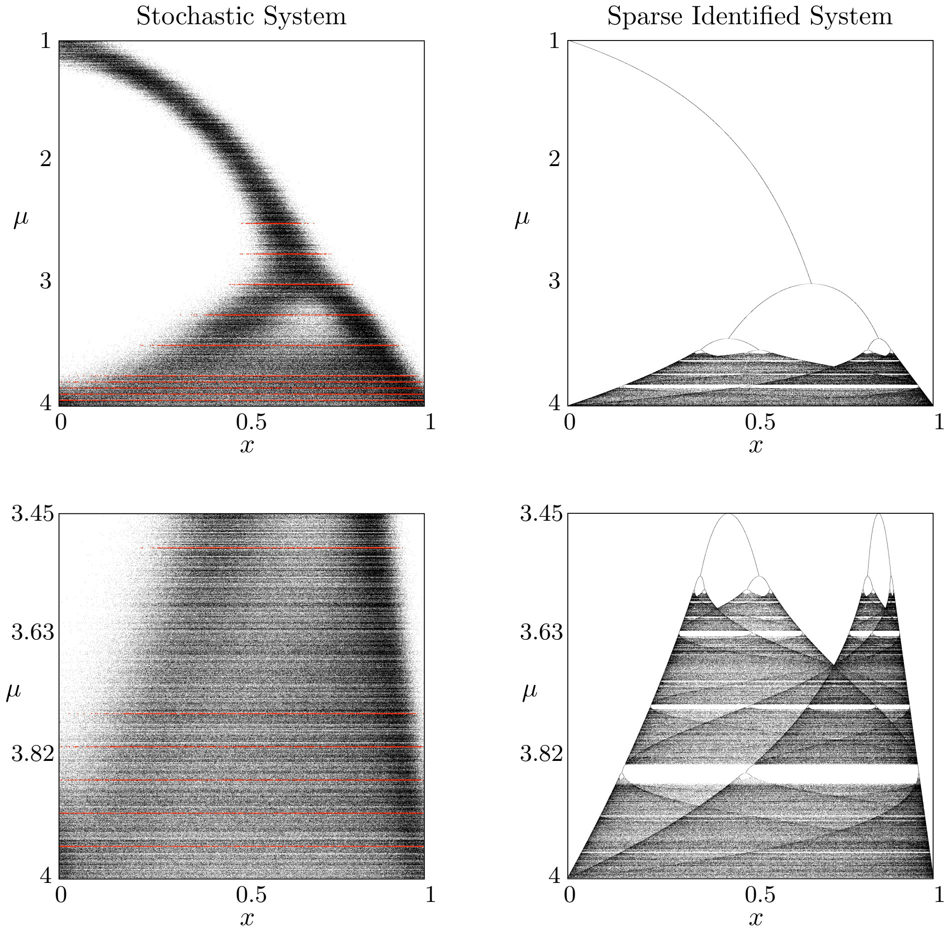


Figure 3.1: From [7], Attracting sets of the logistic map vs. the parameter μ. Data is sampled at rows indicated in red. The forcing η_k is Gaussian with magnitude 0.025.

3.2 Hopf Normal form

Another example to showcase the ability of the sparse dynamics method to identify parameterized normal forms is the Hopf normal form. Here, Noisy data is collected from the Hopf system for various values of the parameter μ.

$$\dot{x} = \mu x + \omega y - Ax(x^2 + y^2) \tag{3.2}$$

$$\dot{y} = -\omega x + \mu y - Ay(x^2 + y^2) \tag{3.3}$$

Data is collected on the blue and red trajectories in Fig. 3.2, and noise is added to simulate sensor noise. The total variation derivative is used to de-noise the derivative for use in the algorithm.

The sparse model identification algorithm correctly identifies the Hopf normal form, with model parameters. The noise-free model reconstruction is shown in Fig. 3.3. Note that although the model terms are correctly identified, with noise in the training data, the actual values of the cubic terms were off by almost 8% so more data collection or reduction of the noise magnitude should be done to improve the model agreement.

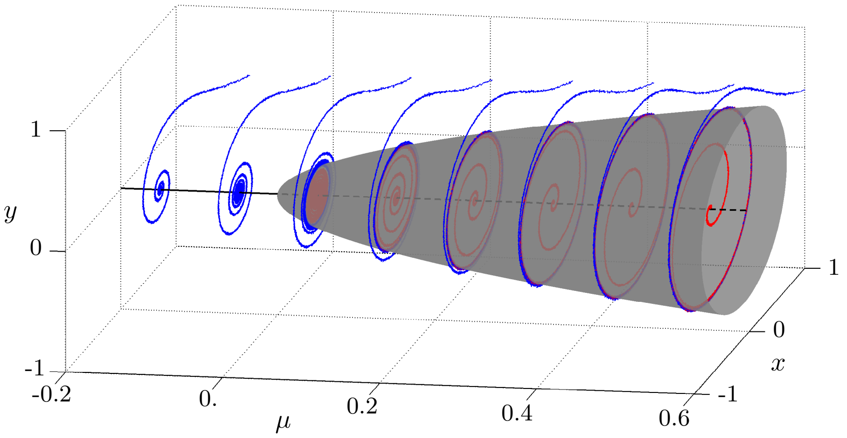


Figure 3.2: From [7], Training data to identify Hopf normal form. Blue trajectories denote solutions that start outside of the fixed point for $\mu < 0$ or the limit cycle for $\mu > 0$, and red trajectories denote solutions that start inside of the limit cycle.

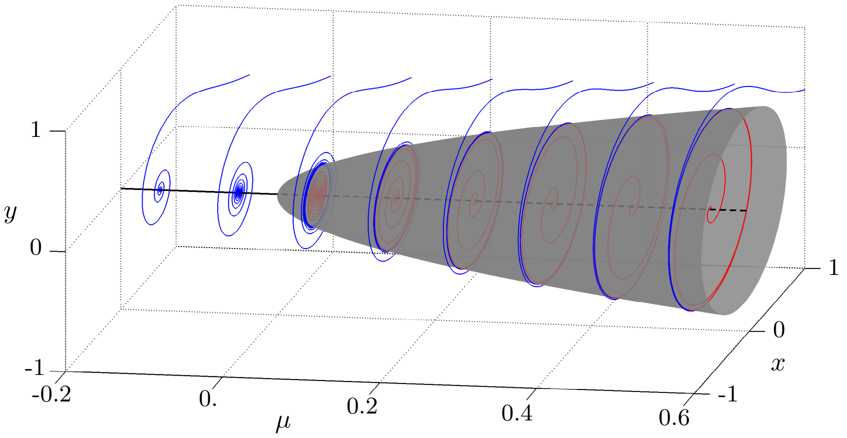


Figure 3.3: From [7], Sparse model captures the Hopf normal form. Initial conditions are the same as in Fig. 3.2

4 Extensions Applications and Limitations

Constrained sparse Galerkin regression Loiseau and Brunton [10] generalised the SINDy framework to incorporate known physical constraints and symmetries in the equations by implementing a constrained sequentially thresholded least-squares optimization. In particular, energy-preserving constraints on the quadratic nonlinearities in the Navier–Stokes equations were imposed to identify fluid systems.

Rational Function Nonlinearities Many dynamical systems, such as metabolic and regulatory networks in biology, contain rational function nonlinearities in the dynamics. these rational function nonlinearities arise due to the separation of timescales. Original SINDy faces issues since general rational functions are not sparse linear combinations of a few basis functions. Kaheman [14] reformulated the dynamics in an implicit ordinary differential equation and modified the optimization procedure accordingly.

Implicit ODEs The optimization procedure above may be generalized to include a larger class of implicit ordinary differential equations in addition to those containing rational function nonlinearities. This can be achieved by updating our library of functions.

Limitations SINDy often faces scaling issues for high-dimensional systems that do not admit a low-dimensional subspace or sub-manifold. Here, the recent linear and nonlinear disambiguation optimization (LANDO) algorithm [16] leverages kernel methods to identify an implicit model for the full nonlinear dynamics, where it is then possible to extract a low-rank DMD approximation for the linear portion linearized about some specified operating condition. This way, the LANDO algorithm robustly extracts the linear DMD dynamics even from strongly nonlinear systems. This was possible by cracking the use of kernels for learning dynamical systems and Koopman representations [15].

References

[1] Robert Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 00359246. URL: http://www.jstor.org/stable/2346178 (visited on 04/10/2023).

[2] PETER J. SCHMID. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656 (2010), pp. 5–28. DOI: 10.1017/S0022112010001217.

[3] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. *Dynamic mode decomposition with control.* 2014. arXiv: 1409.6358 [math.OC].

[4] Jonathan H. Tu et al. *On dynamic mode decomposition: Theory and applications.* 2014. DOI: 10.3934/jcd.2014.1.391. URL: /article/id/1dfebc20-876d-4da7-8034-7cd3c7ae1161.

[5] J. Nathan Kutz, Xing Fu, and Steven L. Brunton. *Multi-Resolution Dynamic Mode Decomposition.* 2015. arXiv: 1506.00564 [math.DS].

[6] Joshua L. Proctor and Philip A. Eckhoff. “Discovering dynamic patterns from infectious disease data using dynamic mode decomposition”. In: *International Health* 7 (2015), pp. 139–145.

[7] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937. DOI: 10.1073/pnas.1517384113. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1517384113.

[8] Scott T. M. Dawson et al. “Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition”. In: *Experiments in Fluids* 57.3 (Feb. 2016). DOI: 10.1007/s00348-016-2127-7. URL: https://doi.org/10.1007%2Fs00348-016-2127-7.

[9] J. Nathan Kutz et al. *Dynamic Mode Decomposition.* Philadelphia, PA: Society for Industrial and Applied Mathematics, 2016. DOI: 10.1137/1.9781611974508. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611974508. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611974508.

[10] Jean-Christophe Loiseau and Steven L. Brunton. *Constrained Sparse Galerkin Regression.* 2016. arXiv: 1611.03271 [physics.flu-dyn].

[11] Travis Askham and J. Nathan Kutz. *Variable projection methods for an optimized dynamic mode decomposition.* 2017. arXiv: 1704.02343 [math.NA].

[12] Omri Azencot, Wotao Yin, and Andrea Bertozzi. *Consistent Dynamic Mode Decomposition.* 2019. arXiv: 1905.09736 [math.NA].

[13] N. Benjamin Erichson et al. “Randomized Dynamic Mode Decomposition”. In: *SIAM Journal on Applied Dynamical Systems* 18.4 (Jan. 2019), pp. 1867–1891. DOI: 10.1137/18m1215013. URL: https://doi.org/10.1137%2F18m1215013.

[14] Kadierdan Kaheman, J. Nathan Kutz, and Steven L. Brunton. “SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476.2242 (Oct. 2020). DOI: 10.1098/rspa.2020.0279. URL: https://doi.org/10.1098%2Frspa.2020.0279.

[15] Steven L. Brunton et al. *Modern Koopman Theory for Dynamical Systems.* 2021. arXiv: 2102.12086 [math.DS].

[16] Peter J. Baddoo et al. “Kernel learning for robust dynamic mode decomposition: linear and nonlinear disambiguation optimization”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 478.2260 (Apr. 2022). DOI: 10.1098/rspa.2021.0830. URL: https://doi.org/10.1098%2Frspa.2021.0830.

[17] Diya Sashidhar and J. Nathan Kutz. “Bagging, optimized dynamic mode decomposition for robust, stable forecasting with spatial and temporal uncertainty quantification”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 380.2229 (June 2022). DOI: 10.1098/rsta.2021.0199. URL: https://doi.org/10.1098%2Frsta.2021.0199.

[18] J. Nathan Kutz Steven L. Brunton. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control 2nd Edition, Kindle Edition.* 2nd ed. Cambridge University Press, 2022. ISBN: 9781009098489,9781009089517. URL: http://gen.lib.rus.ec/book/index.php?md5=506691EE78EF20A2425784847A47DBCB.