# Modeling the Lorenz Attractor with SINDY

Lia Gianfortone

December 2017

# Abstract

# Introduction

System identification techniques that use measured data to recover a system's governing equations are useful for a wide range of applications. The system-identification algorithm proposed in [paper] involves sparse identifications of nonlinear dynamical systems (SINDy). The algorithm relies on the assumption that dynamics of the system depend on only a few linear and nonlinear terms which will guarantee the sparsity of the solution.

# SINDY Algorithm

The algorithm is effective under the assumption that the systems studied with SINDy have dynamics that are depend by only a few terms and are thus sparse in a high-dimensional, nonlinear function space.

We consider systems of the form

$$\frac{d}{dt}\boldsymbol{x}(t) = \dot{x} = \boldsymbol{f}(\boldsymbol{x}(t)) \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state of the system at time $t$ and $\boldsymbol{f}$ consists of the governing equations of the system.

Given $m$ (potentially noisy) measurements of the state $\boldsymbol{x}$ and its derivative $\dot{\boldsymbol{x}}$ over time, we arrange them into data matrices $\boldsymbol{X}$ and $\dot{\boldsymbol{X}}$ as follows

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^T(t_1) \\ \boldsymbol{x}^T(t_2) \\ \vdots \\ \boldsymbol{x}^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix}$$

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

To identify the active terms in the dynamics, we construct the system

$$\dot{\boldsymbol{X}} = \boldsymbol{\Theta}(\boldsymbol{X})\boldsymbol{\Xi} \tag{2}$$

where the state data are input to the library of linear and nonlinear candidate functions, $\boldsymbol{\Theta}(\boldsymbol{X})$, which consists of constant, polynomial, and trigonometric terms that may be chosen based on hypotheses (based on symmetry, physics, etc.) about the system dynamics. [[More about choosing functions]]. The library of functions has the form

$$\boldsymbol{\Theta}(\boldsymbol{X}) = \begin{bmatrix} | & | & | & & | & | & \\ 1 & \boldsymbol{X} & \boldsymbol{X}^{P_2} & \cdots & \sin(\boldsymbol{X}) & \cos(\boldsymbol{X}) & \cdots \\ | & | & | & & | & | & \end{bmatrix} \tag{3}$$

where polynomial cross-terms of degree $i$ are denoted $\boldsymbol{X}^{P_i}$. For example, $\boldsymbol{X}^{P_i}$ contains quadratic nonlinearities in the state $\boldsymbol{x}$,

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

The final term in Eq. (2), $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1 \boldsymbol{\xi}_2 \cdots \boldsymbol{\xi}_n]$ is the desired sparse matrix of coefficients that identifies which of the candidate functions in $\boldsymbol{\Theta}(\boldsymbol{X})$ are active in the system.

Solving for $\boldsymbol{\Xi}$ requires a distinct optimization of each column, corresponding to each of the $n$ dynamical equations that guide the system. Solving for these coefficients requires a distinct optimization for each vector equation,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\Xi}^T(\boldsymbol{\Theta}(\boldsymbol{x}^T))^T. \tag{4}$$

Algorithms to perform the desired sparse regression of Eq. (4) include LASSO, which uses the $L_1$-norm to ensure sparsity, and the sequential least squares method. In this study, sequential least squares was the method of choice.

# Lorenz Attractor

The Lorenz system is an interesting test for the SINDy method. The known dynamics are

$$\dot{x} = \sigma(y - x) \tag{5}$$

$$\dot{y} = x(\rho - z) - y \tag{6}$$

$$\dot{z} = xy - \beta * z. \tag{7}$$

This system demonstrates chaotic behavior under certain conditions and is observably sparse in a Cartesian function space with only two nonlinear terms, $xz$ in $\dot{y}$ and $xy$ in $\dot{z}$.

Data is generated in Matlab using the `ode45` function and the chaotic parameters

$$\sigma = 10 \qquad \beta = 8/3 \qquad \rho = 28. \tag{8}$$

Random Gaussian noise is added to the $\dot{\boldsymbol{X}}$ data to challenge the algorithm. Figure **??** shows the solution generated by `ode45` beside SINDy's interpretation of the system. [[Errors]]

# SINDy for Discrete Systems

The SINDy method can be adapted to determine the dynamics governing discrete systems. Systems

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) \tag{9}$$

are particularly well suited for the SINDy algorithm due to the absence of errors from the measurement or generation of state derivate data. The $m$ data points can be arranged in the two matrices

$$\boldsymbol{X}_1^{m-1} = \begin{bmatrix} \boldsymbol{x_1^T} \\ \boldsymbol{x_2^T} \\ \vdots \\ \boldsymbol{x_{m-1}^T} \end{bmatrix} \boldsymbol{X}_2^{m} = \begin{bmatrix} \boldsymbol{x_2^T} \\ \boldsymbol{x_3^T} \\ \vdots \\ \boldsymbol{x_m^T} \end{bmatrix}. \tag{10}$$

Identifying $\boldsymbol{f}(\boldsymbol{x}_k)$ in Eq. (9) with SINDy consists of selecting a function basis $\boldsymbol{\Theta}(\boldsymbol{x}^T)$ and constructing the relation

$$\boldsymbol{X_2^m} = \boldsymbol{\Theta}(\boldsymbol{X_1^{m-1}})\Xi \tag{11}$$

so that

$$\boldsymbol{f}(\boldsymbol{x}_k) = \Xi^T \boldsymbol{\Theta}(\boldsymbol{x}^T)^T \tag{12}$$

and solving for $\Xi$ with column-by-column optimization.

## Duffing Map

One discrete, nonlinear chaotic relation of interest is the Duffing map,

$$x_{k+1} = y_k \tag{13}$$

$$y_{k+1} = -\beta x_k + \alpha y_k - y_k^3. \tag{14}$$

Choosing $\alpha = 2.75$ and $\beta = 0.2$ guarantees chaotic behavior in the system and our satisfaction in the problem as a test of SINDy for discrete systems. An exact set of solutions is then given by

$$\boldsymbol{f}(\boldsymbol{x}) = \Xi^T \tag{15}$$

## SINDy for Systems with Forcing

## External Forcing

Identifying the dynamics of systems with purely external forcing is a straightforward procedure if the control measurements are known. The control vectors are entered in a data matrix $\boldsymbol{Y}$ as

$$\boldsymbol{Y} = \begin{bmatrix} u(t_1)^T \\ u(t_2)^T \\ \vdots \\ u(t_m)^T \end{bmatrix} \tag{16}$$

and the selection of basis functions in $\boldsymbol{Theta}$ is expanded to include control terms including cross-terms with the state variables. Optimization is then performed on the columns of

$$\dot{\boldsymbol{X}} = \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{Y})\boldsymbol{\Xi} \tag{17}$$

to determine a sparse $\Xi$ that identifies the dynamics of the system regardless regardless of the control that is acting on it, so long as the control is itself not a function of the state.

## Duffing Equation

The Duffing equation describes a second order nonlinear differential equation with external and sinusoidal forcing. The equation models damped and driven oscillators including the motion of a classical particle in a double well potential. The equation is

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma\cos(\omega t). \tag{18}$$

Systems governed by the Duffing equation will exhibit chaotic behavior for certain values of $\alpha$, $\beta$, $\gamma$, and $\delta$. Up until this point, we have not considered using SINDy to determine coefficients in higher order differential equations that govern a measurable system. This is no obstacle as we recall that any (??) $p$-th order differential equation can be separated in to $p$ first order differential equations. As such, the Duffing equation can be rewritten as

$$\dot{x} = v \tag{19}$$

$$\dot{v} = x - x^3 - \gamma v + \delta\cos(\omega t). \tag{20}$$

The dynamics can then be recovered with

$$[\dot{\boldsymbol{X}}\ \dot{\boldsymbol{V}}] = \boldsymbol{\Theta}([\boldsymbol{X}\ \boldsymbol{V}], \boldsymbol{Y})\boldsymbol{\Xi}. \tag{21}$$

## Lorenz Equations with Forcing

## SINDyC Algorithm

## Possible Problems

- wrong candidate functions - coefficients of dynamics are smaller than S.L.S. threshold, $\lambda$ - noisiness of data

Given that computation of $\boldsymbol{\Theta}(\boldsymbol{X})$ grows factorially with $n$, the dimension of the state, this method is less preferable than dynamic mode decomposition (DMD) which uses single value and eigen- decomposition to identify the normal modes of linear systems. Thus if a system is expected or known to be linear, the DMD method would likely be a better choice of system identification.

## 0.1 Further Study

# Conclusion

# Bibliography

Code

Figures