Sparse Identification of Nonlinear Dynamical Systems

Chandramouli Reddy

December 12, 2024

1 Motivation

With progression in time, dynamical processes leave a trace of data behind. The apprehension and interpretation of such data, if not infeasible, is much too cumbersome and error-prone if left solely to the devices of humans. Together with the advances in machine learning and data science, and in the spirit of model discovery, a myriad of algorithms have come to light to aid us in data analysis. Among these, there are algorithms devised to discover the governing equations of non-linear dynamical systems from data. In this report, we present and summarize the workings of one such algorithm, SINDy, an algorithm that strives to reach this aim, featuring parsimony all the while promising a balance between accuracy and complexity of the discovered model.

2 Background

Sparse identification of nonlinear dynamical systems, or SINDy for short, is a machine learning optimization algorithm, that runs to deliver the set of unknown governing equations of a dynamical system based on noisy time-series data[2]. The dynamical system in question is considered to be of the following form

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

where $x(t)$, represents the states of our system at a given time t, while $f(x(t))$ are the dynamical constraints of the system in question, and $\dot{x}(t)$ are the state derivatives which can be accounted for either by direct differentiation or, numerical approximation, and the history of system states can be represented in matrix form.

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

We subsequently construct another matrix, Θ. This would be our library of potential candidate functions that can properly be mapped to the final equation terms. The selection of these candidate functions is completely reliant on us. We have the freedom to accommodate various function bases, that being said, however, for the algorithm to yield the closest result, it is crucial to develop a physics-informed idea when selecting these candidate functions.

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{X} & \mathbf{X^2} & \mathbf{X^3} & \sin(\mathbf{X}) & \cos(\mathbf{X}) \cdots \end{bmatrix}$$

SINDy is based on sparse regression, and hence, it contains another matrix. Specifically, a sparse matrix of coefficients, ζ, which in turn, weighs the candidate function entries from Θ that offer the closest mapping to the governing equations' terms. It is very much analogous to the classical $A = xb$ matrix equation, specifically taking the following form,

$$X = \Theta.\zeta \tag{2}$$

Where we intend to solve for ζ simply by,

$$\zeta = \Theta^{-1}.X$$

It is at this stage that the quality of parsimony comes to light. With a simple matrix multiplication, we can confidently determine the coefficient of each term on the left-hand side. This renders SINDy, an attractive option to consider. That being said, there are several caveats that one needs to bear in mind to reach the most optimal solution.

2.1 Challenges

The efficacy of SINDy is reliant on several factors, which if not carefully handled could yield an undesired result. The dimensionality of the dynamics, noisiness of data, and unsuitable selection of candidate functions can all play a noticeable role in deviating from the correct solution.

Data Oftentimes, we only have $X(t)$ at our disposal, and $\dot{X}(t)$ requires a numerical approximation. Both $X(t)$ and $\dot{X}(t)$ are prone to noise, and ought to be filtered before using them in the calculation. Section 3.1 presents algorithms one could employ to filter out the noise.

High-order Dimensionality In scenarios where the physical system is described in higher dimensions and/or is multivariate, the discretization grid would consequently grow in size, which upon selecting each candidate function in Θ for every state $X(t)$, would computationally be of a factorial order. Therefore, it is pertinent to reduce dimensionality using techniques such as Singular Value Decomposition (SVD).

Candidate functions The selection of this library of functions is at the discretion of the expert using this algorithm. Although it offers the freedom for one to enlist functions, it is important to make a physics-informed decision suitable for the system in question. For instance, when dealing with a fluid flow system, it would be appropriate to consider functions up to quadratic order. That being said, having the freedom to select elements of $\Theta(t)$, one would have the option to verify and diagnose the relationship between the coordinates and coefficients by varying the basis of the function space.

Correct coordinates In circumstances where we are partially aware of the coordinates' correctness and/or have only a limited number of variables available, one could employ time-delay coordinates from the time-series data.

3 Algorithm

A sequential thresholded least squares algorithm has been used for sparse regression. The algorithm starts with a least squares solution to (2) (solving for ζ), and thresholding all the values less than an appropriate thresholding value λ to zero. A new least squares problem is now identified for the nonzero indices and the new coefficients are thresholded again. This process is continued until the nonzero coefficients converge. The algorithm is remarkably efficient and the solution converges in small iterations. The following snippet shows the pseudo-code of this procedure.

```

z = theta\dxdt # coefficient matrix

for k=1:10

    s_index = z < lambda; # extract indices of elements smaller than lambda
    z[index] = 0 # threshold

    for i = 1:n
        b_index = not(s_index(:,i)) # excluding s_index

    z(b_index,i) = theta(:,b_index)\dxdt(:,i)
```

3.1 Denoising

The numerical derivative of a noisy signal significantly amplifies the fluctuations, making it virtually impossible for the algorithm to describe the dynamics of the system accurately even for a small amplitude of noise. Thus, making it an important step to filter the noise from the first derivative while retaining the information from the original signal that characterizes the system.

Variation regularization [1] The author's preferred choice of denoising the derivative, is an optimization approach that aims at penalizing the random fluctuations caused by the noise while withholding all possible information from the original signal. The optimization function is defined as follows:

$$\hat{x} = \arg\min_{x}\{1/2\|y-x\|_2^2 + \lambda \mathrm{R}(x)\} \tag{3}$$

where

$$\|y-x\|_2^2$$

is the data fidelity term that denotes the difference between the original and noisy, thus retaining the information from the original system. $R(x)$ is the Regularization term that is defined as $\mathrm{R}_{\mathrm{TV}}(x) = \|\nabla x\|_1$ penalizing the fluctuations. however, it is essential to normalize the two terms to be consistent with different sample sizes and define a suitable parameter λ.

Fourier transform thresholding This technique, often referred to as spectral analysis, allows us to decompose a time-varying signal into its constituent frequency components[3]. By applying the Fast Fourier Transform (FFT) algorithm, we can efficiently compute the amplitude and phase of these frequency components. plotting the power spectrum of the data then separates the random frequencies from the relevant ones. The data is then thresholded appropriately eliminating the noise and retaining the relevant information. The data is then transformed back into the time domain. Allows for significant savings in computational costs compared to the optimization.

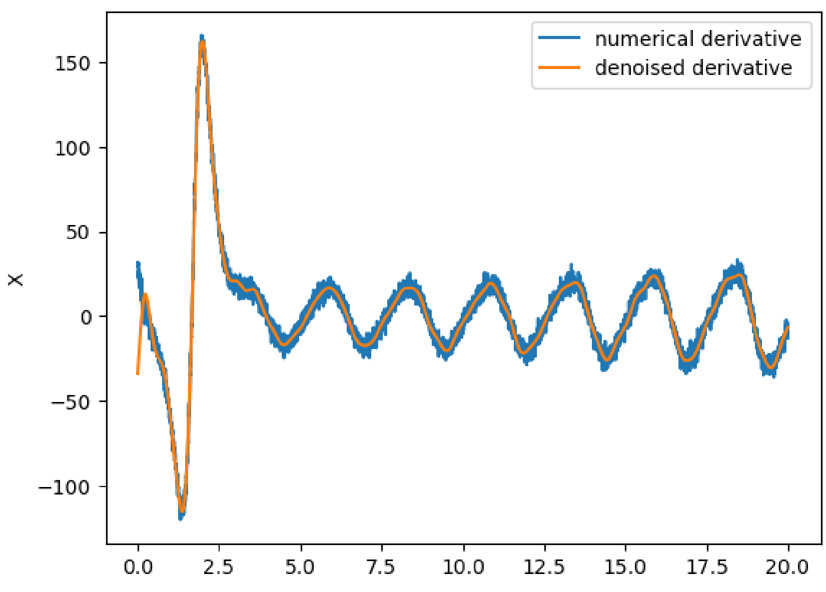

Figure 1: Denoised first derivative using Fourier transform thresholding

4 Examples

This section exemplifies the application of SINDy on two different systems, namely, a simple harmonic oscillator and the Lorenz system.

4.1 Harmonic Oscillator

The states of a harmonic oscillator (i.e., $x(t)$) would be the position of the mass, $q(t)$, at any time t. This, in turn, would imply momentum of the mass, $p(t)$, would represent state differentiation at any time t. Knowing the equation of a simple harmonic oscillator, $f(x) = -Kx$, we can numerically approximate the momentum at any time t. We used velocity verlet scheme for this purpose. Upon calculation of position at every time step, some magnitude of random Gaussian noise is augmented. By varying the noise intensity we can gauge the integrity of SINDy's solution. When constructing Θ, we need to take into account the function basis of the final solution. Here we applied SINDy with polynomial and trigonometric functions of q and p as the function bases.

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{q} & \mathbf{p} & \mathbf{cos(q)} & \sin(\mathbf{q}) & \mathbf{cos(p)} & \sin(\mathbf{p}) \cdots \end{bmatrix}$$

After using the algorithm and reconstructing the terms from ζ's entries, we found that SINDy is very robust to high amplitudes of noise. All the values in ζ have been reduced to zero except for the one corresponding to p itself.

4.2 Lorenz Attractor

The Lorenz system stands as a good example to test how well SINDy can arrive at the correct solution for non-linear equations. A Lorenz system with the following governing equations,

$$\dot{x} = \sigma(x - y) \tag{4}$$

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

$$\dot{z} = xy - \beta z \tag{6}$$

For this example, we use the standard parameters $\sigma = 10$, $\beta = 8/3$, $\rho = 28$ and initial conditions $[x, y, z]^T = [-8, 7, 27]^T$, the data is generated for $t = 0$ to $t = 20$ with a timestep $\Delta t = 0.001$.

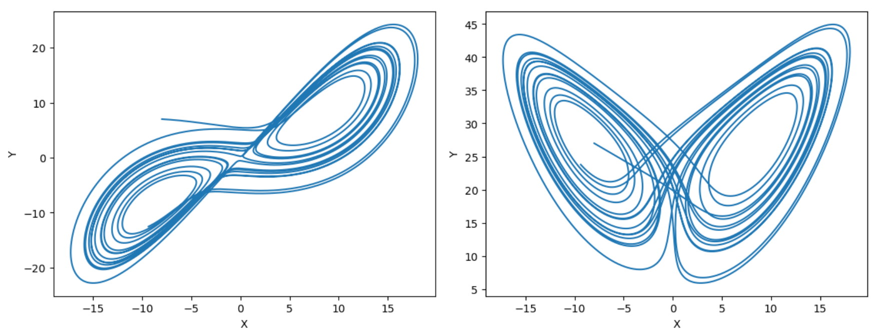

Figure 2: Planar view of Chaotic Lorenz system

Analogous to what we did for the harmonic oscillator, we add random Gaussian noise to each step of the trajectory, and numerically approximate the derivatives corresponding to each state variable. With a function library primarily comprised of polynomials, after denoising and applying the algorithm we arrived at the following matrix of coefficients for $\eta = 0.01$,

$$\zeta = \begin{bmatrix} 0 & 0 & 0 \\ -9.94492931 & 27.98524914 & 0 \\ 9.94418517 & -0.99973869 & 0 \\ 0 & 0 & -2.69590081 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.99835793 \\ 0 & -0.99947343 & 0 \\ 0 & 0 & 0 \\ '' & '' & '' \\ 0 & 0 & 0 \end{bmatrix}$$

The result is by no means perfect. It has some noise and slight discrepancy, but it comes very close to the true solution, otherwise.

5 Results

The data simulating the Chaotic Lorenz system has been generated using the Runge Kutta scheme with added Gaussian random noise. The first derivative is calculated numerically. Total variation regularization has been applied to denoise the first derivative. Although the data is denoised, a good trade-off between reducing the noise and holding the original information has not been reached. The fidelity term changed much faster than the regularization term as the sample size changed. This necessitated normalization of both the terms and finding a suitable parameter λ. The Fourier transform thresholding however yielded the desired output in this example.

Polynomial functions of the order x through x^5 have been used as the dictionary of basis functions. The robustness of the algorithm has been checked at different noise levels ($\eta = 0.01, ...10$). The coefficients have been successfully reduced to zero except for the constant functions, which encountered reasonably small errors at higher noise amplitudes. The system is reconstructed as $\Theta^T.\zeta$, where Θ is the dictionary of the basis functions and not the data matrix itself. The system is then regenerated using the same initial conditions as the original data and the same integration scheme.

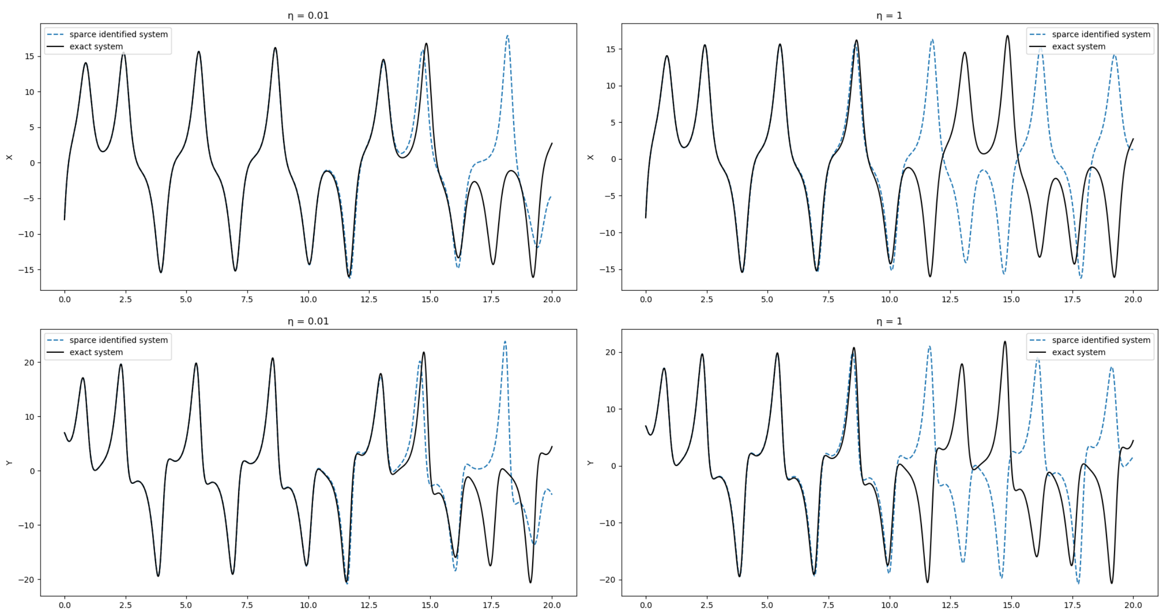

Figure 3: Dynamo view of trajectories of the Lorenz system for the illustrative case where x and $\dot{x}$ are measured with noise

The $L2$ error in terms of X rapidly grows after a period of time even at low noise amplitudes, as is the case with the authors, however, this is attributed to the highly chaotic nature of the Lorenz attractor and the numerical integration scheme and it can be concluded that the SINDy has indeed captured the dynamics of the system accurately at all noise amplitudes.

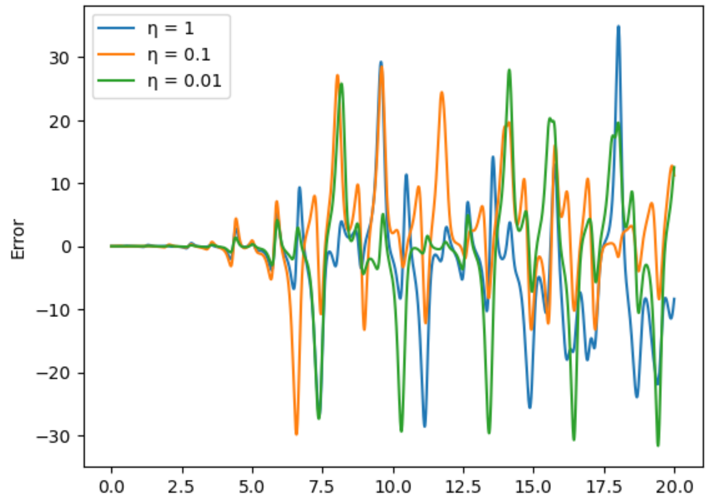

Figure 4: error vs time for sparse identified systems generated from noisy data

References

[1] Fan Zhang Linwei Fan. "Brief review of image denoising techniques". In: *Visual Computing for Industry, Biomedicine, and Art* (2019). DOI: https://doi.org/10.1186/s42492-019-0016-7..

[2] Joshua L. Proctor Steven L. Brunton and J. Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *PNAS* 113.15 (2016), pp. 3932–3937.

[3] Dong Li Yinjie Lin Ping Tan. "An FFT-based beam profile denoising method for beam profile distortion correction". In: *Nuclear Inst. and Methods in Physics Research* (2023).