Model Selection for Infectious Disease Dynamics

April Zhi Zhou

March, 2020

1 Introduction

Brunton et al. introduced the method of sparse identification of nonlinear dynamics (SINDy) to recover the governing equations of dynamical systems given time series data. The method can also be used for model construction with experimental data. SINDy determines the nonlinear dynamics terms by conducting sparse regression on a library of nonlinear functions. To recover/construct a n-dimensional dynamical system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$ given $\mathbf{x}(t)$ and $\mathbf{x}'(t)$ at $t = t_1, t_2, \cdots, t_m$, SINdy constructs matrices $\mathbf{X}, \dot{\mathbf{X}} \in \mathbb{R}^{n\times m}$ and builds a library $\mathbf{\Theta}(\mathbf{X})$ consisting candidate nonlinear functions of the columns of $\mathbf{X}$. The sparse vector of coefficients are determined by solving the sparse regression problem: $\dot{\mathbf{X}} = \mathbf{\Theta}(\mathbf{X})\mathbf{\Xi}$. Then, the governing equations of the dynamical system is constructed to be $\dot{\mathbf{x}}_k = \mathbf{f}_k)(\mathbf{x}) = \mathbf{\Theta}(\mathbf{x}^T)\xi_k$, for $k = 1, 2, \cdots, n$, where ξ_k denotes the k-th column of $\mathbf{\Theta}$.

In this report, we tested the SINDy model selection algorithm on a variety of infectious disease models, namely, the SIR model with constant transmission rate, noisy transmission rate, and decreasing transmission rate. Also, we collected data on the reported COVID-19 infection, recovery and death in Hubei province, China, and used SINDy to construct a model for the disease spreading process.

2 SIR Systems

Infectious diseases are a major cause of death in human history. A disease with a high transmission rate is likely to become a pandemic due to its ability to affect a large number of people in a relatively short time. COVID-19 is a very good example. A few months after its first emergence in November, 2019, COVID-19 has spread to at least 114 countries in the world. The SIR (susceptible-infected-removed) model is a mathematical model that studies the disease spreading dynamics. It is given as follows:

$$\dot{S} = -\beta IS \tag{1}$$

$$\dot{I} = -\beta IS - \gamma I \tag{2}$$

$$\dot{R} = \gamma I, \tag{3}$$

where S, I, and I represent the ratio of susceptible, infected, removed populations to the total population, respectively, β denotes the transmission (infection) rate and γ denotes the removal rate. We make the assumption that the total population remains unchanged, that is, $S + I + R = 1$.

To test SINDy on the basic SIR model, we solved equations (1) - (2) with $S(0) = 0.9, I(0) = 0.1$ $\beta = 1.1$ and $\gamma = 0.5$ using MATLAB function `ode45`, and sampled data from the numerical solutions. Using polynomial of degrees 1 to 3 for the library of candidate nonlinear functions, SINDy was able to correctly determine the coefficients β and γ. In Figure 1, we plotted the numerical solutions of the SIR model and the model recovered by SINDy. The dynamics appeared to be well captured but the algorithm is sensitive to noise in the derivatives.

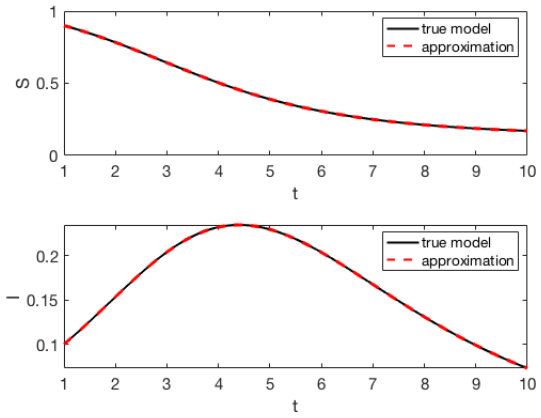

Figure 1: Numerical solution to the SIR model with $\beta = 1.1, \gamma = 0.5$ compared with the numerical solution to the model recovered by SINDy

2.1 SIR systems with randomly perturbed transmission rate

In real life, it is unlikely that the disease transmission rate remains constant. In this section, we consider transmission rates that are perturbed by normally distributed noises. In Figure 2, we plotted the randomly perturbed transmission rate $\beta(t) = 1.1 + \epsilon_t$, with $\epsilon \sim \mathcal{N}(0, \sigma^2), \sigma^2 = 0.2$ and $\sigma^2 = 0.5$. To construct time series data, we solved the SIR system with $\beta = \beta(t)$ using the forward Euler's method. Since random noise is added to β and the system is no longer deterministic, SINDy was unable to recover the nonlinear terms in SIR governing equations. Though, it was able to construct a new model that captures the disease evolution accurately (see Figure 3).

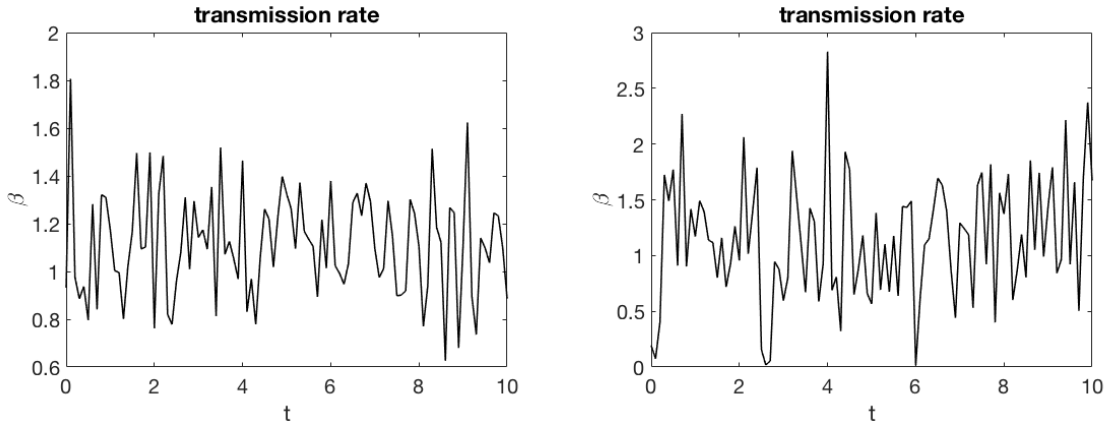

Figure 2: Transmission rate $\beta = 1.1 + \epsilon_t, \epsilon \sim \mathcal{N}(0, \sigma^2)$, with $\sigma^2 = 0.2$ (left) and 0.5 (right)

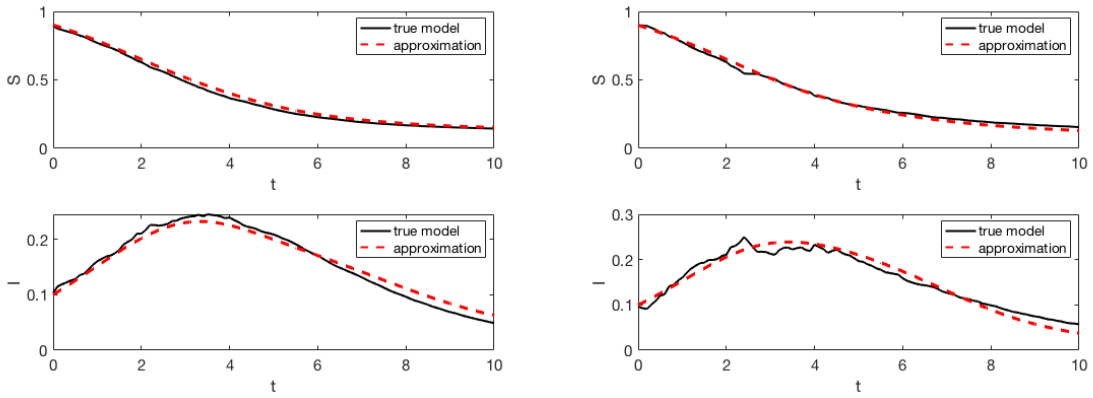

Figure 3: Numerical solution to the stochastic SIR model with $\beta(t) = 1.1 + \epsilon_t, \gamma = 0.5$ compared with the numerical solution to the model constructed by SINDy

2.2 SIR systems with decreasing transmission rate

In this section, we consider disease transmission rates decreasing over time. This is motivated by the governments' responses of limiting social interactions among citizens to reduce disease transmissions, such as travel bans and shutdowns of businesses and schools. If we let $\beta_1 <$ denotes the reduced transmission rate after the government's orders, and let $\beta_2 > 1$ denote the original transmission rate. Then, consider function

$$\beta(t) = \frac{\beta_1 + \beta_2}{2} + \frac{\beta_1 - \beta_2}{2} \tanh\left(\frac{t - \alpha T}{\xi}\right),$$

where T denotes the final time, α determines how fast the government takes action (small α means fast action), and ξ determines the smoothness of the rate drop (see Figure 4).

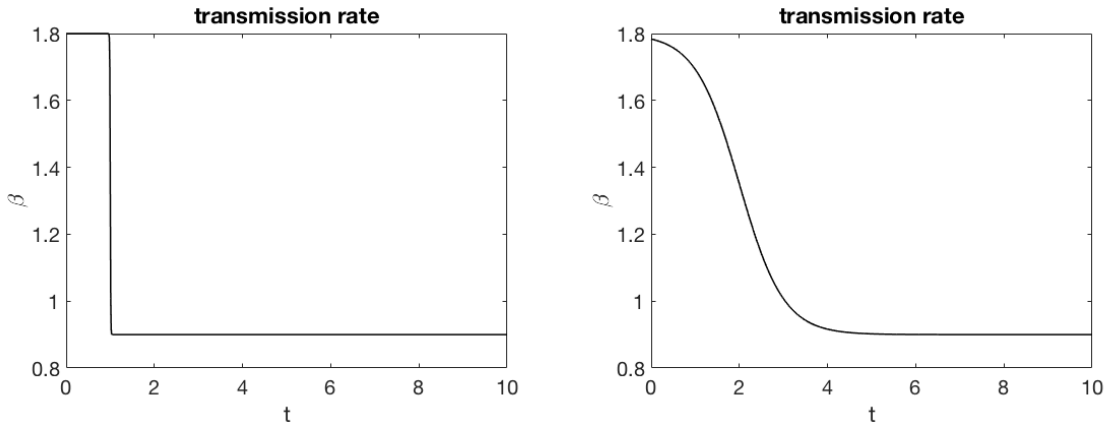

Figure 4: Transmission rates with $\alpha = 0.1$, $\xi = 0.01$ (left) and $\alpha = 0.2, \xi = 1$ (right)

We consider SIR models with transmission rate with $\beta = 0.1, \xi = 0.01$ (which corresponds to fast response to disease spreading) and $\alpha = 0.2, \xi = 0.1$ (which which corresponds to slow and gradual response) (see Figure 4). To implement SINDy, we approximated the derivatives using fourth order central differencing and used polynomials of degrees 1-2 for the function library. Note that the exact terms in the governing equations cannot be recovered by SINDy due to the varying coefficient β. We used SINDy to construct a new model that describes the disease evaluations - the numerical results to the SIR model with decreasing transmission rate and the model constructed by SINDy are plotted in Figure 5. In both cases, the model constructed by SINDy performed well in capturing the original models' dynamics.

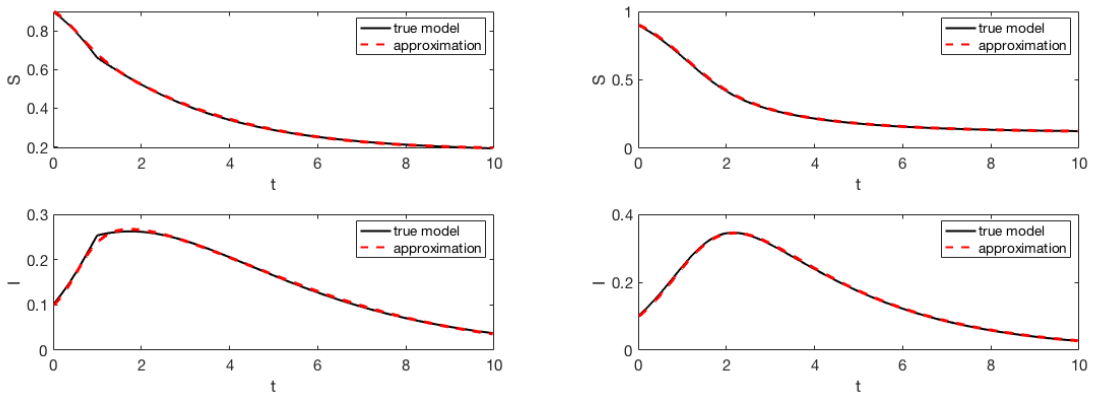

Figure 5: Comparisons between the numerical solutions of the SIR models with decreasing transmission rates and the numerical solutions of the models constructed by SINDy

3 Model construction with COVID-19 data

From the JHU CSSE (Johns Hopkins University Center for Systems Science and Engineering) website, we were able to obtain the reported COVID-19 data in Hubei province, China, from January 22 to March 14, 2020, which gives us the numbers of infections, recoveries, and deaths for 53 days. We consider both recoveries and deaths as the removed population in this section. Also, since most of the infections in the Hubei province have occurred in the city Hubei, which has an 11 million population, we set the total population to be $N = 10^7$.

To avoid SINDy failing due to insufficient sampling, we performed a nonlinear OLS regression on the collected data to produce more data points (see Figure 6). Again, using fourth order central differencing to approximate the derivatives and polynomials of order 1 to 3 for the function library, the numerical solution to the SINDy-constructed model is plotted in Figure 7 along with the COVID-19 data and its regression. The SINDy approximation predicts that the number of infections decreases to zero as time progresses.

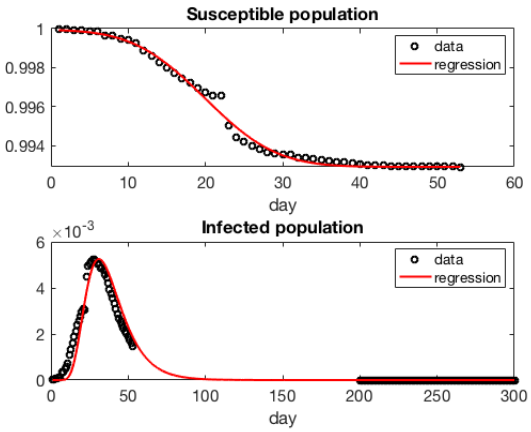

Figure 6: COVID-19 data and nonlinear OLS regression

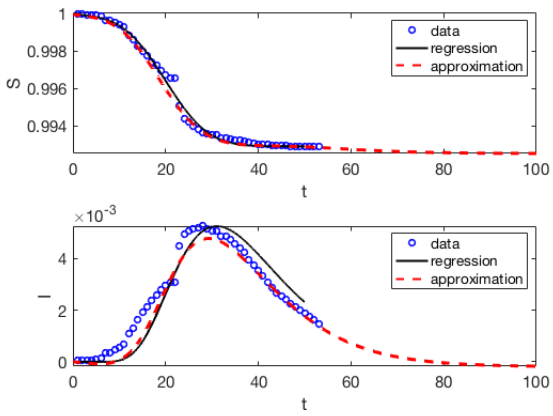

Figure 7: Numerical solution to the SINDy-constructed model along with COVID-19 data and its nonlinear OLS regression: $N = 10^7$

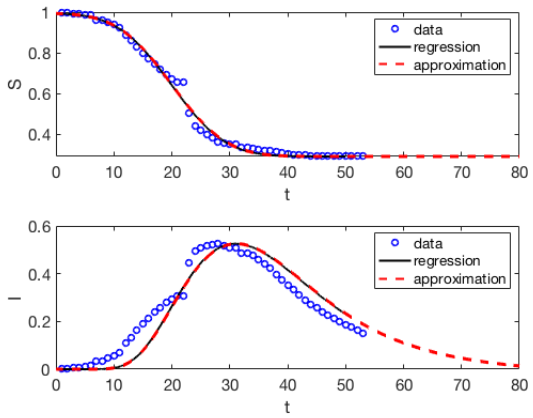

Figure 8: Numerical solution to the SINDy-constructed model along with COVID-19 data and its nonlinear OLS regression: $N = 10^5$

4 Conclusion

In this report, we investigated SINDy's ability to recover and construct the governing equations of infectious disease models. We tested SINDy on the time series data sampled from the standard SIR model, the stochastic SIR model with randomly perturbed transmission rates, the SIR model with decreasing transmission rates, and the nonlinear OLS regression of COVID-19 data in Hubei province, China. In general, the SINDy algorithm appears to be sensitive in noise in the derivatives of the state variables. Apart from noise sensitivity, SINDy performs well and gives reason predictions.

A SINDy implementation

For the SINDy implementation, we follow the algorithm below:

initialize state $\mathbf{x}(t)$ and derivative $\dot{\mathbf{x}}(t)$ at sampling times $t_1, t_2, \cdots, t_m$, a list of of nonlinear candidate functions
construct matrices for state $\mathbf{x}(t)$ and derivative $\dot{\mathbf{x}}(t)$ (if $\dot{\mathbf{x}}(t)$ is not given, approximate it numerically from $\mathbf{x}(t)$):

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m), \end{bmatrix}, \quad \dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m), \end{bmatrix}$$

construct a library $\boldsymbol{\Theta}(\mathbf{X})$ consisting of candidate nonlinear functions of the columns of $\mathbf{X}$:

$$\boldsymbol{\Theta}(\mathbf{X}) = \begin{bmatrix} 1 & \mathbf{X} & \mathbf{X}^{p_2} & \cdots & \sin \mathbf{X} & \cos \mathbf{X} \end{bmatrix}$$

construct a sparse regression problem (Algorithm 2) to determine sparse vectors of coefficients $\boldsymbol{\Xi} = [\xi_1, \xi_2, \cdots, \xi_n]$:

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

where $\mathbf{Z}$ is the noise matrix with independent identically distributed Gaussian entries with zero mean. It is added to account for the noise introduced by derivative approximation
construct equations:

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

Algorithm 1: SINDy implementation

B Sparse regression implementation

Given $\dot{\mathbf{X}}, \boldsymbol{\Theta}(\mathbf{X}), \boldsymbol{\Xi}, \eta, \mathbf{Z}$, as previously defined and a sparsification parameter λ (we use $\lambda = 0.025$) First perform the least-squares regression,

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi} + \eta \mathbf{Z} \tag{4}$$

for $i = 1 : 10$ **do**
  Set all coefficients in $\boldsymbol{\Xi}$ that are $< \lambda$ equal to 0.
  Then repeat regression on the remaining terms.
end

Algorithm 2: Sparse regression implementation