CSE 8803 - EPI Project: Discovering Epidemiological Differential Equations from Data

Sabrina Edwards-Swart
Georgia Institute of Technology
Atlanta, GA, United States
sjes3@gatech.edu

Atticus Rex
Georgia Institute of Technology
Atlanta, GA, United States
arex8@gatech.edu

ACM Reference Format:
Sabrina Edwards-Swart and Atticus Rex. 2023. CSE 8803 - EPI Project: Discovering Epidemiological Differential Equations from Data. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ABSTRACT

This project seeks to discover governing differential equations directly from data to determine the overall course of an epidemic. The two methodologies used to accomplish this will be the popular SINDy algorithm, developed by Brunton and Kutz et al. as well as the symbolic regression model applied to the derivative of the model. The group successfully used both methods to recover the SIR model from synthetic data with noise artificially added and to predict COVID-19 data. The PySR model, when effectively regularized showed promise in being able to robustly recover governing differential equations in the presence of noisy data.

1 INTRODUCTION

Models are essential to computational epidemiology to predict the behavior of a disease outbreak, to identify stability and equilibrium points, and to explain the underlying dynamics of an outbreak. The availability of data is an essential component to modeling the outbreak of disease and real-world data tends to be sparse and incomplete. Ordinary Differential Equation (ODE) models have been used to successfully model the spread of disease in ideal conditions for hundreds of years, operating under assumptions such as perfect mixing of a population, lack of mutations, lack of interventions, and homogeneous population demographics like age, susceptibility, and so-on [7].

Network Models are another approach to modeling infectious disease. These use graphs/networks to model the propagation of disease from one entity to another [10]. This allows for a population to be much more heterogeneous and model specific interactions between cliques or individuals. However, the number of parameters in a network model tends to be much higher than ODE models. In class, we learned of the EINN (Epidemiology Informed Neural Network) framework which trains neural networks as universal function approximators on epidemiological constraints baked into loss functions [11]. While neural networks perform extremely well with large amounts of data, epidemiological data is often noisy and in low or incomplete quantity. Further, neural networks are not nearly as interpretable as ODE models [7, 11].

The emergence of data-driven modeling to discover underlying differential equations from data has enjoyed numerous breakthroughs since the advent of digital computing [8, 9] . The Sparse Identification of Nonlinear Dynamics algorithm originally proposed by Brunton et al. provides a powerful framework for discovering nonlinear dynamics from noisy data. Model Predictive Control (MPC) shows how to implement epidemiological constraints such as infection thresholds and hospital capacities to quantify real-world constraints on an epidemiological model [4] . Further, symbolic regression algorithms that employ metaheuristic optimization have also shown promise in not only accurately modeling physical systems, but providing interpretable equations to explain the underlying dynamics. Further, these models can be regularized by restricting the sparsity or number of terms to generalize quite well with far fewer parameters than the aforementioned network models [3].

When modeling the spread of an infectious disease, there is generally a tradeoff between the complexity of the model and its ability to generalize. Make a model too simple and it won't give an accurate enough representation of the system. Make a model too complex and it runs the risk of overfitting to the training data. This is why person-to-person models are exceedingly difficult to calibrate and scale accurately.

2 PROBLEM STATEMENT

This project seeks to improve upon current methods for discovering governing equations of epidemiological spread. Specifically, we seek to combine advances in symbolic regression methods and the SINDy framework using a more diverse library of candidate functions to discover more robust ODE models for predicting the spread of infectious disease. Specifically, we seek to validate the use of these models with synthetic data in order to ensure their capability to recover governing differential equations. Subsequently, we will then apply this model to COVID-19 data in the Atlanta Metropolitan area to examine its ability to scale to real-world applicaitons.

3 CURRENT METHODS

Horrocks et al. uses SINDy to model the temporal dynamics of Measles, Varicella, and Rubella in various countries [7]. The results were promising, but limited; SINDy makes use of a library of candidate functions to account for nonlinearity in the data [6]. However, the candidate functions used for the final models of SINDy in this

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

paper were only of polynomial nonlinearity up to order 3. There is some serious upside to experimenting with sinusoidal terms in this kind of analysis as shown in [12] to model seasonal changes and variant mutations in epidemiological spread. Horrocks et al. cites not wanting to overfit by not using higher order terms, but this causes the model to perform poorly on certain datasets. Further, the model was quite sensitive to noise.

Interestingly, Horrocks et al. also uses spectral density plots in the frequency domain to gain insight into the cyclical nature of the epidemics. The results from this analysis were unconclusive with some diseases able to be analyzed in this way but others not holding up well.

Symbolic regression is a new method proposed in [5] for discovering underlying equations from data. There has been limited work in discovering ODEs using symbolic regression and also limited work in applying SR to epidemiological data. However, the tournament selection methods and other optimization schemes used in SR have had success in physical applicaitons. Further, constraints can be placed on the complexity of the equation sets in Symbolic Regression to regularize and avoid overfitting [3].

We were unable to find a paper that applied Symbolic Regression to discovering governing equations to epidemiological dynamics, however this ability to regularize and control the complexity of the equations is desirable and mirrors the aims of the MDLInfer algorithm.

4 DATA COLLECTION AND SYNTHESIS

Improving upon the results found in Horrocks et al makes sense in demonstrating the utility of the methods. This data is available in the Materials and Methods section of that paper [7]. To demonstrate the modern utility of data-driven symbolic regression/SINDy methods, COVID-19 data is available in high volumes from multiple sources. COVID-19 case and death counts per day provide a good baseline approximation for the recovery and infection rates. This data is widely available through the CDC in .csv form along with key demographic data as well [2]. Lastly, to augment the raw disease data, we'd like to include the following data sources to capture changing behaviors and infection rates:

- **Search Engine data for symptoms** Include how many people are searching for symptoms of the disease over time to help give an indicator of the severity. This data is available on Google Trends from 2004 onward which could be quite useful in seeing how many people are searching for symptoms of the disease.
- **Temperature and Precipitation Data** Include the average temperature and the number of hours of precipitation each day as data points. This information can be found on the NOAA website [1].

We have been able to download the data from Google's Open-Data for COVID and isolate just the Atlanta Metropolitan Area. We have also been able to isolate google searches for just the Atlanta Metropolitan Area, though it's difficult to say how accurate these terms are. That said, everything is compiled in a GitHub document and cleaned. The group used a Savitsky-Golay filter to smooth the COVID Epidemiological data such as cases and deaths, so the algorithm could have a more accurate representation of the derivative. This is part of the preprocessing stage. Preprocessing is important because, without it, the model would have to process through irrelevant data at every iteration, making the model untenably slow.

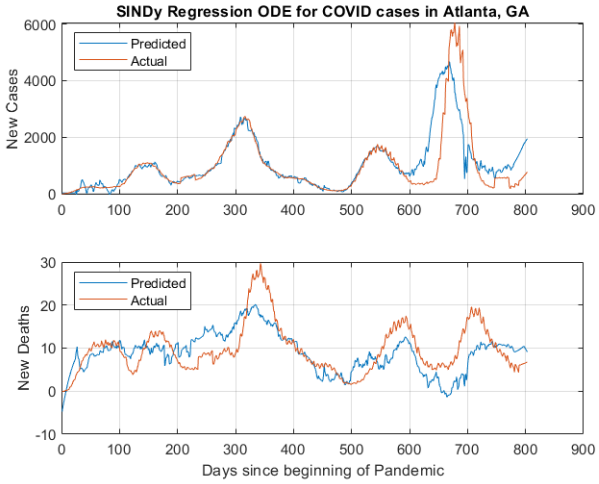

Figure 1: The prediction of the SINDy Algorithm on COVID Cases and Deaths. The Algorithm was trained on days 0-600 and allowed to propagate forward for days 600 to 800.

We will also simulate SIR data using the governing SIR ODE model, which will be discussed in detail in the Results section of the paper.

5 ALGORITHMS & TECHNIQUES

We will have to do significant preprocessing of the infection and recovery rates to infer real rates of reporting and recovery. We will use the method in [6] which formulates a reporting rate with birth and death rates to approximate infection and recovery rates.

We will divide the population at the city level into three groups: susceptible, infected and dead. We will try to perform minimal inference and calibrate the models on deaths rather than try to extrapolate a recovered population.

SINDy Framework

The SINDy framework assumes that the underlying dynamical system can be modeled in terms of their derivatives in the following general form:

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

We use a matrix of snapshots of a state, x, over a given time, the only difference being we also need to sample the derivative of x. This can be illustrated as:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_N \\ | & | & & | \end{bmatrix}^T, \mathbf{X} = \begin{bmatrix} | & | & & | \\ \dot{x}_1 & \dot{x}_2 & \dots & \dot{x}_N \\ | & | & & | \end{bmatrix}^T$$

And then the library of nonlinear candidate functions can be modeled by $\Theta(\mathbf{X})$. These functions are entirely up to the user. The

more functions, the more computational cost, but the higher probability that the model will converge to the correct governing equations. An example of $\Theta(\mathbf{X})$ might be:

$$\Theta(\mathbf{X}) = \begin{bmatrix} 1 & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \dots & \sin(\mathbf{X}) & \dots & \ln(\mathbf{X}) & \dots & e^{\mathbf{X}} \end{bmatrix}$$

In this case, P_2, P_3 and so on represent polynomial combinations of a certain order. Once this matrix of candidate functions evaluated at every single datapoint is formed, the following sparse regression equation can be solved:

$$\mathbf{X} = \Theta(\mathbf{X})\Xi$$

where Ξ is a matrix of weights that shows how many variables of $\Theta(\mathbf{X})$ are present in the dynamics. The sparse linear regression involves some sparsity parameter, λ, to cut off terms that are not very present. The idea is that the least squares regression is solved and an initial Ξ is obtained. Then, every term in Ξ that is less than λ is set to zero. And the least squares regression is repeated, but *only* onto the terms that have not been set to zero. This process is repeated until only a handful of terms are remaining and the rest are zero.

Possibly the most overlooked component of this algorithm is the ability to sample the derivative. The original SINDy paper introduced noisy signals into their analysis and illustrated how the model still held up reasonably well with various noise-levels. However, to smooth the noise, the authors used the TVD algorithm described in the introduction. This is an *incredibly* computationally costly algorithm to run at the scale that it was used in this paper, having hundreds of gradient descent steps involving matrices of size 100,000 x 100,000 individually for the x, y, and z coordinates of the Lorenz System [4].

What makes epidemiological systems so applicable to the SINDy algorithm is the fact that they report the derivative of total infections and deaths by reporting counts per day as opposed to total. Our hope is that this will greatly increase the accuracy of the learned dynamics as opposed to numerically calculating derivatives.

Symbolic Regression

Symbolic Regression has been performed in various ways for hundreds of years. Recent developments have shown great promise in using trees of function operations and Genetic Programming techniques to optimize symbolic representations of functions. In Figure 2, we see the computation tree and the crossover process for creating variability in the solutions.

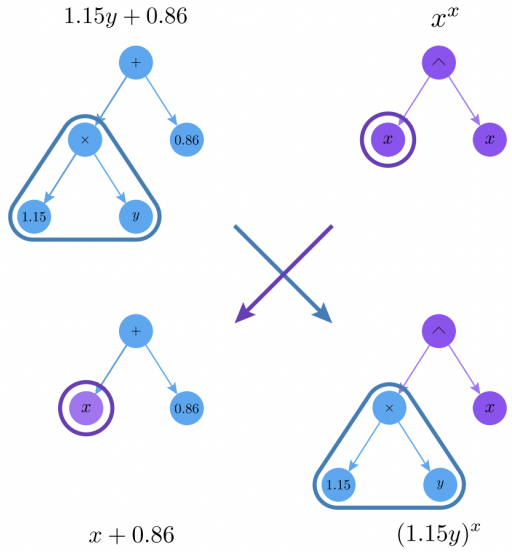

Figure 2: A representation of the function trees and a crossover method to induce variety in the candidate solutions.

In this study, we will be using the PySR Optimization framework which makes use of a Tournament Selection (TS) framework which is depicted in Figure 10. This consists of a population of random trees selected from the available function operations and takes the fittest trees from this population. With the fittest trees, the algorithm either mutates one node of the tree by changing the operation, crosses two trees as shown in Figure 2, simplifies the trees by combining like operations (think addition and addition), or optimizes the coefficients in the solution via gradient descent [3].

We will compare the results in this method to the SINDy framework to evaluate which is more effective at generalizing and predicting disease dynamics.

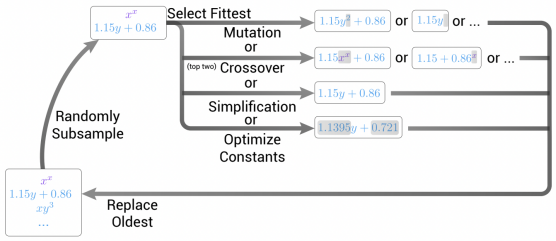

Figure 3: The Tournament Selection optimization method that

6 RESULTS

6.1 Synthetic Data Validation

To establish a baseline efficacy for the SINDy and SR algorithms to determine symbolic differential equations, we produced the following curve from the standard SIR ODE model for infectious diseases, formulated by:

$$\frac{dS}{dt} = -\beta S \cdot I \tag{1}$$

$$\frac{dI}{dt} = \beta S \cdot I - \gamma I \tag{2}$$

$$\frac{dR}{dt} = \gamma I \tag{3}$$

We then used a 4th order Runge Kutte Solver to simulate the progression of these dynamics with $\beta = 0.1, \gamma = 0.01, S_0 = 0.99, I_0 = 0.01$ and $R_0 = 0.0$, with some normally distributed random noise added to the model. This is depicted in Figure 4.

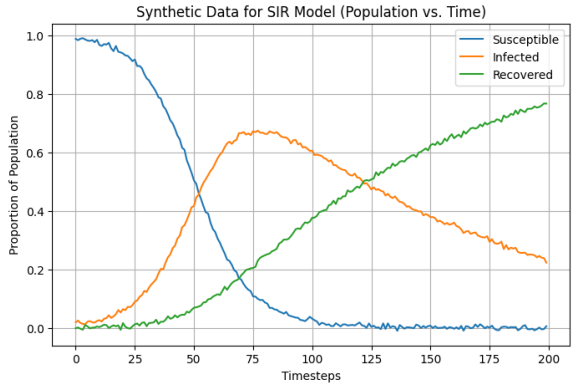

Figure 4: Synthetic SIR model simulation (Noise: $\mathcal{N}(\mu = 0, \sigma = 0.005)$).

6.2 SINDy Results on Synthetic Data

We regressed the SINDy algorithm on the synthetic data. One significant challenge of the SINDy algorithm is the need for a numerical sample of the derivative of the desired signal. In practice, numerically differentiating a noisy signal with a finite-difference derivative means a derivative in which the noise is amplified. To get around this, we used a forward-difference derivative and then smoothed it with a Savitsky-Golay Filter to denoise the derivative. Figure 5.

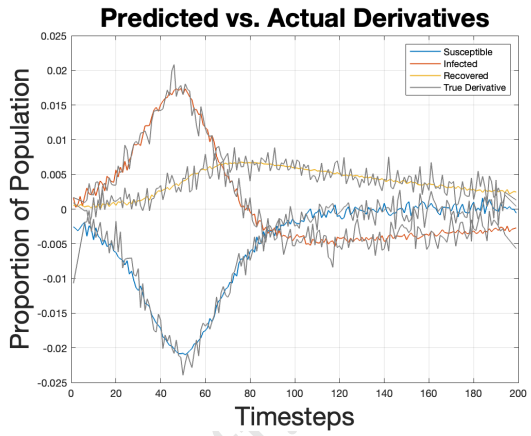

Figure 5: SINDy Derivative Prediction (MAE = 0.000789).

Once we solved for the Ξ matrix, we used it within a 4th Order Runge Kutte solver to simulate the system from the same initial conditions. The result is shown in Figure 6.

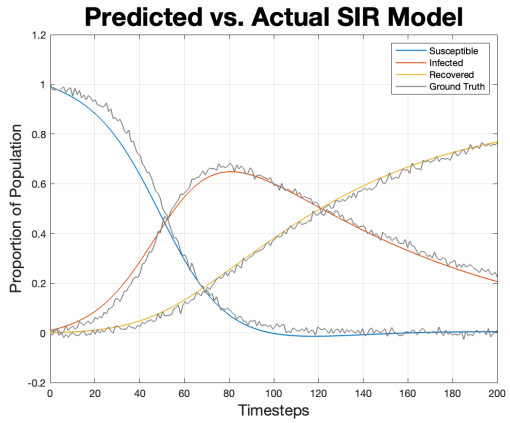

Figure 6: Simulated Epidemic using SINDy Model (MAE = 0.0582).

In our best model, we produced a Mean Absolute Error (MAE) of 0.0582 between the ground-truth data. With this model, the Ξ matrix corresponds to the active terms within our candidate functions. The computed Ξ matrix is shown below:

Table 1: SINDy Reconstruction of SIR ODEs

Terms	S	I	R	S^2	SI	SR	I^2	IR	R^2
dS/dt	0	0	0	0	-0.09412	0	0	0	0
dI/dt	0	-0.0078	0	0	0.0892	0	0	0	0
dR/dt	0	0.0098	0	0	0	0	0	0	0

While this is rather encouraging, there are some major considerations that will be addressed in the discussion section.

- **Sensitivity to Noise**: The SINDy Algorithm is *highly* sensitive to noisy data, and in this case, the clarify of the output matrix Ξ, was highly dependent on an effective window length for the SG-Filtering algorithm. Further, past a noise level of 0.02, we could not reliably reproduce the SIR Model.
- **Amount of Data**: To produce the model in the results, we had to train the model on more like 400-500 days of pandemic data under different initial conditions, which doesn't have great implications on the ability to apply this to real-world data that could be much noising and be far more reactive to external measures.
- **Optimization Method**: The original SINDy paper by Brunton et al. uses Sparse Linear Regression to force the majority of the coefficients to zero. However, we found much better results employing the MDL framework and forcing the Ξ matrix to have a certain number of nonzero terms. We forced the Ξ matrix to five nonzero terms and it converged to the correct solution eventually. However, we had to tweak many parameters to force it to converge to the correct solution. Imperfect knowledge of the underlying dynamics may significantly impede the performance/ability to converge onto correct solutions.

Symbolic Regression Results on Synthetic Data

The Symbolic Regression algorithm is much more computationally costly with the tournament selection method, usually taking multiple minutes to run, however has powerful nonlinear optimization techniques and regularization methods that make this algorithm quite powerful if correctly applied. When regressed on the numerical derivatives of the SIR model, the algorithm produced the following symbolic results–the following equations are in the exact form the output returned them (Substituting S, I, and R for readability):

$$\frac{dS}{dt} = (((-0.09922531 * S) + -0.00058495) * I)$$

$$\frac{dI}{dt} = (((S + -0.09877744) * I) * 0.09526928)$$

$$\frac{dR}{dt} = ((I * -0.0007062112) + 0.010293879 * I)$$

Simplifying these equations yields:

$$\frac{dS}{dt} = -0.09923SI + -0.000584I$$

$$\frac{dI}{dt} = 0.095269SI - 0.009410I$$

$$\frac{dR}{dt} = 0.009587I$$

This result is almost identical to the governing SIR ODEs, with one extraneous term in dS/dt. This extraneous term has a very low coefficient, however, and should not significantly affect results. Let us examine what we get when we simulate this ODE compared to the actual SIR model:

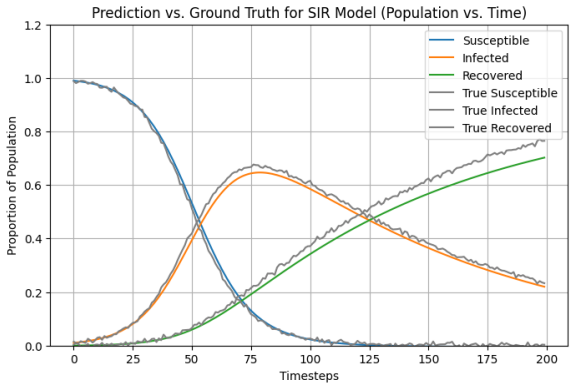

Figure 7: Simulated ODE with Symbolic Regression-Created SIR Model (MAE = 0.0276.)

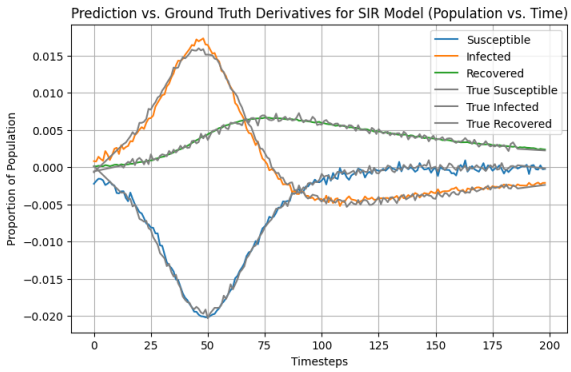

Figure 8: Plot of Symbolic Regression-Approximated Derivative vs. Smoothed Numerical Derivative of SIR equation (MAE = 0.000193)

As shown in figures 7 and 8, the Symbolic Regression ODE is highly accurate of the original results, demonstrating the effectiveness of the model to be used to model real-world data.

The group was also interested in how the algorithm performed when noise was amplified. The following figure shows the resulting dynamics simulated from ODEs when the noise is amplified to have a standard deviation of 6% of the population (a lot of noise!):

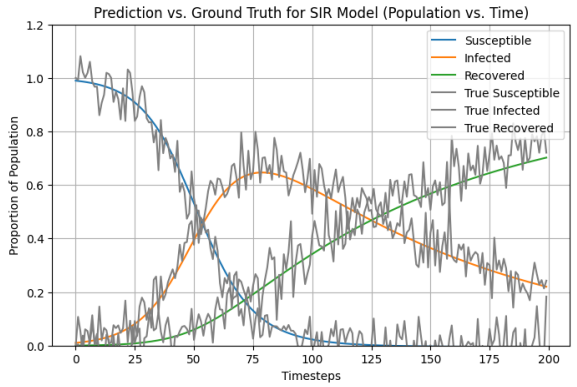

Figure 9: Symbolic Regression-Produced ODE with highly noisy data ($\mu = 0, \sigma = 0.06$, MAE = 0.0389 compared to noiseless data)

As shown in figure 9, the ODEs produced from the Symbolic Regression was able to successfully reproduce the dynamics to a very high degree of accuracy compared to the noiseless training data.

Candidate functions produced:

$$\frac{dS}{dt} = -0.0864SI + 0.000612$$
$$\frac{dI}{dt} = 0.0822SI + -0.00796I$$
$$\frac{dR}{dt} = 0.00845I + 0.000303$$

While the coefficients differ a bit from the true governing coefficients by about 0.01-0.02 and there are small random numerical intercepts added on, the structure of the governing ODE is still preserved, which is highly desirable for real-world experimentation. Further, we wished to test the ODEs from the noisy data on different initial conditions to test any overfitting to training data. This is shown in

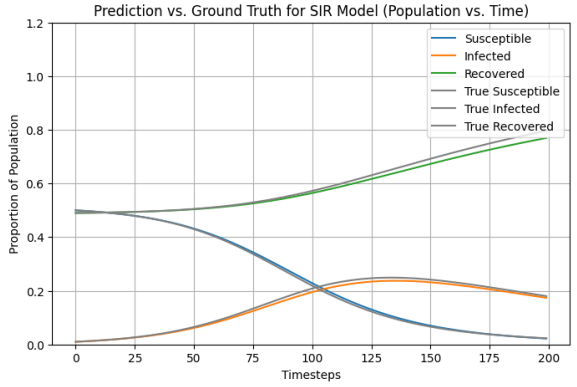

Figure 10: Symbolic Regression-Produced ODE simulation with different initial conditions compared with ground truth. SR algorithm was trained on highly noisy data (MAE = 0.0427, $\mu = 0, \sigma = 0.06$)

Additionally, we ran the symbolic regression algorithm on the data with time as the input data and S, I, and R as the target labels for both noisy and ideal generated datasets. While the predictions for the noisy data were slightly worse than the predictions based on the model run with one of SIR as the target label set and the other two as the input data (for example, when prediction X, I and R were the input data, the predictions for the ideal datasets were dramatically incorrect. This brings up a possible point of interest for further study on how the amount of noise effects convergence speed and ability. Figure 11 displays these results for the I data.

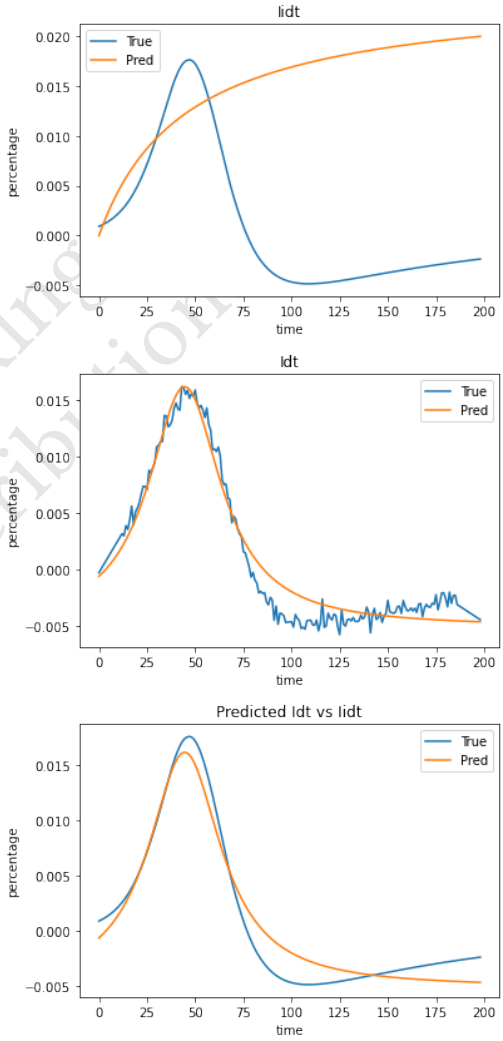

Figure 11: The first graph is a graph of ideal data vs the predictions from a model trained on the ideal data (RMSE = .2460). The second graph is a graph of noisy data vs the predictions from a model trained on the noisy data (RMSE = .0199). The third graph is a graph of ideal data vs the predictions from a model trained on the noisy data (RMSE = .0266). For contrast, the same model trained with noisy S and R to predict I had an RMSE of .0149 compared to noisy data and .0134 compared to ideal data.

SINDy Results on Real-World Data

To demonstrate the effectiveness of this algorithm on real-world data, we used the aforementioned COVID-19 pandemic data from Atlanta, GA to attempt to produce an autoregressive model with the derivative. The model was trained on 600 days of actual COVID Data which was augmented with weather and symptom-search data. The model was then tested on an additional 300 days of data which contained a large spike which had not been seen in the training data. The trained SINDy model was able to produce the dynamics shown in Figure 12.

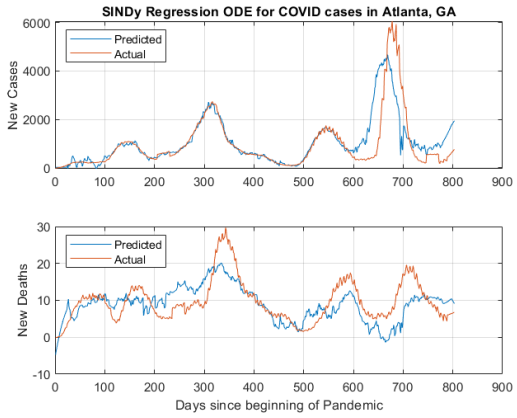

Figure 12: SINDy Model Prediction (Test MAE = 567.29 for new cases, and 4.14 for new deaths.)

What is interesting is the model was able to predict a spike larger than it had ever seen before in the derivative, indicating an ability to generalize well to unseen data. The SINDy algorithm produced governing equations that weighted the search data quite heavily, but were too long to include in this report (There were over 40 terms in each equation).

Symbolic Regression on Real-World Data

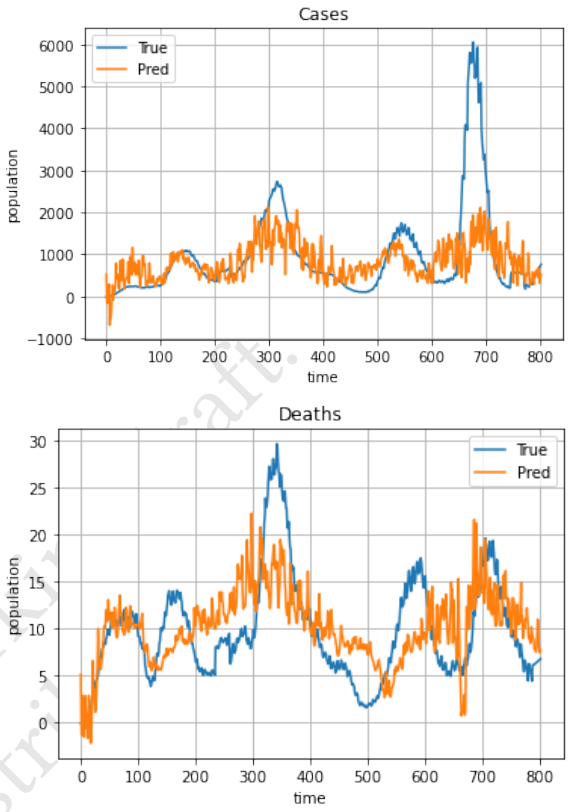

Figure 13: Symbolic Regression Model Prediction (Test RMSE = 21475.38 for new cases, and 51.31 for new deaths.

Similarly to the SINDy model, we used the aforementioned COVID-19 pandemic data from Atlanta, GA to attempt to produce an autoregressive model with the derivative; additionally, the model was trained on 600 days of actual COVID Data which was augmented with weather and symptom-search data. The model was then tested on an additional 103 days of data to determine how effective it was at predicting future cases and deaths. Despite the cases prediction having a much higher RMSE than the deaths graph, to the naked eye, the cases graph appears to be a better predictor than the deaths graph.

7 CONCLUSION AND DISCUSSION

Advantages of SINDy

- **Computational Cost**: Because the SINDy algorithm is effectively solving a linear least squares problem with some regularization, it is very quick to compute for large datasets, making it an ideal candidate for physical experiments with high temporal resolution as used in Brunton et al. [4]. It should be said, however, that computing the Total Variation Regularized Derivative as the authors did in the original SINDy paper is highly computational costly and direct differentiation methods should be explored.

- **Simplicity of Results**: The SINDy algorithm produces highly interpretable results that make displaying equations like in **Table 1** effective in research applications.

Limitations of SINDy

- **Sensitivity to Noise**: The SINDy Algorithm is *highly* sensitive to noisy data, and in this case, the clarify of the output matrix Ξ, was highly dependent on an effective window length for the SG-Filtering algorithm. Further, past a noise level of 0.02, we could not reliably reproduce the SIR Model.
- **Quantity of Data**: To produce the model in the results, we had to train the model on more like 400-500 days of pandemic data under different initial conditions, which doesn't have great implications on the ability to apply this to real-world data that could be much noising and be far more reactive to external measures.
- **Optimization Method**: The original SINDy paper by Brunton et al. uses Sparse Linear Regression to force the majority of the coefficients to zero. However, we found much better results employing the MDL framework and forcing the Ξ matrix to have a certain number of nonzero terms. We forced the Ξ matrix to five nonzero terms and it converged to the correct solution eventually. However, we had to tweak many parameters to force it to converge to the correct solution. Imperfect knowledge of the underlying dynamics may significantly impede the performance/ability to converge onto correct solutions.
- **Inflexibility of Candidate Functions** Not only does the user have to specify which candidate functions that the function should investigate, which means all polynomial candidate functions up to some order, and any others, but there is also no ability to nest candidate functions, nor use exponential functions. This is a major drawback of this algorithm compared to Symbolic Regression.

Advantages of Symbolic Regression

- **Nested Candiate Functions**: The ability of the SR algorithm to not only nest functions within other functions, but also optimize the coefficients within the nested functions makes this a much more robust algorithm for reliably finding good approximations of underlying ODEs. For example, if SINDy wanted to use sinusoidal functions, it cannot modify the ω on the inside of the $\sin(\omega x)$ function, nor can it put any other coefficients or operators within it. SR does not suffer from this problem.
- **Robust Regularization**: As described previously, the SINDy algorithm was rather difficult to regularize for a desired number of coefficients. With the SR algorithm, we are able to specify the max-depth of the computational tree as well as the maximum complexity so only candidate soultions that satisfy this criteria are chosen. This turns out to be much more powerful in constraining the complexity of the overall model instead of using some sort of linear regression regularization technique as SINDy does.
- **Quantity of Data**: Like the SINDy algorithm the SR algorithm suffers when the quantity of data is low. However, unlike the SINDy algorithm the SR library was able to recover a good approximation of the underlying differential equations from one instance of data, making it a much more robust candidate for real-world use.

Limitations of Symbolic Regression

- **Computational Cost/Complexity**: Because this optimization algorithm runs on a population-based tournament selection algorithm that includes mutations, crossovers, and gradient descent optimization with multiple populations of many agents, this is a *costly* algorithm to run. Further, as the complexity and depth of the candidate functions increase, the amount of time to convergence increases exponentially. Given the random nature of this algorithm, we aren't sure at what rate the runtime grows, but in our experience, adding one or two more candidate functions, increasing the population size by 50% or increasing the model complexity by 50 % would more than triple the runtime of the algorithm and would not guarantee convergence.
- **Extraneous Candidate Functions**: The more candidate functions the algorithm is allowed to use means that the model has a much more robust hypothesis class of composite functions to use. This being said, however, this also makes the probability of convergence onto a random combination of functions that happens to have a low MAE quite high. Initially, we used all kinds of candidate functions like logs, square roots, exponential functions, sinusoidal terms, and so-on. However, this produces so many exponentially more options for the function to iterate through that not only does it heavily extend computational time, it usually converged onto some mess of nested nonlinear functions that happened to approximate the derivative well.
- **Sensitivity to Noise**: Like SINDy, the Symbolic Regression algorithm is still sensitive to noise and the more noise, the less likely the algorithm converges to the correct solution. This can be fixed with some smoothing of the derivative and the original funciton, but it still has potential to sway the accuracy of the results quite a bit. However, symbolic regression sometimes performs better with a small amount of noise than with no noise.

REFERENCES

[1] [n. d.]. *Past Weather by Zip Code - Data Table | NOAA Climate.gov.* http://www.climate.gov/maps-data/dataset/past-weather-zip-code-data-table

[2] CDC. [n. d.]. *Cases, Data, and Surveillance.* https://www.cdc.gov/coronavirus/2019-ncov/covid-data/about-COVID-data.html

[3] Miles Cranmer. [n. d.]. Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl. https://doi.org/10.48550/arXiv.2305.01582 arXiv:2305.01582 [astro-ph, physics:physics]

[4] Urban Fasel, Eurika Kaiser, J. Nathan Kutz, Bingni W. Brunton, and Steven L. Brunton. [n. d.]. SINDy with Control: A Tutorial. https://doi.org/10.48550/arXiv.2108.13404 arXiv:2108.13404 [math]

[5] Sébastien Gaucel, Maarten Keijzer, Evelyne Lutton, and Alberto Tonda. [n. d.]. Learning Dynamical Systems Using Standard Symbolic Regression. In *Genetic Programming* (Berlin, Heidelberg, 2014) *(Lecture Notes in Computer Science)*, Miguel Nicolau, Krzysztof Krawiec, Malcolm I. Heywood, Mauro Castelli, Pablo García-Sánchez, Juan J. Merelo, Victor M. Rivas Santos, and Kevin Sim (Eds.). Springer, 25–36. https://doi.org/10.1007/978-3-662-44303-3_3

[6] Jonathan Horrocks. [n. d.]. Sparse Identification of Epidemiological Models from Empirical Data. ([n. d.]).

[7] Jonathan Horrocks and Chris T. Bauch. [n. d.]. Algorithmic discovery of dynamic models from infectious disease data. 10, 1 ([n. d.]), 7061. https://doi.org/10.1038/

s41598-020-63877-w Number: 1 Publisher: Nature Publishing Group.

[8] Liron Simon Keren, Alex Liberzon, and Teddy Lazebnik. [n. d.]. A computational framework for physics-informed symbolic regression with straightforward integration of domain knowledge. 13, 1 ([n. d.]), 1249. https://doi.org/10.1038/s41598-023-28328-2 Number: 1 Publisher: Nature Publishing Group.

[9] William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. [n. d.]. Contemporary Symbolic Regression Methods and their Relative Performance. https://doi.org/10.48550/arXiv.2107.14351 arXiv:2107.14351 [cs]

[10] Tae Jin Lee, Masayuki Kakehashi, and Arni S. R. Srinivasa Rao. [n. d.]. Chapter 8 - Network models in epidemiology. In *Handbook of Statistics*, Arni S. R. Srinivasa Rao and C. R. Rao (Eds.). Data Science: Theory and Applications, Vol. 44. Elsevier, 235–256. https://doi.org/10.1016/bs.host.2020.09.002

[11] Alexander Rodríguez, Jiaming Cui, Naren Ramakrishnan, Bijaya Adhikari, and B. Aditya Prakash. [n. d.]. EINNs: Epidemiologically-informed Neural Networks. https://doi.org/10.48550/arXiv.2202.10446 arXiv:2202.10446 [physics, q-bio, stat]

[12] Yoshiyasu Takefuji. [n. d.]. Fourier analysis using the number of COVID-19 daily deaths in the US. 149 ([n. d.]), e64. https://doi.org/10.1017/S0950268821000522