

MATH181 Project: Automatic Identification of Nonlinear Systems

Theo Rode

1 Introduction

This project discusses the procedure of recovering the underlying equations for the dynamics given noisy data collected by the system. So called “physics-inspired” or “physics-informed” AI. The goal of this field is to gain a physical understanding of systems where we can measure various trajectories, but the dynamics are unknown. Though this is additionally useful in situations where complex systems have dynamical properties that can be modeled on a smaller scale. In particular, the focus of this project is on discussing the results demonstrated in a paper published in 2016 by Steven Brunton, Joshua Proctor, and Nathan Kutz who propose an algorithm for reconstructing dynamics dubbed SINDy (Sparse Identification of Nonlinear Dynamics) [3]. This first section aims to discuss the field more broadly and relevant papers to the results proposed in the aforementioned paper as well as more generally in the field.

The authors of [3] give two foundational papers for recovering the dynamics of nonlinear systems: a paper by Josh Bongard and Hod Lipson [2] and one by Michael Schmidt and Hod Lipson [5], in 2007 and 2009 respectively. Bongard and Lipson [2] outline a method for “reverse engineering” the dynamics of nonlinear systems utilizing an active learning system. The main process of the algorithm is a technique labeled **symbolic regression**. In [2], symbolic regression is made usable for complicated (nonlinear) systems by introducing three concepts: partitioning, automated probing, and snipping. Partitioning is a process by which the algorithm is able to model each of the variables in the system separately using Stochastic optimization. Each of the variables are integrated by substituting in representations of the other variables rather than integrating the variables together. This method significantly reduces the overhead when working with high dimensional systems. Automated probing allows the algorithm to explore the system through active learning. After candidate models are created, the algorithm attempts to create many “tests” which are judged against their ability to disprove as many of the candidate models as possible. Then, the test which rejects as many of the candidate models as possible is used to judge the performance of all candidate models in order to pick the most capable ones. From this, the next generation of candidate models can be generated. Finally, snipping is a process to reduce the complexity of the generated models and prevent over-fitting. In particular, when creating another

generation of models, occasionally certain subexpressions will be replaced with a value of the subexpression picked uniformly from its range. This acts to reduce situations where the models will create complex subexpressions which take on narrow ranges of values, demonstrating over-fitting to the data. Together, these innovations on the process of symbolic regression allowed Bongard and Lipson to create a framework that was applicable to an array of real-world systems. The framework demonstrated the ability to find nonlinearities and interdependencies as well as an ability to scale successfully. However, the authors note that current limitations include not yet demonstrating scalability to much larger biological models and an inability to work on data where certain variables are unobservable, as a key component of the framework was the active learning [2].

Schmidt and Lipson present a different algorithm for reconstructing dynamics that is still based on **symbolic regression** [5]. This paper focuses on presenting a new metric for comparing the accuracy of various candidate equations in order to better search for invariants within the system. In particular, once candidate equations are generated (initially randomly from a set of “building blocks”), the partial derivative pairings of every pair of variables are compared to the numerically calculated partial derivative pairings in order to select the equations which model these pairs the closest. The next generation of equations is then generated from these best equations probabilistically. The algorithm will return a set of equations when they reach a certain accuracy to the observed data, calculated from a subset of the data randomly selected to be withheld from training. Furthermore, these equations are prioritized according to a measured parsimony, effectively complexity which is calculated by the number of terms in the expression. The authors were able to demonstrate that this algorithm could reconstruct invariants from numerous physical systems effectively. Additionally, they noticed that with a limited set of building blocks, the algorithm found approximations to functions like sines and cosines, utilizing the available building blocks, when such functions were part of the physical invariant [5].

2 Mathematical Background

For the SINDy algorithm, the authors aimed to reframe the problem of recovering nonlinear dynamics using **sparse regression** and **compressed sensing** [3]. Both sparse regression and compressed sensing broadly refer to the process of finding a regression solution to a problem such that the solution is sparse in some space [3]. Brunton et al. relate that utilizing sparse regression is a strategy to reduce the noise present in the fitting of systems [3]. The particular assumptions for sparse regression come from the idea that in most cases, the dynamics for a particular system will be governed by a small set of given basis functions. Therefore, the sparsity also gives advantages for computation as we can increase the size of the function space dramatically without needing significantly more computation [3].

The authors of [3] propose an iterative algorithm for sparse regression. In particular, they start with a least-squares fit to the data, and then iteratively

remove coefficients below a given threshold. By refitting with the remaining, nonzero terms iteratively, the algorithm can arrive at a sparse solution [3].

The SINDy algorithm relies on the availability of both sampled data, \mathbf{X} , and the sampled derivative, $\dot{\mathbf{X}}$. However, the authors note that the availability of derivative measurements are not guaranteed and therefore propose using the **total variation regularized derivative** to mitigate the noise created from numerically calculating $\dot{\mathbf{X}}$ from \mathbf{X} [3]. In particular, they cite the algorithm used by Rudin et al. [4]. Here, Rudin et al. use a framework for computing a desired, denoised output, u , from a noisy input, u_0 , by assuming that $u = u_0 + n$ for some white noise function n . It is assumed that n has a mean of 0 with some given standard deviation. The minimization problem corresponds to minimizing $\int_{\Omega} \sqrt{u_x^2 + u_y^2} dA$, which is constrained by $\int_{\Omega} u dA = \int_{\Omega} u_0 dA$ under the assumption of the additive white noise. It is then proposed to solve for u through either an iterative algorithm (i.e. simulated annealing) or, preferably, a PDE solver to find local minimum [4].

In order to address the computational difficulties present in higher-dimensional problems, Brunton et al. suggest using **proper orthogonal decomposition (POD)** in order to perform dimension reduction [3]. Berkooz et al. outline the process of POD on a space of functions [1]. In particular, POD aims to produce a set of vectors (eigenfunctions) and corresponding eigenvalues which best represent a given vector field with coefficients that are uncorrelated (part of a random process) [1].

2.1 SINDy

The most basic systems considered for the algorithm SINDy are those of the form

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)).$$

In order to use the concepts of sparse regression and compressed sensing, discussed above, the authors assume that \mathbf{f} is sparse in some function space [3]. The state vector, $\mathbf{x}(t)$, is collected over some time period and $\dot{\mathbf{x}}$ is either sampled alongside $\mathbf{x}(t)$ or numerically calculated from it. These samplings create the following matrices:

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

and the immediately following $\dot{\mathbf{X}}$. We can then construct a library of candidate functions:

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

Each column of $\Theta(\mathbf{X})$, therefore, represents a candidate function for \mathbf{f} . This therefore sets up the sparse regression problem

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

In particular, $\mathbf{\Xi} = [\boldsymbol{\xi}_1 \quad \boldsymbol{\xi}_2 \quad \cdots \quad \boldsymbol{\xi}_n]$ where each $\boldsymbol{\xi}_i$ is a sparse vector dictating which functions in the function library are active for each of the row equations in \mathbf{f} .

References

- [1] Gal Berkooz, Philip Holmes, and John L. Lumley. “The proper orthogonal decomposition in the analysis of turbulent flows”. In: *Annual review of fluid mechanics* 25.1 (1993), pp. 539–575.
- [2] Josh Bongard and Hod Lipson. “Automated reverse engineering of nonlinear dynamical systems”. In: *Proceedings of the National Academy of Sciences* 104.24 (2007), pp. 9943–9948.
- [3] Steven L. Bruton, 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: <https://doi.org/10.1073/pnas.1517384113>.
- [4] Leonid Rudin, Stanley Osher, and Emad Fatemi. “Nonlinear total variation based noise removal algorithms.” In: *Physica D: nonlinear phenomena* 60 (1992), pp. 259–268.
- [5] Michael Schmidt and Hod Lipson. “Distilling free-form natural laws from experimental data”. In: *science* 324.5923 (2009), pp. 81–85.