

INVESTIGATING SINDY AS A TOOL FOR CAUSAL DISCOVERY IN TIME SERIES SIGNALS

Andrew O'Brien, Rosina Weber, Edward Kim

College of Computing & Informatics, Drexel University

ABSTRACT

The SINDy algorithm has been successfully used to identify the governing equations of dynamical systems from time series data. In this paper, we argue that this makes SINDy a potentially useful tool for causal discovery and that existing tools for causal discovery can be used to dramatically improve the performance of SINDy as tool for robust sparse modeling and system identification. We then demonstrate empirically that augmenting the SINDy algorithm with tools from causal discovery can provide engineers with a tool for learning causally robust governing equations.

Index Terms— Sparsity, causality, dynamical systems, SINDy

1. INTRODUCTION

The Sparse Identification of Nonlinear Dynamics (SINDy) algorithm was proposed by Brunton et al. [1] as a tool for learning the governing equations of a dynamic system from data collected from the system as it evolved through time. It has been used extensively in the engineering and signal analyses communities to recover governing equations in a diverse set of circumstances[2] [3] [4].

Within the artificial intelligence community, a related set of problems related to causality have recently generated considerable interest [5] [6] [7]. Among them is the ability to take as input a data set and return as output the causal relations between variables in that data set is a central challenge known as causal discovery. A wide variety of algorithms are currently in use within the field [8].

Peters et al. [9] formalized the connection between learning a set of ordinary differential equations that govern a system and identifying the causal relations between the variables within a system. In their framework, a variable x can be said to be a direct cause of a variable y if and only if x is present in the ordinary differential equation governing the change in y . Therefore, an algorithm can successfully learn the ordinary differential equations governing a system only if it is also learning the causal relations within a system. This makes SINDy a potentially valuable and, to the best of our knowledge, underexplored tool for causality researchers. The connection should also open the door to scientists, engineers, and

signal process researchers to use the recently developed tools of causal discovery to improve the performance of SINDy as a system identification tool.

In this paper we compare SINDy's ability to identify causal relationships between variables against a sampling of prominent causal discovery algorithms. We first show that under the assumption that all included variables are part of the same dynamical system, SINDy achieves similar performance to the comparison algorithms. Once this assumption is relaxed, we show the SINDy's performance degrades substantially. Finally, we show that augmenting SINDy with a comparison algorithm can make SINDy a state-of-the-art tool for causal discovery and a substantially more robust tool for engineers and signal process researchers performing sparse modeling and system identification.

2. BACKGROUND

2.1. Sparse Regression

Regression is a branch of machine learning that takes some data set $\{(X_i, Y_i)\}_{i=1}^N$ and some parameterized function $f(\cdot, \beta)$ and attempts to use the data set to find a set of parameters β^* such that $f(X_i; \beta^*) \approx Y_i$. Typically $X_i, \beta \in \mathbb{R}^{k \times 1}$ and $Y_i \in \mathbb{R}$. A common approach to regression is assuming f is linear and trying to find the set of parameters that minimize the l_2 norm of the difference vector between the model predictions and the ground truth. This approach is shown in Equation 1 where the rows of X are the transpose of the X_i data vectors and the elements of Y are the Y_i scalars.

$$\beta^* = \underset{\beta}{\operatorname{argmin}}(\|X^T \beta - Y\|_2) \quad (1)$$

There are common problems with this approach [10]. If the regression is under-determined, there are often multiple solutions to Equation 1. In the high-dimensional case, β is often hard to interpret for stakeholders making use of the model solutions. Finally, solutions with large amounts of parameters frequently do not generalize well to out-of-sample data. All of these problems can be solved by imposing an l_0 sparseness penalty to the optimization problem in Equation 1. In practice, it is difficult to perform this optimization directly so the

l_1 norm is used as a relaxation of the l_0 norm as depicted in Equation 2.

$$\beta^* = \operatorname{argmin}_{\beta} (\|X\beta - Y\|_2 + \lambda \|\beta\|_1) \quad (2)$$

2.2. SINDy

The SINDy algorithm takes as input time series data and attempts to reconstruct the dynamical system that generated the data [1]. An autonomous dynamical system is formally defined as $\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t); \beta)$ where t is the time, $\mathbf{x}(t)$ is the state of the system at time t , β is a set of system parameters, and \mathbf{f} is a governing function describing how the system state is changing at time t . Governing functions of dynamic systems are often sums of a few simple functions [11]. SINDy treats learning these governing functions as a sparse regression problem.

Formally, the algorithm takes as input two matrices $\mathbf{X} = [x(t_1), \dots, x(t_m)]^T$ and $\dot{\mathbf{X}} = [\dot{x}(t_1), \dots, \dot{x}(t_m)]^T$ where $x(t_1) \in \mathbb{R}^{nx1}$ is the measurement of the state of the system at time t , $\dot{x}(t_1) \in \mathbb{R}^{nx1}$, is the measurement (or approximation) of the derivative of the system at time t . From \mathbf{X} , a library of candidate functions is constructed and denoted by $\theta(\mathbf{X})$. The optimization in Equation 3 is then performed for each of the N columns of $\dot{\mathbf{X}}$.

$$\xi_k = \operatorname{argmin}_{\xi'_k} (\|\dot{\mathbf{X}}_k - \theta(\mathbf{X})\xi'_k\|_2 + \lambda \|\xi'_k\|_1) \quad (3)$$

Brutnon et al. solve the optimization problem in Equation 3 using the sequential thresholding least squares (STLS) proposed in [1]. The STLS algorithm works by first performing a standard least squares regression of $\dot{\mathbf{X}}$ on θ to get initial values for ξ_k . A threshold c is then chosen such that any element of ξ_k less than c is set to 0. Then a least squares regression is again performed of $\dot{\mathbf{X}}$ onto the columns of $\theta(\mathbf{X})$ that have non-zero coefficients in ξ_k . The process of least squares regression and thresholding is continued until the values of ξ_k converge.

2.3. Causality

The dominant causal framework in artificial intelligence consists of a structural causal model (SCM) M and a corresponding directed causal graph G_M [12]. $M = (U, V, F)$ where V is a set of endogenous variables, U is a set of exogenous variables, and F is a set of assignment functions that assign values to the variables in V using the values of the variables in V and U . A probability distribution over U induces a probability distribution over V as well. For every element of U and every element of V , there is a node in G_M . There is a directed edge from a node n_i to a node n_j if and only if variable i is an argument of variable j 's assignment function f_j . We say that i is a direct cause of j if and only if i is a parent of j in graph G_M .

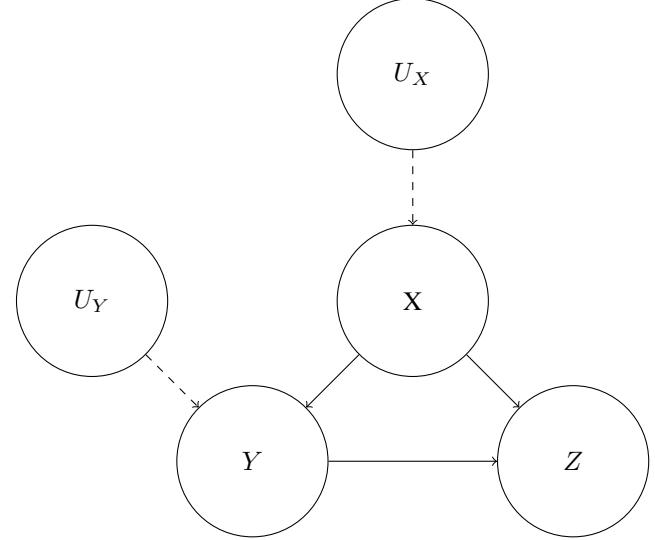


Fig. 1. The causal graph corresponding to the structural equations in Equation 4. Solid lines indicate causal relations between endogenous variables and dashed lines indicate causal relations between an exogenous and an endogenous variable.

For illustrative purpose, consider the following simple example. Imagine some system with variables X, Y, Z, U_X , and U_Y . U_X and U_Y are exogenous variables with standard normal probability distributions. X 's value is a function of U_X . Y is a linear function of X and U_Y . Z is a linear function of Y and X . The following is a potential SCM, $M = (U, V, F)$ where $V = \{X, Y, Z\}$, $U = \{U_X, U_Y\}$, and $F = \{f_X, f_Y, f_Z\}$ such that

$$\begin{aligned} X &:= f_X(U_X) = 10 * U_X + 3, U_X \sim N(0, 1) \\ Y &:= f_Y(X, U_Y) = 2X + U_Y, U_Y \sim N(0, 1) \\ Z &:= f_Z(X, Y) = 5X + 4Y \end{aligned} \quad (4)$$

The corresponding causal graph G_M is given in Figure 2.

The causal framework can be extended to dynamical systems [9]. Consider a set of d ordinary differential equations and initial values assignments. These can be written in as assignment equations with the form given in Equation 5. The causal graph is defined analogously for the dynamical case. This causal interpretation of dynamical systems allows for a causal interpretation of the ordinary differential equations learned by the SINDy algorithm and thus to construct a causal graph from them.

Causal discovery is the process of taking data generated from the distribution defined by a SCM and inferring either the model's causal graph or the structural equations of that model [6]. The problem is difficult because the SCM is underdetermined by the data. Many causal identification algorithms attempt to solve this problem by making assumptions about

the underlying SCM or causal graph and the algorithms can be classified according to the kind of assumption they make and how much they are trying to infer [8]. In causal identification with time series data, methods often assume either that conditional independence in the data implies conditional independence in the causal graph, the structural equations in the SCM have a particular parametric form and the exogenous variables have a particular probability distribution, or assume that the structural equations are a dynamical system with a reconstructable state space, [5] [13].

$$\frac{d}{dt}x_t^i := f^i(x_t^{PA_i}), x_0^i := \alpha_0^i \quad (5)$$

2.4. Comparison Algorithms

In our paper, we use several leading causal discovery algorithms for comparison.

The Convergent Cross Mapping (CCM) algorithm [14] starts by using Takens's theorem [15] as a justification to reconstruct the system manifold, M , twice using time lags from two system variables $x(t)$ and $y(t)$. The reconstructions using $x(t)$ lags and $y(t)$ lags are denoted M_x and M_y respectively. If the ability to predict $M_y(t)$ from points on $M_x(t)$ in some neighborhood around t increases as the number of sampled points increases, then $y(t)$ is a cause of $x(t)$.

The premise behind the Granger Causality (GC) algorithm [16] is that a variable $x(t)$ is a direct cause of $y(t)$ if and only if the past values of $x(t)$ contain unique information about $y(t)$. In practice, bivariate Granger Causality is used whereby if a linear regressive model using the time lags of $y(t)$ and $x(t)$ can produce better predictive value than just the lags of $y(t)$ then $x(t)$ is said to be a cause of $y(t)$.

The PCMCI algorithm [17] is an extension of the PC algorithm [18]. The PC algorithm assumes that conditional independence in the data implies conditional independence in the causal graph. This assumption is called causal faithfulness. The central idea of the algorithm is that, under the assumption of causal faithfulness, variables that are conditionally independent in the data set have no edge connecting them in the graph. Therefore repeated conditional independence tests can be used to "prune" a complete directed graph.

The PCMCI algorithm discovers the causal graph from time series data. It proceeds in two steps. First, it runs the PC algorithm to learn the parent nodes of all the nodes in the graph. Next, it runs an additional conditional independence test to eliminate false positive edges.

The Linear Non-Gaussian Acyclic Model [19] assumes the assignment functions in the SCM are linear and that the exogenous variables have a non-Gaussian probability distribution. Under these assumptions, it can be proven that there is a unique SCM that corresponds to the probability distribution that generated the data and it can be discovered by using independent component analysis.

3. METHODOLOGY

Two sets of experiments were run to test SINDy's ability as an algorithm for causal identification. A final experiment was run to test the ability of existing causal identification techniques to improve SINDy's performance. For the first set of experiments, a diverse set of 6 dynamical systems were chosen which are described in table 1. For each dynamical system, 10 simulations were run. For each simulation, the Runge-Kutta algorithm [20] was used to simulate the system for 1000 time steps. The SINDy algorithm and 4 causal identification algorithms were used to construct the causal graph of the system from the data. The 4 comparison causal identification algorithms were chosen on the basis of diversity of technique, breadth of use, and performance (Section 3.1).

The ground truth causal graph was constructed for the dynamical systems using the framework described in [9]. The ground truth causal graph and learned causal graphs were represented as adjacency matrices. The Hamming distance between the learned adjacency matrix and the ground truth matrix was used as the metric to measure the quality of the learned causal graphs. The Hamming distance between the ground truth causal graph and the learned graph is computed for each algorithm after each simulation of the dynamical system. The average Hamming distance for all 10 simulations is computed for each of the six dynamical systems.

The second set of experiments are similar except Gaussian noise variables are added to double the size of the system. This represents the case in which the scientist using these techniques is uncertain as to which variables are part of the system.

For the third set of experiments, a causal discovery algorithm, PCMCI, was first run as a preprocessing step. Missing edges in the causal graph learned by PCMCI were then encoded as a set of constraints the SINDy algorithm had to observe. The new constraint augmented SINDy algorithm was run on the same set of simulated dynamical systems as in Experiment 1 and Experiment 2.

4. RESULTS & CONCLUSION

The results from Experiment 1 are described in Table 2. On average, SINDy identified the correct graph as well or better than 2.67 of the other four techniques. It had the best average performance in two of the six systems. This is evidence that SINDy is a strong tool for causal identification in dynamical systems under the assumptions that all system variables and only system variables have been measured.

In Experiment 2 (Table 3), SINDy performed much worse relative to the other algorithms. It achieved the best average performance on 0 systems and on average only performed as well .67 comparison algorithms per dynamical system. To the best of our knowledge, the degradation of SINDy's performance in the presence of non-system variables is a weak-

System	System Equations
Lorenz [21]	$\dot{x} = 10(y - x)$
	$\dot{y} = x(28 - z)$
	$\dot{z} = xy - \frac{8}{3}z$
Mankiw-Romer-Weil [22]	$\dot{k} = .01(kh)^{\frac{1}{3}} - .06k$ $\dot{h} = .01(kh)^{\frac{1}{3}} - .06h$
FitzHugh-Nagumo [23]	$\dot{v} = v(.1 - v)(v - 1) - w + 5$ $\dot{w} = .01v - .02w$
Lotka-Volterra [24]	$\dot{n}_1 = n_1(1 - n_1 - 1.09n_2 - 1.52n_3)$
	$\dot{n}_2 = .72n_2(1 - n_2 - .44n_3 - 1.36n_4)$
	$\dot{n}_3 = 1.53n_3(1 - 2.33n_1 - n_3 - .47n_4)$
	$\dot{n}_4 = 1.27n_4(1 - 1.21n_1 - .52n_2 - 1.53n_3 - n_4)$
Pendulum [25]	$\dot{u} = v$ $\dot{v} = -.76\sin(u)$
SIR[26]	$\dot{s} = -\frac{.715}{60}is$
	$\dot{i} = \frac{.715}{60}is - .285i$
	$\dot{r} = .285i$

Table 1. Dynamical systems used to test SINDy as a tool for causal inference.

ness of the algorithm that is currently absent from the literature. It also suggests that engineering and signal processing researchers could greatly improve the performance of SINDy by first using established causal identification techniques as a pruning step when unsure of the exact number of variables in the system. Experiment 3 tested this and demonstrated that the deficiency could be remedied using PCMCi as a preprocessing step. When preprocessing was done, (augmented) SINDy achieved best performance on four of the six systems and on average outperformed 3.67 of the four other methods.

This state-of-the art performance on causal identification provides strong evidence that using SINDy with a preprocessing step is a potentially very useful technique for artificial intelligence researchers in causal identification and engineers in system identification.

This study used numerical experiments only. In the future, data from real dynamic systems should be used.

	PCMCi [17]	LINGAM [19]	GC [16]	CCM [14]	SINDy [1]
Lorenz	.11	.34	.11	.11	.01
MRW	.65	.70	.00	.35	.40
FN	.00	.33	.00	.00	.03
LV	.24	.41	.19	.19	.22
Pendulum	.50	.88	.50	.50	.43
SIR	.49	.40	.44	.44	.43

Table 2. Hamming distance for SINDy and comparison methods for each system

	PCMCi [17]	LINGAM [19]	GC [16]	CCM[14]	SINDy [1]
Lorenz	.06	.43	.35	.65	.64
MRW	.19	.44	.43	.57	.73
FN	.05	.52	.31	.65	.75
LV	.11	.51	.34	.67	.80
Pendulum	.21	.68	.43	.75	.88
SIR	.62	.68	.44	.35	.42

Table 3. Hamming distance for SINDy and comparison methods for each system with noise variables

System	Augmented SINDy
Lorenz	.07
MRW	.13
FN	.14
LV	.11
Pendulum	.17
SIR	.15

Table 4. Hamming distance for SINDy after PCMCi was run as a pre-pruning step

5. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1954364.

6. REFERENCES

- [1] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *PNAS*, vol. 113, no. 15, pp. 3923–3937, 2016.
- [2] Kathleen Champion, Peng Zheng, Aleksandr Y. Aravkin, Steve L. Brunton, and J. Nathan Kutz, “A unified sparse optimization framework to learn parsimonious physics-informed models from data,” *IEEE Access*, vol. 8, pp. 169259–169271, 2020.
- [3] Niall M. Mangan, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz, “Inferring biological networks by sparse identification of nonlinear dynamics,” *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 169259–169271, 52–63.
- [4] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz, “Data-driven discovery of partial differential equations,” *Science Advances*, vol. 3, no. 4, pp. e1602614, 2017.
- [5] Jakob Runge, Sebastian Bathiany, Erik Bolt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D. Mahecha, Jordi Munoz-Mari, Egbert H. Van Nes, Jonas Peters, Rick

Quax, Markus Reichstein, Marten Scheffer, Bernhard Scholkopf, Peter Spirtes, George Sugihara, Jie Sun, Kun Zhang, and Jakob Zscheischler, “Inferring causation from time series in earth system sciences,” *Nature Communications*, vol. 10, 2019.

[6] Peter Spirtes and Kun Zhang, “Causal discovery and inference: concepts and recent methodological advances,” *Applied Informatics*, vol. 3, no. 3, 2016.

[7] Kadierdan Kaheman, Steven L Brunton, and J Nathan Kutz, “Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data,” *Machine Learning: Science and Technology*, vol. 3, 2022.

[8] Christina Heinze-Deml, Marloes H. Maathuis, and Nicolai Meinshausen, “Causal structure learning,” *Annual Review of Statistics and Its Application*, vol. 5, pp. 371–391, 2018.

[9] Jonas Peters, Stefan Bauer, and Niklas Pfister, “Causal models for dynamical systems,” 2020.

[10] Steven L. Brunton and J. Nathan Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2 edition, 2022.

[11] Brian M. de Silva, David M. Higdon, Steven L. Brunton, and J. Nathan Kutz, “Discovering of physics from data: Universal laws and discrepancies,” *Frontiers in Artificial Intelligence*, vol. 3, 2020.

[12] Judea Pearl, *Causality: Models, Reasoning, and Inference*, Cambridge University Press, 2 edition, 2009.

[13] Raha Moraffah, Mansooreh Karami Paras Sheh, Anchit Bhattacharya, Qianru Wang, Anique Tahir, Adrienne Raglin, and Huan Liu, “Causal inference for time series analysis: Problems, methods, and evaluation,” *Knowledge and Information Systems*, vol. 63, pp. 3041–3085, 2021.

[14] George Sugihara, Robert May, Hao Ye, Chih hao Hsieh, Ethan Doyle, Michael Fogarty, and Stephan Munch, “Detecting causality in complex ecosystems,” *Science*, vol. 338, no. 496, pp. 496–500, 2012.

[15] Floris Takens, “Detecting strange attractors in turbulence.,” in *Dynamical Systems and Turbulence*, Warwick, England, 1980, pp. 366–381.

[16] C.W.J. Granger, “Investigating causal relations by econometric models and cross-spectral methods,” *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.

[17] Jakob Runge, Peer Norwack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic, “Detecting and quantifying causal association in large nonlinear time series datasets,” *Science Advances*, vol. 5, no. 11, pp. eaau4996, 2019.

[18] Peter Spirtes, Clark Glymour, and Richard Scheines, *Causation, Prediction, and Search*, MIT Press, 2 edition, 2001.

[19] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen, “A linear non-gaussian acyclic model for causal discovery,” *Journal of Machine Learning Research*, vol. 7, pp. 2003–2030, 2006.

[20] William H. Press, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, W.H. Press, 3 edition, 2007.

[21] Edward Norton Lorenz, “Deterministic nonperiodic flow,” *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[22] N. Gregory Mankiw, David Romer, and David N. Weil, “A contribution to the empirics of economic growth,” *The Quarterly Journal of Economics*, vol. 107, no. 2, pp. 407–437, 1992.

[23] Eugene M. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*, The MIT Press, 2010.

[24] J A Vano, J C Wildenberg, M B Anderson, J K Noel, and J C Sprott, “Chaos in low-dimensional lotka–volterra models of competition,” *Nonlinearity*, vol. 19, no. 10, pp. 2391–2404, 2006.

[25] Steve H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, CRC Press;, 2 edition, 2015.

[26] Stefan Thurner, Rudolf Hanel, and Peter Klimek, *Introduction to the Theory of Complex Systems*, Oxford University Press;, 2018.