Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 52 (2018) 275–290

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

Equations of mind: Data science for inferring nonlinear dynamics of socio-cognitive systems

Rick Dale[a,*], Harish S. Bhat[b]

[a] Department of Communication, UCLA, United States
[b] Applied Mathematics, UC Merced, United States

Received 14 April 2018; received in revised form 15 June 2018; accepted 30 June 2018
Available online 26 July 2018

Abstract

Discovering the governing equations for a measured system is the gold standard for modeling, predicting, and understanding complex dynamic systems. Very complex systems, such as human minds, pose stark challenges to this mode of explanation, especially in ecological tasks. Finding such "equations of mind" is sometimes difficult, if impossible. We introduce recent directions in data science to infer differential equations directly from data. To illustrate this approach, the simple but elegant example of sparse identification of nonlinear dynamics (SINDy; Brunton, Proctor, & Kutz, 2016) is used. We showcase this method on known systems: the logistic map, the Lorenz system, and a bistable attractor model of human choice behavior. We describe some of SINDy's limitations, and offer future directions for this data science approach to cognitive dynamics, including how such methods may be used to explore social dynamics.

1. Introduction

Differential equations define the time evolution of a dynamical system. Their precision inspires some to see such mathematical formulation as critical to scientific understanding. This perspective on differential equations found prominent expression in the dynamical systems approach to cognition of the 1990s (Port & Van Gelder, 1995; Van Gelder, 1995), and was the subject of vigorous debate (Bechtel, 1998; Eliasmith, 1996): "Dynamical systems governed by differential equations are a particularly interesting and important subcategory, not least because of their central role in the history of science." (Van Gelder, 1995, p. 368) Simon (1992) famously expressed an even stronger position, arguing that cognitive explanation is founded on "difference equations" which characterize much cognitive systems research still:

> "For systems that change through time, explanation takes the form of laws acting on the current state of the system to produce a new state – endlessly. Such explanations can be formalized with differential or difference equations. A properly programmed computer can be used to explain the behavior of the dynamic system that it simulates. Theories can be stated as computer programs." (Simon, 1992, p. 160)

Nowadays this mode of mathematical description and explanation permanently inhabits many realms of cognitive science.[1] It was well established even before this recent

* Corresponding author.
E-mail address: rdale@ucla.edu (R. Dale).

https://doi.org/10.1016/j.cogsys.2018.06.020

[1] A reviewer helpfully pointed out that description and explanation should not be confounded, and that equations alone rarely fulfill our conventional notions of "explaining" systems. For simplicity, we do not distinguish between these modes of scientific inquiry – describing and explaining – but assume that differential equations are considered, by a great many, to be important for both.

debate. From the firing of single nerve cells (Hodgkin & Huxley, 1952) and the control of an entire physical body (Beek, Turvey, & Schmidt, 1992; Kugler, Kelso, & Turvey, 1980) to multi-agent models (Richardson et al., 2016), systems of differential equations have long captured a wide variety of psychological phenomena. When we have a set of differential equations for a system, we can predict its time evolution, understand its controlling variables, and identify how system variables interact. These dynamic equations can also participate with other forms of cognitive explanation, such as *mechanistic* explanations of how a cognitive architecture is composed of various particular parts and their interactions (Kaplan & Bechtel, 2011).

Despite their power, differential equations are not always easy to identify. Identification of governing equations can involve an interacting cycle of mathematical invention and empirical tinkering. Guided by intuition, a scientist can happen upon a formulation that generates a covering law (Hempel, 1966). Consequences of this covering law can be explored to consider other formulae in other domains of application. The literature on this is deep and colorful, and excellent reviews of the philosophy and history of science abound (Brush, 1974; Hempel, 1966; Hirsch, 1984; Kuhn, 1962).

Cognitive scientists continue to study and model this psychological process of identifying scientific generalizations and natural law (Addis, Sozou, Lane, & Gobet, 2016; Klahr & Simon, 1999; Langley, 1987). A complementary approach, made possible by computational tools of the day, is to use data and algorithms together to automatically recover dynamical laws. This is what we consider here in this paper. There is an emerging domain, growing rapidly with the advent of data science and machine learning, to precisely recover differential equations from raw data. This offers considerable potential to researchers interested in the dynamics of socio-cognitive systems. It may be possible to use these tools for new and explicit descriptions of system dynamics, even when the data are noisy, and especially when there are plenty of data to be found (a common circumstance these days: Paxton & Griffiths, 2017).

There has been considerable prior work on equation discovery. Motivated by the same points we raise above, researchers over the past two decades have explored different frameworks for automatic recovery of governing equations. Below we first briefly review this past work through influential examples. After this, we introduce a recent simple and elegant formulation of equation discovery (SINDy; Brunton, Proctor, & Kutz, 2016). Based only on transformation of time series data, and simple sparse regression, a researcher can recover equations for their measured systems. In some simple cases, these equations may reflect a full reconstruction of a system's underlying dynamics. More complex cases present other challenges, but in these more complex situations SINDy may still be useful. Below, we introduce SINDy and then showcase how it works on a number of example systems. We also outline its key limitations. After this, we summarize a few outstanding issues in these domains, including how SINDy and related methods could be expanded in the future to help recover governing equations of social systems.

1.1. Some background

There has been considerable prior work on equation discovery. Classic work in cognitive science itself can be found in Langley (1981), who used symbolic cognitive models to infer equations from data. His early model, BACON.3, is meant to capture some important aspects of human scientific activity. More recently, Langley and colleagues (Langley, Sanchez, Todorovski, & Dzeroski, 2002) have also used time series data in an Inductive Process Modeler that can fix certain parameters on population dynamics models. These general approaches fall under the rubric of symbolic machine learning, as a kind of heuristic search. For example, process models of biological systems can include a space of parameters that describe the relationship among variables (Džeroski & Todorovski, 2008). A heuristic search navigates this parameter space under certain constraints to best fit a set of data.

Crutchfield, Shalizi, and others have developed a hidden Markov approach that generates a directed graph that represents a theory of a system from a time series of its behavior (Crutchfield, 1994, 2011; Shalizi & Crutchfield, 2001; Shalizi & Shalizi, 2004). This framework finds transitions between system states in coarse-grained representation of the time series. The result is a kind of compact theory which can describe the time evolution of the system. It also provides descriptive measures of the system, such as its computational complexity. This modeling framework can be used to simulate the relationship between measurement level and theory, and can be likened to a cognitive agent seeking to explain and model a system's dynamics (Crutchfield, 1994; Dale & Vinson, 2013).

There are many related techniques, both in cognitive science and in other realms of the physical sciences. An excellent review can be found in Sozou, Lane, Addis, and Gobet (2017). Much work used clever analysis of time series with *assumed* form of laws to recover particular systems (Bezruchko, Karavaev, Ponomarenko, & Prokhorov, 2001; Bünner, Meyer, Kittel, & Parisi, 1997; Crutchfield & McNamara, 1987; Smith, 1992).

With the advent of large matrix libraries, advanced regression methods are now possible. Schmidt and Lipson (2009) use symbolic regression and motion tracking of physical systems to derive various equations of motion. Example systems included chaotic systems, such as double pendula. Their approach involves extraction of motion time series, and then seeking invariances (correlation structure) among the measured variables according to candidate symbolic functions. The symbolic functions are found via a search through a space of candidates, generated randomly and gradually winnowed down based on best fit (see their Fig. 2). This method is closely related to the one we showcase below, with the primary difference that in SINDy candidate functions are defined comprehensively as a search through *all* possible functions defined by a set of features of interest to the researcher. Modeling more complex systems, Pikovsky has shown how time series of measurements from a neural network can be used to reconstruct the neural network itself (Pikovsky, 2016). Pikovsky's method can reconstruct a connection matrix using time-difference neuron states through solving for a linear system with singular value decomposition (similar to the regression-based method used here).

In many of the examples reviewed here, researchers estimate derivatives numerically. This differencing is key in these approaches (and the one we illustrate below). Recent research has sought to overcome limitations in differencing raw data. For a given signal (e.g., a noisy time series), one will typically find that differentiation amplifies noise while integration filters noise out. Chen, Shojaie, and Witten (2017) have shown how to learn dynamical systems without using numerical differencing or differentiation. In their work, they use the time-integrated or integral equation form of the dynamical system.

Equation discovery seeks to find a dynamical system that best fits a given data set. Each dynamical system is specified by one or more functions – the space of all such functions is typically infinite-dimensional. As in many other nonparametric problems, this leads to a model selection problem. As we increase the dimensionality of the space over which we search for a best-fitting dynamical system, we will decrease training error. However, this fit to training data comes at the expense of generalization to new data. Using techniques from compressed sensing, non-convex optimization, and the statistics of chaotic systems, recent work has investigated conditions under which equation discovery techniques converge to the correct underlying dynamical system (Tran & Ward, 2017; Zhang & Schaeffer, 2018). A recent approach also seeks to find lower-order models of network dynamics by using Bayesian model comparison (Daniels & Nemenman, 2015). These papers reflect an exciting new direction of this work. They will help refine the selection of models among many that may be formulated for a given set of complex data.

Some recent research in cognitive science is inspired by this data-driven reconstruction of lawful regularities. Using first-principle Newtonian mechanics, a "mental landscape" can also be reconstructed via behavioral data (O'Hora, Dale, Piiroinen, & Connolly, 2013; Zgonnikov, Aleni, Piiroinen, O'Hora, & Bernardo, 2017). In this approach, researchers collected a series of computer mouse trajectories towards two possible decisions, at the top left or top right of a computer screen. These computer-mouse data are represented as x, y-coordinates, starting from a set of fixed coordinates ($x = 0,\ y = 0$). Each time series is a decision, with the mouse moving to one final decision point on the left ($x = -A,\ y = B$) or right ($x = +A,\ y = B$) on the computer screen. O'Hora et al. (2013) and Zgonnikov et al. (2017) treat these movements as a kind of "descent" into an attractor on an uneven surface. These attractors model a decision as starting from the peak of a hill, and falling into one of two valleys. Assuming a set of equations with the form of Newtonian mechanics, these decision surfaces can be estimated from these time series data.

Many statistical approaches to model and explore complex data are related to these techniques. For example, the large and still growing application of structural equation modeling (SEM) by social scientists is fundamentally about both exploring and confirming theoretical hypotheses from complex response data (Keith, 2005). SEM models tend to be structural, rather than dynamic, in nature. However, many other still common quantitative methods are closely related to our goals. The notion of a model as a scientific explanation of some phenomenon cannot be neatly distinguished from general statistical practices (Stigler, 2016, Chap. 6). In signal processing and statistical modeling, for example, methods such as Kalman filters, time series regression modeling, and other applications of hidden Markov approaches, offer a rich array of choices (for brief reviews see Brockwell, 2014; Rydén, 2015).

It should therefore be emphasized that many statistical modeling techniques have relevance to understanding underlying relationships. What is unique about this recent trend in data science is to (i) find methods that have some relative transparency of output, (ii) relate output to low-dimensional lawful regularities, which express (iii) dynamical equations that govern a system's behavior. Surely HMMs and other techniques can be placed under this designation. But the synergy among (i)–(iii) reflects a distinct trend. Our brief review of this trend shows a long-standing interest in techniques that have these properties. Recently, rather extensible out-of-the-box methods are now available, and these may increase accessibility to researchers in many areas, such as the social and cognitive sciences. Indeed, with the emergence of machine learning techniques for training models on very large datasets over very large feature sets, it is now possible to fit models with few assumptions about their form. We use a recent example based on this "data science" approach to recovering nonlinear dynamics.

2. Present study

An emerging approach to model building in the computational and social sciences is to exploit the ready availability of high-density measurements and machine learning algorithms to estimate models. We describe one of these techniques, showcase it on simple and known systems, and then develop ideas for how it might be expanded to raw data. Importantly, we offer full source code in R, along with simulations, that can reconstruct these demonstrations, and serve as a foundation for further methods development.[2]

[2] We provide an easy-to-install R library `sindyr`, https://github.com/racdale/sindyr, along with extended summaries leading the reader through each of the demonstrations below.

As a demonstration, we describe a framework called sparse identification of nonlinear dynamics (SINDy) introduced by Brunton et al. (2016). It is perhaps the simplest kind of regression framework for doing equation recovery. Put simply, SINDy takes a set of time series, transforms them in given ways, differentiates them, and then conducts multiple regression. By iteratively thresholding the solutions of ordinary least squares problems, SINDy converges on a set of regression coefficients that define ordinary differential equations (ODEs). SINDy thus serves as an elegant illustration of this data science approach.

As a sparse regression technique, SINDy assumes that a small number of potential variables govern a system's dynamics. It is also very easy to deploy and extend. Brunton, Kutz, and others have shown simple extensions can integrate system forcing (Brunton et al., 2016), model complex fluid dynamics (Rudy, Brunton, Proctor, & Kutz, 2017), integrate latent variables extracted from high dimensional data (Loiseau & Brunton, 2018; see also Schmid, 2010), and extract graph structures of variable relationships (Mangan, Brunton, Proctor, & Kutz, 2016). It is therefore elegantly simple, extensible, and easy to deploy. This is why we showcase SINDy here. It is important to note that we describe some of its limitations below, motivating our discussion about potential future developments.

In the remainder of this paper, we infer dynamic governing equations from simulated data directly using SINDy, in known systems. To make SINDy and related tools relevant to cognitive data, we also propose metrics that may be useful. Indeed, the equations inferred by SINDy may serve as interesting dependent variables themselves, describing the underlying changes in the relationships among interacting variables. In the next section, we introduce this method in more detail, before applying it to sample systems.

2.1. Data processing and application of SINDy

SINDy is a regression-based technique for inferring equations from multivariate time series. It is based on two guiding assumptions. First, governing equations are often "sparse," in the sense that determining the time evolution of one system variable may be a function of a relatively smaller subset of the overall space of potential feature combinations (e.g., higher-order polynomial interaction terms). Second, linear regression can be used to predict estimated first-order derivatives of the system – this is surprisingly sufficient to recover even nonlinear relations. SINDy uses the observed data to estimate derivatives for the system variables, using a method called total-variation regularized derivative (Chartrand, 2011). SINDy can be described in one compact equation, adapted here from Brunton et al. (2016):

$$\frac{d\mathbf{X}}{dt} = f(\mathbf{X})\mathbf{B} \tag{1}$$

X is a matrix of state vectors, with rows reflecting individual samples, and columns the individual state variables. The function f returns a transformation of the observed state variables, such as polynomial terms, trigonometric transformations, and so on. These are typically up to the researcher. Because SINDy is a sparse regression approach, the number of candidate transformations can be very large in number. This is illustrated in Fig. 1.

B is a matrix of coefficients. It is assumed by SINDy that these coefficients should mostly be 0. SINDy uses a thresholding process as **B** is fit from data. Only coefficients that survive this thresholding process retain non-zero values. The remainder are set to 0. This is called sparsification. SINDy can be used with a variety of sparsification techniques, such as the well-known LASSO (Tibshirani, 1996). In the demonstrations here, we use an implementation that carries out a simple sequential least-squares thresholding (Brunton et al., 2016; Quade, 2018), which is simply to say that regression coefficients not above some researcher-fixed threshold (ϵ) are set to 0 iteratively as the model is refit until it converges (i.e., the thresholding ceases to change the model). This process is illustrated in Fig. 2.

Solving for **B** requires predicted variables, on the left side of Eq. (1). In the SINDy formulation, predicted variables are first-order derivatives estimated from the observed data. The simplest way to estimate first-order derivatives is the method of finite differences, which we use here (illustrated also in Fig. 1). We smooth this method by 2 samples in the following way, with **x** representing one column of m-by-n data matrix **X**, and x_t a value at a given time sample t:

$$\frac{d\mathbf{x}}{dt} = \left\langle \frac{x_2 - x_1}{S}, \ldots, \frac{x_k - x_{k-2}}{2S}, \ldots, \frac{x_m - x_{m-1}}{S} \right\rangle, \quad k = 3, \ldots, m-1 \tag{2}$$

The parameter S is simply the time interval of the source sample rate, allowing SINDy to fit real time. Coefficients of SINDy's output will obviously be impacted by this parameter, and so the researcher typically has to attend to the scales of both input and output variables in order to threshold appropriately.

With $f(\mathbf{X})$ and $d\mathbf{X}/dt$ in hand, we solve for **B** via ordinary least squares regression, i.e., by minimizing the squared, column-wise, two-norm error between the left- and right-hand sides of Eq. (1). It is instructive to look under the hood on the mathematics of this procedure, as it also incorporates some critical ingredients of how SINDy and related methods will work. For an n-dimensional vector $\mathbf{v} = (v_1, v_2, \ldots, v_N)$, let

$$\|\mathbf{v}\|_2^2 = \sum_{j=1}^{n} v_j^2$$

be the squared two-norm. Let $\mathbf{B}_j$ denote the j-th column of the matrix **B**, and let $d\mathbf{X}_j/dt$ denote the j-th column of the matrix $d\mathbf{X}/dt$. We find $\mathbf{B}_j$ by minimizing

$$\left\| \frac{d\mathbf{X}_j}{dt} - f(\mathbf{X})\mathbf{B}_j \right\|_2^2$$

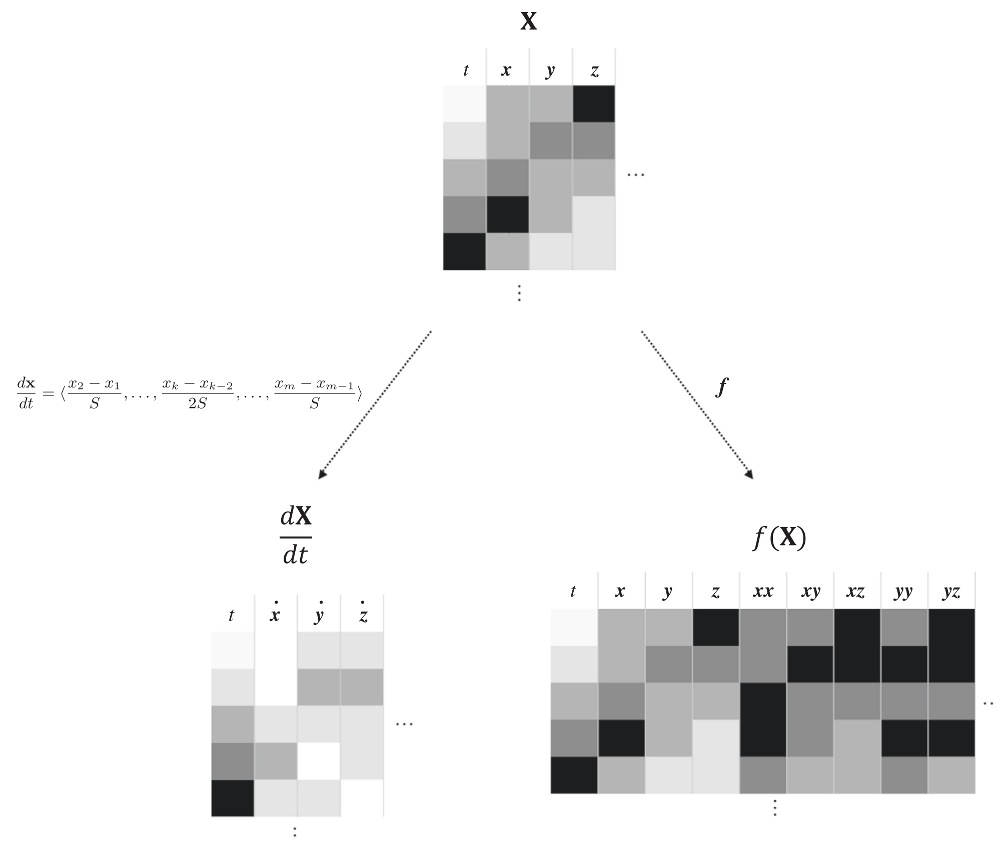


Fig. 1. A representation of the first step of the SINDy algorithm. The researcher collects samples of data in matrix **X**. These might be several variables (**x**, **y**, etc.) collected at a given sample rate. The researcher then obtains a representation of the first-order derivative of the data (bottom left), and then uses a feature function *f* to generate a set of derived variables of interest (bottom right). The function *f* in many cases may simply multiply these original variables (generating polynomial terms). Note: Cell shading is meant to connote variation in any such data matrices.

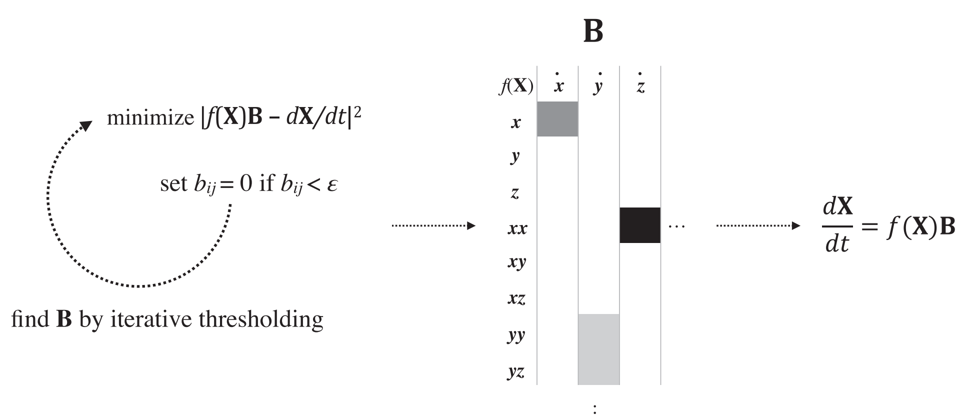


Fig. 2. In the second step for SINDy, an iterative thresholding approach is taken to fit a matrix of coefficients, **B**. **B** is a sparse matrix, where all coefficients remaining are above some threshold, ϵ. Once this matrix **B** is established, we can fully state the estimated dynamic system.

with respect to $\mathbf{B}_j$. This is a least squares regression problem with solution

$$\mathbf{B}_j = \left(f(\mathbf{X})^T f(\mathbf{X})\right)^{-1} f(\mathbf{X})^T \frac{d\mathbf{X}_j}{dt}. \quad (3)$$

In the theory of linear least squares regression, this is called the solution to the *normal equations*. The following three remarks are important for our purposes here:

1. We have assumed that the matrix $f(\mathbf{X})$ has full column rank. As long as the user chooses a set of linearly independent transformations of state variables, this condition will be satisfied. For instance, choosing all polynomial transformations up to a fixed order will result in a $f(\mathbf{X})$ matrix with full column rank.
2. Let $\mathbf{A} = f(\mathbf{X})^T f(\mathbf{X})$ and let $\mathbf{C} = f(\mathbf{X})^T d\mathbf{X}/dt$. Then $\mathbf{B}$ is the solution to the system $\mathbf{AB} = \mathbf{C}$. This solution can be

obtained numerically using a variety of matrix factorization algorithms that have been well-known and built into tools such as `MATLAB` and `R` for some time. In `MATLAB`, this procedure is often carried out with what is known as the backslash operator. In our implementation, we use the `mldivide` function from the `pracma` library in `R`. In particular, full inversion of the **A** matrix is unnecessary.

3. In case $f(\mathbf{X})$ does not have full column rank, the solution $\mathbf{B}_j$ still exists and is given by $\mathbf{B}_j = f(\mathbf{X})^{\dagger} d\mathbf{X}_j/dt$, where $f(\mathbf{X})^{\dagger}$ is the Moore–Penrose pseudoinverse of $f(\mathbf{X})$. This pseudoinverse exists for any possible matrix $f(\mathbf{X})$ and can be computed via the singular value decomposition (SVD).

The first linear regression step in SINDy, illustrated in Fig. 2, is to compute $\mathbf{B}_j$ as above for each column j. The matrix **B** will typically be filled with non-zero coefficients. SINDy then forces to 0 any value in the matrix that is below the threshold ϵ. Let us call the resulting matrix $\mathbf{B}^{\text{old}}$.

SINDy then refits **B**. Suppose we are trying to find the j-th column of this new **B**. We form a version of $f(\mathbf{X})$ that retains only those columns of $f(\mathbf{X})$ that correspond to non-zero entries of the j-th column of $\mathbf{B}^{\text{old}}$. Using this version of $f(\mathbf{X})$, we compute $\mathbf{B}_j$ as in Eq. (3). Carrying this out for all columns j, we form **B**. This procedure is continued until convergence, and is therefore a kind of iteratively thresholded linear regression.

In a nutshell, this is SINDy. We now demonstrate how it works on a series of known systems, including one related to cognitive dynamics.

3. Demonstration of SINDy in known systems

We first revisit two simple examples from Brunton et al. (2016). We show that the classic logistic map can be fit with SINDy. By sampling a subset of values of its single control parameter, SINDy recovers a close approximation of the closed-form update equation. We then showcase something similar with the Lorenz system. Sampling its three system variables under parameter settings that generate chaotic behavior, SINDy can accurately recover the differential equations, including nonlinear terms, that generate its behavior. Following these demonstrations, a known system recently used to model cognition is fit with SINDy. We use a model from Tuller, Case, Ding, and Kelso (1994) and adapted by Duran and Dale (2014) to capture forced-choice dynamics in a cognitive task.

3.1. Logistic map

The logistic map is among the simplest dynamic systems capable of showing a wide range of interesting behavior, such as bifurcation and transition into chaotic behavior. It is defined by the following discrete update equation over a single state variable:

$$x_{t+1} = ax_t(1 - x_t) \tag{4}$$

The control parameter a determines this univariate system's behavior, revealing various regimes such as a point attractor, period bifurcation, and chaos. In Fig. 3, we show how this model is run under different values of this parameter. When a enters certain value ranges, the system variable of the logistic map x_t can reveal several interesting behaviors, including chaos (illustrated with $a = 2.1$ and 3.95). In Fig. 3, we show the different values that x_t can take on under values of a. For low values of a, the logistic map behaves like a point attractor: It converges on a single value. For example, when we set $x = .01$ and run it under parameter $a = 2.1$, the system converges on a particular value near $x \approx 0.55$. As a increases, x_t starts to take on two or more oscillating values. Once a reaches a threshold of about 3.57, the system enters a chaotic regime, taking on many values. This simple model has been applied to a variety of dynamics, including population change and social dynamics (see Richardson, Dale, & Marsh, 2014, for a brief review). Here we reconstruct this simple equation using SINDy in a manner consistent with Brunton et al. (2016).

SINDy can reconstruct this basic formula through the sparse regression technique describe above. To do this, we initialize the parameter a to the value 2.1 and slowly increment it to 3.99. As we increase a, at each value for a, we measure the logistic map for 200 iterations, collecting x_t and a as variables. Collecting these data into two columns (a, x) creates a data matrix (**X** in Fig. 1) of 7599 rows by 2 columns.[3] As in Brunton et al. (2016), we define the first-order derivative of this system as $\frac{dx}{dt} = x_{t+1}$. Equipped with this raw data and after computing the derivative, we produce a feature matrix $f(\mathbf{X})$ that contains all individual and multiplicative polynomial terms up to the third order (illustrated in Fig. 4). We then solve for the coefficient matrix **B** through iteratively thresholded linear regression, as described in the prior section. SINDy is surprisingly accurate in recovering the logistic map.[4]

3.2. The Lorenz system

The Lorenz equations, famous in the atmospheric sciences, are also well known to dynamical systems researchers in cognitive science, as they again illustrate various properties of dynamic systems. These equations reveal multiplicative (nonlinear) relationships among three state

[3] It is not a multiple of 200 because, as noted, we look one step ahead to estimate $d\mathbf{X}/dt$, and so lose one data sample.

[4] SINDy does assume a number of critical details about the sampled system, which will have to be remarked on later in more complex measurements – SINDy does require the researcher to collect adequately diverse measurements under functionally relevant conditions. By "functional," we mean that the sampled data will in some comprehensive manner sample the array of behaviors that a mathematical function will exhibit. In the case of the logistic map, SINDy requires a good sampling of values of a to work adequately. Similarly, the Lorenz reconstruction works only in an adequate sample that captures the kind of interval variable that reflects the functional structure of the governing equations.

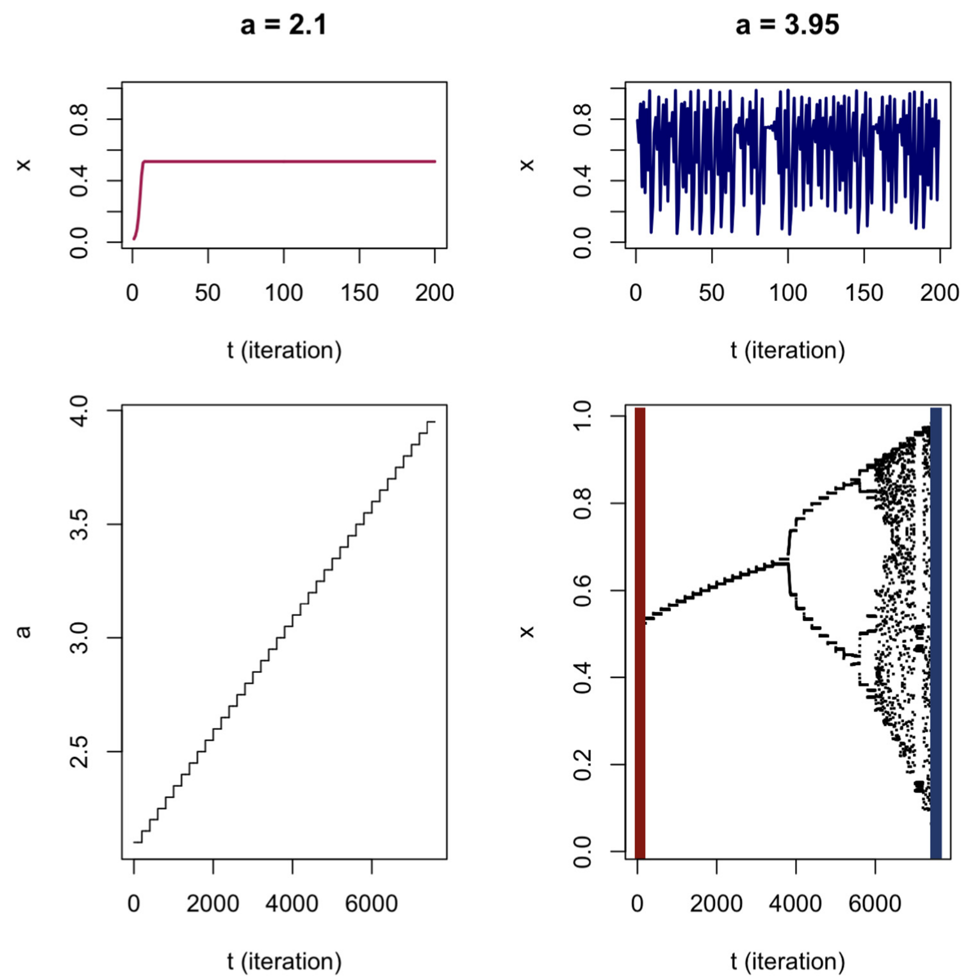


Fig. 3. Top: We illustrate iterating the logistic map under two values of the control parameter a. When $a = 2.1$ and x begins at .01, the logistic map converges on a particular value. When a increases, the behavior of x can become more complex, and under $a = 3.95$ it displays a chaotic regime, traversing a wide range of values. Bottom: We stored values of x_t as the logistic map was iterated over time. After each 200 iterations, we slightly incremented the control parameter a by.05 to obtain a sampling of its range of possible behaviors. The two bottom graphs represent the data **X** submitted to SINDy. The blue and red colors shown are for convenience – indicating in which region the examples (Top) derive from the overall data (Bottom).

variables (x, y, z). The relationship among these variables is expressed in the following way:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = x(r - z) - y \quad (5)$$
$$\dot{z} = xy - \beta z$$

The variables σ, r, and β determine the shape of the system's dynamics, and can be given interpretations relevant to the form of atmospheric flow. After setting these parameters to particular values, and determining an initial state (x_0, y_0, z_0), these equations precisely define the time evolution of the system. We illustrate the famous "butterfly" attractor in Fig. 5. This behavior of the Lorenz system can be clearly seen under various settings, such as control parameters $\sigma = 10$, $r = 28$, and $\beta = 2.6$. We show this in Fig. 5. These three variables also serve as the data input **X** for SINDy.

We ran the Lorenz system under these parameters for 50,000 time steps (each of size 10^{-3}), and use only these three variables to populate **X** and compute derivatives and parameters. The recovery of these equations via SINDy is conducted in a similar way, using the method of finite differences described in the prior section. The sparsification of **B** leads to a very good fit, illustrated in Fig. 6. As in the logistic example, we show a regeneration of the Lorenz butterfly under these reconstructed parameters by initializing (x_0, y_0, z_0) to the first step of the observed data. This is shown in the simulated data, in Fig. 6. It bears a close resemblance to the underlying dynamics that we expected.

The logistic map and Lorenz are well-known demonstrations, shown in Brunton et al. (2016), but we reconstruct them here for illustration. In the next section we showcase how this data science approach could be used

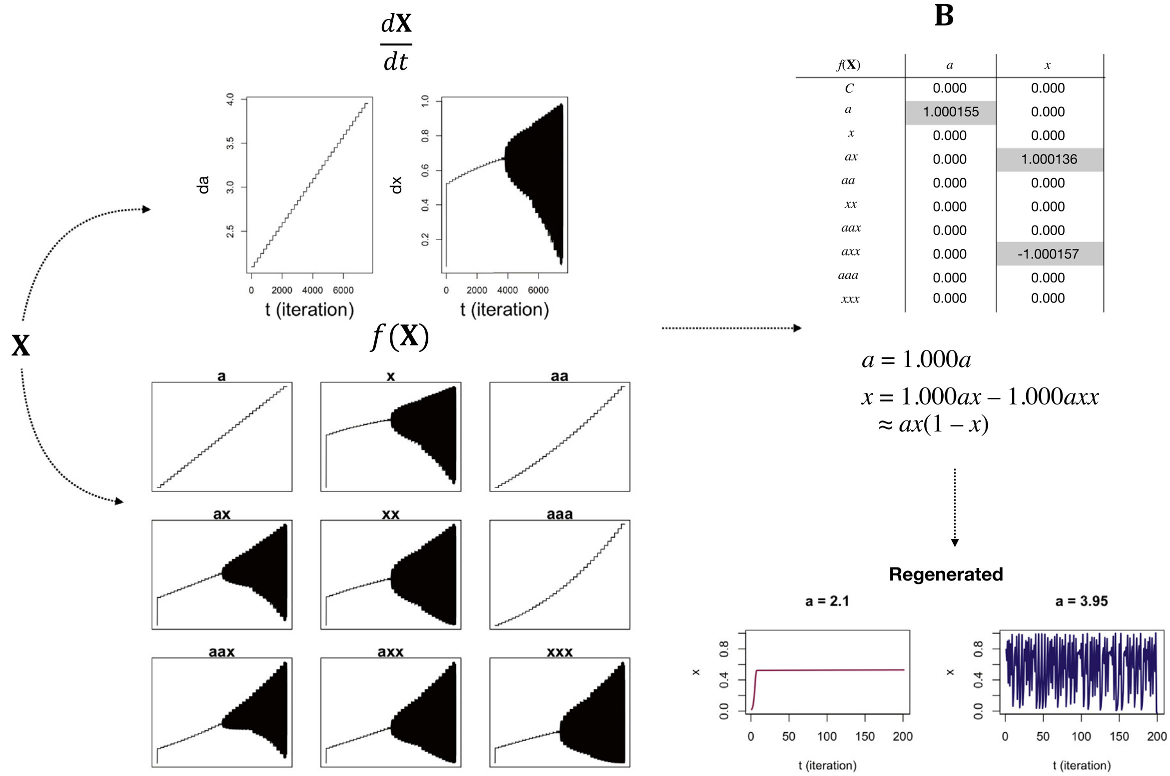


Fig. 4. Left: The series of samples of (a, x) gathered in **X** are transformed into a set of third-order polynomial features, shown in $f(\mathbf{X})$, and first-order derivatives. Because this system is discretely updated, $d\mathbf{X}/dt$ is taken to be the values one iteration ahead. Top right: These are then used to reconstruct a set of coefficients in **B**, shown in the top right. The equation for the logistic map can be approximately recovered by **B**, to a good approximation. Bottom right: This raw formulation in **B** (i.e., *all* decimal places, direct from the fit) can then be used to generate system behavior under the same values for a in Fig. 3.

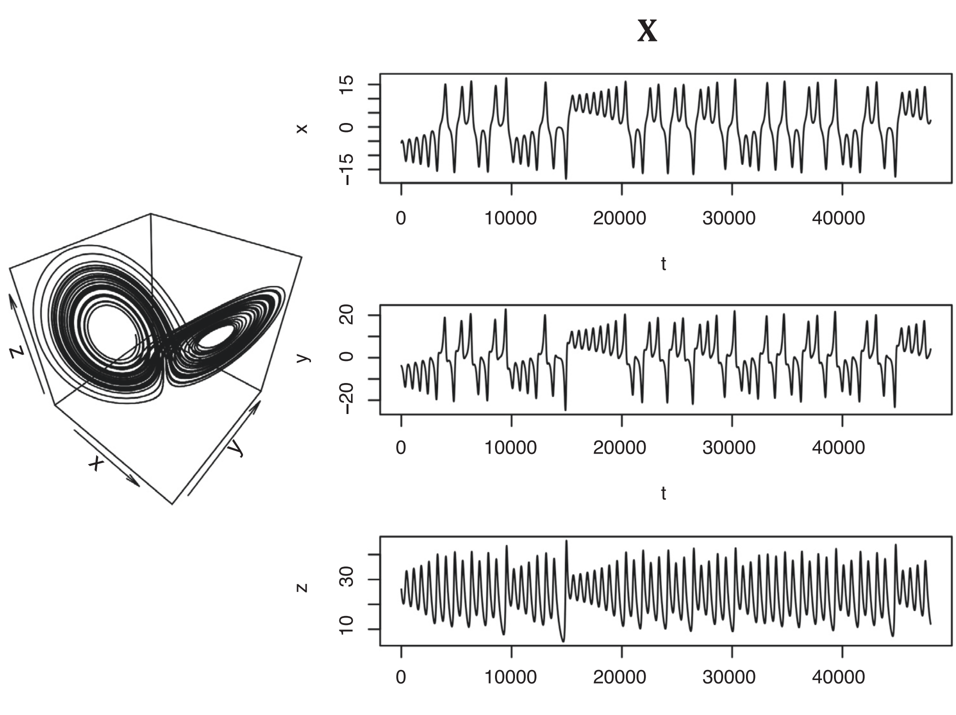
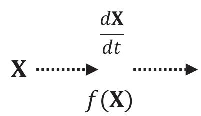


Fig. 5. We accumulated 50,000 time steps (each of size 10^{-3}) of the Lorenz state variables x, y, and z. We use a common setting for the control parameters, generating a chaotic regime: $\sigma = 10, r = 28$, and $\beta = 2.6$. Below **X**, we have plotted the data used as input to the SINDy algorithm.



B

$f(\mathbf{X})$	$\dot{x}$	$\dot{y}$	$\dot{z}$
C	0.000	0.000	0.000
x	-10.011	28.251	0.000
y	9.999	-1.120	0.000
z	0.000	0.000	-2.604
xy	0.000	0.000	1.000
xz	0.000	-1.004	0.000
xx	0.000	0.000	0.000
yz	0.000	0.000	0.000
yy	0.000	0.000	0.000
zz	0.000	0.000	0.000
xyz	0.000	0.000	0.000
xxy	0.000	0.000	0.000
xyy	0.000	0.000	0.000
xxz	0.000	0.000	0.000
xzz	0.000	0.000	0.000
xxx	0.000	0.000	0.000
yyz	0.000	0.000	0.000
yzz	0.000	0.000	0.000
yyy	0.000	0.000	0.000
zzz	0.000	0.000	0.000

$$\dot{x} = -10.01x + 10y \qquad \approx \sigma(y - x)$$

$$\dot{y} = 28.25x - 1.12y - 1.00xz \qquad \approx x(r - z) - y$$

$$\dot{z} = -2.60z + xy \qquad \approx xy - \beta z$$

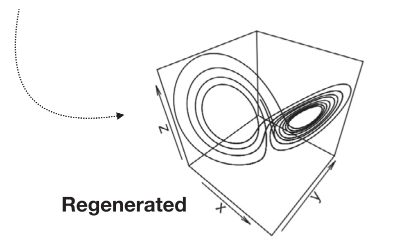


Fig. 6. We use the method of finite differences described above, and generate the third-order polynomial feature space. Running the sparse regression technique generates a close approximation to the coefficients **B**, which define the set of equations for the Lorenz. Beneath the reconstructed equations, we show a simulation of the Lorenz system under these reconstructed values. To do this, we set (x, y, z) to the first values in **X**, then iterate only under our reconstructed model for 10,000 steps. The result is shown here in the 3D plot.

on a human experimental context by drawing from a dynamic simulation of choice behavior.

3.3. Recovering bistable attractor models of cognitive dynamics

A well-known approach to capturing human behavior is to use equations that define a two-well attractor. The Haken-Kelso-Bunz model of bimanual coordination is perhaps the best known (Haken, Kelso, & Bunz, 1985). Another that is well suited to SINDy is that of Tuller et al. (1994). This model is defined by a potential landscape equation in which two minima describe the movement of a system into one of two stable states. This general strategy has offered a fruitful means of capturing the dynamics of many cognitive and perceptuomotor processes (e.g., among many: Frank, Richardson, Lopresti-Goodman, & Turvey, 2009; Raczaszek, Tuller, Shapiro, Case, & Kelso, 1999; Schmidt, Carello, & Turvey, 1990; Tuller et al., 1994; van Rooij, Bongers, & Haselager, 2002), including high-level cognitive processes, such as choosing a social perspective during interaction (Duran & Dale, 2014). The Tuller et al. (1994) variant of this approach is based on a potential landscape defined by the equation:

$$V = kx - \frac{x^2}{2} + \frac{x^4}{4} \tag{6}$$

This is visualized in Fig. 7. Duran and Dale (2014) use the first-order derivative of this function to define movement across the single state variable in the following way:

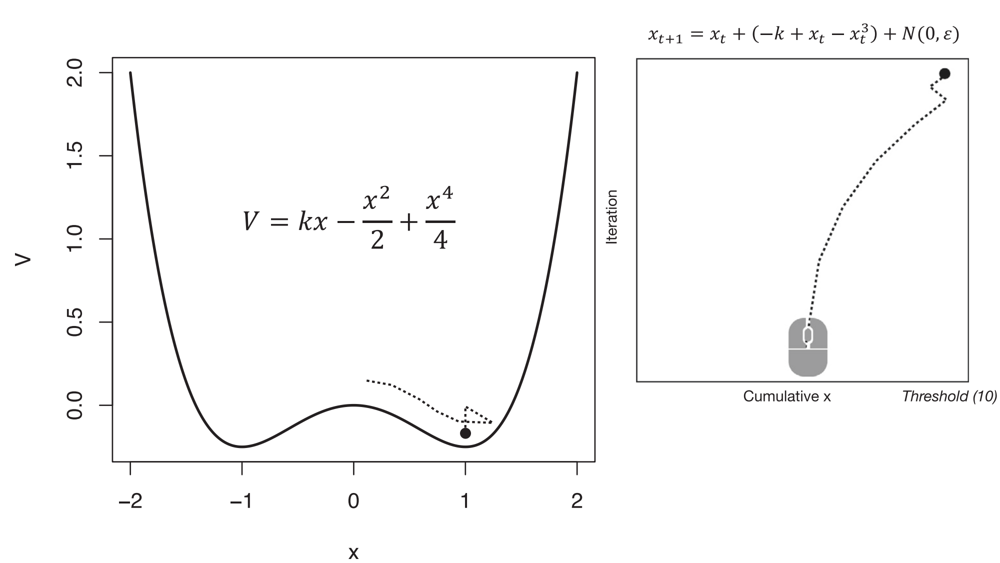
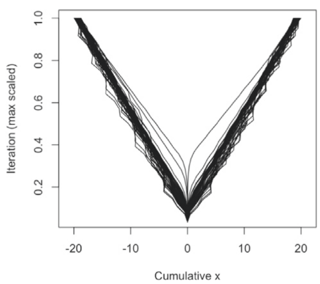


Fig. 7. On the left, the potential landscape from Tuller et al. (1994) that can be used for iteratively modeling a two-well attractor system. On the right, we illustrate how this can be used to model behavior, such as tracking human activity through the computer mouse to a decision point.



$\frac{d\mathbf{X}}{dt}$ $f(\mathbf{X})$

B

$f(\mathbf{X})$	k	x
C	0.00	0.00
k	1.0000	-1.0017
x	0.00	1.9996
kx	0.00	0.00
kk	0.00	0.00
xx	0.00	0.00
kkx	0.00	0.00
kxx	0.00	0.00
kkk	0.00	0.00
xxx	0.00	-0.9998

$k = 1.00k$

$x = -1.00k + 1.9996x - 0.9998xxx$

$\approx x + (-k + x - x^3)$

Fig. 8. On the left, we illustrate the rise to threshold of 100 decisions, collected as two system variables (k and x). We treat this as a discretely updated system like the logistic map, and define the first-order derivative in the same manner. This approximately recovers the update equations of the two-well system. The model in this case uses random values of k between -0.5 and 0.5 and noise of 0.01.

$$x_{t+1} = x_t + \alpha(-k + x_t - x_t^3) + N(0, \sigma) \quad (7)$$

Here $N(0,\sigma)$ represents a source of Gaussian noise with mean 0 and standard deviation of σ. α is a parameter that determines how rapidly the simulation settles. For our data, we chose $\alpha = 1$ to simplify the equation we are reconstructing. When initialized at the center (saddle point) of the potential, this system variable will begin to change until it converges towards the minima seen in Fig. 7. The settling process of this simple formulation models behavioral or cognitive dynamics towards a decision, and has been used to model multiple timescales in cognitive processing (Duran & Dale, 2014), such as the movement of the arm or computer mouse towards an option (Spivey & Dale, 2006). This is also illustrated on the right side of Fig. 7. The model can be seen as implementing a kind of decision carried out by hand, moving towards a target to render a response. Duran and Dale (2014) utilized this framework to model this kind of dynamic tracking of decision processes, specifically how participants chose a perspective in a social task.

We simulated many such iterations to a threshold. Akin to a psychological experiment, we simulated 100 trials of this settling process, and reconstructed this equation for a single extended performance by the model. For each trial, we randomly selected k to be between $-.5$ and $.5$. Here k is a parameter that reflects the "tilt" of this potential. If k is negative, it slightly tilts the potential towards a $x > 0$ outcome; when it is positive, toward a $x < 0$ outcome. For each iteration, after randomly choosing k, the model was initialized at $x = 0$. For this first reconstruction, we assume only a small source of noise, with $\sigma = .01$ (revisited below).

Using the update Eq. (7), we track the value of x as it changes. This x then accumulates in a sum term ($\sum x$), and when that sum reaches + or -20, the threshold is reached, and the process begins again. We carried out this process 100 times. A visualization of these trajectories is shown in Fig. 8. We collected k and x as our system variables, and used SINDy to reconstruct the original update equation. Importantly, when we combine the model's measurements across each trial, we omit the time points that would align the derivative at the last time sample with the start of the next trial.[5] Results are shown in Fig. 8.

These surprising results should be considered alongside the time series modeled here. In this two-well decision model, these simulated trials are often very short, with fewer than 20 to 30 time iterations for the model. When only 100 of them are collected from this model, we nevertheless obtain a close approximation of the governing update rule for the state variable.

This model can also be simulated as a kind of cumulative diffusion process, in which we stop the iteration once a cumulative sum of system states achieves a threshold of some constant (illustrated in Fig. 7, right). Indeed, the methods we introduce here may be useful in fitting such diffusion models of reaction time (Ratcliff, Van Zandt, & McKoon, 1999), which can also be fit using statistical methods (e.g., Vandekerckhove & Tuerlinckx, 2007). Procedures for fitting these models have been central since their inception (Link, 1975), and a technique like SINDy may be an especially flexible framework in which to explore their variants.

4. Extending related methods, and other issues

But what if we don't know the governing equations underlying some data? In fact, what if it seems reasonable to suspect that the equations won't be simple, and perhaps not even stable, over a period of time? In addition, what are the impacts of noise? Brunton et al. (2016) showed that in these model systems, introduction of small amounts of noise does not disrupt equation discovery. In our own explorations of SINDy, we find that noise at higher levels – levels perhaps reasonable for data in cognitive science – may render SINDy's output unstable. Here we explore some of these general issues. We briefly describe several ways methods like SINDy may be enhanced. First, we consider more direct integration of stochasticity. Second, we propose use of SINDy-like techniques not as pure equation

[5] This small adjustment to the data is critical for a strong fit of the choice model.

discovery but as system description. In each case, we explore modifications of the model systems above.

4.1. Rigorous integration of stochasticity and dynamic fields

One challenge to a data science method like SINDy is handling stochasticity gracefully. SINDy's output can be impressive for some models, but we find that wider exploration of SINDy under noise leads to more graded outcomes. We show this in Fig. 9 below. In fact, in the human behavioral simulation in Fig. 8 above, we find that SINDy is fairly stable in the face of some noise, but can become quite unstable as noise increases. Fig. 9 shows goodness of fit statistics of the obtained equations with those expected (fit **B** with expected **B**).

These data suggest that SINDy is indeed quite sensitive to noise in some contexts, and reconstruction is not assured. A recent data science approach, that seeks to overcome this limitation, is to infer stochastic dynamical systems from data. In this framework, drift and diffusion processes are integrated with the general form of differential equations in order to better accommodate high amounts of noise, potential irregularities in sampling, missing data, and more. The resulting technique has been termed density tracking by quadrature (DTQ; Bhat et al., 2018; Bhat, Madushani, & Rawat, 2016b). The underlying, continuous-time model can be expressed as

$$d\mathbf{X}_t = f_{\text{drift}}(\mathbf{X}_t)dt + f_{\text{diffusion}}(\mathbf{X}_t)d\mathbf{W}_t. \quad (8)$$

Here $d\mathbf{W}_t$ is an increment of Brownian motion, which we can think of as a mean zero, variance dt Gaussian noise process. The presence of the $d\mathbf{W}_t$ term causes the resulting model to be inherently probabilistic. Given an initial condition $\mathbf{X}_0$, the solution at time $t, \mathbf{X}_t$, is a random variable. Note that when f_{drift} or $f_{\text{diffusion}}$ are nonlinear, the distribution of $\mathbf{X}_t$ can be multimodal and/or non-Gaussian.

The two f functions encode the drift and diffusion components of the model; these functions are allowed to depend nonlinearly on the state variables of the system. We estimate these functions using a matrix of transformations and regression coefficients, just as in SINDy. Note that the SINDy model considered above is a special case of this model in which $f_{\text{diffusion}}$ is identically zero.

Because the model is probabilistic, given time series data, we can use the model to derive a likelihood function. The parameters in this likelihood will be regression coefficients whose values pin down the functional forms of f_{drift} and $f_{\text{diffusion}}$. We then find the parameters that numerically maximize the log likelihood. After we fit the model in this way, we can use it to estimate the probability $P(X_{t_i})$, namely the probability density around some state of the system at time t_i.

This new method can be described as a kind of dynamic field inference, in which the probability distribution around system states at time increments can be computed. Bhat and colleagues have had success in applying this to complex behavioral data (Bhat, Madushani, & Rawat, 2016a), and designed an R package that currently works on low-dimensional systems (Bhat et al., 2016b). This framework can accommodate model selection by penalizing the likelihood function, equivalent to putting a prior distribution on the coefficients. This approach is highly suitable for modeling systems with significantly lower signal-to-noise ratios than accommodated by SINDy.

One obvious way to accommodate noise is to collect more data. This simple intuitive approach is illustrated in the middle panel of Fig. 9. Let us assume again a high level of noise (noise, $\sigma = 1.5$). In the middle panel, we show the impact of trial number on model fit. By increasing the number of trials, for a fixed value of the noise parameter, the fit is again rendered perfectly, with RMSE near 0.

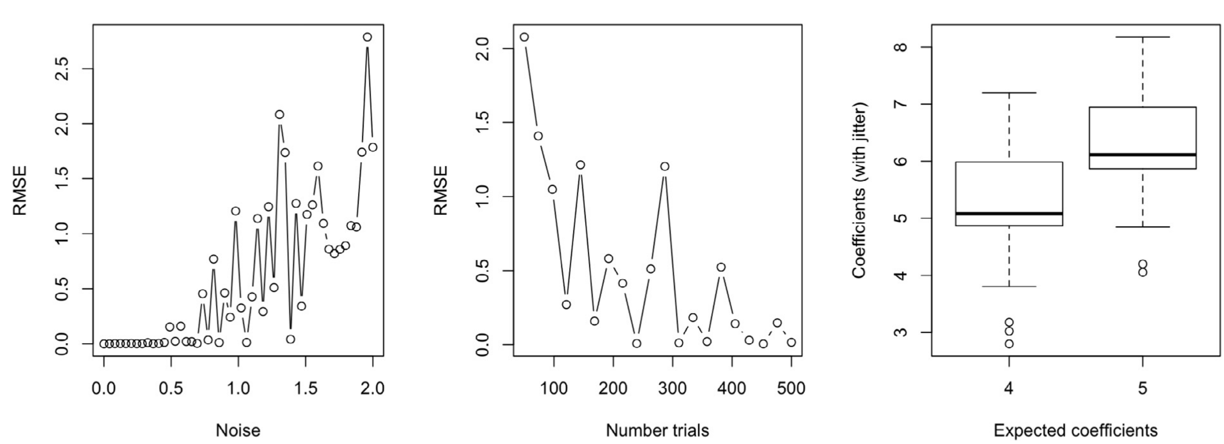


Fig. 9. Left: We estimate the same model in Fig. 8, but under increasing levels of noise. As noise is increased to a fraction of the system cumulative threshold ($threshold = +/-20$), the SINDy fit becomes highly unstable, indicated by the increase in root mean squared error (RMSE). Middle: When we collect more trials of this dynamic system, SINDy is much more robust to the high level of noise used. Here we show RMSE dropping to near 0 if several hundred trials are collected, even under high noise, $\sigma = 1.5$. Right: If we explore variants of the Tuller et al. model (1994), then even under high levels of noise, SINDy obtains the *relative* complexity of these models. This suggests SINDy could be used as a descriptive tool for system complexity.

4.2. New descriptive measures of systems

As noted in the prior section, injection of growing noise can introduce considerable issues in these methods. In much human behavioral data, in which we expect considerable variability, this will pose challenges to SINDy. There are few existing applications of SINDy to raw data in real research contexts. Our demonstration of SINDy here is no exception to this trend, as we have focused on model systems. In our initial explorations of SINDy with raw data, equations are not stable. In this case, it could be that SINDy is simply incorrect about recovered equations. An additional possibility is that the governing processes of a human cognitive system are dynamically changing themselves. This would be suggested by graph-dynamic approaches proposed by Saltzman and colleagues (e.g., Saltzman & Caplan, 2015; Saltzman & Munhall, 1992). We could bridge their proposal, likening their graph-theoretic structures to fit matrices such as **B**, leading to the expectation that **B** itself is changing. In other words, system descriptions of an extended cognitive task should expect that context and task will lead to radical drift in control dynamics.

In these cases, SINDy could be used instead as a descriptive model. Let us take the same model from Section 3.2, and use a level of noise that, as shown in Fig. 9, causes the fit to become unreliable (noise, $\sigma = 1.5$). We define a different model with an additional influence of a second-order polynomial term:

$$x_{t+1} = x_t + (-k + x_t + x_t^2 - x_t^3) + N(0, \sigma) \tag{9}$$

In Fig. 9, right panel, we show that SINDy can distinguish between these models in terms of complexity. We ran 100 models under each condition, and estimated the complexity by counting the non-zero coefficients. These two samples can now be treated as independent and their complexity compared statistically. In this case, SINDy's output implies that the more complex model has a greater average reconstructed complexity ($M = 6.34$) than the simpler prior model ($M = 5.29$), $t(198) = 7.9$, $p < .00001$.

Adapting SINDy in this way may permit researchers to see the models as an *approximate* description of the system's underlying dynamics, rather than a reflection of the precise underlying equations. By applying it in a windowed fashion over data or simulations, this approach may also permit identification of the onset or offset of particular control variables as a system's dynamics are changing. It is also suitable to a heavily experimental discipline such as cognitive science, in which SINDy estimates could be compared on a relative rather than absolute basis, to compare conditions. This could facilitate a solution to the problem that began this section. If one can neither know nor estimate the precise equations, we can at least get a sense of their relative complexity under different contexts. We could draw an analogy here with the concept of Kolmogorov complexity, often difficult to define operationally except under particular contexts. In the context of estimated differential equations, Kolmogorov complexity could be analogized as the number of non-zero coefficients. This sort of approach was taken by Crutchfield (1994), under the term computational complexity, based on summary measures of a model recovered from data.

4.3. Social dynamics: coupled logistic map

Applications of SINDy have extended to the reconstruction of biochemical networks (Mangan et al., 2016). The intuition that drives this application is that the terms in reconstructed equations reflect relationships across system components. If two state variables input to SINDy come from two different systems (or more), we can use the recovered terms of the equations to define a graph structure. We demonstrate an application of this here. We use a model of social dynamics initially proposed by Buder (1991). He used coupled logistic maps to describe how two model "conversants" may follow each other in a behavioral space. They are defined by two state variables x and y, and influence each other in the following way:

$$\begin{aligned} x_{t+1} &= a_x x_t(1 - x_t)(1 - \pi_{y\to x}(x_t - y_t)) \\ y_{t+1} &= a_y y_t(1 - y_t)(1 - \pi_{x\to y}(y_t - x_t)) \end{aligned} \tag{10}$$

The parameter $\pi_{x\to y}$ reflects the amount of connectivity from system x to system y. When x and y maximally influence each other, $\pi_{x\to y} = \pi_{y\to x} = 1$. When $\pi_{x\to y} = \pi_{y\to x} = 0$, these equations reduce to the standard logistic map shown above, under the same control parameter a. For simplicity, we assume the two control parameters a_x and a_y are equal. In each equation, the right-most term consists of a difference. For y to x, the term in question is $(1 - \pi_{y\to x}(x_t - y_t))$. This term can be interpreted as forcing x to move *towards* y according to their difference. The systems therefore move about their phase spaces and mutually influence each other. Here we demonstrate that SINDy can detect the presence and nature of *influence* in this simple model of social dynamics. We also show that the terms of the SINDy output, in **B**, can be visualized in a graph-theoretic form, as a network of social dynamics.

We ran the same kind of simulation as for the logistic map above. Varying a from 2.4 to 4 (Buder, 1991) in increments of .01, we extract three state variables of this system (a, x, y), and then seek a reconstruction through SINDy. This serves as an interesting additional test of SINDy, because the polynomial order of this "social" system is quite high. When expanding the definition of x_{t+1} in (10), simplifying with $\pi_{y\to x} = 1$, we obtain

$$x_{t+1} = ax_t - 2ax_t^2 + ax_t^3 + ax_t y_t - ax_t^2 y_t \tag{11}$$

This already requires a polynomial term up to the 4th order. Indeed, in the original Buder (1991) formulation, and in other applications of this model (Dale, Warlaumont, & Richardson, 2011), we assume that the discrete updates to these systems starts with x, and then with

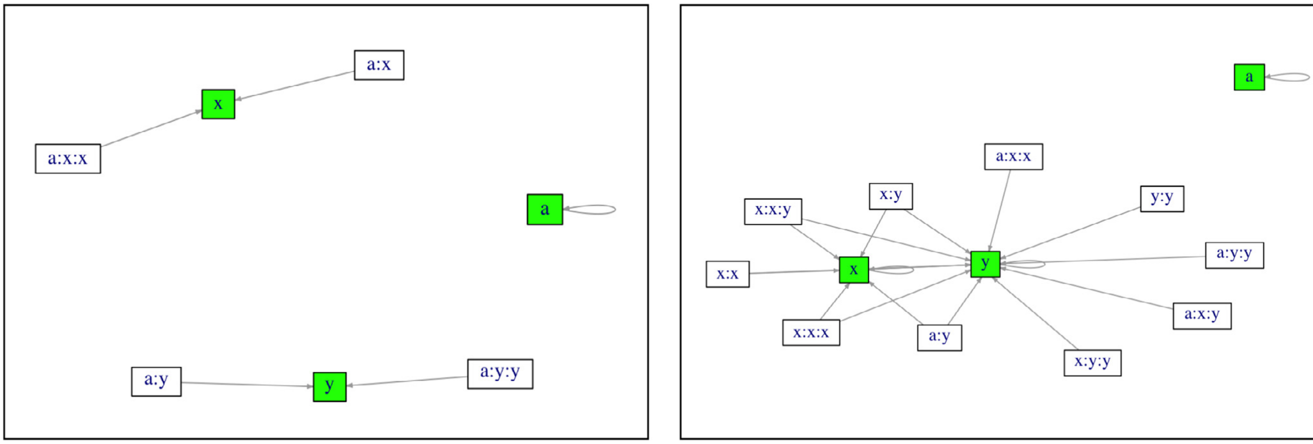


Fig. 10. Left: When connectivity π is set to 0, the system reduces to a fully reconstructed logistic map, with x and y independent. Right: When connectivity rises to 1, the x and y variables are intertwined – with mutual influences of various kinds. In general, we did not find that the terms on the right could precisely reconstruct the coupled maps. Nevertheless, the nature of these terms and their interactions can be used as a new summary measure and model of the interactions taking place in the coupled system.

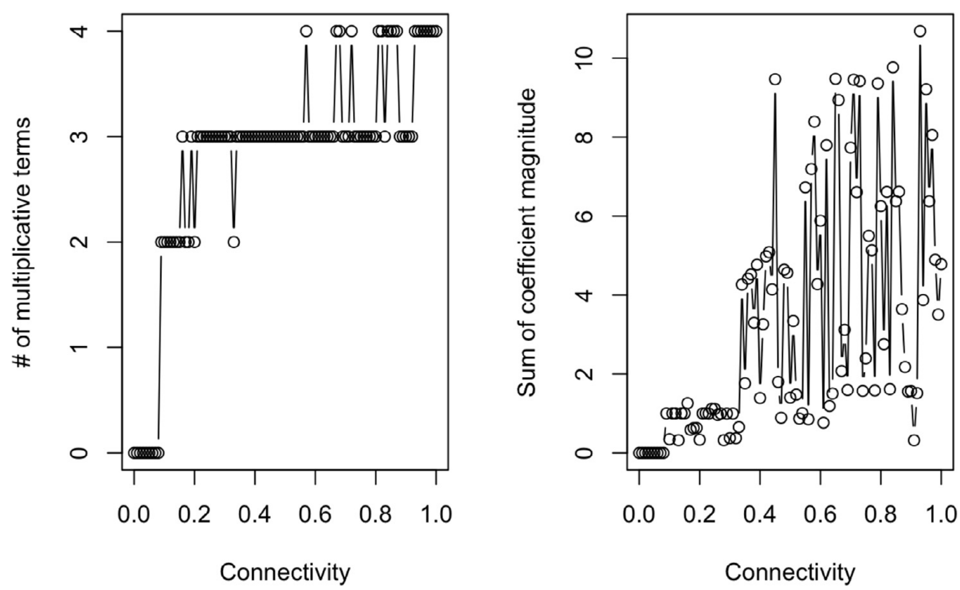


Fig. 11. Using SINDy as a source of summary measures about social influences, we can estimate connectivity from **B** in two ways. Left: We can count the number of terms with non-zero coefficients in **B** that contain *both* x and y. As connectivity π in the simulation increases, the number of such terms increases in step. Right: We can also take the absolute magnitude of the influence between the systems, taking a sum over the *values* of multiplicative terms of **B**. This measure is less stable, but reveals the same trend.

y. This means that the output from x_{t+1} should then be used to expand the defining equations for the subsequent update of y. The result is a formulation that requires a 7th order polynomial:

$$y_{t+1} = ay_t - 2ay_t^2 + ay_t^3 + ay_t\underline{(x_{t+1})} - ay_t^2\underline{(x_{t+1})} \tag{12}$$

This is several orders of magnitude greater than the fits we sought above with SINDy. In general, SINDy can find the fit for x more effectively than that y. The y variable update seems now to *particularly* challenge SINDy's sparsification. For reasons we discuss further below in the next section, when we have many polynomial terms of this kind it "washes out" the coefficients across a much larger set of columns of **B**, and thus the sparsification may omit key terms.

Nevertheless, and as suggested in the prior section, SINDy can be used as a source of summary measures. We illustrate this in Fig. 10, showing the graph structure that emerges from the SINDy algorithm.[6] As the connectivity parameter π rises from 0 to 1, we obtain a concomitant *complexity* in the model. This complexity can be explored for *multiplicative* terms that contain both x and y, indexing a nonlinear connection between the systems. These nonlinear connections, of course required by ground-truth reconstruction of the system, can be seen in the graph structures. We show

[6] The `sindyr` package has a built-in graphing tool using `igraph` that readers can explore themselves. This will generate these network diagrams in `R` automatically if requested.

the precise numbers of such multiplicative terms in Fig. 11 across values of connectivity π.

In sum, seeking summary measures from SINDy, and visualizing **B**, permit exploration of interactive dynamics between social models. In principle such an approach could be taken for human data in which two or more participants generate time series of various behaviors. SINDy may supply a new and interesting basis on which to explore such social dynamics.

5. Conclusion

There are many emerging approaches to "computational scientific methods," using machine learning or other techniques to recover equations from data (Sozou et al., 2017). The model we showcase here has certain desirable properties. It is extremely simple to deploy. It requires only a handful of adjustments to raw data, and the process of finding coefficients **B** does not require high-performance computing resources. With these in hand, a high-dimensional space of features, obtained from raw data **X**, can be explored. Few assumptions about the underlying form of the governing equations are needed. And this is implemented in just a few lines of code. Our demonstrations also illustrate the value in manipulating and exploring dynamic models as subjects of study in and of themselves. By creating new datasets under varying conditions, we are able to validate and expand these new analysis techniques. Such a strategy is an important part of expanding the toolkit of dynamic systems methods (cf. discussion in this issue: Spivey, 2018).

We did demonstrate that the approach can suffer under noise. It also requires distinct application in discrete dynamic systems, illustrated in the logistic map and the Lorenz system. In the former model, we take the first-order derivative to be the next values of the system variables. In the latter, we use numeric differentiation. These modeling choices are not necessarily clearly motivated, and still have to be set by the researcher. In addition, the thresholding of the coefficients may be better implemented by standard regularization techniques, such as LASSO (Tibshirani, 1996).

We proposed three possible extensions of this specific approach. By more elegantly integrating stochasticity in a continuous-field approach, we could obtain a more compelling recovery of the source system, including systems that are not sparse, but rather radically interactive. Second, we showcased how something like SINDy could serve as a basis for new descriptive measures of a system being studied in the lab. In the cognitive context, for example, SINDy's output may help describe the relative complexity of behavior in different contexts. Finally, we demonstrated that SINDy can be used to explore multiple systems, and map out potential interactive relationships that lie between them. Here models get considerably more complex, and the value of using SINDy as a source of new aggregate measures may be again useful.

These illustrations and concerns raised lead to a set of important pointers for application of SINDy to a real research context. To a great extent, these are open to future investigation. Application of these techniques to raw, noisy datasets is an important next step. When doing so, the researcher should bear in mind the following three concerns, as practical recommendations that are preliminary in nature.

1. *Time series length*. SINDy can work with surprisingly short time series (in our models 20–30 samples, across just 100 trials), but the more densely sampled the dynamics, the better the observed fit. The general lesson here is that if the observed time series thoroughly explores the underlying system's phase space—including stable/unstable equilibria, periodic orbits, and attractors—then SINDy will have a much better chance of reconstructing the right equations of motion. For example, in the logistic map case, if we only sampled from control parameters that had a point attractor, the fit is based on less variance than if we sampled from values of the parameter in chaotic regimes.
2. *System drift*. If the system's underlying dynamics are changing, if the researcher suspects drift, a single application of SINDy is unlikely to be interpretable. In this case, the researcher can explore a sliding-window approach. Such an option is offered in the `sindyr` package that the authors created. A single time series can be segmented using a sliding window. The researcher can explore the extent to which drift is taking place, and may derive new dependent variables that align with an experimental task or stimuli.
3. *Unknown and complex models*. We had to include parameters a (for the logistic map) and k (for the choice model), but an experimenter may not know the underlying control variables. In these cases, an experimenter may be able to propose a set of parameters (similar to priors, in the Bayesian sense), and adjust a SINDy model to capture behavioral dynamics. Researchers using trial conditions could propose specific fixed control variables to input into SINDy. Finally, in both unknown *and* complex systems, the threshold selection will not be obvious, and should be done by exploration. With each added polynomial order, the threshold may need to be scaled down to avoid iterative removal of viable terms. This is because, as the dimensionality of the feature space increases, coefficient magnitudes tend to be lower and more evenly distributed. This may be due to the iterative thresholding by least squares; a LASSO approach or the vector-field extensions we have described would be an important next step in general.

There is much left to do, of course. For example, a systematic comparison of these methods is still needed. It would be of value simply to compare the predictions of these methods with basic machine learning tools. A simple autoregressive model may compete with fits from recovered equations, suggesting that while equations provide a certain epistemological inspiration (such as expressing nonlinearities), their practical role could be humbler. This is especially true in the application of these methods to raw behavioral data from the lab or beyond it. The interactions among brain, behavior, and environment are robust (Chemero, 2011; Favela & Chemero, 2015). As noted above, human behaviors may undergo sudden drift in the equations that best fit data, through perturbation from environmental sources or even from dynamic adaptation to a task. For example, Dixon, Kelty-Stephen, and others (Dixon & Bangert, 2004; Dixon, Stephen, Boncoddo, & Anastas, 2010; Stephen, Dixon, & Isenhower, 2009) have studied how the cognitive system may show a pronounced "representational reorganization" during problem solving. In some of these studies, reorganization is detected through dynamics of body movement (such as movements of the hand). This work suggests that these data science approaches may need gentler summary measures, rather than pure formulaic outcomes.

In this paper, we focused on introducing this data science approach under very limiting assumptions about known systems. A crucial next step in all these domains is to move into raw data, perhaps even data for which the governing equations are not at all clear. This could have profound theoretical relevance for cognitive systems research (Favela, 2014). Finding lower-order descriptions that can encapsulate the high-dimensional complexity of human behavior is a long sought goal of our field. Using emerging data science tools may offer new ideas regarding what lower-order descriptions can do in a wide variety of contexts in which these data are collected.

Acknowledgments

HSB acknowledges support from the National Science Foundation award DMS-1723272.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.cogsys.2018.06.020.

References

Addis, M., Sozou, P. D., Lane, P. C., & Gobet, F. (2016). Computational scientific discovery and cognitive science theories. In *Computing and philosophy* (pp. 83–97). Springer.

Bechtel, W. (1998). Representations and cognitive explanations: Assessing the dynamicist's challenge in cognitive science. *Cognitive Science, 22*(3), 295–318.

Beek, P. J., Turvey, M. T., & Schmidt, R. C. (1992). Autonomous and nonautonomous dynamics of coordinated rhythmic movements. *Ecological Psychology, 4*(2), 65–95.

Bezruchko, B. P., Karavaev, A. S., Ponomarenko, V. I., & Prokhorov, M. D. (2001). Reconstruction of time-delay systems from chaotic time series. *Physical Review E, 64*(5), 056216.

Bhat, H. S., & Madushani, R. W. M. A. (2018). *Density tracking by quadrature for stochastic differential equations*, March (Tech. Rep. No. arXiv: 1610.09572 [stat.CO]). ArXiV. Retrieved from https://arxiv.org/abs/1610.09572.

Bhat, H. S., Madushani, R. W. M. A., & Rawat, S. (2016a). Bayesian inference of stochastic pursuit models from basketball tracking data. In *Bayesian Statistics in Action* (pp. 127–137). Springer.

Bhat, H.S., Madushani, R.W.M.A., & Rawat, S. (2016b). *Rdtq: Density tracking by quadrature [Computer software manual]*. <http://cran.r-project.org/package=Rdtq> (R package version 0.1).

Brockwell, P. J. (2014). Time series analysis. In *Wiley StatsRef: Statistics reference online*. John Wiley & Sons, Ltd.

Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences, 113*(15), 3932–3937.

Brush, S. G. (1974). Should the History of Science Be Rated X?: The way scientists behave (according to historians) might not be a good model for students. *Science, 183*(4130), 1164–1172.

Buder, E. H. (1991). A nonlinear dynamic model of social interaction. *Communication Research, 18*(2), 174–198.

Bünner, M. J., Meyer, T., Kittel, A., & Parisi, J. (1997). Recovery of the time-evolution equation of time-delay systems from time series. *Physical Review E, 56*(5), 5083.

Chartrand, R. (2011). Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics, 2011*, 164564.

Chemero, A. (2011). *Radical embodied cognitive science*. MIT Press.

Chen, S., Shojaie, A., & Witten, D. M. (2017). Network reconstruction from high-dimensional ordinary differential equations. *Journal of the American Statistical Association, 112*(520), 1697–1707. https://doi.org/10.1080/01621459.2016.1229197.

Crutchfield, J. P. (1994). The calculi of emergence: computation, dynamics and induction. *Physica D: Nonlinear Phenomena, 75*(1-3), 11–54.

Crutchfield, J. P. (2011). Between order and chaos. *Nature Physics, 8*(1), 17–24.

Crutchfield, J. P., & McNamara, B. S. (1987). Equation of motion from a data series. *Complex Systems, 1*(417–452), 121.

Dale, R., & Vinson, D. W. (2013). The observer's observer's paradox. *Journal of Experimental & Theoretical Artificial Intelligence, 25*(3), 303–322.

Dale, R., Warlaumont, A. S., & Richardson, D. C. (2011). Nominal cross recurrence as a generalized lag sequential analysis for behavioral streams. *International Journal of Bifurcation and Chaos, 21*(04), 1153–1161.

Daniels, B. C., & Nemenman, I. (2015). Automated adaptive inference of phenomenological dynamical models. *Nature Communications, 6*, 8133.

Dixon, J. A., & Bangert, A. S. (2004). On the spontaneous discovery of a mathematical relation during problem solving. *Cognitive Science, 28*(3), 433–449.

Dixon, J. A., Stephen, D. G., Boncoddo, R., & Anastas, J. (2010). The self-organization of cognitive structure. *Psychology of learning and motivation* (Vol. 52, pp. 343–384). Elsevier.

Duran, N. D., & Dale, R. (2014). Perspective-taking in dialogue as self-organization under social constraints. *New Ideas in Psychology, 32*, 131–146.

Džeroski, S., & Todorovski, L. (2008). Equation discovery for systems biology: finding the structure and dynamics of biological networks from time course data. *Current Opinion in Biotechnology, 19*(4), 360–368.

Eliasmith, C. (1996). The third contender: A critical examination of the dynamicist theory of cognition. *Philosophical Psychology, 9*(4), 441–463.

Favela, L. H. (2014). Radical embodied cognitive neuroscience: addressing grand challenges of the mind sciences. *Frontiers in Human Neuroscience, 8*, 796.

Favela, L. H., & Chemero, A. (2015). The animal-environment system. *Perceptual and Emotional Embodiment: Foundations of Embodied Cognition, 1*, 59.

Frank, T. D., Richardson, M. J., Lopresti-Goodman, S. M., & Turvey, M. T. (2009). Order parameter dynamics of body-scaled hysteresis and mode transitions in grasping behavior. *Journal of Biological Physics, 35* (2), 127–147.

Haken, H., Kelso, J. S., & Bunz, H. (1985). A theoretical model of phase transitions in human hand movements. *Biological Cybernetics, 51*(5), 347–356.

Hempel, C. (1966). *Philosophy of natural science* (1st ed.). Upper Saddle River, NJ: Prentice Hall.

Hirsch, M. W. (1984). The dynamical systems approach to differential equations. *Bulletin of the American Mathematical Society, 11*(1), 1–64.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology, 117*(4), 500–544.

Kaplan, D. M., & Bechtel, W. (2011). Dynamical models: An alternative or complement to mechanistic explanations?. *Topics in Cognitive Science 3*(2), 438–444.

Keith, T. Z. (2005). *Multiple regression and beyond* (1st ed.). Allyn & Bacon.

Klahr, D., & Simon, H. A. (1999). Studies of scientific discovery: Complementary approaches and convergent findings. *Psychological Bulletin, 125*(5), 524.

Kugler, P. N., Kelso, J. S., & Turvey, M. T. (1980). On the concept of coordinative structures as dissipative structures: I. Theoretical lines of convergence. *Advances in psychology* (Vol. 1, pp. 3–47). Elsevier.

Kuhn, T. S. (1962). *The structure of scientific revolutions*. University of Chicago Press.

Langley, P. (1981). Data-driven discovery of physical laws. *Cognitive Science, 5*(1), 31–54.

Langley, P. (1987). *Scientific discovery: Computational explorations of the creative processes*. MIT Press.

Langley, P., Sanchez, J. N., Todorovski, L., & Dzeroski, S. (2002). Inducing process models from continuous data. *Proceedings of the Nineteenth International Conference on Machine Learning*, 347–354.

Link, S. W. (1975). The relative judgment theory of two choice response time. *Journal of Mathematical Psychology, 12*(1), 114–135.

Loiseau, J.-C., & Brunton, S. L. (2018). Constrained sparse Galerkin regression. *Journal of Fluid Mechanics, 838*(March), 42–67.

Mangan, N. M., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications, 2*(1), 52–63.

O'Hora, D., Dale, R., Piiroinen, P. T., & Connolly, F. (2013). Local dynamics in decision making: The evolution of preference within and across decisions. *Scientific Reports, 3*, 2210.

Paxton, A., & Griffiths, T. L. (2017). Finding the traces of behavioral and cognitive processes in big data and naturally occurring datasets. *Behavior Research Methods, 49*(5), 1630–1638.

Pikovsky, A. (2016). Reconstruction of a neural network from a time series of firing rates. *Physical Review E, 93*(6), 062313.

Port, R. F., & Van Gelder, T. (1995). *Mind as motion: Explorations in the dynamics of cognition*. MIT Press.

Quade, M. (2018). *Sparsereg: A collection of modern sparse (regularized) linear regression algorithms* <https://github.com/loiseaujc/sparsereg> (original-date: 2018-02-23T22:49:26Z) Retrieved 2018-03-21.

Raczaszek, J., Tuller, B., Shapiro, L. P., Case, P., & Kelso, S. (1999). Categorization of ambiguous sentences as a function of a changing prosodic parameter: A dynamical approach. *Journal of Psycholinguistic Research, 28*(4), 367–393.

Ratcliff, R., Van Zandt, T., & McKoon, G. (1999). Connectionist and diffusion models of reaction time. *Psychological Review, 106*, 261–300.

Richardson, M. J., Dale, R., & Marsh, K. L. (2014). Complex dynamical systems in social and personality psychology: Theory, modeling and analysis. In H. Reis & C. Judd (Eds.), *Hand-book of research methods in social and personality psychology* (pp. 253–282). Cambridge University Press.

Richardson, M. J., Kallen, R. W., Nalepka, P., Harrison, S. J., Lamb, M., Chemero, A., Schmidt, R. C. (2016). Modeling embedded interpersonal and multiagent coordination. In *COMPLEXIS* (pp. 155–164).

Rudy, S. H., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Science Advances, 3* (4), e1602614.

Rydén, T. (2015). Hidden Markov models. In *Wiley StatsRef: Statistics reference online* (pp. 1-10). John Wiley & Sons, Ltd.

Saltzman, E., & Caplan, D. (2015). A graph-dynamic perspective on coordinative structures, the role of affordance-effectivity relations in action selection, and the self-organization of complex activities. *Ecological Psychology, 27*(4), 300–309.

Saltzman, E., & Munhall, K. G. (1992). Skill acquisition and development: The roles of state-, parameter-, and graph-dynamics. *Journal of Motor Behavior, 24*(1), 49–57.

Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics, 656*, 5–28.

Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science, 324*(5923), 81–85. April.

Schmidt, R. C., Carello, C., & Turvey, M. T. (1990). Phase transitions and critical fluctuations in the visual coordination of rhythmic movements between people. *Journal of Experimental Psychology: Human Perception and Performance, 16*(2), 227.

Shalizi, C. R., & Crutchfield, J. P. (2001). Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics, 104*(3), 817–879.

Shalizi, C. R., & Shalizi, K. L. (2004). Blind construction of optimal nonlinear recursive predictors for discrete sequences. In *Proceedings of the 20th conference on uncertainty in artificial intelligence* (pp. 504–511).

Simon, H. A. (1992). What is an explanation of behavior? *Psychological Science, 3*(3), 150–161.

Smith, L. A. (1992). Identification and prediction of low dimensional dynamics. *Physica D: Nonlinear Phenomena, 58*(1-4), 50–76.

Sozou, P. D., Lane, P. C. R., Addis, M., & Gobet, F. (2017). Computational scientific discovery. In *Springer handbook of model-based science* (pp. 719–734). Cham: Springer.

Spivey, M. J. (2018). Discovery in complex adaptive systems. *Cognitive Systems Research, 51*, 40–55.

Spivey, M. J., & Dale, R. (2006). Continuous dynamics in real-time cognition. *Current Directions in Psychological Science, 15*(5), 207–211.

Stephen, D. G., Dixon, J. A., & Isenhower, R. W. (2009). Dynamics of representational change: Entropy, action, and cognition. *Journal of Experimental Psychology: Human Perception and Performance, 35*(6), 1811.

Stigler, S. M. (2016). *The seven pillars of statistical wisdom*. Harvard University Press.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Tran, G., & Ward, R. (2017). Exact recovery of chaotic systems from highly corrupted data. *Multiscale Modeling & Simulation, 15*(3), 1108–1129.

Tuller, B., Case, P., Ding, M., & Kelso, J. A. (1994). The nonlinear dynamics of speech categorization. *Journal of Experimental Psychology: Human Perception and Performance, 20*(1), 3.

Vandekerckhove, J., & Tuerlinckx, F. (2007). Fitting the Ratcliff diffusion model to experimental data. *Psychonomic Bulletin & Review, 14*(6), 1011–1026.

Van Gelder, T. (1995). What might cognition be, if not computation? *The Journal of Philosophy, 92*(7), 345–381.

van Rooij, I., Bongers, R. M., & Haselager, W. P. F. (2002). A non-representational approach to imagined action. *Cognitive Science, 26*(3), 345–375.

Zgonnikov, A., Aleni, A., Piiroinen, P. T., O'Hora, D., & Bernardo, M. d. (2017). Decision landscapes: Visualizing mouse-tracking data. *Royal Society Open Science, 4*(11), 170482, November.

Zhang, L., & Schaeffer, H. (2018). On the convergence of the SINDy algorithm (Tech. Rep. No. arXiv: 1805.06445 [math.OC]), May. Retrieved from https://arxiv.org/abs/1805.06445.