

EQUATION DISCOVERY FROM DATA: PROMISE AND PITFALLS, FROM RABBITS TO MARS

G. M. DONOVAN AND Q. SU

(Received 7 April, 2022)

Abstract. The problem of equation discovery seeks to reconstruct the underlying dynamics of a time-varying system from observations of the system, and moreover to do so in an instructive way such that we may understand these underlying dynamics from the reconstruction. This article illustrates one type of modern equation discovery method (sparse identification of nonlinear dynamics, or SINDy) in the context of two classic problems. The presentation is in a tutorial style intended to be accessible to students, and could form a useful module in undergraduate or graduate courses in modelling, data analysis, or numerical methods. In this style we explore the strengths and limitations of these methods. We also demonstrate, through use of a carefully constructed example, a new result about the relationship between the reconstructed and true models when a naïve polynomial basis is used.

1. Scope

This article is written in a tutorial style, intended to be accessible to students with a background in differential equations¹, linear algebra, and numerical methods. Modern equation discovery methods are not standard fare in applied mathematics teaching, and this is an effort to make them accessible to students, even at the undergraduate level. It is written in a relatively informal style, with a few notes to instructors as to how the presentation might be adapted to different student backgrounds. Code is also available in both MATLAB and python so that students can begin to use these methods themselves, without having to start from scratch.

2. Motivation

Suppose that, from some observable system $\mathbf{x}(t) \in \mathbb{R}^n$, we have only a set of observations of this system at discrete times, denoted $\mathbf{x}(t_k)$ for $k = 1 \dots m$. The problem of *equation discovery* is: can we reconstruct, from these observations, the dynamical system which governs the underlying behaviour of this system? Importantly, we seek not just to *predict* the future timecourse beyond the observations (i.e. in a black-box fashion) but to gain *understanding* of the underlying system; if we are able to do so, it will also allow prediction of future behaviour, and much more. See Fig. 1 for a schematic illustration of the concept.

To be more specific, let's suppose that \mathbf{x} arises from a system of ordinary differential equations

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}(t)). \quad (2.1)$$

¹The method can be formulated for difference equations as well, if needed; see also the supplemental material to reference [1].

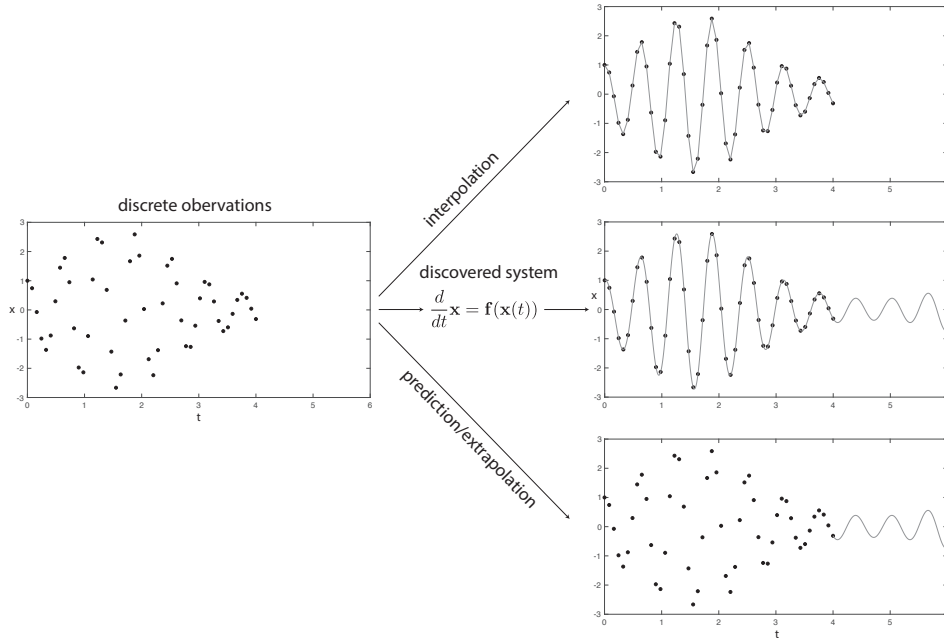


FIGURE 1. Schematic illustration of the equation discovery process and its relationship with prediction and interpolation.

From a set of observations at discrete times, can we “discover” the (hidden) right-hand side \mathbf{f} which governs the dynamics? Moreover, we seek to do so in an informative way, where the discovered \mathbf{f} tells us about the underlying dynamics of the problem, rather than merely allows extrapolation/prediction from the observations.

Here we will consider equation discovery in the context of two classical problems:

- (1) The mutually oscillating populations of predator-prey systems, and the classic data of the lynx and snowshoe hare from the Hudson bay company in the 19th and 20th centuries, motivating the famous Lotka-Volterra equations, and
- (2) The orbits of Earth and Mars, specifically as reflected in Tycho Brahe’s 16th century observations and Kepler’s subsequent interpretation of elliptical heliocentric orbits to explain the observed retrograde motion.

Together these two problems illustrate the interplay between data and the discovery of the underlying dynamics, and provide a series of examples illuminating both the promise and many of the potential pitfalls of the equation discovery approach.

3. Sparse Identification of Nonlinear Dynamics (SINDy)

We will focus on one particular approach to equation discovery, known as sparse identification of nonlinear dynamics (or SINDy) [1]. The central idea is to select a library of candidate nonlinear functions and attempt to use these to represent the right-hand side of our dynamical system. Ideally the final representation will be

sparse (or *parsimonious*); that is, even though we may have many candidate functions, only a few of the candidate functions will be active in the final representation.

We start with our discrete observations $\mathbf{x}(t_k)$ for $k = 1 \dots m$ and stack these observations to construct the *observation matrix*

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \dots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \dots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \dots & x_n(t_m) \end{bmatrix} \quad (3.1)$$

where $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$. That is, across the columns we have the different variables of the system, while down the rows we have the different observation times, and hence all of our observations are encoded in this matrix. In reality of course, missing and corrupt data are likely to occur; we will not go into the details here but methods to deal with this can be found in refs [1, 3].

We will also need the time derivatives of the observations. In some situations these might be available directly as part of the observations, but more often they will need to be approximated numerically². We stack these up into the observed derivatives matrix in the same way as we did with the observations

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \dots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \dots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \dots & \dot{x}_n(t_m) \end{bmatrix}. \quad (3.2)$$

Now we choose the library of candidate (nonlinear) functions $g_i(\cdot)$ and apply them to the columns of the observation matrix \mathbf{X} to form

$$\Theta(\mathbf{X}) = \begin{bmatrix} 1 & g_1(\mathbf{X}) & g_2(\mathbf{X}) & \dots \end{bmatrix}. \quad (3.3)$$

Let's suppose for now that we select low-order polynomials for our candidate basis functions. Then g_1 gives linear terms, g_2 quadratic terms, etc³. In essence, Θ encapsulates all possible right-hand sides allowed by our chosen library: each column of Θ represents a possible term in the right-hand side (RHS) of the dynamical system.

What remains to be done is to figure out how much each of these possible terms contributes to the observations. To do so we create parameter vectors ξ_i for each variable $i = 1 \dots n$ which determine the activity of each possible term, and the ξ_i can be determined through our observation matrices. We also stack the parameter vectors up into a matrix

$$\Xi = [\xi_1 \ \xi_2 \ \dots \ \xi_n] \quad (3.4)$$

²For the moment, imagine approximating the derivative using finite differences. However, there are subtleties when noise is present, and this is not actually a good approach in practice most of the time; see Appendix A.

³Selection of these functions is important; more on this later. We will also write the form of Θ out explicitly in the next section.

which allows us to set up a linear system relating the observed derivatives, candidate functions applied to the observations, and coefficients as

$$\underbrace{\dot{\mathbf{X}}}_{\text{observed derivatives}} = \underbrace{\Theta(\mathbf{X})}_{\text{library of RHS terms}} \underbrace{\Xi}_{\text{coefficients}}. \quad (3.5)$$

Take a moment to consider the dimensions of this system. We have n dimensions in the dynamical system, m discrete time observations from that system, and p library functions for the possible RHS. Thus $\dot{\mathbf{X}}$ is m -by- n , $\Theta(\mathbf{X})$ is m -by- p , and Ξ is p -by- n . We require $n < p < m$, and typically $p \ll m$, so this is an *overdetermined* linear system (for each column of $\dot{\mathbf{X}}$ and Ξ). As such we expect no unique solution, but instead undertake regression (for each column) to determine the best-fit solution. We hope to obtain from the regression

- (1) that the reconstructed system using the candidate functions (i.e. $\frac{d}{dt}\mathbf{x} = \Xi^T \Theta^T(\mathbf{x})$) adequately reproduces the observations, which is to say that the “true” underlying dynamics can be captured by our library functions, and
- (2) that Ξ is sparse, meaning that only a few of the candidate functions are active (most terms have coefficient zero).

If successful, the first means that we have captured the dynamics, and the second means that we have done so usefully in the sense of being able to understand the system through the action of a small number of terms. To achieve this, we need not just a standard regression of the overdetermined linear system, but instead a *sparse* regression method.

Briefly, to understand the idea of sparse regression, consider a regression problem $b = A\xi$ where the linear system is overdetermined. If we are thinking of this as being a column of Eq. 3.5, then $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times p}$, and $\xi \in \mathbb{R}^p$ with $p \ll m$. In order to obtain a *sparse* regression, one option is to impose a so-called *regularisation* term (with coefficient λ) so that

$$\xi = \underset{y}{\operatorname{argmin}} \left(\underbrace{\|Ay - b\|_2}_{\text{least-squares residual}} + \underbrace{\lambda \|y\|_1}_{\text{regularisation term}} \right). \quad (3.6)$$

This method is called least absolute shrinkage and selection operator (LASSO) [16]. The first term to be minimised is the usual sum-of-squares of the residual; thus setting $\lambda = 0$ gives traditional least squares regression, and solutions will typically not be sparse. By increasing the parameter λ we can increase the sparsity in the regression at some cost to the accuracy (in terms of the traditional least-squares residual.) However, note that as we increase λ , eventually the solution is $\xi = 0$ (and the “reconstruction” is the trivial solution). Thus the selection of λ is a trade-off, sacrificing accuracy for sparsity.

It is worth noting here that LASSO is often not used in practice, largely because of its computational cost. An alternative to LASSO, used here and in ref [1], is the *sequential thresholded least-squares algorithm*, which also provides a sparse regression, controlled by the sparsity parameter λ . In order not to dwell over-long on these differences here, details of the algorithm are given in Appendix B.

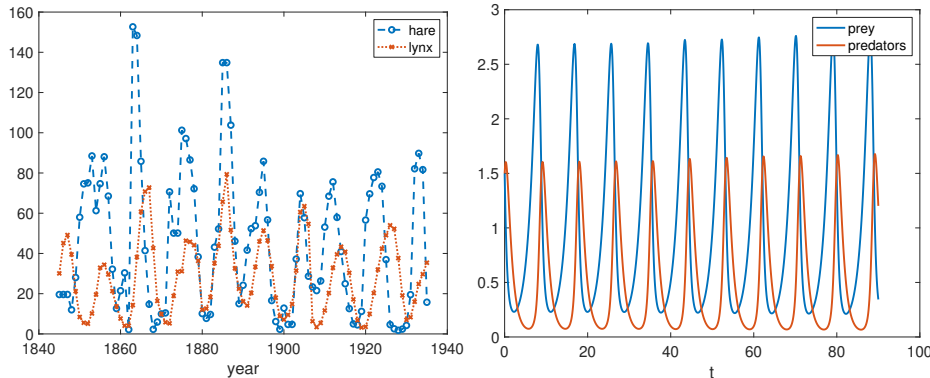


FIGURE 2. Left panel: the classic Hudson Bay Company lynx-hare population data (thousands). Right panel: predator-prey model trajectory with parameters $\alpha = 2/3$, $\beta = 4/3$, $\delta = 1$, and $\gamma = 1$.

4. Predator-Prey Systems

We begin with an example of the classical predator-prey system. We give a relatively brief presentation here, though for more detail the reader is referred to reference [13].

The interactions between predator and prey species are often motivated by the classic dataset of furs sold to the Hudson Bay Company between the mid 19th and mid 20th centuries. The two species involved are hares (snowshoe hare, *Lepus americanus*) and lynx (Canada lynx, *Lynx canadensis*), and assuming the fur numbers as a proxy for population, show both populations apparently locked in periodic cycles of roughly 10 year duration; see Fig. 2, left panel.

The canonical model of predator-prey interaction is given by the Lotka-Volterra equations⁴

$$\frac{dx}{dt} = \alpha x - \beta xy \quad (4.1)$$

$$\frac{dy}{dt} = \delta xy - \gamma y \quad (4.2)$$

where x denotes the population of the prey species and y the predator species. In this form the model makes a number of assumptions, including (but not limited to):

- in the absence of predators, the prey population grows exponentially without bound ($\frac{dx}{dt} = \alpha x$ when $y \equiv 0$).
- in the absence of prey, the predator species dies out, i.e. has no other food source ($\frac{dy}{dt} = -\gamma y$ when $x \equiv 0$).

⁴Alfred Lotka proposed the model in the context of chemical reactions in 1910 [11], and Vito Volterra arrived at the same model in 1926 when studying fish stocks [19]. Neither was motivated to do so by the lynx and hare data, which was still in the process of being collected, but let's not have facts get in the way of a good story.

- the interaction (predation) between the populations takes a simple product form (the $-\beta xy$ and $+\delta xy$ terms).

A typical solution trajectory from this model is shown in the right panel of Fig. 2, showing qualitatively similar population cycles – note that the parameter values $\alpha = 2/3$, $\beta = 4/3$, $\delta = 1$, and $\gamma = 1$ are chosen to be “nice” and not to fit the data.

Ultimately we would like to discover the governing equations directly from the data, potentially bypassing the model entirely. However, we will start by taking snapshots from the model system (known as *synthetic data*) and attempting to reconstruct the underlying dynamics (which, conveniently we already know.) While in some sense this is clearly cheating, using the synthetic data when the real data is sitting right there, it will help to illustrate a number of useful points.

Using 1000 discrete observations over the period from $t = 0$ to $t = 90$, we have timestep $\Delta t = 0.09$ and $m = 1000$ observations. We thus stack up our m -by- n observation matrix \mathbf{X} , which in this case is 1000×2 , and the corresponding derivative observation matrix $\dot{\mathbf{X}}$ of the same size.

We then need to select our library of nonlinear functions from which to reconstruct the RHS of the dynamical system. Choosing polynomial terms up to order 2, we have for this two-dimensional system $\mathbf{x} = [x \ y]^T$ and so

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | \\ 1 & x & y & x^2 & xy & y^2 \\ | & | & | & | & | & | \end{bmatrix} \quad (4.3)$$

and hence we have $p = 6$ possible RHS terms (recall that we have arranged the discrete snapshots “down” the columns). Computing this matrix and then the sparse regression, we obtain the parameter values such that the sparse reconstructed system is, explicitly written out,

$$\frac{dx}{dt} = 0 \quad +0.6621x \quad +0y \quad +0x^2 \quad -1.3254xy \quad +0y^2 \quad (4.4)$$

$$\frac{dy}{dt} = 0 \quad +0x \quad -0.9887y \quad +0x^2 \quad +0.9907xy \quad +0y^2. \quad (4.5)$$

Comparing this with the original system (which we know, because we’re cheating and using synthetic data)

$$\frac{dx}{dt} = \frac{2}{3}x - \frac{4}{3}xy \quad (4.6)$$

$$\frac{dy}{dt} = xy - y \quad (4.7)$$

we see that we have reconstructed the true underlying dynamics very closely, with parameter values within 2% or so, and importantly with zero coefficients for the inactive terms. Given the quality of this reconstruction, we expect the reconstructed system to reproduce the data very well, and indeed it does, as shown in Fig. 3.

Flushed with this success, one might ask: what about the real data? Did we need Lotka and Volterra, or could SINDy have discovered this model directly from the data? So, we repeat the process, but now with the 91 datapoints of the real Hudson Bay Company dataset, and $\Delta t = 1$ yr. Again constructing the snapshot matrices and using the same polynomial basis we have $n = 2$ dimensions in the system, $p = 6$ possible RHS terms and $m = 91$ observations, so even with an order

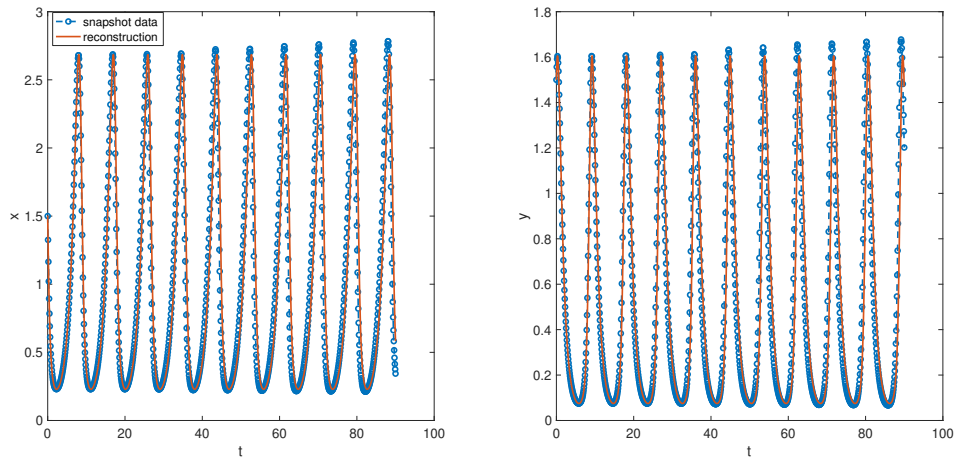


FIGURE 3. Reconstruction of synthetic predator-prey data. Left panel: prey (x); right panel: predators (y). The snapshot data are given by the blue circles and the reconstruction by the red lines.

of magnitude fewer samples, we still have our overdetermined system. Constructing the sparse regression we find

$$\frac{dx}{dt} = -6.2767 \quad +0x \quad +0.1324y \quad +0x^2 \quad +0xy \quad +0y^2 \quad (4.8)$$

$$\frac{dy}{dt} = 13.8098 \quad -0.4690x \quad +0y \quad +0x^2 \quad +0xy \quad +0y^2. \quad (4.9)$$

The regression is sparse, with only 4 active terms, but does not correspond to the Lotka-Volterra model. And checking the reconstruction (Fig 4), we find that the reconstructed system does a very poor job indeed of reproducing the data. Oh dear – maybe we do need modellers after all.

So, our effort to reconstruct the (assumed) underlying predator prey dynamics from the lynx and hare data failed. Why? Perhaps the most obvious potential problem is the quality of the data, both in terms of the number of observations, and the degree of noise present. There is also a more subtle potential problem. Let’s return to the synthetic data case to understand what’s gone wrong.

In order to understand the problems with the quality of data, we will use again our synthetic predator-prey data (Fig. 3) but now degrade that data (again synthetically) in order to see how this affects our ability to reconstruct the underlying dynamics. The two obvious things to do are to add noise, and to reduce the number of observations (increase the timestep). Once we are adding noise stochastically, we’ll need to repeat the simulation and look at the *probability* of convergence, and we can do this for a range of noise levels and timesteps. As we vary the noise level and the timestep of the snapshot data, how often do we discover the correct underlying dynamics? The results are shown in Fig. 5. The original, non-degraded synthetic data (Fig. 3) corresponds to a timestep multiplier of 1 and a noise level of 0. In this lower-left region of Fig. 5, with a relatively small timestep and low noise level, SINDy virtually always converges to the “true” underlying dynamics.

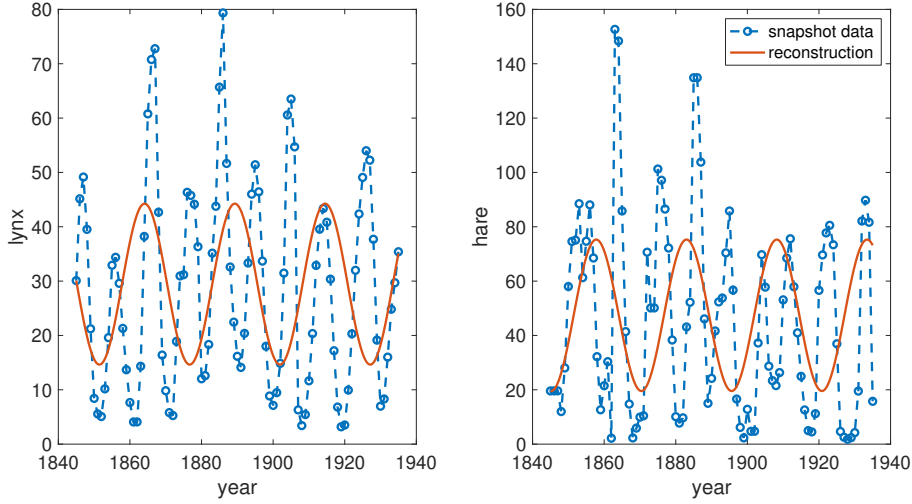


FIGURE 4. Failed reconstruction of Hudson Bay lynx-hare data.

However, as we increase the noise level and/or the timestep, we transition through a region of partial convergence (that is, it works sometimes) into the black zone in which the method almost never finds the true underlying dynamics. So we can see that there are clear requirements in terms of data quality to recover the underlying dynamics, even with synthetic data. Where would the actual lynx/hare data lie on this figure? The timestep multiplier is about 10 (roughly an order of magnitude fewer observations) and the noise level, at a rough estimate, is at least 10^0 if not larger. So data of this quality lies very far from the convergence zone, and it is unsurprising that we are unable to reconstruct the underlying dynamics directly from the lynx-hare data.

On the basis of data quality alone, you may be convinced that we shouldn't expect this approach to work. But let's consider also a further subtlety: in the above analysis we implicitly assumed that the lynx-hare data is actually, underneath all that noise, governed by Lotka-Volterra dynamics. What if, although the basic idea may be correct, the details are somewhat different? For example, a variation (or perhaps successor) to the Lotka-Volterra model is the Rosenzweig-MacArthur model⁵[14] where now

$$\frac{dx}{dt} = rx \left(1 - \frac{x}{K}\right) - \frac{sxy}{1 + s\tau x} \quad (4.10)$$

$$\frac{dy}{dt} = d \frac{sxy}{1 + s\tau x} - cy \quad (4.11)$$

still with x being the prey population and y the predators, and now with parameters r, K, s, τ, c and d . Comparing with Eq. 4.1-4.2 we see some similarities (the $-cy$ term playing the same role as the $-\gamma y$ term for predator decay, and the rx and αx terms for prey growth) but also differences. One change is the logistic-type carrying capacity introduced by $-rx^2/K$ which introduces a carrying capacity K

⁵Technically, with a Holling type-II functional response [17].

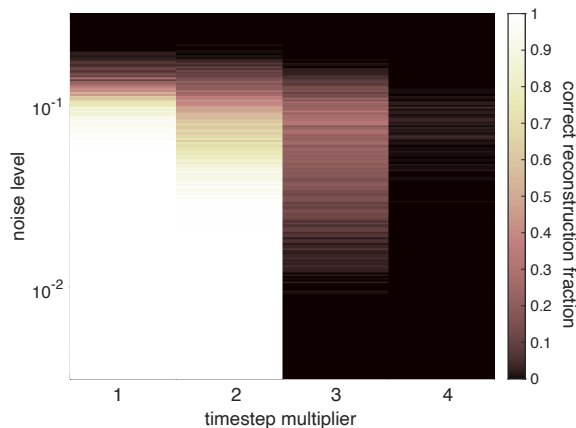


FIGURE 5. Reliability of SINDy reconstruction of degraded synthetic data as noise level and timestep are varied. Colour shows the observed convergence fraction. Note the log-scale on the vertical axis; noise level is the standard deviation of additive Gaussian white noise. Convergence is defined as coefficient reconstruction within 1% of the true model parameters.

for the prey species⁶. The other change to note is that the functional form of the predation terms has changed: where previously we had interaction terms $\propto xy$ we now have terms of the form $\frac{xy}{1+x}$. Now, as a thought experiment: what if the underlying dynamics for the lynx-hare data arise from the Rosenzweig-MacArthur model instead of the Lotka-Volterra model?

The key thing to remember is that we were seeking system reconstructions *in which the right-hand side of the dynamical system could be represented by low-order polynomial terms*. For the Lotka-Volterra model, this works fine, because the underlying dynamics *are* low-order polynomial terms. What about the Rosenzweig-MacArthur model? Well, the carrying capacity term presents no problem, because it's just quadratic in x and so fits neatly in our low-order polynomial basis. The new form of the predation terms, however, is more problematic: what are dynamics of the form $\frac{xy}{1+x}$ going to do to our attempted polynomial reconstruction?

Well, bad things. But what, exactly? To understand this, we need to know about Taylor series, and multivariate Taylor series. If you immediately feel comfortable with the concept, feel free to skip to Eq. 4.14; if not, here is a brief primer.

⁶While previously in the absence of predators the prey increased without bound, they now equilibrate at the carrying capacity K . If you are not familiar with the logistic growth model, set $y \equiv 0$ and you will see that we now have a fixed point at $x = K$ in addition to the one at $x = 0$.

4.1. Crash course in multivariate Taylor series⁷. You will be aware that some functions can be represented as sums of polynomials, for example the exponential

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \quad (4.12)$$

This is an example of *Taylor series*, which may allow us to express a function as an infinite sum, with coefficients in terms of the derivatives of the function:

$$F(x) = \sum_{n=0}^{\infty} \frac{F^{(n)}(a)}{n!} (x-a)^n = F(a) + \frac{F'(a)}{1!} (x-a) + \frac{F''(a)}{2!} (x-a)^2 + \dots \quad (4.13)$$

where $F^{(n)}$ denotes the n^{th} derivative. If we truncate the infinite sum to a finite number of terms (say N terms), then the resulting *Taylor polynomial* is a polynomial of degree N and gives us an approximation to F near the point $x = a$.

What does this have to do with our reconstruction of the lynx-hare data? Well, we are seeking reconstructions of the dynamics in terms of low-order polynomials. So if the dynamics are non-polynomial, but we ask for a polynomial reconstruction, what we will get (if it works) might be *the Taylor polynomial which approximates the dynamics*. The concept is complicated somewhat by the fact that we are not working in one dimension, and the non-polynomial interaction term is not just a function of a single variable, but of both x and y . Fortunately, the concept of Taylor series extends naturally to several variables. In two dimensions, about the point $(0, 0)$ for simplicity, we have

$$F(x, y) = F(0, 0) + F_x(0, 0)x + F_y(0, 0)y + \frac{1}{2} [F_{xx}(0, 0)x^2 + 2F_{xy}(0, 0)xy + F_{yy}(0, 0)y^2] + \dots \quad (4.14)$$

where the subscripts on F denote the partial derivatives⁸. So if we try to represent non-polynomial dynamics on a polynomial basis, we might expect our reconstruction will look like the Taylor polynomial (of the order of the basis we have chosen), and the error in the reconstruction will depend on the accuracy of the Taylor approximation. However, even assuming that the approximation is reasonably good, this presents another problem: recall that we seek *sparse* (or parsimonious) solutions in which most coefficients are zero, with the aim of having only a few active terms whose action we can readily understand. If we are capturing instead a Taylor polynomial arising from approximation of non-polynomial dynamics, we have two potential issues:

- (1) If the Taylor coefficients are not sparse, our reconstruction will not be sparse. And there is no reason to expect them to be so in general.
- (2) Even if (1) is satisfied, our ability to interpret the reconstruction usefully is limited by the less-intuitive Taylor polynomial form.

⁷This section can be adapted, amended, or omitted depending on the background of the students. It is written here to give an intuitive feel for the effect of the Taylor expansion on this method for students without background in Taylor series, but of course many subtleties are omitted.

⁸For an audience not familiar with partial derivatives, the following development can instead utilise the simple y -dependence of the predation term and hence the single-variable expansion of $x/(1 + s\tau x)$.

To see this, consider the Taylor expansion for our Rosenzweig-MacArthur predation term:

$$\frac{sxy}{1+s\tau x} = sxy - s^2\tau x^2y + s^3\tau^2x^3y - s^4\tau^3x^4y + s^5\tau^4x^5y + \dots \quad (4.15)$$

Where would we truncate this expansion? How useful would a reconstruction of this form be? This emphasises that the selection of the candidate functions for representing the dynamics is important, and that naive expansion in terms of low-order polynomials may not be suitable.

4.2. Relationship between truncated and reconstructed model. This section explores in more detail the relationship between the truncated Taylor series model and the reconstructed SINDy model using a polynomial basis⁹.

It is often assumed that, when using a polynomial basis for SINDy reconstruction, even if the true dynamics are non-polynomial, the reconstruction will be the truncated Taylor series of the true model. In this section we demonstrate that this assumption is an oversimplification, and that the relationship between the truncated model and the reconstructed model is more complicated than simple Taylor truncation.

Consider again the Rosenzweig-MacArthur model, which for this section we take to be the *true model*

$$\frac{dx}{dt} = rx \left(1 - \frac{x}{K}\right) - \frac{sxy}{1+s\tau x} \quad (4.16)$$

$$\frac{dy}{dt} = d\frac{sxy}{1+s\tau x} - cy. \quad (4.17)$$

If we expand this as a Taylor series and truncate the series at third order, we obtain the so-called *truncated model*

$$\frac{dx}{dt} = rx - \frac{r}{K}x^2 - sxy + s^2\tau x^2y \quad (4.18a)$$

$$\frac{dy}{dt} = dsxy - ds^2\tau x^2y - cy. \quad (4.18b)$$

Note that, if we take $\tau \rightarrow 0$ and $K \rightarrow \infty$, we recover the Lotka-Volterra system. In particular, we are interested here in the role of the parameter τ , which controls the extent to which the dynamics are non-polynomial¹⁰. The idea, then, is to use τ as a *homotopy parameter*: by varying τ we can transition from polynomial dynamics ($\tau = 0$, the Lotka-Volterra system) through systems which are “close” to polynomial (τ small) and finally systems which are strongly non-polynomial (τ large). However, simply varying τ alone is not a satisfactory solution, because we wish to maintain the periodic solutions as τ is varied.

⁹For use as a tutorial, depending on the level of the students it may be advisable to skip this section and proceed directly to the next example (Sec 5).

¹⁰While the carrying capacity term is also new, because it is of polynomial form it presents no particular difficulties.

Stability analysis of the Rosenzweig-MacArthur model (e.g. [15]) provides the conditions we need to stay within the regime of periodic solutions as

$$d > c\tau, \quad (4.19)$$

$$K > \frac{d + c\tau}{s\tau(d - c\tau)} \quad (4.20)$$

and thus we select $r = \frac{2}{3}$, $s = \frac{3}{4}$, $c = 1$, $d = \frac{4}{3} > c\tau$, and $K = K(\tau) = \frac{d+c\tau}{s\tau(d-c\tau)} + 1$ to maintain these solutions for $\tau \in (0, 1]$.

Now we can vary τ freely within this range in order to explore the relationship between the true model, the truncated model, and the (SINDy) reconstructed model using a naïve polynomial basis (when the polynomial basis and the truncated model are of the same order, in this case up to cubic terms). When $\tau = 0$, we expect all three to be essentially identical, because the true model is reduced to polynomial form. However, as τ increases, the changes are instructive; Fig. 6 shows several measures of the models as τ is varied. The top panel shows the error between the true model and the reconstructed model, i.e. the basic measure of the method’s effectiveness. The second and third panels show the relationships with the *truncated* model, first between the truncated model and the reconstructed model in the second panel, and the true model and truncated model in the third. Finally, the bottom panel shows the minimum and maximum of each solution.

For small values of τ , as expected, all three models are essentially identical¹¹. As τ is increased, all errors increase gradually at first, as might be expected given that we are making the true model increasingly “non-polynomial” while still using a polynomial basis for the reconstruction. However, note the point at which the truncated model undergoes qualitative change from periodic to equilibrium solution (the vertical, dashed grey line is a visual aid); at this point the true model and the truncated model diverge, and the reconstructed model tracks the former and not the latter. Thus the intuitive explanation about using a polynomial basis to reconstruct non-polynomial dynamics – namely, that one obtains the truncated Taylor series of the non-polynomial model – is, at best, incomplete. However, a complete understanding of this relationship remains an area of future work.

5. The Orbit of Mars

Our second example is that of the orbits of the planets, as studied by Tycho Brahe and Johannes Kepler in the late 16th and early 17th centuries. Between 1582 and 1600, Tycho Brahe collected meticulous observational data on the declination¹² of Mars, shown in Fig. 7. The characteristic “retrograde” motion is clear to see, in which Mars occasional appears to reverse course briefly, before returning to its previous trajectory. This retrograde motion could be partially explained by Copernicus’ heliocentric model, but because that model assumed circular orbits about the sun rather than elliptical, there were large prediction errors. In 1593 the error between observation and Copernican prediction reached 5 degrees, and caused sufficient shock to be referred to as a “catastrophe”.

¹¹Values of τ smaller than 10^{-2} were computed but are not shown for clarity.

¹²What, precisely, is declination? For the moment it is sufficient to have a vague notion that it is some sort of angle of Mars as observed from Earth; the precise definition will become important later on but it is not particularly informative to delve into the details now.

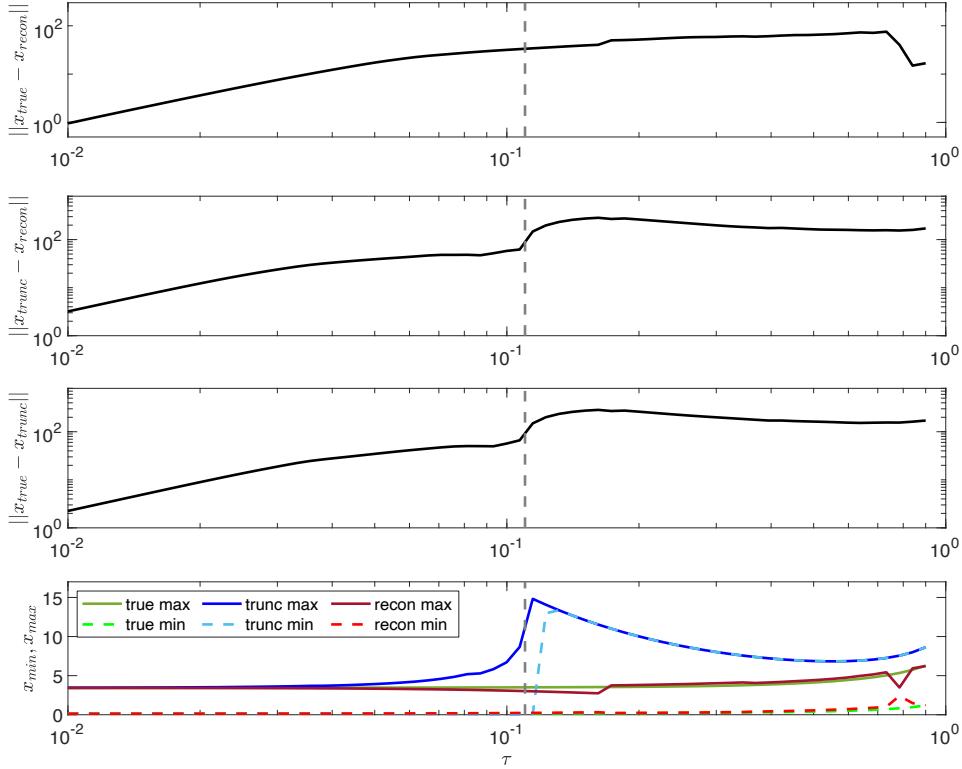


FIGURE 6. Relationship between true model, truncated model, and reconstructed model as the homotopy parameter τ is varied. Note that as $\tau \rightarrow 0$ the true model is of polynomial form and hence identical to the truncated model. The vertical, dashed grey line is a visual aid to indicate the location of the divergence between the reconstructed model and the truncated model. Top panel: observation error between the true model and the reconstructed model; second panel: observation error between the truncated model and the reconstructed model; third panel: observation error between true model and truncated model; bottom panel: minimum and maximum of each solution. Note that this panel is in a linear scale, as opposed to the logarithmic scale in the upper panels.

Upon Tycho Brahe's death in 1601 these observations passed to Kepler, whose subsequent analysis allowed the formulation of Kepler's famous three laws of planetary motion. Although these are based on geometry and not the underlying physics, they established the elliptical orbits, solved the Martian catastrophe, cemented the heliocentric view, and paved the way for Newton's later solution in physical rather than geometric terms. This makes an instructive example for our efforts to reconstruct the underlying dynamics from observational data. What can we reconstruct from Tycho Brahe's original observations? Can we discover from these data any of the underlying dynamics?

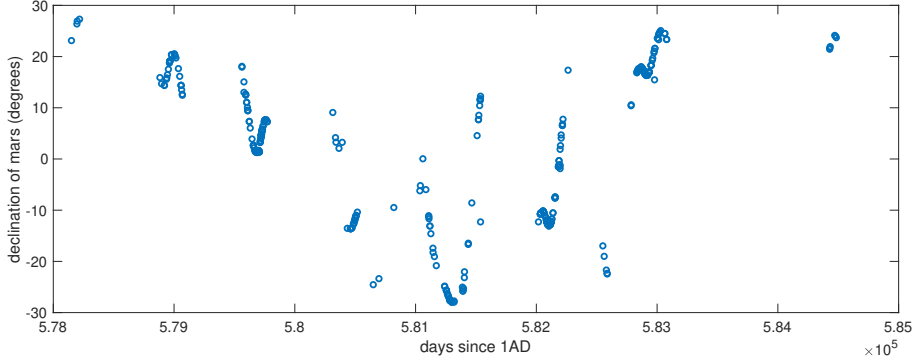


FIGURE 7. Tycho Brahe’s original observations of the declination of Mars, collected between November 1582 and March 1600. To avoid difficulties with calendar systems time is expressed in days since 1 AD.

It will perhaps not surprise you, having read the previous section on lynx and here, to learn that a naive reconstruction of the declination of Mars in terms of low-order polynomials is not productive: the reconstruction is neither accurate nor sparse, and certainly not informative as to the true underlying dynamics. In order to understand why this is so, we will use a series of simpler examples. Let’s start with the so-called Kepler problem for the orbit of a single planet in polar coordinates about the sun, given by

$$\ddot{r} = \frac{L^2}{m^2 r^3} - \frac{k}{mr^2} \quad (5.1)$$

$$\dot{\theta} = \frac{L}{mr^2} \quad (5.2)$$

with parameters L , m , and k for angular momentum, mass, and a constant of gravitational attraction, respectively. You may have noticed the presence of the inverse square law, i.e. that the force of the attraction varies as the inverse square of the distance. The usual procedure is to change independent variable from t to θ , and then apply the transformation $u = 1/r$, solving the resulting quasi-linear equations to obtain the orbits in terms of conic sections. We will not use that procedure here, but instead to maintain consistency with our previous framework, we will reduce the system to first order as

$$\dot{r} = \rho \quad (5.3)$$

$$\dot{\rho} = \frac{L^2}{m^2 r^3} - \frac{k}{mr^2} \quad (5.4)$$

$$\dot{\theta} = \frac{L}{mr^2}. \quad (5.5)$$

Thus the system is explicitly in the form of Eq. 2.1 with $\mathbf{x} = [r \ \rho \ \theta]^T$.

As before, we generate some synthetic data from this system (using very simple “nice” parameters $k = L = m = 1$), and then try to infer the underlying dynamics from that, using a polynomial basis. The resulting reconstruction is shown in Fig. 8,

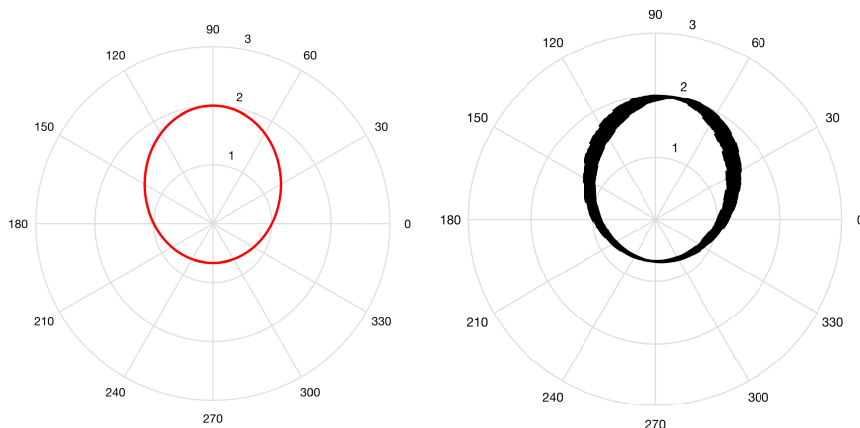


FIGURE 8. Kepler problem reconstruction from synthetic data with naive polynomial basis. Left: snapshots, right: reconstruction.

and the reconstructed equations are¹³

$$\dot{r} = 1.0000\rho \quad (5.6)$$

$$\dot{\rho} = 1.9469 - 1.3549r - 3.0172\rho^2 + 0.1590r^2 \quad (5.7)$$

$$\dot{\theta} = 4.0907 - 3.1496r - 2.2220\rho^2 + 0.6163r^2. \quad (5.8)$$

So this has not worked particularly well – the orbits are drifting, and our reconstruction does not match the true underlying system. What has gone wrong? The obvious thing is that our true dynamics are not in the polynomial basis which we are trying to use for reconstruction, so perhaps we are getting Taylor approximations to the $1/r^2$ and $1/r^3$ terms (in Eqs. 5.4 and 5.5) which are not particularly accurate or informative. (The simple polynomial form $\dot{r} = \rho$ is reconstructed without difficulty.)

Of course, we can “fix” this by tailoring our library of nonlinear basis functions. We chose them to be low order polynomials, but we can choose them to be something else; what if we include some $1/r^q$ terms? If we do, this recovers the true governing equations almost exactly, with resulting excellent reconstruction (Fig. 9) and reconstructed equations

$$\dot{r} = 1.0000\rho \quad (5.9)$$

$$\dot{\rho} = -1.0000\frac{1}{r^2} + 1.0000\frac{1}{r^3} \quad (5.10)$$

$$\dot{\theta} = 1.0000\frac{1}{r^2}. \quad (5.11)$$

But, how did we know to do this? Only because we knew the answer already – in essence, that we already know that the inverse square law is in there! We can't even include all of the $1/x^q$ type terms without causing problems because the $1/\rho^q$

¹³In the previous section we wrote out all of the coefficients explicitly, but now we omit the zero coefficient terms.

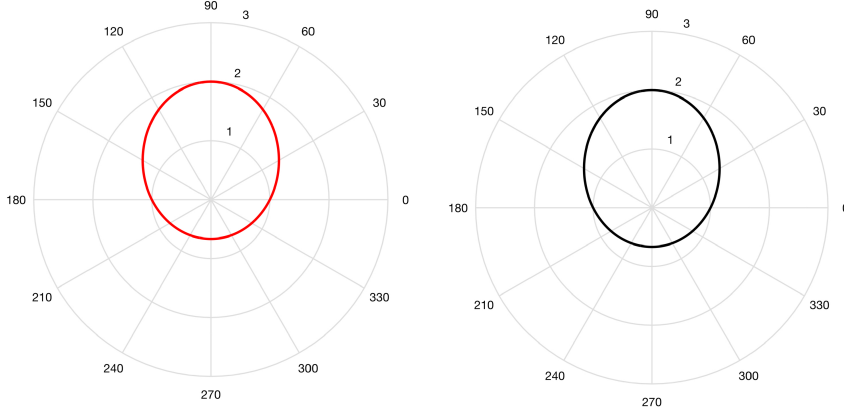


FIGURE 9. Kepler problem reconstruction from synthetic data with polynomial basis augmented with $1/r^q$ terms. Left: snapshots, right: reconstruction.

terms blow up (when ρ passes through 0), so in order to make this work we really do need to know specifically about the $1/r^q$ terms.

Even with just a single planet, and synthetic data, we have already been forced to augment our naive polynomial basis in way which may not have been obvious *a priori*, and we are still some way from even being able to attempt to capture declination. In order to get closer, let's now consider a system with two planets: two copies of what we had above, with variables r_A, θ_A and parameters k_A, L_A and m_A for planet A, and corresponding B subscripts for planet B. So far, then, we have a six-dimensional system: $(r_A, \rho_A, \theta_A, r_B, \rho_B, \theta_B)$. To get closer to declination, we will also add in an observation for the relative observation angle of planet B from planet A, as

$$\phi = \arccos\left(\frac{r_B^2 - r_A^2 - a^2}{-2r_A a}\right) \quad (5.12)$$

$$\text{where } a = \sqrt{r_A^2 + r_B^2 - 2r_A r_B \cos(\theta_B - \theta_A)} \quad (5.13)$$

thus giving a seventh variable. (This isn't yet Earth, Mars, and declination, but it's a toy problem working in that direction.)

Again we generate synthetic data from this system and attempt a reconstruction based on that, including not just our polynomial library functions but also the $1/r_A^q$ and $1/r_B^q$ terms. The parameters are still conveniently idealised with $k_A = L_A = m_A = k_B = L_B = 1$ and $m_B = 1/2$. Attempting reconstruction of this system

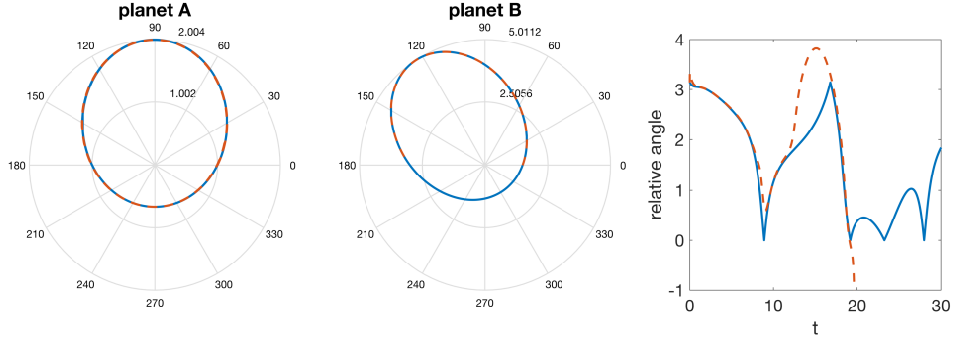


FIGURE 10. Attempted reconstruction of idealised two-planet system, using a polynomial basis augmented with $1/r_A^q$ and $1/r_B^q$ terms. Blue curves are snapshot data, and dashed red is the reconstruction.

yields Fig. 10 and

$$\begin{aligned}
 \dot{r}_A &= 1.0001\rho_A \\
 \dot{\rho}_A &= -0.9984\frac{1}{r_A^2} + 0.9980\frac{1}{r_A^3} \\
 \dot{\theta}_A &= -0.7493 + -0.9982\rho_A^2 + 1.9992\frac{1}{r_A} \\
 \dot{r}_B &= 1.0000\rho_B \\
 \dot{\rho}_B &= -1.9974\frac{1}{r_B^2} + 3.9934\frac{1}{r_B^3} \\
 \dot{\theta}_B &= 1.9990\frac{1}{r_B^2} \\
 \dot{\phi} &= 4.6009 \times 10^3 - 697.3503\rho_A - 310.4976r_A - 174.4954\theta_A - 201.9708\rho_B - 972.0965r_B \\
 &\quad + 305.3059\theta_B - 91.5882\phi - 57.4168\rho_A^2 + \dots - 5.5448 \times 10^3\frac{1}{r_B^2} + 3.7874 \times 10^3\frac{1}{r_B^3}
 \end{aligned}$$

where terms coloured red match the known underlying dynamics (within 1%), and the rest either do not match (in the case of $\dot{\theta}_A$) or we don't yet know the true underlying form ($\dot{\phi}$)¹⁴. In terms of the orbits themselves, we have obtained a reasonable reconstruction, although it is imperfect in the sense that we have not obtained the correct expression for θ_A (even though the θ_A dynamics are reconstructed well over this observational period.) The relative angle ϕ has proved much more problematic: not only is the reconstruction poor, except perhaps for very short times, but the coefficients are not sparse (in fact they're all nonzero) and the reconstruction is not informative in any way.

In order to see why, consider what we are attempting to reconstruct: $\dot{\phi}$. We haven't written this out explicitly (and won't do so!) but consider that we could,

¹⁴We omit the rest of the quadratic terms. All the coefficients are nonzero, but you already get the point about this reconstruction without writing them all out!

and what would be involved. We would start by taking Eq. 5.12 and differentiating with respect to time, substituting 5.13 and 5.3–5.5 as required. In the first step, differentiating $\arccos(\cdot)$ gives us $\frac{-1}{\sqrt{1-(\cdot)^2}}$, so we’re clearly not in a nice polynomial basis, and things only get worse from there as we chain out all of the other dependencies. It’s not even informative to try to write it all out, except to say that it’s horrendous, and so we shouldn’t be surprised when we get a non-sparse approximation to it which is neither accurate nor informative.

So things are not looking good in our quest to outdo Kepler using equation discovery, and we are still missing a few things:

- (1) We’ve so far just used nice, made-up parameters and initial conditions, rather than the actual physical parameters of the system, which has glossed over some potential issues.
- (2) We still don’t actually have a measure of the declination of Mars, just a fabricated angle which is qualitatively similar but quantitatively quite different.
- (3) We are using a synthetic system which includes observations of the positions of the planets, not just the declination on its own.
- (4) We are using clean, noise-free synthetic data with very frequent observations, as opposed to the real data, which is both noisy and infrequent¹⁵.

Obtaining reasonably accurate parameters and initial conditions for our orbiting planets isn’t too difficult: NASA’s Jet Propulsion Laboratory (JPL) provides the publicly and freely available *JPL Horizons* service with a wealth of data [7], including historical predictions. Recall that our period of interest is 1582–1600; intriguingly, and perhaps not coincidentally, JPL Horizons provides data for Earth’s orbit back to 1583 and Mars’ back to 1600¹⁶. Thus we can ascertain that at 1600-Jan-02 00:00 Mars’ configuration was at $r = 1.639672647881$ AU (astronomical units) and $\dot{r} = 1.2000278$ km/s, where 1 AU is approximately the average distance from the Earth to the sun, and specifically $1 \text{ AU} = 149,597,870.700$ km. Notice the mismatch of the magnitude in these units! So, what units should we use? Does it matter?

The answer is that it does. Up to this point we have implicitly assumed that our non-zero reconstruction coefficients are of the same magnitude. Consider again our sparse regression method and the parameter λ (Eq. 3.6). We are in effect penalising parameters which would be “small” in a traditional least-squares regression, and instead forcing them to be zero. Obviously this relies on the true values not being small. Thus if we choose our system scaling such that some variables (and their coefficients) are much larger (or smaller) than others, this will cause difficulties with appropriate enforcement of sparseness; we run the risk of either zeroing out “true” small coefficients, or keeping spurious fit-error coefficients which should be eliminated¹⁷.

¹⁵Relatively speaking, of course. Tycho Brahe was an amazingly good observer and the data is renowned for its accuracy in a historical context.

¹⁶This is actually sufficient, as with this model we can run the orbits backwards in time just as well as forwards by changing $t \rightarrow -t$. Of course we are not really using a good orbital model, just one sufficient to illustrate this method.

¹⁷This can be at least partially addressed by normalising the columns of Θ . It is also worth noticing that in practice, LASSO is sometimes not used, due to computational cost, in favour of

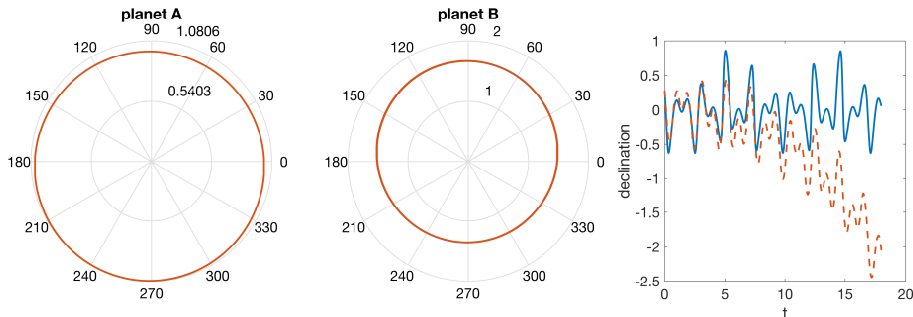


FIGURE 11. Attempted reconstruction of the two-planet system from synthetic data, using a polynomial basis augmented with $1/r_A^q$ and $1/r_B^q$ terms. Blue curves are snapshot data, and dashed red is the reconstruction. Declination is expressed in radians.

To cut a long story short, we’ll choose units of AU, Earth years, and Earth masses (5.97×10^{24} kg) to ensure that our target coefficients are of the same magnitude, at least for the orbits themselves. That leaves the thorny question of how to calculate declination. First, we need to say what it actually is. From the perspective of an observer on Earth, the concept is relatively straightforward. In the same way that positions on the Earth’s surface can be described by latitude and longitude, positions in the sky (the ‘celestial sphere’) are described by *declination*, which in this analogy corresponds to latitude, and *right ascension*, which corresponds to longitude. Latitude is expressed relative to the equator, and similarly declination is expressed, usually in degrees, relative to the celestial equator. Thus one can see from Tycho Brahe’s data in Fig. 7 that the declination of Mars varied from about 30 degrees above the celestial equator to about 30 degrees below during the 1582–1600 observation period.

Thus declination is a natural measurement from the perspective of an Earth-based observer. It is not, however, nice to express in terms of heliocentric polar coordinates, which is the nice way to express the orbits. In order to calculate declination from the positions of the planets, the usual procedure is to change from heliocentric polar (or spherical) coordinates to geocentric Cartesian coordinates, taking into account the axial tilt of the earth and the orbital angle of Mars (it’s not quite in-plane with the Earth’s orbit), before changing back to geocentric spherical coordinates. With the changes of coordinate system it is not even easy to write as an expression akin to the observational angle (Eq. 5.12) that we used in our toy model. That said, we can calculate it, and so it can be used in our synthetic snapshot data.

Using then something close to the real orbital parameters of Earth and Mars, and calculation of declination, we can again generate synthetic data and attempt to reconstruct the underlying dynamics, resulting in Fig. 11 and the “discovered” equations

an alternative sparse regression method known as sequential thresholded least-squares [1] in which the sparseness parameter λ itself is dimensional; in this case the scaling is all the more important.

$$\begin{aligned}
\dot{r}_A &= 1.0003\rho_A \\
\dot{\rho}_A &= 1.2418r_A + 4.2957r_B - 88.4075\frac{1}{r_A} + 55.3138\frac{1}{r_A^2} + 7.2831\frac{1}{r_A^3} \\
&\quad + 57.7100\frac{1}{r_B} - 57.4511\frac{1}{r_B^2} + 21.4183\frac{1}{r_B^3} \\
\dot{\theta}_A &= 6.3729\frac{1}{r_A^2} \\
\dot{r}_B &= -12.8429r_A + \dots \\
\dot{\rho}_B &= 292.9121r_A + \dots \\
\dot{\theta}_B &= 53.3507r_A + \dots \\
\dot{\delta} &= 9.4484\rho_A - 5.5540 \times 10^3 r_A - 11.0393\theta_A + 2.6393 \times 10^3 r_B + 19.5811\theta_B \\
&\quad - 1.4895 \times 10^4 \frac{1}{r_A} + 3.1831 \times 10^3 \frac{1}{r_A^2} + 988.7002 \frac{1}{r_A^3} + \dots
\end{aligned}$$

where δ is the declination and (...) omits many incorrect and unpleasant terms. Suffice it to say, this reconstruction is neither accurate nor informative as to the underlying dynamics, and this is in a synthetic case where we have given ourselves every opportunity to succeed. By using synthetic data, we gave ourselves high-frequency observations where the data is low-frequency and sporadic; we have noiseless data where the true observations are not; we gave ourselves additional observational variables, and augmented our polynomial basis with $1/r^q$ terms, both of which were implicitly informed by the fact that we know the underlying physics. Even with all of these advantages, there is no overcoming the fact that declination does not have a “nice” expression in heliocentric coordinates, and we’re trying to reconstruct something which just doesn’t exist.

6. Additional Examples and Code Availability

For readers interested in more examples, it is useful to note that the supplemental material from reference [1] has more useful examples, if not exactly presented in a tutorial style. Software implementations are available both as MATLAB code [1] and a python implementation [5].

7. Discussion

Using two classic examples we have explored the potential for equation discovery to reconstruct unknown underlying dynamics from snapshot data. The method has potential as a powerful tool, but also has substantial limitations, as we have explored with these examples. The first caveat is that equation discovery is only suitable for problems with lots of very high-quality data, both in terms of the frequency of observations and the levels of noise. However, it is worth noting that the state-of-the-art is being pushed back in this regard; two recent variations, *ensemble SINDy* [6] and *uncertainty quantification SINDy* [8] both are able to reconstruct something close to Lotka-Volterra dynamics from the Hudson Bay dataset, at least under certain assumptions.

The more subtle difficulty is that equation discovery cannot really operate successfully in situations in which we know nothing about the underlying dynamics

and expect the discovery process to give us everything. The reasons for this are that choosing the right basis requires at least some insight into the system under consideration; naively applying a low-order polynomial basis to whatever set of observations comes immediately to hand will only result in a useful reconstruction with a great deal of luck. On the other hand, if we know enough about the underlying dynamics of the system, then we don't really need equation discovery at all – we can just model the system in the usual way¹⁸. So there is something of a Goldilocks effect, where we need to know the right amount about the system: not too little (because we'll fail) and not too much (because then there is no real need). In some situations one can use an auto-encoder to select an appropriate basis transformation [2], but this comes at the cost of sacrificing the interpretability which, to these authors at least, is at the heart of the equation discovery problem.

The version of SINDy used here is not precisely the latest and greatest; these methods are under continuing development. For example, SINDy-PI [10], where PI stands for the 'parallel implicit', offers several improvements. It relaxes the data quality requirements, and so goes some way to addressing our first caveat. The implicit formulation also allows more naturally for rational basis functions. Other more recent developments include ensemble- and uncertainty quantification-SINDy (mentioned above) as well as a weak formulation with reduced data requirements [12]. There are also issues around data filtering which we have not discussed here but can be important in practice [1]. These methods, however, are beyond the desired scope of this article.

Appendix A: Numerical Differentiation

In many cases, snapshot observational data will not contain direct information about the time derivative and it is necessary to approximate the derivative numerically. One might suppose that it is sufficient to use a simple finite difference approach, i.e. given observations $x(t_i)$ and $x(t_{i+1})$, we can approximate

$$\dot{x}(t_i) = \frac{x(t_{i+1}) - x(t_i)}{\Delta t} + \mathcal{O}(\Delta t) \quad (7.1)$$

where $\mathcal{O}(\Delta t)$ indicates the approximation error terms. In the absence of noise, this approach is often adequate, but of course real data always has at least some noise.

But suppose we have noise in our observations: that is, we might observe not the "true" values ($x(t_i)$ and $x(t_{i+1})$) but only noisy observations $\tilde{x}(t_i) = x(t_i) + \eta_i$ and $\tilde{x}(t_{i+1}) = x(t_{i+1}) + \eta_{i+1}$. In that case, when we attempt to approximate the derivative using 7.1 we would have

$$\dot{\tilde{x}}(t_i) \approx \frac{x(t_{i+1}) + \eta_{i+1} - (x(t_i) + \eta_i)}{\Delta t} \quad (7.2)$$

$$= \frac{x(t_{i+1}) - x(t_i)}{\Delta t} + \frac{\eta(t_{i+1}) - \eta(t_i)}{\Delta t} \quad (7.3)$$

$$= \dot{x}(t_i) + \mathcal{O}(\Delta t) + \frac{\eta(t_{i+1}) - \eta(t_i)}{\Delta t}. \quad (7.4)$$

¹⁸Indeed, if we know enough about the system to constrain our library to contain only the active terms, then we are essentially just parameter fitting.

That is, we now have an additional error term arising from the noise, and in fact this can be very large, at least relative to the first two terms – to the point that this method often produces total nonsense. So, in the presence of noise, we have to do something more sophisticated. For further details, see e.g. [18, 9, 4].

Appendix B: Sequential Thresholded Least-Squares Algorithm

In practice, the LASSO sparse regression method (see Eq. 3.6) is often not used due to computational cost in favour of the *sequential thresholded least-squares algorithm* due to [1]. The idea is relatively simple: any regression coefficients $< \lambda$ are set to zero, and the associated basis functions are removed; then the regression is repeated onto the reduced basis. As an algorithm, this is:

```

obtain  $\Xi$  by least-squares regression of  $\dot{\mathbf{X}}$  onto  $\Theta$ .
for number of thresholding iterations do
  for  $j$  from 1 to  $n$  (number of state variables) do
    set any elements of  $\xi_j < \lambda$  to zero
    remove the associated basis functions to obtain the reduced basis  $\tilde{\Theta}$ 
    obtain an updated  $\tilde{\xi}_j$  by regression of  $\dot{\mathbf{X}}_j$  onto the reduced basis  $\tilde{\Theta}_j$ 
  end for
end for
return  $\tilde{\Xi}$ 

```

See also the supplemental material to reference [1] for a MATLAB implementation.

References

- [1] S. L. Brunton, J. L. Proctor and J. Nathan Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proc. Natl. Acad. Sci. USA **113** (15) (2016), 3932–3937.
- [2] K. Champion, B. Lusch, J. N. Kutz and S. L. Brunton, *Data-driven discovery of coordinates and governing equations*, Proc. Natl. Acad. Sci. USA **116** (45) (2019), 22445–22451.
- [3] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton and J. N. Kutz, *A unified sparse optimization framework to learn parsimonious physics-informed models from data*, IEEE Access, 8:169259–169271, 2020.
- [4] R. Chartrand, *Numerical differentiation of noisy, nonsmooth data*, ISRN Appl. Math. (2011), Article ID 164564.
- [5] B. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. Brunton, *Pysindy: A python package for the sparse identification of nonlinear dynamical systems from data*, J. of Open Source Software **5** (49) (2020), 1–4.
- [6] U. Fasel, J. N. Kutz, B. W. Brunton and S. L. Brunton, *Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control*, Proc. R. Soc. A **478** (2260) (2022), 20210904.
- [7] J. D. Giorgini, *Status of the JPL horizons ephemeris system*, IAU General Assembly **29** (2015), 2256293.

- [8] S. M. Hirsh, D. A. Barajas-Solano and J. N. Kutz, *Sparsifying priors for bayesian uncertainty quantification in model discovery*, R. Soc. Open Sci. **9** (2) (2022), 211823.
- [9] M. S. Hosseini and K. N. Plataniotis, *Derivative kernels: Numerics and applications*, IEEE Transactions on Image Processing **26** (10) (2017), 4596–4611.
- [10] K. Kaheman, J. N. Kutz, and S. L. Brunton, *SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics*, Proc. R. Soc. A **476** (2242) (2020), 20200279.
- [11] A. J. Lotka, *Contribution to the theory of periodic reactions*, The Journal of Physical Chemistry **14** (3) (1910), 271–274.
- [12] D. A. Messenger and D. M. Bortz, *Weak SINDy: Galerkin-based data-driven model selection*, Multiscale Modeling & Simulation **19** (3) (2021), 1474–1497.
- [13] J. D. Murray, *Mathematical Biology: I. An Introduction*, Biomathematics 19, Springer, New York, 2007.
- [14] M. L. Rosenzweig and R. H. MacArthur, *Graphical representation and stability conditions of predator-prey interactions*, The American Naturalist **97** (895) (1963), 209–223.
- [15] H. L. Smith, *The Rosenzweig-MacArthur predator-prey model*, available online at <https://math.la.asu.edu/halsmith/Rosenzweig.pdf>.
- [16] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society Series B (Methodological) **58** (1) (1996), 267–288.
- [17] P. Turchin, *Complex Population Dynamics: a Theoretical/Empirical Synthesis*, Monographs in Population Biology 35, Princeton University Press, Princeton, 2003.
- [18] F. Van Breugel, J. N. Kutz and B. W. Brunton, *Numerical differentiation of noisy data: A unifying multi-objective optimization framework*, IEEE Access, **8**:196865–196877, 2020.
- [19] V. Volterra, *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi*, Società anonima tipografica “Leonardo da Vinci”, 1926.

G. M. Donovan
University of Auckland,
Private Bag 92019,
Auckland 1142,
New Zealand
g.donovan@auckland.ac.nz

Q. Su
University of Auckland,
Private Bag 92019,
Auckland 1142,
New Zealand
sqin055@aucklanduni.ac.nz