

Data-driven Evolution Equation Reconstruction for Parameter-Dependent Nonlinear Dynamical Systems

David W. Sroczynski,^[a] Or Yair,^[b] Ronen Talmon,^{*,[b]} and Ioannis G. Kevrekidis^{*,[c]}

Abstract: When studying observations of chemical reaction dynamics, closed form equations based on a putative mechanism may not be available. Yet when sufficient data from experimental observations can be obtained, even without knowing what exactly the physical meaning of the parameter settings or recorded variables are, data-driven methods can be used to construct minimal (and in a sense, robust) realizations of the system. The approach attempts, in a sense, to circumvent physical understanding, by building intrinsic “information geometries” of the observed data, and thus enabling prediction without physical/chemical knowl-

edge. Here we use such an approach to obtain evolution equations *for a data-driven realization of the original system* – in effect, allowing prediction based on the informed interrogation of the agnostically organized observation database. We illustrate the approach on observations of (a) the normal form for the cusp singularity, (b) a cusp singularity for the nonisothermal CSTR, and (c) a random invertible transformation of the nonisothermal CSTR, showing that one can predict even when the observables are not “simply explainable” physical quantities. We discuss current limitations and possible extensions of the procedure.

Keywords: kinetics · reaction mechanisms · model development · machine learning · diffusion maps

1. Introduction

Obtaining predictive dynamical equations from data lies at the heart of science and engineering modeling, and is the linchpin of our technology. In mathematical modeling one typically progresses from observations of the world (along with experience and some serious thinking!) first to choice of relevant variables and parameters, then to equations for a model, and then to the analysis of the model to make predictions. Good mathematical models give good predictions (and inaccurate ones do not) – but the computational tools for analyzing them are the same: algorithms that are typically based on closed form equations. Today we increasingly witness the development of mathematical techniques that appear to circumvent the experience and the serious thinking that goes into selection of variables and parameters and the derivation of meaningful equations; these techniques “jump” directly from data to predictions through mathematical tools, but arguably without physical understanding.

Nonlinear system identification (the recovery of the underlying structure of nonlinear dynamical systems from data) has been the subject of considerable research over the years, and the appearance of useful and inspired techniques addressing the problem is accelerating recently in the literature as a result of machine learning developments. Methods to automatically find nonlinear differential equations,^[1,2] methods to discover governing equations from time-series,^[3,4] equation-free modeling approaches,^[5] and methods for empirical dynamic modeling^[6] testify to this acceleration. In a recent paper^[7] we presented a technique with roots in manifold learning,^[8,9,10,11] which involved metric learning and approximation^[12] as well tensor-geometry learning;^[13,14,15] the technique was able to

create minimal realizations of nonlinear dynamical systems from data (in the spirit of Kalman^[16] and Moore,^[17] see also Lall et al.^[18]).

The purpose of this contribution, after starting with a succinct discussion of the basis and structure of the technique, is to extend it beyond the discovery of useful realizations of parameters and state variables for nonlinear dynamical systems. The extension consists of outlining, implementing, and demonstrating the use of an approach for obtaining predictions in time: going beyond the discovery of minimal effective parameter space and state space realizations, and constructing (approximate) effective dynamical evolution equations in this state space.

The paper is organized as follows: Section 2 starts with a reasonably self-contained description of the important components of the approach introduced in Yair et al.^[7] for data driven construction of parameter space and state space realizations.

[a] D. W. Sroczynski
Department of Chemical and Biological Engineering
Princeton University
Princeton, NJ 08540

[b] O. Yair, R. Talmon
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa 32000, Israel
E-mail: ronem@ee.technion.ac.il

[c] I. G. Kevrekidis
Department of Chemical and Biomolecular Engineering; Applied Mathematics and Statistics; and Urology (JHMS)
Johns Hopkins University
Baltimore, MD 21218
E-mail: yannisk@jhu.edu

We then describe our contribution: the extension to enable prediction in time through the data-driven construction of (effective differential) evolution equations in this “discovered” state space.

The examples have been chosen with chemical engineering applications in mind: the first “textbook” example is the data-driven recovery of the cusp normal form^[19,20] from data (the basic singularity underlying steady state multiplicity in chemical systems). The second example comes from the foundations of chemical reaction engineering: an example of multiplicity in the context of the classical Uppal, Ray, and Poore nonisothermal CSTR with a single first order reaction.^[21,22] The third example bends a little towards machine learning (as opposed to chemical understanding): predictive equations are constructed in terms of “not-so-human” variables: random (yet invertible!) nonlinear observations combining the chemical bedrock variables for kinetics: concentration and temperature. We demonstrate that the approach can still detect the nature of the singularity and its unfolding. We then close with a discussion of some of the open problems and possible extensions and implications of this class of approaches.

2. Computational Methods

2.1 Iteratively Refined Informed Metrics

Consider a nonlinear, deterministic system with an underlying state space \mathbf{X} described by d_x state variables, which we observe through some observation function h in a space \mathbf{Y} described by d_y observation variables. The governing equations are given by

$$\begin{aligned} \frac{dx}{dt} &= f(x; p) \\ y &= h(x), \end{aligned} \quad (1)$$

where \mathbf{p} represents an *unknown* list of system parameters. We begin from the viewpoint that, in this setting, the current state of any experiment, be it physical or computational, is determined by three components: (a) the parameter settings for the experiment (the *trial* as we will refer to it), (b) the initialization of the state variables, and (c) the time elapsed since the initialization. With no prior knowledge of the dimensionality of either the parameter space \mathbf{P} or the state space \mathbf{X} , we wish to uncover a coordinate system for each space (a realization) that captures its effective dimensionality. We will do this in a data-driven way, solely through observations in \mathbf{Y} ; while the observation space itself may be higher or lower dimensional than that of the true state space, the variability of trajectories in the observation space carries information about the variability in state space. The use of time-delay coordinates, taking advantage of the Takens^[23] (and through it of the Whitney^[24,25]) embedding theorem, will allow us to use the history of the observations along a trajectory to

create rich enough embeddings of the observations^[26] to be “one-to-one” with the unknown “true” dynamics.

To develop this new set of embedding coordinates, we will consider data from N_p sets of experiments, or *trials*, each with a different and unknown parameter setting \mathbf{p} . We will make no explicit assumptions about how these parameter settings are drawn from some distribution; we only – nonrigorously – assume that these trials cover the parameter space of interest uniformly enough and densely enough that behaviors at intermediate parameter values can be accurately interpolated from the ones in our database. In other words, the trials that we have are *sufficiently representative* of all possible dynamic behaviors in the parameter regime of interest. The mathematics for quantifying and testing these assertions, as well as the sensitivity of the results to other “equally representative” ensembles of initial conditions, are beyond the scope of this paper (see for example Coifman and Lafon^[14]). Note that the uniformity assumption is much less significant than the sufficient density assumption: we need enough information to be ultimately able to interpolate across our database once we have “made sense” of it and usefully organized/ordered it. For each of the N_p trials we run experiments (computations) from a set of N_v initial system states, which are unknown but common across all trials. We can think of this as having N_v *measurement channels*; and then for each experiment (each trial) we observe \mathbf{y} at a set of N_t *predetermined* (for this paper) time points. This produces a data set which can be viewed as an $N_p \times N_v \times N_t$ tensor of observations. Each element of this 3D tensor represents a point in observation space for the experiment at the corresponding parameter, initialization, and time indices.

Suppose that we are first interested in determining data-driven embedding coordinates for the parameter space. One approach is to use diffusion maps,^[14,27] a manifold learning algorithm which relies on the construction of a set of weights (similarities) between each possible pair of data samples. A weight of 1 indicates that two samples are identical, while a weight close to 0 indicates that two samples are very dissimilar. In this case, the set of samples that we would like to use is the set of the N_p parameter settings from our experiments. However, for each parameter setting, what we have access to is an $N_v \times N_t$ matrix of observations, i. e., the set of all observation trajectories from that parameter setting. For a given parameter setting \mathbf{p} , we will write this observation matrix as a vector \mathbf{y}_p of length $N_v \times N_t$ by column-stacking the matrix elements.

Given this set of N_p observations, the weights between each pair are then given by

$$w_{ij} = \exp\left(-\frac{d(y_{p_i}, y_{p_j})^2}{\varepsilon^2}\right), \quad (2)$$

where $d(\dots)$ represents a chosen distance metric, and ε represents a distance scale below which samples are consid-

ered similar. We are mostly concerned with this distance metric d ; it is common practice to use the Euclidean distance or the l_1 norm of the difference if there is no *a priori* reason to use another metric. In our case, taking into account the coupling (that is, the similarity, to be discussed below) between different initial states and different time points provides additional information about the observations that could be exploited to develop a more “informed” metric.^[28] To accomplish this, we *append new coordinates* to each observation vector y_p . Let $F(y_{p_i})$ denote the appended vector of coordinates. Accordingly, the l_1 distance between the new samples is defined by:

$$d(y_{p_i}, y_{p_j}) = \|y_{p_i} - y_{p_j}\|_1 + \|F(y_{p_i}) - F(y_{p_j})\|_1. \quad (3)$$

In the current implementation, each element of the appended vector F consists of the inner product of the original sample y_p with *some* basis function g of length $N_v \times N_t$, which is explicitly given by:

$$F_i(y_p) = \sum_{v=1}^{N_v} \sum_{t=1}^{N_t} g_i(v, t) y_p(v, t). \quad (4)$$

There exist a variety of basis function options to choose from; in the spirit of data-driven geometry learning, we use a multi-level clustering approach employing a hierarchical tree.^[28,29] At the bottom level of the tree, each sample belongs to its own tree leaf. For each level moving up, some leaves, and then tree folders, are merged to create larger and larger folders, until reaching the root of the tree, where all the samples belong to a single root folder. The clustering required to construct the folders at each level of the tree is based on a metric between the samples.

Similarly to the observation vectors corresponding to the parameters, when we want to create a useful parametrization of the state variables, for each initial condition we have an $N_p \times N_t$ matrix of observations, i.e., the set of all trajectories starting at that initial condition. For a given initial condition index v , we will denote the column-stack representation of this observation matrix by y_v . To start with, we simply use the l_1 norm between these samples y_v as our metric for clustering. Once we have a set of tree folders $\{I_l\}$ in multiple levels for the initial conditions, we analogously construct a set of tree folders $\{J_l\}$ for the time samples y_t (i.e., the set of observations for all parameter settings and all initial conditions for a particular time). Here, l and l' are indices which refer to specific folders in our set. For each possible pairing of a folder of initial conditions I_l and a folder of time samples $J_{l'}$, we can construct a basis function that is an indicator function of whether a particular entry in the observation vector has an initial condition index which is in I_l and a time index which is in $J_{l'}$. Formally, we define

$$g_{l,l'}(v, t) = \begin{cases} 1 & v \in I_l, t \in J_{l'} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

When using such basis functions in the computation of inner products, as in (4), this choice harnesses the average of observations with a similar underlying character (in the sense that their initial conditions and time indices are similar) to the formulation of the informed metric. The justification for choosing this type of basis function is further discussed in the references.^[7,15,28,29] Other choices of basis functions are possible and may be more effective; we are currently exploring using the eigenvectors of diffusion maps as our basis functions.

At this point, we have constructed an informed metric for the parameter space which considers the underlying geometry of both the state space (indexed by the initial conditions) and the time evolution, and we have established the framework for iteratively refining the informed metric by cycling through the three viewpoints of the data (parameters, variables, and time, iteratively). Specifically, we can now construct an informed metric for the initial conditions samples y_v by constructing basis functions (which must now be of length $N_p \times N_t$) based on clustering trees of the parameters and time samples, using the same method as above. However, instead of using the “uninformed” l_1 norm for the parameters samples, we use the informed metric obtained during the previous iteration. Once we have obtained, after this second step, the informed metric for the initial conditions, we then proceed to construct an informed metric for the time samples (based on the previous informed metric of the parameters and the previous informed metric of the initial conditions). We proceed iteratively, updating our informed metric for each “tensor axis” (parameters, variables, or time) by using the informed metrics for the other two axes obtained in the preceding iteration step. In practice, we have empirically found that this iterative procedure converges to a useful result after only a few (two to four) iterations. Clearly, this convergence should be the subject of careful mathematical study.

2.2 Reconstructing the Equations

The procedure described above has already been successfully used in reconstructing a Takens-Bogdanov parameter space from unordered observations,^[7] as well as in the study of experimental neuroscience data for which no model exists.^[28] Sample code, including examples from reference 7, is available at <https://github.com/oryair/InformedGeometry-CoupledPendulum>. The present paper goes beyond this “data organization” phase and attempts to reconstruct evolution equations in the realization of the dynamical system that results from the data organization: dynamical equations in the identified state and parameter space that allow us to be predictive about future observations.

Once we are satisfied, after a number of iterations of the algorithm, with its convergence, the result is an informed metric for each axis of our observation tensor. We can now use these metrics for diffusion maps, and this will give us useful embedding coordinates for each axis. The distances between points in these new coordinate systems reflect how the parameters, the initial conditions, and the elapsed time influence the observations. The dimensionality of the parameter embedding indicates the effective dimensionality of the underlying parameter space, and the dimensionality of the initial conditions embedding indicates the effective dimensionality of the underlying state space. In effect, the process gives us a *minimal realization* of the underlying system, and we suspect (although we cannot prove at this point) that beyond minimal, this realization may have features of robustness/balance (in the spirit of Moore^[17] and Lall et al.^[18]).

With these new embedding coordinates acting as stand-ins for the true state variables, we would like to understand how they evolve over time. For simplicity, we will first consider the case where we have data for only a single parameter setting; in this case, our 3D observation tensor becomes a 2D observation matrix, and our basis functions become vectors of length N_v and N_r . To develop a model for the time derivative in the embedding space, we would like to know where each trajectory would be embedded if it started k timesteps later, where k is small enough such that the evolution can be considered well-approximated locally as linear. In general, to embed a new trajectory into our embedding space, we can use the Nyström extension,^[30] a well-known algorithm for calculating diffusion map coordinates for new samples which are not contained in the training database. In our case here, we can do something “neater”: having chosen k , we will take two sub-trajectories from each of the original trajectories, one consisting of the first $(N_r - k)$ time samples, and the second consisting of the last $(N_r - k)$. Thus for each initial condition, we have a trajectory starting at that initial condition, and a shifted trajectory that starts k time steps later. If we include all of these length $(N_r - k)$ trajectories (concatenated along the initial conditions axis), we get a $2N_v \times (N_r - k)$ matrix; we have effectively doubled the number of initial conditions, although we have shortened the time sampling by k . If we run our iterative metric construction on this matrix, not only do we get an embedding for the state space, but we can also see how each corresponding pair of sub-trajectories has shifted in the embedding space, which gives us an estimate for the time derivative at each of the original initial conditions. With this table of shifts, any functional regression method can be used to approximate a global model for the derivative. In the examples below we simply use kriging.^[31] The estimated derivatives at all of the original initial conditions are fed as training data to the kriging algorithm, which returns a model that can give a maximum-likelihood estimation (based on an empirically derived Gaussian process model) of the derivative at any point in the embedding space. To generate the kriging model, we use the DACE toolbox in Matlab, for which the

software and documentation can be found at the internet URL: <http://www2.imm.dtu.dk/projects/dace/>.^[32]

In the full 3D case, there are (surmountable) obstacles to using this method. We cannot, for example, simply include all the shifted trajectories in the iterative metric construction, because the shifted trajectories will not start at the same initial condition across all parameter settings. However, after constructing the metric and getting an embedding, we can embed the shifted trajectories for each given parameter setting separately using a Nyström-like weighted average. In principle, interpolation of the resulting time-derivative models in the parameter embedding is also possible, although this will be the subject of future research. Here we will restrict ourselves to the derivation of approximate evolution equations for a single parameter value.

3. Results and Discussion

3.1 Recovered Embeddings

Our first illustrative example is the normal form of a cusp bifurcation, to which we have added a second linearly-decaying state variable (in anticipation of the 2D CSTR example following). In the two-dimensional parameter space of the cusp unfolding, two branches of a saddle-node bifurcation collide, resulting in transition from three critical points to a single critical point. The governing equations we used are

$$\begin{aligned}\dot{x}_1 &= \beta_1 + \beta_2 x_1 - x_1^3 \\ \dot{x}_2 &= -x_2.\end{aligned}\quad (6)$$

Figure 1 shows the bifurcation map for this system, and Figure 2 illustrates the bifurcation through two representative phase portraits. For $\beta_2 \leq 0$, only one stable critical point is

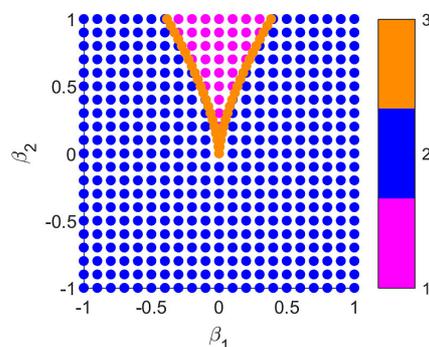


Figure 1. Bifurcation map for the normal form of the cusp bifurcation. In region 1 (pink), there are three critical points, one of which is unstable. In region 2 (blue), there is one stable critical point. Along the saddle-node branches (orange), the unstable critical point collides with one of the stable critical points, resulting in a semi-stable point.

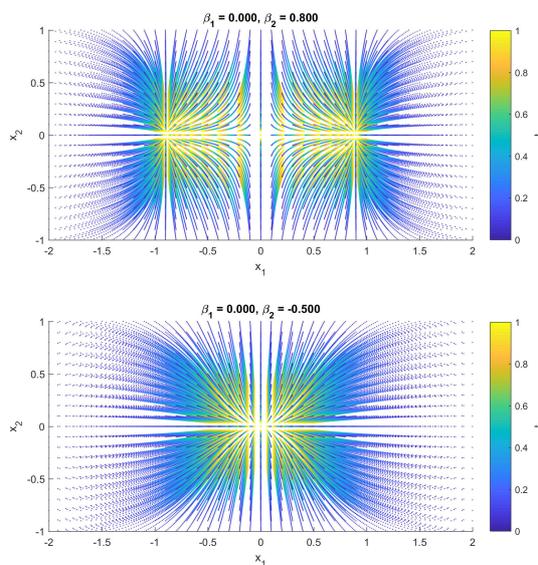


Figure 2. Example phase plots colored by time for parameter settings (top) in the regime with three critical points and (bottom) in the regime with a single stable critical point.

possible. For $\beta_2 > 0$, there exist three critical points for a range of values centered at $\beta_1 = 0$.

To generate the data, 441 parameter settings were selected on a regular grid with β_1 and β_2 each varying from -1 to 1 . An additional 40 parameter settings, included solely for visual clarity, were selected on the two saddle–node branches, for a total $N_p = 481$. A total of $N_v = 861$ initial conditions were selected on a regular grid with x_1 varying from -2 to 2 and x_2 varying from -1 to 1 . For each combination of parameter settings and initial conditions, the trajectory was integrated over 1 time unit with $N_t = 101$. Figure 2 shows two sample phase portraits. The first is in the regime with three critical points, the outer two of which are stable. The second is in the regime with a single stable critical point.

It is important to know that we do not know what the parameter settings are for each trial – we only have labels for trials (1, 2, 15, ...) and these labels are randomly distributed across the true parameter space. Similarly, we do not know what the values of the initial conditions are – we have only labels for each pair of time series, and these labels are randomly distributed over the true state space. The main difficulty of our task is *not the interpolation* – that is easy. The main difficulty is the correct ordering of the indices, by similarity, reconstructing a coherent parameter space and state space realization, so that nearby parameter indices exhibit similar phase portraits, and nearby initial condition indices give rise to similar observation trajectories.

With these data, two iterations of the informed metric construction were deemed sufficient for our purposes. Figure 3 compares the original bifurcation map to the new map uncovered by our informed metric. With no prior knowledge of the dimensionality of the parameter space, our algorithm

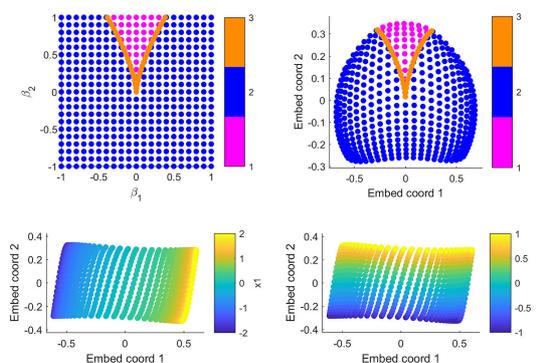


Figure 3. (top left) The original bifurcation map for the normal form of the cusp bifurcation. (top right) The parameter-realization embedding from our informed metric, colored by the true parameter regimes. (bottom) The variables-realization embedding colored by the initial condition in (left) x_1 and (right) x_2 .

can uncover a new embedding for the parameters axes that is obviously (visually) homeomorphic to the “true” parameters. Figure 3 also shows how, again with no knowledge of the true dimensionality, our informed embedding of the variables axes is one-to-one with the true state variables.

For a second illustrative example, one more in the spirit of this Festschrift, we will look at another cusp bifurcation, this time occurring in the well-studied model for a single, first order, exothermic reaction in a CSTR with a cooling jacket. Following Uppal et al.,^[21,22] under the approximations of high activation energy and cooling temperature equal to feed temperature, the governing equations can be non-dimensionalized to

$$\begin{aligned}\dot{x}_1 &= -x_1 + Da(1 - x_1)e^{x_2} \\ \dot{x}_2 &= -x_2 + BDa(1 - x_1)e^{x_2} - \beta x_2.\end{aligned}\quad (7)$$

Here, Da , B , and β are dimensionless parameters, x_1 is a dimensionless fractional conversion varying from 0 to 1, and x_2 is a dimensionless deviation from the feed temperature. To generate our data, we fix $B = 6$ and vary β and Da in an elliptical regime around the cusp point at $\beta = 0.5$ and $Da = 0.137$. As in the previous example, we also included a set of extra parameter setting samples on the saddle-node curve branches, for a total of $N_p = 445$. The initial conditions were sampled in an elliptical domain around the cusp point steady state at $x_1 = 0.5$ and $x_2 = 2.0$, with $N_{ic} = 519$. To show the utility of the approach, we ran our algorithm on both the original state space observations, as well as on observations through a nonlinear transformation function defined by

$$y_i = h_i(x) = 1/[1 + \exp(-\alpha_i^T x)] \quad (8)$$

where α_i is a random “observation vector”. In this example, $\alpha_1 = [0.6294, -0.7460]^T$ and $\alpha_2 = [0.8116, 0.8268]^T$. Figure 4 shows the recovered bifurcation maps, as well as the recovered

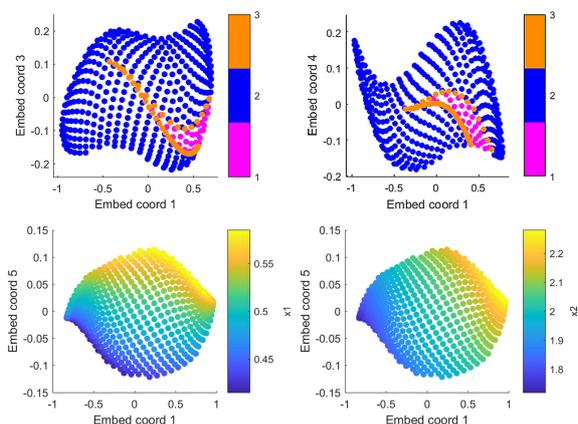


Figure 4. (top) The CSTR parameter embedding from our informed metric, colored by the true parameter regimes, based on observing (left) the true state variables and (right) the state variables passed through a nonlinear observation function. (bottom) The CSTR state variables embedding colored by the initial condition in (left) x_1 and (right) x_2 .

embedding for state space. While the embeddings are slightly more distorted than in the simpler case of the cusp normal form, they are still visually homeomorphic, and the key information is retained. The value of this last exercise is in demonstrating that one can ultimately be predictive even if the measurements do not carry an immediate physical meaning (they are neither “concentrations” nor “temperatures”, but functions of the two); the key is that there is “sufficient information” that the observations are one-to-one with the true system variables. This is not, of course, known *a priori*, and must be tested for at the end of the process. Here this test simply involved making sure that, on the data, the transformation from physical variables to observations is one to one (that the determinant of the Jacobian of Eq. 8 does not change sign, and that its magnitude remains bounded and bounded away from zero; in other words, that the transformation is bi-Lipschitz).^[33]

3.2 Reconstructed Equations

To extend this work to the construction of data-driven evolution equations, and thus to data-driven prediction, we look at data from the cusp normal form for the case of a single parameter setting ($\beta_1=0.0$ and $\beta_2=0.8$) in the three critical point regime. We use the same grid of initial conditions and time sampling as above; however, for each trajectory we also include the corresponding shifted trajectory from $t=0.2$ to $t=1.2$ time units. Having doubled the number of initial conditions, our data matrix is now 1722×101 . After running our iterative metric construction for two iterations, we have now embedded the original trajectories AND the shifted trajectories simultaneously. For each original trajectory, we obtain a time derivative estimate at the corresponding point in

embedding space based on the embedding of the corresponding shifted trajectory (see Figure 5). In other words, the set of shifts in embedding space from each original trajectory to its corresponding shifted trajectory gives us a set of point estimates of the derivative vector field. We use these point estimates to train a kriging^[31] model which gives maximum-likelihood estimates of the temporal evolution at any point in the embedding space. In Figure 5 (middle/bottom), we compare the results of integrating our dynamical model – fitted through kriging – to “the truth”: to using the Nyström extension to embed shifted trajectories, that were calculated using the original equations, in the new space. The results are visually close, yet one can clearly observe some gradual

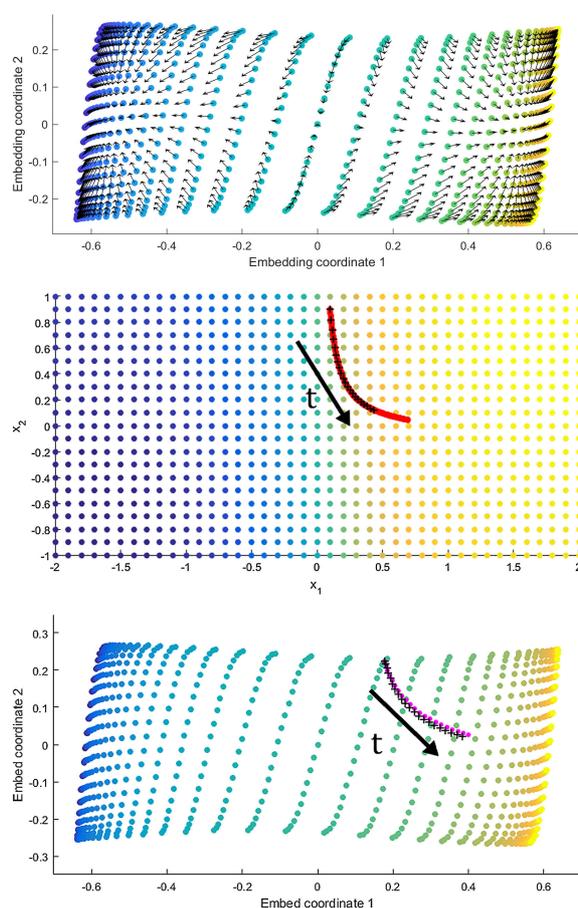


Figure 5. (top) The recovered embedding of state space when we include both original and shifted trajectories. The embeddings of the 861 trajectories with the original initial conditions are shown as circles colored by the initial condition in x_1 . For each original trajectory, a small black arrow is drawn from the embedding of the original trajectory to the embedding of the corresponding shifted trajectory. (middle) The original grid of initial conditions in state space, with a trajectory of length 3 time units (red) as well as 21 subsets of that trajectory, each of length 1 time unit, mapped to their initial conditions (black crosses). (bottom) The same trajectories embedded in the recovered space with the Nyström extension (black crosses) and a trajectory calculated using our kriging model (magenta circles).

deviation. This is likely because the derivative estimate “fed” to kriging is just a first order estimate based on a time step of 0.2. Figure 6 shows how the error in the final location can be improved somewhat by reducing the time shift of the included trajectories. For a small enough time shift, however, there will be no more improvement, and in fact the error will begin to increase. This is clearly the result of the interplay of the accuracy of the numerics of (a) the numerical integrator that produced the data, (b) the derivative estimation, (c) the kriging interpolation, and (d) most importantly, the scale of the diffusion map kernel. We will explore the numerical side of this important problem in future work, yet the value of the “proof of concept” presented above – i.e. that we can go from agnostic measurements to predictions, using only similarity between data obtained through trials – remains: the process circumvents physical understanding of the mechanisms involved, at the price of (extensive) computation and storage. It appears to produce certainly minimal (and quite possibly robust) realizations of the unknown system – in this case, the nonisothermal CSTR.

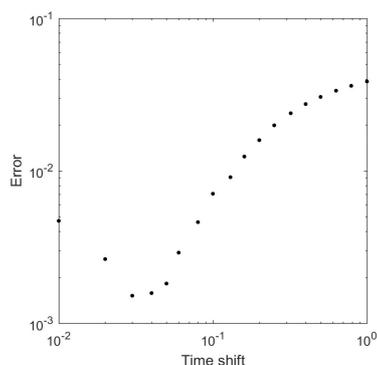


Figure 6. Log-log plot of the error in the reconstructed trajectory against the time shift for the included shifted trajectories. Error is quantified as the distance in embedding space between the Nyström embedding of the true trajectory and the final result of integrating the kriging model.

4. Conclusion

In this paper, we demonstrated a method for recovering numerical approximations for the evolution equations governing nonlinear chemical (and not only chemical!) dynamical systems from agnostic, initially unorganized observation data. We illustrated our approach on the normal form of a cusp bifurcation, as well as a cusp bifurcation found in a standard CSTR model. We organize the data along the dimensions of parameter settings (inputs), initial conditions (state variables), and time, and we iteratively constructed an informed metric for each type of variation. The recovered representations of both the parameter space and the underlying state space are homeomorphic to the true system, and provide – as a byproduct – the effective parameter and state dimensionality

of the data. Our main contribution was the demonstration of a method for modeling the dynamic behavior in the recovered state space – in effect, using the system realization to make predictions.

It is clear that while this type of technique is still at its infancy, it holds the promise of automating feature and equation extraction from raw observations. Even though the mathematics and numerical analysis of these approaches have not yet been completely worked out, this appears to be a reasonably straightforward, if highly nontrivial, task – a subcase of the so-called “manifold completion” problem,^[7,30,34] an extension of the “matrix completion” problem. The obvious issues of how much data one has to have to guarantee prediction confidence intervals, how to collect additional data when a given guarantee level is not met, and how to test with confidence the effective dimensionality of the relevant parameter and state spaces, are all the subject of current research in several communities, from mathematics and computer science to the domain science practitioners (in our case, to chemical engineers like the scientist we honor here).

To the mind of the authors, a prevailing question is less a quantitative and more a philosophical one: how much effort should be put in mapping the data-driven realizations to humanly interpretable, even chemically meaningful or mechanistic realizations? If this is possible, it would make the predictions obtained in a data-driven way much more physically credible. A practical partial solution is to establish homeomorphisms (and even diffeomorphisms) *on the data* between data-driven realization variables and humanly interpretable ones.^[33,35] If we know the one-to-one correspondence between, say, “embed var 1, embed var 2” and “concentration, temperature” on the data, that is certainly a big reconciliation step between chemical understanding and data mining. Yet the data driven approach has the chance of “working” even when the right-hand-side of the equation does not fit any obvious simple chemical mechanism – the only requirement is that the data-driven variables parametrize the manifold of the observed behaviors. In other words, that the right-hand-side of the equations is a graph of a function above these variables. We can safely say that this subject of “explainable AI”, or XAI as it is referred to,^[36] the subject of rationalization of the results of machine learning – in this case, the subject of discovering the chemical mechanism underpinning data-driven prediction schemes – will be an important topic of discussion, and of scientific research, during the next several years.

The authors, and especially Yannis Kevrekidis, wish to acknowledge a debt of gratitude to Prof. Sheintuch for his work over the years in chemical reaction dynamics and nonlinear dynamics more generally. His choice of problems (from the first Sheintuch and Schmitz papers^[37]), his style and approach, and the pervasive quality of his work, have been a reference point and an inspiration over many years and encounters. As Skip Scriven would say, “Excelsior!”.

Acknowledgements

The work of IGK and DS was partially supported by the US NIH (NIBIB), NSF (CDS&E) and by DARPA (MODyL).

References

- [1] J. Bongard, H. Lipson, *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 9943–9948.
- [2] M. Schmidt, H. Lipson, *Science* **2009**, *324*, 81–85.
- [3] J. P. Crutchfield, B. S. McNamara, *Complex Syst.* **1987**, *1*, 417–452.
- [4] G. Doretto, A. Chiuso, Y. N. Wu, S. Soatto, *Int. J. Computer. Vis.* **2003**, *51*, 91–109.
- [5] S. L. Brunton, J. L. Proctor, J. N. Kutz, *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 3932–3937.
- [6] I. G. Kevrekidis, C. W. Gear, J. H. Hyman, P. G. Kevrekidis, O. Runborg, C. Theodoropoulos, *Commun. Math. Sci.* **2003**, *1*, 715–762.
- [7] O. Yair, R. Talmon, R. R. Coifman, I. G. Kevrekidis, *Proc. Natl. Acad. Sci. USA* **2017**, *114*, E7865–E7874.
- [8] G. Sugihara, R. May, H. Ye, C. Hsieh, E. Deyle, M. Fogarty, S. Munch, *Science* **2012**, *338*, 496–500.
- [9] J. B. Tenenbaum, V. de Silva, J. C. Langford, *Science* **2000**, *290*, 2319–2323.
- [10] S. T. Roweis, L. K. Saul, *Science* **2000**, *290*, 2323–2326.
- [11] D. L. Donoho, C. Grimes, *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 5591–5596.
- [12] J. I. Ankenman, PhD thesis, Yale University (USA), **2014**.
- [13] M. Belkin, P. Niyogi, *Neural. Comput.* **2003**, *15*, 1373–1396.
- [14] R. R. Coifman, S. Lafon, *Appl. Comput. Harmon. Anal.* **2006**, *21*, 5–30.
- [15] M. Gavish, R. R. Coifman, *Appl. Comput. Harmon. Anal.* **2012**, *33*, 354–369.
- [16] R. E. Kalman, *J. Soc. Ind. Appl. Math. Ser. A* **1963**, *1*, 152–192.
- [17] B. Moore, *IEEE Trans. Autom. Control* **1981**, *26*, 17–32.
- [18] S. Lall, J. E. Marsden, S. Glavaski, *Proceedings of the IFAC World Congress*, International Federation of Automatic Control, New York, pp. 473–478.
- [19] J. Guckenheimer, P. J. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, vol. 42, Springer, New York, **2013**.
- [20] J. Guckenheimer, Y. A. Kuznetsov, *Scholarpedia* **2007**, *2*, 1852.
- [21] A. Uppal, W. H. Ray, A. B. Poore, *Chem. Eng. Sci.* **1974**, *28*, 967–985.
- [22] A. Uppal, W. H. Ray, A. B. Poore, *Chem. Eng. Sci.* **1976**, *31*, 205–214.
- [23] F. Takens in *Dynamical Systems and Turbulence, Warwick 1980* (Eds.: D. Rand, B. S. Young), *Lecture Notes in Math.*, 898, Springer-Verlag, Berlin, **1981**, pp. 366–381.
- [24] H. Whitney, *Ann. of Math.* **1936**, *37*, 645–690.
- [25] H. Whitney, *Ann. of Math.* **1944**, *45*, 220–246.
- [26] T. Sauer, J. A. Yorke, M. Casdagli, *J. Stat. Phys.* **1991**, *65*, 579–616.
- [27] S. S. Lafon, PhD thesis, Yale University (USA), **2013**.
- [28] G. Mishne, R. Talmon, R. Meir, J. Schiller, M. Lavzin, U. Dubin, R. Coifman, *IEEE J. Selected Topics in Sig. Proc.* **2016**, *10*, 1238–1253.
- [29] G. Mishne, R. Talmon, I. Cohen, R. Coifman, Y. Kluger, *IEEE Transactions on Sig. and Inf. Proc. over Networks* **2017**, preprint.
- [30] S. Lafon, Y. Keller, R. R. Coifman, *IEEE Transactions on Pattern Anal. and Mach. Learn.* **2006**, *28*, 1784–1797.
- [31] N. Cressie, *Mathematical Geosciences* **1990**, *22*, 239–252.
- [32] S. N. Lophaven, H. B. Nielsen, J. Søndergaard, report no. IMM-TR-2002-12, Technical University of Denmark, Kgs. Lyngby, Denmark, **2002**.
- [33] B. E. Sunday, M. Haataja, I. G. Kevrekidis, *Phys. Rev. E* **2009**, *80*, 031102.
- [34] C. J. Dsilva, *Development* **2015**, *142*, 1717–1724.
- [35] T. A. Frewen, I. D. Couzin, A. Kolpas, J. Moehlis, R. R. Coifman, I. G. Kevrekidis in *Coping with Complexity: Model Reduction and Data Analysis* (Eds. A. N. Gorban, D. Roose), *Lecture Notes in Comp. Sci. and Eng.*, 75, Springer-Verlag, Berlin, **2011**, pp. 302–310.
- [36] C. Kuang, “Can A. I. be Taught to Explain Itself,” *The New York Times Sunday Magazine*, **November 26, 2017**, MM46.
- [37] M. Sheintuch, R. A. Schmitz, *Chemical Reaction Engineering – Houston ACS Symp. Series*, **1978**, *65*, 487–497.

Received: December 31, 2017

Accepted: March 1, 2018

Published online on April 6, 2018