Discovering Equations from Data

Melissa R. McGuirl

February 28, 2019

Outline

Problem Set-Up

Inputs

Observation data sampled at discrete times t_i: $\{x(t_i)\}_{i \in I}$, where $x(t_i) \in \mathbb{R}^n$

Goal

Learn $f(x(t))$ such that $\frac{dx}{dt} = f(x(t))$

Approach 1: Sparse Identification of Non-linear Dynamical Systems (SINDy)

SINDy References/Sources:

- K. Champion, S. Brunton, J. N. Kutz, *Discovery of Non-linear Multiscale Systems: Sampling Strategies and Embeddings*, SIAM Journal of Applied Dynamical Systems (2019)
- S. Brunton, J. Proctor, J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, PNAS (2016)
- `https://www.youtube.com/watch?v=gSCa78TIldg`

Approach 1: Sparse Identification of Non-linear Dynamical Systems (SINDy)

Given measurement data $\{x(t_i)\}_{i \in I}$, can we accurately learn $f(x(t))$ so that $\frac{dx}{dt} = f(x(t))$?

SINDy Assumptions

1. We have the full state measurements
2. f only has a few active terms, i.e. f is sparse is the space of all possible functions of $x(t)$

SINDy Algorithm

Case 1: Also assume data sampled from uniscale dynamical system

SINDy: Step 1

Collect measurement data and form a state space matrix:

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

SINDy Algorithm

SINDy: Step 2

Numerically approximate $\dot{\mathbb{X}}$ to get

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

Common methods for approximating the derivative:

- Total variation regularized derivative (see R. Chartrand, *Numerical Differentiation of Noisy, Nonsmooth Data*, ISRN Applied Mathematics, 2011)
- Finite difference methods

SINDy Algorithm

SINDy: Step 3

Construct library $\Theta(\mathbb{X})$ of candidate nonlinear functions of $\mathbb{X}$:

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

where

$$\mathbb{X}^{P_2} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^2(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & \cdots & x_n^2(t_m) \end{bmatrix}$$

and similarly for general $\mathbb{X}^{P_q}$.

SINDy Algorithm

SINDy: Step 4

Sparse regression on $\dot{\mathbb{X}} = \Theta(\mathbb{X})\Sigma$ to solve for $\Sigma = [\sigma_1, \ldots, \sigma_n]$ coefficients, $\sigma_i \in \mathbb{R}^p$.

Let $\lambda > 0$ be the sparsity threshold

1. Initial guess: solve $\dot{\mathbb{X}} = \Theta(\mathbb{X})\Sigma$ via ordinary least squares
2. If $\Sigma(i,j) < \lambda$ set $\Sigma(i,j) = 0$
3. for k = 1, 2, . . . , n
   - solve $\dot{\mathbb{X}}(:,k) = \Theta(\mathbb{X})(:,\Sigma(:,k) > \lambda)\Sigma(\Sigma(:,k) > \lambda), k)$ via least squares
4. Repeat steps 2-3 until coefficients do not change (or for a fixed number of iterations)

SINDy Summary

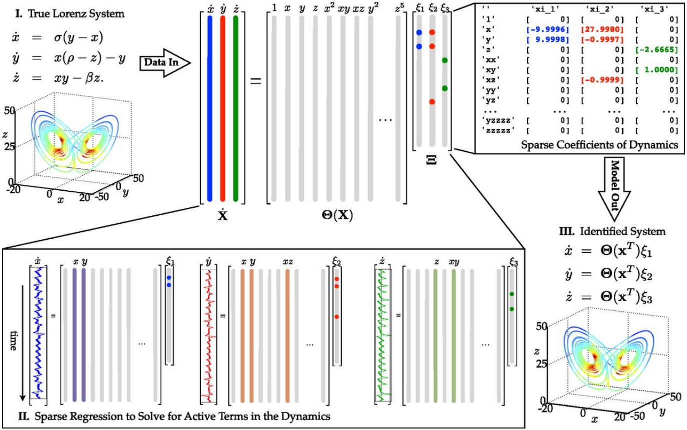

Figure 1: SINDy algorithm. Figure from Figure 1 of *Discovering governing equations from data by sparse identification of nonlinear dynamical systems* (S. Brunton, J. Proctor, J.N. Kutz, PNAS, 2016).

SINDy Parameter Choices

- Differentiation method and corresponding parameters (e.g., for the total variation regularized derivative you need to specify the regularization term)
- Polynomial order for $\Theta(\mathbb{X})$ and whether or not to include sine/cosine bases functions
- Sparsity threshold λ

SINDy Case 2 Algorithm

Case 2: Assume data sampled from multi-scale dynamical system

Refined Problem Set-Up: nonlinear systems with linear coupling

Given measurement data $\{u(t_i)\}_{i \in I}$ and $\{v(t_i)\}_{i \in I}$, with $u(t_i) \in \mathbb{R}^n$ (fast variables) and $v(t_i) \in \mathbb{R}^l$ (slow variables), we try to learn functions $f(u(t))$ and $g(v(t))$ and $C \in \mathbb{R}^{n \times l}$, $D \in \mathbb{R}^{l \times n}$ such that

$$\tau_{\text{fast}} \dot{u} = f(u) + Cv$$
$$\tau_{\text{slow}} \dot{v} = g(v) + Du$$

SINDy Case 2 Algorithm

SINDy Multiscale Approach: Burst Sampling

- Sampling scheme: Collect sample measurements in short bursts with a small step size, spread out of over a long duration
- Repeat SINDy with burst sampling scheme

Burst Sampling

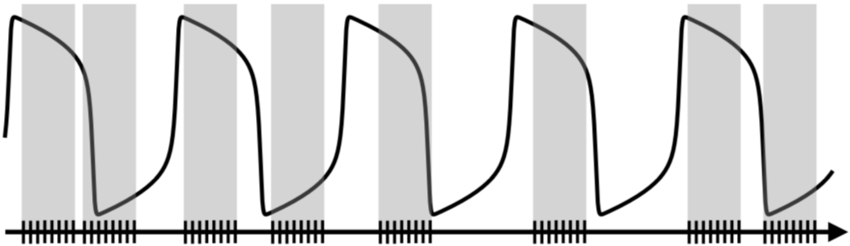

Figure 2: Burst Sampling. Figure from Figure 3(a) in *Discovery of Non-linear Multiscale Systems: Sampling Strategies and Embeddings* (K. Champion, S. Brunton, J.N. Kutz, SIAM Journal of Applied Dynamical Systems, 2019).

SINDy Case 2 (multiscale dynamics) Additional Parameter Choices

- Burst size (number of samples per burst)
- Duration over which to sample (about 2 times the period of the slow dynamics, at least)
- Total number of bursts to collect
- Placement of bursts (select burst times as Poisson arrival times or select them uniformly and then add noise to the left/right)

Aside: What about PDEs?

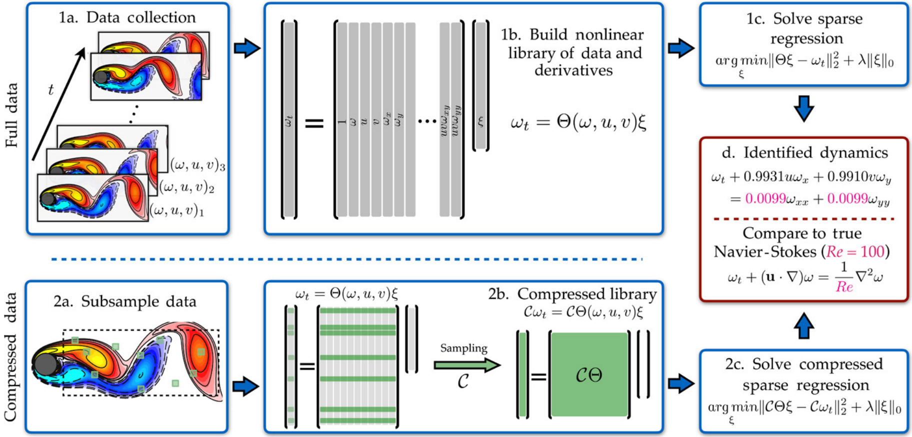

Figure 3: PDE-FIND algorithm. Figure from Figure 1 in *Data-driven Discovery of partial differential equations* (S. Rudy, S. Brunton, J. Proctor, J. N. Kutz, Science Advances, 2017). Also see `https://www.youtube.com/watch?v=oI3grKBxkqM&feature=youtu.be+target%3D`

Approach 2: Hankel alternative view of Koopman (HAVOK) analysis

HAVOK References/Sources:

- S. Brunton, B. Brunton, J. Proctor, E. Kaiser, J. N. Kutz, *Chaos as an intermittently forced linear system*, Nature Communications (2017)
- K. Champion, S. Brunton, J. N. Kutz, *Discovery of Non-linear Multiscale Systems: Sampling Strategies and Embeddings*, SIAM Journal of Applied Dynamical Systems (2019)
- `https://www.youtube.com/watch?v=Q8VzAtGGlDQ`

Approach 2: Hankel alternative view of Koopman (HAVOK) analysis

HAVOK Assumptions

1. We only have one state measurement variable $\{x(t_i)\}_{i \in I}$, $x(t_i) \in \mathbb{R}$
2. Data comes from an r-dimensional chaotic dynamical system

HAVOK Algorithm

HAVOK: Step 1

Collect measurement data and form a state space vector:

$$\mathbb{X} = \begin{bmatrix} x(t_1) \\ x(t_2) \\ \vdots \\ x(t_m) \end{bmatrix}$$

HAVOK Algorithm

HAVOK: Step 2

Form Hankel matrix:

$$\mathbb{H} = \begin{bmatrix} x(t_1) & x(t_2) & \dots & x(t_p) \\ x(t_2) & x(t_3) & \dots & x(t_{p+1}) \\ \vdots & \vdots & \ddots & \vdots \\ x(t_q) & x(t_{q+1}) & \dots & x(t_m) \end{bmatrix}$$

HAVOK Algorithm

HAVOK: Step 3

Apply singular value decomposition to Hankel matrix:

$$\mathbb{H} = U \Sigma V^*$$

HAVOK: Step 4

Find optimal rank r (M. Gavish and D. L. Donoho *The optimal hard threshold for singular values is* $4/\sqrt{3}$, IEEE Transactions on Information Theory (2014))

HAVOK Algorithm

HAVOK: Step 5

Apply SINDy algorithm (with degree 1) using first r columns of V as measurement data:

$$V_r = [V_1, V_2, \ldots, V_r]^T$$

HAVOK Algorithm

HAVOK: Step 6

Separate out rth variable. Build state-space model from first $(r-1)$ terms and use rth term as a forcing term:

$$\frac{d\mathbb{V}}{dt} = A\mathbb{V}(t) + BV_r(t),$$

where $\mathbb{V} = [V_1, V_2, \ldots, V_{r-1}]^T$, and A and B are coefficient matrices found from SINDy in step 5

HAVOK Summary

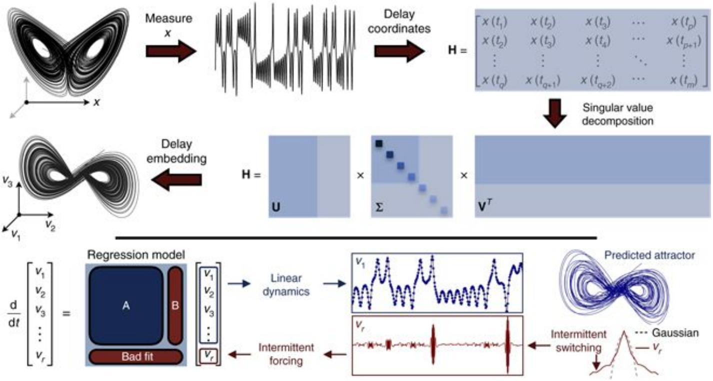

Figure 4: HAVOK analysis. Figure from Figure 1 in *Chaos as an intermittently forced linear system* (S. Brunton, B. Brunton, J. Proctor, E. Kaiser, J. N. Kutz, Nature Communications, 2017).

HAVOK Parameter Choices

- q (HAVOK matrix dimension)
- Differentiation method and corresponding parameters
- Sparsity threshold λ

Other HAVOK approaches

What if we have quasiperiodic data instead of chaotic data?

- Methods for tackling this case with both uniscale and multiscale dynamics are presented in *Discovery of Non-linear Multiscale Systems: Sampling Strategies and Embeddings* (K. Champion, S. Brunton, J. N. Kutz, SIAM Journal of Applied Dynamical Systems, 2019)
- Methods rely on dynamic mode decomposition of Hankel matrix and shifted Hankel matrix
- Return to these methods after we cover dynamic mode decomposition?

Open source code

- MATLAB SINDy and PDE-FIND code available here:
  http://faculty.washington.edu/kutz/page26/
- MATLAB HAVOK code available here:
  http://faculty.washington.edu/sbrunton/HAVOK.zip
- Modified comparison code on Dropbox

Group Activities

1. Try MATLAB code with favorite dynamical systems
2. Play around with parameter and analyze how performance changes
3. Add noise to system data and see how methods perform
4. Download relevant data from `https://www.kaggle.com/datasets` (or try your own data sets) and see if we can learn interesting/accurate governing equations for the data

References

- S. Brunton, B. Brunton, J. Proctor, E. Kaiser, J. N. Kutz, *Chaos as an intermittently forced linear system*, Nature Communications (2017)
- S. Brunton, J. Proctor, J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, PNAS, (2016)
- K. Champion, S. Brunton, J. N. Kutz, *Discovery of Non-linear Multiscale Systems: Sampling Strategies and Embeddings,* SIAM Journal of Applied Dynamical Systems (2019).
- R. Chartrand, *Numerical Differentiation of Noisy, Nonsmooth Data*, ISRN Applied Mathematics (2011).
- M. Gavish and D. L. Donoho *The optimal hard threshold for singular values is* $4/\sqrt{3}$, IEEE Transactions on Information Theory (2014).
- S. Rudy, S. Brunton, J. Proctor, J. N. Kutz, *Data-driven Discovery of partial differential equations*, Science Advance (2017).