Discovering governing equations using noisy measurements through projection-based denoising and second order cone programming

Jacqueline Wentz, Alireza Doostan

University of Colorado at Boulder

Machine Learning/Deep Learning Conference, July 2022

Funding: DOE Predictive Science Academic Alliance Program (PSAAP)

Discover governing equations by assuming basis expansion

- Goal is to find governing equations of an ODE system

Discover governing equations by assuming basis expansion

- Goal is to find governing equations of an ODE system
- In system with two states u_1 and u_2, for example, we first assume:

$$\begin{aligned}\dot{u}_1 &= c_{1,1} + c_{1,2}u_1 + c_{1,3}u_2 + c_{1,4}u_1^2 + c_{1,5}u_1u_2 + \dots \\ \dot{u}_2 &= c_{2,1} + c_{2,2}u_1 + c_{2,3}u_2 + c_{2,4}u_1^2 + c_{2,5}u_1u_2 + \dots\end{aligned} \tag{1}$$

Discover governing equations by assuming basis expansion

- Goal is to find governing equations of an ODE system
- In system with two states u_1 and u_2, for example, we first assume:

$$
\begin{aligned}
\dot{u}_1 &= c_{1,1} + c_{1,2}u_1 + c_{1,3}u_2 + c_{1,4}u_1^2 + c_{1,5}u_1u_2 + \dots \\
\dot{u}_2 &= c_{2,1} + c_{2,2}u_1 + c_{2,3}u_2 + c_{2,4}u_1^2 + c_{2,5}u_1u_2 + \dots
\end{aligned} \tag{1}
$$

- Suppose we have N state measurements $\rightarrow u_1(t_i)$ and $u_2(t_i)$ for $i = 1, 2, \dots, N$

Discover governing equations by assuming basis expansion

- Goal is to find governing equations of an ODE system
- In system with two states u_1 and u_2, for example, we first assume:

$$\begin{aligned} \dot{u}_1 &= c_{1,1} + c_{1,2}u_1 + c_{1,3}u_2 + c_{1,4}u_1^2 + c_{1,5}u_1u_2 + \ldots \\ \dot{u}_2 &= c_{2,1} + c_{2,2}u_1 + c_{2,3}u_2 + c_{2,4}u_1^2 + c_{2,5}u_1u_2 + \ldots \end{aligned} \tag{1}$$

- Suppose we have N state measurements $\rightarrow$ $u_1(t_i)$ and $u_2(t_i)$ for $i = 1, 2, \ldots, N$
- Find $\boldsymbol{c}_i$ for $i = 1, 2$ such that

$$\Theta \boldsymbol{c}_i = \dot{\boldsymbol{u}}_i, \quad \text{for } i = 1, 2 \tag{2}$$

where

$$\Theta = \begin{bmatrix} 1 & u_1(t_1) & u_2(t_1) & u_1^2(t_1) & \ldots \\ 1 & u_1(t_2) & u_2(t_2) & u_1^2(t_2) & \ldots \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & u_1(t_N) & u_2(t_N) & u_1^2(t_N) & \ldots \end{bmatrix}, \qquad \dot{\boldsymbol{u}}_i = \begin{bmatrix} \dot{u}_i(t_1) \\ \dot{u}_i(t_2) \\ \vdots \\ \dot{u}_i(t_N) \end{bmatrix} \tag{3}$$

Existing approaches for coefficient recovery fail at large noise levels

- Sparse identification of nonlinear dynamics (SINDy): Brunton et al. 2016
  1. Estimate the derivative, $\dot{\boldsymbol{u}}$, using measurements
  2. Assume coefficient vector is sparse
  3. Solve the following, using sequential thresholding least squares:

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}'} \|\Theta\boldsymbol{c}' - \dot{\boldsymbol{u}}\|_2^2 + \lambda\|\boldsymbol{c}'\|_1 \tag{4}$$

Existing approaches for coefficient recovery fail at large noise levels

- Sparse identification of nonlinear dynamics (SINDy): Brunton et al. 2016
  1. Estimate the derivative, $\dot{\boldsymbol{u}}$, using measurements
  2. Assume coefficient vector is sparse
  3. Solve the following, using sequential thresholding least squares:

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}'} \|\Theta \boldsymbol{c}' - \dot{\boldsymbol{u}}\|_2^2 + \lambda \|\boldsymbol{c}'\|_1 \tag{4}$$

- Challenges:
  1. **Derivative estimation in the presence of noise**
  2. Determining optimal λ

Existing approaches for coefficient recovery fail at large noise levels

- Sparse identification of nonlinear dynamics (SINDy): Brunton et al. 2016
  1. Estimate the derivative, $\dot{\boldsymbol{u}}$, using measurements
  2. Assume coefficient vector is sparse
  3. Solve the following, using sequential thresholding least squares:

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}'} \|\Theta \boldsymbol{c}' - \dot{\boldsymbol{u}}\|_2^2 + \lambda \|\boldsymbol{c}'\|_1 \tag{4}$$

- Challenges:
  1. **Derivative estimation in the presence of noise**
  2. Determining optimal λ
- Research Question: Can we improve coefficient recovery in the presence of noise?
  1. Novel denoising strategy
  2. Approach for finding the coefficients/derivative simultaneously.

Example dynamic system: Van der Pol oscillator

- Non-conservative oscillator with non-linear damping:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0. \qquad (5)$$

Example dynamic system: Van der Pol oscillator

- Non-conservative oscillator with non-linear damping:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0. \qquad (5)$$

- Equivalent ODE: $x \to u_1$ and $\dot{x} \to u_2$

$$\begin{aligned} \dot{u}_1 &= u_2 \\ \dot{u}_2 &= -u_1 + \mu u_2 - \mu u_1^2 u_2 \end{aligned} \qquad (6)$$

Example dynamic system: Van der Pol oscillator

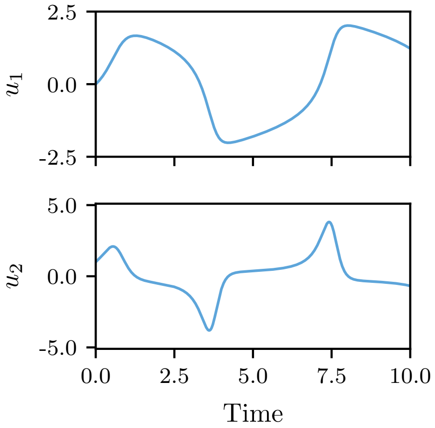
- Non-conservative oscillator with non-linear damping:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0. \tag{5}$$

- Equivalent ODE: $x \to u_1$ and $\dot{x} \to u_2$

$$\begin{aligned} \dot{u}_1 &= u_2 \\ \dot{u}_2 &= -u_1 + \mu u_2 - \mu u_1^2 u_2 \end{aligned} \tag{6}$$

- Example dynamics: $\mu = 2$, $\boldsymbol{u}^{(0)} = [0, 1]$



Example dynamic system: Van der Pol oscillator

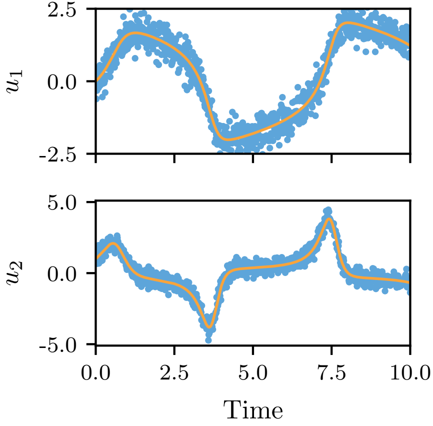
- Non-conservative oscillator with non-linear damping:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0. \qquad (5)$$

- Equivalent ODE: $x \to u_1$ and $\dot{x} \to u_2$

$$\begin{aligned} \dot{u}_1 &= u_2 \\ \dot{u}_2 &= -u_1 + \mu u_2 - \mu u_1^2 u_2 \end{aligned} \qquad (6)$$

- Example dynamics: $\mu = 2$, $\boldsymbol{u}^{(0)} = [0, 1]$
- Noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$, e.g., $\sigma^2 = 0.1$.



Projection-based state denoising: Method

- Use the assumed basis given by the columns of Θ.

Projection-based state denoising: Method

- Use the assumed basis given by the columns of Θ.
- We apply quadrature techniques to

$$\Theta \boldsymbol{c} = \dot{\boldsymbol{u}}, \tag{7}$$

to obtain,

$$\Phi \boldsymbol{d} = \boldsymbol{u} \quad \text{where} \quad \Phi = \begin{bmatrix} \mathbf{1} & T\Theta \end{bmatrix}. \tag{8}$$

Projection-based state denoising: Method

- Use the assumed basis given by the columns of Θ.
- We apply quadrature techniques to

$$\Theta \boldsymbol{c} = \dot{\boldsymbol{u}}, \tag{7}$$

to obtain,

$$\Phi \boldsymbol{d} = \boldsymbol{u} \quad \text{where} \quad \Phi = \begin{bmatrix} \mathbf{1} & T\Theta \end{bmatrix}. \tag{8}$$

- This implies we expect $\boldsymbol{u}$ to be in the column space of Φ.

Projection-based state denoising: Method

- Use the assumed basis given by the columns of Θ.
- We apply quadrature techniques to

$$\Theta \boldsymbol{c} = \dot{\boldsymbol{u}}, \tag{7}$$

  to obtain,

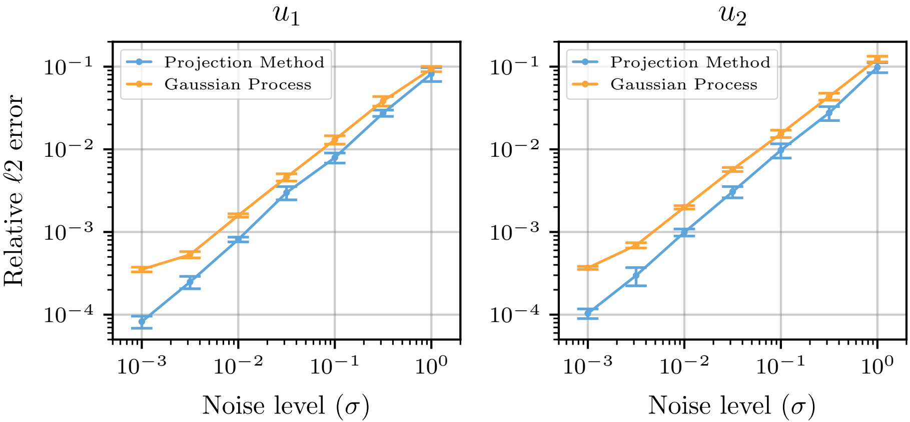
$$\Phi \boldsymbol{d} = \boldsymbol{u} \quad \text{where} \quad \Phi = \begin{bmatrix} \mathbf{1} & T\Theta \end{bmatrix}. \tag{8}$$

- This implies we expect $\boldsymbol{u}$ to be in the column space of Φ.
- Project data onto this expected subspace:

$$\tilde{\boldsymbol{u}} = P_{\Phi} \boldsymbol{u} \tag{9}$$

  where P_{Φ} is projection operator calculated using $\boldsymbol{u}$.

Projection-based denoising: Results for Van der Pol oscillator



Learn coefficients and derivative simultaneously

- Write coefficients in terms of derivative:

$$\Theta \boldsymbol{c} = \dot{\boldsymbol{u}} \quad \rightarrow \quad \boldsymbol{c} = (\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T \dot{\boldsymbol{u}}. \tag{10}$$

Learn coefficients and derivative simultaneously

- Write coefficients in terms of derivative:

$$\Theta \boldsymbol{c} = \dot{\boldsymbol{u}} \quad \rightarrow \quad \boldsymbol{c} = (\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T \dot{\boldsymbol{u}}. \tag{10}$$

- Set B as estimator of $(\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T$.

Learn coefficients and derivative simultaneously

- Write coefficients in terms of derivative:

$$\Theta \boldsymbol{c} = \dot{\boldsymbol{u}} \quad \rightarrow \quad \boldsymbol{c} = (\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T \dot{\boldsymbol{u}}. \tag{10}$$

- Set B as estimator of $(\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T$.
- Second order cone program (SOCP) to find initial condition, u_0, and derivative, $\dot{\boldsymbol{u}}$:

$$\begin{aligned} \text{minimize}_{u_0, \dot{\boldsymbol{u}}} \quad & \|B\dot{\boldsymbol{u}}\|_1 && \text{(sparsity of coefficients)} \\ \text{subject to} \quad & \|D\dot{\boldsymbol{u}}\|_2 \leq C && \text{(smooth derivative)} \\ & \left\| \begin{bmatrix} \mathbf{1} & T \end{bmatrix} \begin{bmatrix} u_0 \\ \dot{\boldsymbol{u}} \end{bmatrix} - \tilde{\boldsymbol{u}} \right\|_2 \leq \gamma && \text{(match a priori smoothed data).} \end{aligned} \tag{11}$$

Learn coefficients and derivative simultaneously

- Write coefficients in terms of derivative:

$$\Theta \boldsymbol{c} = \dot{\boldsymbol{u}} \quad \rightarrow \quad \boldsymbol{c} = (\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T \dot{\boldsymbol{u}}. \tag{10}$$

- Set B as estimator of $(\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T$.
- Second order cone program (SOCP) to find initial condition, u_0, and derivative, $\dot{\boldsymbol{u}}$:

$$\begin{aligned} \text{minimize}_{u_0, \dot{\boldsymbol{u}}} \quad & \|B\dot{\boldsymbol{u}}\|_1 && \text{(sparsity of coefficients)} \\ \text{subject to} \quad & \|D\dot{\boldsymbol{u}}\|_2 \le C && \text{(smooth derivative)} \\ & \left\| \begin{bmatrix} \mathbf{1} & T \end{bmatrix} \begin{bmatrix} u_0 \\ \dot{\boldsymbol{u}} \end{bmatrix} - \tilde{\boldsymbol{u}} \right\|_2 \le \gamma && \text{(match a priori smoothed data).} \end{aligned} \tag{11}$$

- Estimate C using projection-based denoising result.

Learn coefficients and derivative simultaneously

- Write coefficients in terms of derivative:

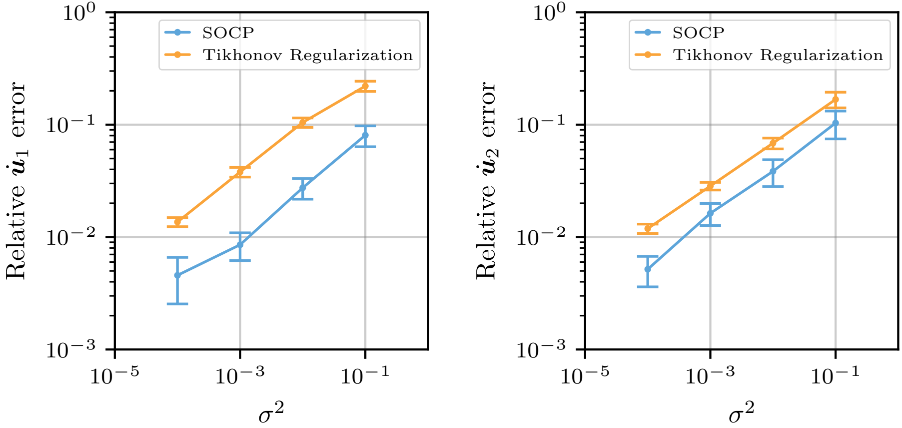
$$\Theta \boldsymbol{c} = \dot{\boldsymbol{u}} \quad \rightarrow \quad \boldsymbol{c} = (\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T \dot{\boldsymbol{u}}. \tag{10}$$

- Set B as estimator of $(\tilde{\Theta}^T \Theta)^{-1} \tilde{\Theta}^T$.
- Second order cone program (SOCP) to find initial condition, u_0, and derivative, $\dot{\boldsymbol{u}}$:

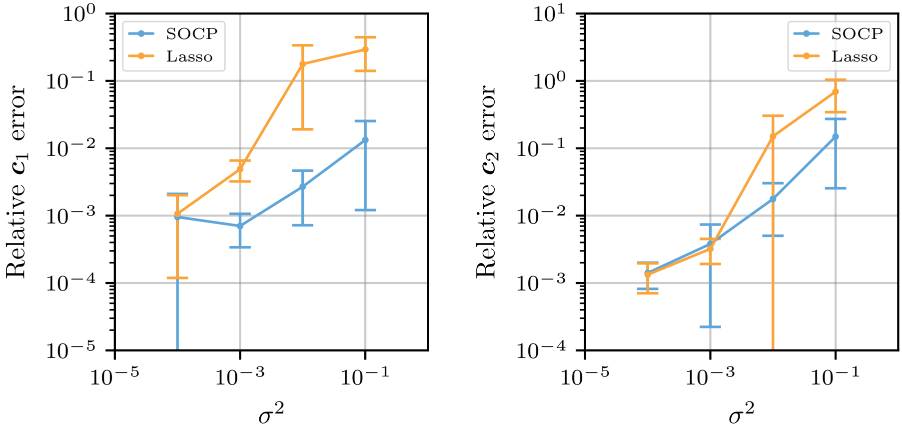
$$\begin{aligned} \text{minimize}_{u_0, \dot{\boldsymbol{u}}} \quad & \|B \dot{\boldsymbol{u}}\|_1 && \text{(sparsity of coefficients)} \\ \text{subject to} \quad & \|D \dot{\boldsymbol{u}}\|_2 \leq C && \text{(smooth derivative)} \\ & \left\| \begin{bmatrix} \mathbf{1} & T \end{bmatrix} \begin{bmatrix} u_0 \\ \dot{\boldsymbol{u}} \end{bmatrix} - \tilde{\boldsymbol{u}} \right\|_2 \leq \gamma && \text{(match a priori smoothed data).} \end{aligned} \tag{11}$$

- Estimate C using projection-based denoising result.
- Find γ using the corner point of Pareto curve.

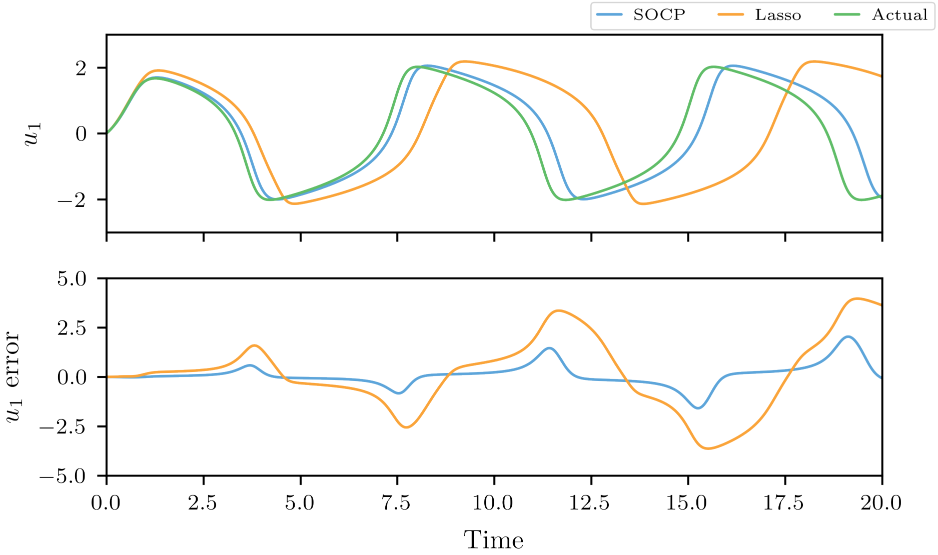
SOCP improves derivative estimation compared with Tikhonov regularization



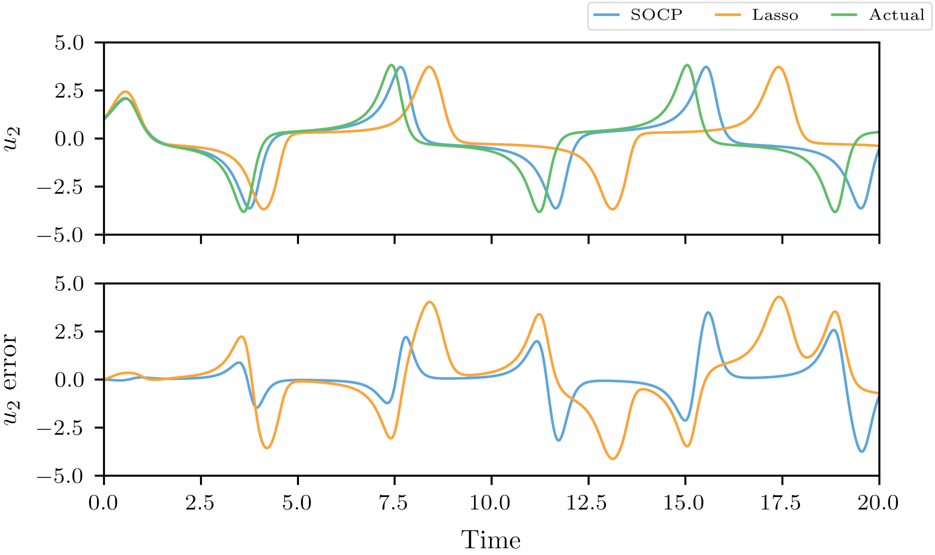
SOCP improves coefficient estimation compared with Lasso approach



Example prediction results: Van der Pol oscillator u_1



Example prediction results: Van der Pol oscillator u_2



Conclusions

- Existing equation discovery methods fail at high noise levels.
- Presented two improvements:
  1. Projection-based denoising strategy.
  2. SOCP to learn derivative/coefficients simultaneously.
- Led to improved derivative and coefficient estimation.
- Future Work:
  1. Compare approach to other versions of SINDy (i.e, Weak-Sindy).
  2. Consider these methods in the context of PDEs.

Conclusions

- Existing equation discovery methods fail at high noise levels.
- Presented two improvements:
  1. Projection-based denoising strategy.
  2. SOCP to learn derivative/coefficients simultaneously.
- Led to improved derivative and coefficient estimation.
- Future Work:
  1. Compare approach to other versions of SINDy (i.e, Weak-Sindy).
  2. Consider these methods in the context of PDEs.

Thank You!

References

Brunton, Steven L. et al. (2016). "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences of the United States of America* 113.15, pp. 3932–3937. ISSN: 10916490. DOI: 10.1073/PNAS.1517384113. arXiv: 1509.03580.