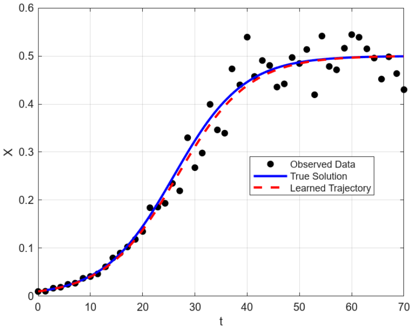
Weak-form Equation Learning with wSINDy: An Introduction and Application to Biological Systems

Rainey Lyons

Joint work with Vanja Dukic and David Bortz
University of Colorado Boulder

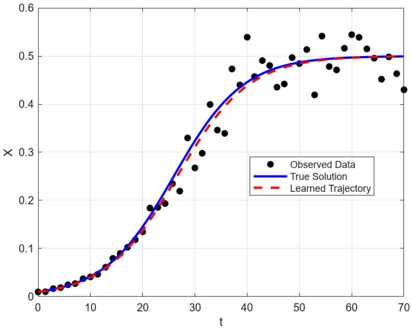
22 Janurary 2026



Set up: Say we have (possibly noisy) measurements of some (vector of) state variable(s) $\{u_k\}_{k=1}^K$ which we assume follow some unknown dynamical system $\dot{\mathbf{u}} = \mathbf{F}(\mathbf{u})$.

Goal: Recover **F(u)** from the data.

Equation Learning

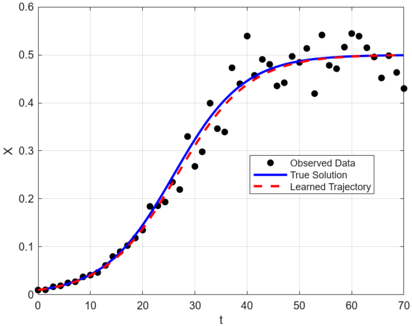


Set up: Say we have (possibly noisy) measurements of some (vector of) state variable(s) $\{u_k\}_{k=1}^{K}$ which we assume follow some unknown dynamical system $\dot{\mathbf{u}} = \mathbf{F}(\mathbf{u})$.

Goal: Recover $\mathbf{F(u)}$ from the data.

Hurtles: We want to avoid overfitting to the data and we want our dynamical system to be somehow *interpretable* to the problem at hand.

Equation Learning



Set up: Say we have (possibly noisy) measurements of some (vector of) state variable(s) $\{u_k\}_{k=1}^K$ which we assume follow some unknown dynamical system $\dot{\mathbf{u}} = \mathbf{F}(\mathbf{u})$.

Goal: Recover $\mathbf{F(u)}$ from the data.

Hurtles: We want to avoid overfitting to the data and we want our dynamical system to be somehow *interpretable* to the problem at hand.

Additionally, this involves learning both *terms* and *parameters* in the equations.

Why equation learning in the biological sciences?

- **Mechanistic understanding (not just prediction):** Biological dynamics arise from *interactions* (feedback, saturation, regulation). But we often lack reliable first-principles models, making it hard to propose correct governing equations from scratch.

Why equation learning in the biological sciences?

- **Mechanistic understanding (not just prediction):** Biological dynamics arise from *interactions* (feedback, saturation, regulation). But we often lack reliable first-principles models, making it hard to propose correct governing equations from scratch.
- **Interpretability & scientific hypotheses:** Discovered terms can map to known biology (e.g., growth, inhibition, enzyme kinetics), producing testable hypotheses and revealing missing mechanisms or unmodeled couplings.

Why equation learning in the biological sciences?

- **Mechanistic understanding (not just prediction):** Biological dynamics arise from *interactions* (feedback, saturation, regulation). But we often lack reliable first-principles models, making it hard to propose correct governing equations from scratch.
- **Interpretability & scientific hypotheses:** Discovered terms can map to known biology (e.g., growth, inhibition, enzyme kinetics), producing testable hypotheses and revealing missing mechanisms or unmodeled couplings.
- **Generalization across conditions:** Equations often transfer better than black-box models when experimental conditions change (dose, environment, genotype), enabling *extrapolation* and robust forecasting.

Why equation learning in the biological sciences?

- **Mechanistic understanding (not just prediction):** Biological dynamics arise from *interactions* (feedback, saturation, regulation). But we often lack reliable first-principles models, making it hard to propose correct governing equations from scratch.
- **Interpretability & scientific hypotheses:** Discovered terms can map to known biology (e.g., growth, inhibition, enzyme kinetics), producing testable hypotheses and revealing missing mechanisms or unmodeled couplings.
- **Generalization across conditions:** Equations often transfer better than black-box models when experimental conditions change (dose, environment, genotype), enabling *extrapolation* and robust forecasting.
- **Design & control:** With an explicit dynamical model we can run *in silico* perturbations, estimate key parameters, and optimize interventions (drug scheduling, synthetic circuits, bioprocess control).

SINDy (Brunton et. al. PNAS 2016)

Idea: Discover governing equations $\dot{\mathbf{u}} = \mathbf{F}(\mathbf{u})$ directly from measurement data,

$$\mathbf{U} = \left[\mathbf{u}(t_1) \ldots \mathbf{u}(t_m)\right]^T,$$

via sparse regression on the system

$$\dot{\mathbf{U}} = \Theta(\mathbf{U})\,\mathbf{w},$$

where $\Theta(\mathbf{U})$ represents a library of candidate functions for the model, i.e.,

$$\Theta(\mathbf{U}) = \left[f_1(\mathbf{u}) \quad f_2(\mathbf{u}) \quad f_3(\mathbf{u}) \quad \cdots \quad \right].$$

Key ideas: Dynamics are represented with only a few active terms. In other words,

$$\mathbf{F}(\mathbf{u}) = \sum_{i=1}^{I} \mathbf{w}_i^\star\, f_i(\mathbf{u}), \quad \text{for sparse } \mathbf{w}^\star.$$

This $\mathbf{w}^\star$ is approximated using equation error techniques, e.g., minimizing

$$\|\dot{\mathbf{U}} - \Theta(\mathbf{U})\mathbf{w}\|_2 + \lambda\|\mathbf{w}\|_0.$$

SINDy for a toy problem

Say we are observing the trajectory of the logistic equation $\dot{u} = au + bu^2$. Then, we might suspect the right-hand side of the equation to polynomial and construct a monomial library from our observations $\{u_k\}_{k=1}^K$,

$$\Theta(\mathbf{U}) = \begin{bmatrix} u_1 & u_1^2 & u_1^3 & \dots & u_1^M \\ u_2 & u_2^2 & u_2^3 & \dots & u_2^M \\ \vdots & \vdots & \vdots & \dots & \vdots \\ u_K & u_K^2 & u_K^3 & \dots & u_K^M \end{bmatrix} \quad \text{and} \quad \dot{\mathbf{U}} = \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \vdots \\ \dot{u}_k \end{bmatrix}.$$

Then we want the solution of the sparse regression problem

$$\min_{\mathbf{w}} \|\dot{\mathbf{U}} - \Theta(\mathbf{U})\mathbf{w}\|_2 + \lambda\|\mathbf{w}\|_0$$

to be $\mathbf{w}^\star = \left[a, b, 0, \dots, 0\right]^T$.

Modified Sequential Thresholding Least Squares

Algorithm 1: MSTLS

Input: Data $\mathbf{U}$, library $\Theta(\mathbf{U})$

Initialize active set $S \leftarrow \{1, \dots, m\}$;

repeat

Solve weighted least squares:

$\mathbf{w}_S \leftarrow \arg\min_{\mathbf{w}_S} \|\Theta_S \mathbf{w}_S - \dot{X}\|_2^2$;

Threshold: remove terms with

$|w_j| \notin [\lambda, \frac{1}{\lambda}] \quad \Rightarrow \quad S \leftarrow S \setminus \{j\}$;

$|w_j| \notin [\lambda \max\{1, \frac{\|b\|}{\|\Theta_j\|}\}, \frac{1}{\lambda} \min\{1, \frac{\|b\|}{\|\Theta_j\|}\}] \quad \Rightarrow \quad S \leftarrow S \setminus \{j\}$;

until S *stabilizes*;

Output: Sparse coefficient matrix $\mathbf{w}$

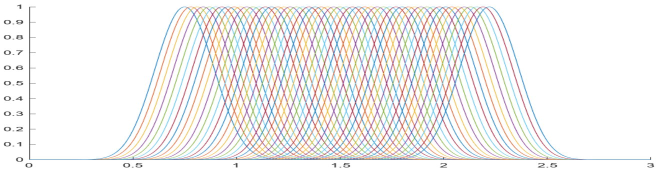
We can also assume that $\mathbf{u}$ *weakly* satisfies the dynamical system, i.e.,

$$-\langle \dot{\phi}, \mathbf{U} \rangle = \langle \phi, \mathbf{F}(\mathbf{U}) \rangle,$$

for any smooth compactly supported test function ϕ.

Then, using a collection of test functions $\{\phi_n\}_{n=1}^N$, we construct and solve the linear system

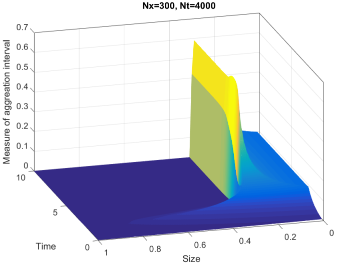
$$\mathbf{b} := -\begin{pmatrix} \langle \dot{\phi}_1, \mathbf{U} \rangle \\ \vdots \\ \langle \dot{\phi}_N, \mathbf{U} \rangle \end{pmatrix} = \begin{pmatrix} \langle \phi_1, f_1(\mathbf{U}) \rangle & \dots & \langle \phi_1, f_I(\mathbf{U}) \rangle \\ \vdots & \ddots & \vdots \\ \langle \phi_N, f_1(\mathbf{U}) \rangle & \dots & \langle \phi_N, f_I(\mathbf{U}) \rangle \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_I \end{pmatrix} =: \mathbf{G}(\mathbf{U})\mathbf{w}.$$

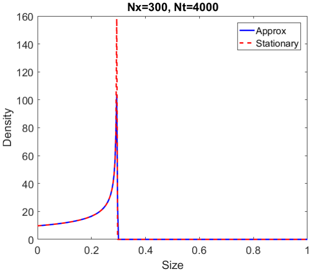


Why use the weak form?

There are two reasons we might prefer the weak form over the strong form:

- The true solution **u** may not be smooth.

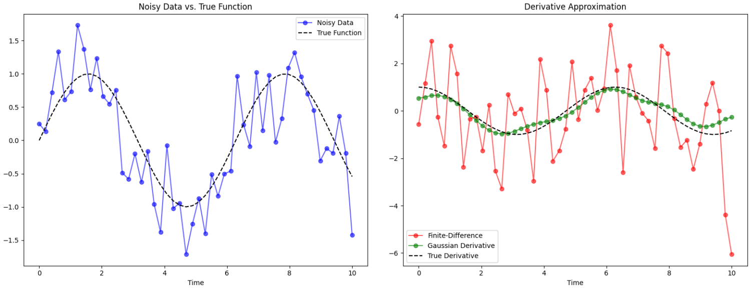




- Approximating derivatives in the presence of noise is unstable:

$$\mathbb{E}(|D_h^c \mathbf{u} - \dot{\mathbf{u}}|^2) = \mathcal{O}\left(\frac{\sigma^2}{h^2}\right).$$

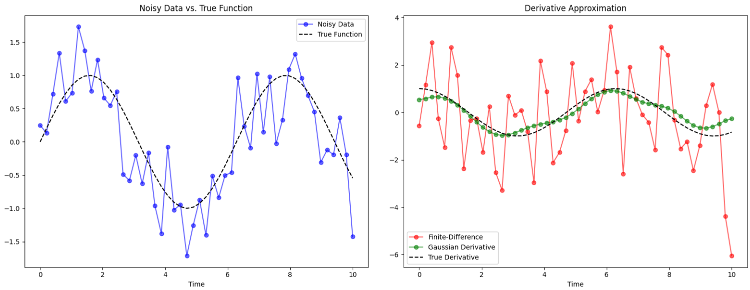
Directly approximating $\dot{u}$ amplifies noise.



Finite differences vs. Gaussian smoothing on noisy data

The problem with noisy derivatives

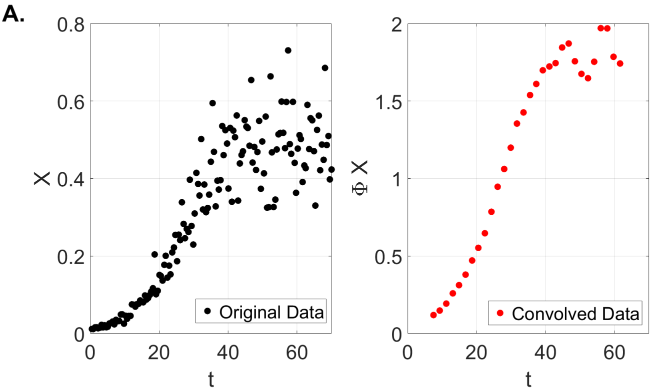
Directly approximating $\dot{u}$ amplifies noise.

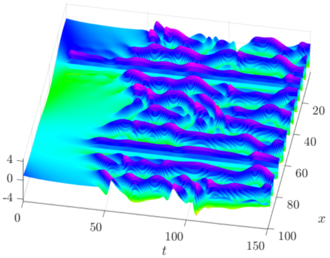


Finite differences vs. Gaussian smoothing on noisy data

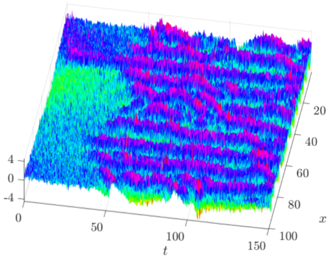
With Gaussian-like test functions, the weak-form simultaneously smooths the data while preserving the equational relationship:

$$-\langle \dot{\phi}, \mathbf{u} \rangle = \langle \phi, \mathbf{F}(\mathbf{u}) \rangle.$$


A.
0.8
0.6
X 0.4
0.2
0
0
20
40
60
t
Original Data
2
1.5
ΦX 1
0.5
0
0
20
40
60
t
Convolved Data



Noise-free



50% Noise

Noise level	Max. Coefficient Error	Identification Rate
0%	8.1×10^{-7}	100%
25%	0.017	100%
50%	0.070	100%
100%	0.31	96.1%

DAM & DMB, JCompPhys, 443:110525, Oct. 2021

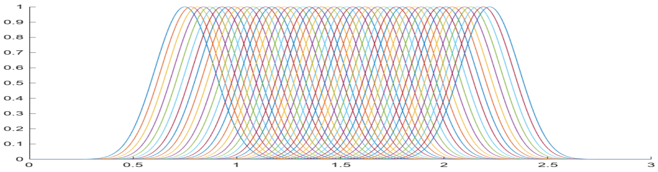
Properties of ϕ

- *Smoothness* of the test function allows you to avoid derivative calculations of noisy data.
  For example, approximating

$$\dot{u} \text{ v.s. } -\langle\dot{\phi}, u\rangle.$$

- *Compact support* of the smooth test function allows for highly accurate computations of integrals.
  For $f \in C_c^p$,

$$\left|\text{trapz}(x, f) - \int f \, \mathrm{d}x\right| \sim \mathcal{O}(\Delta x^p).$$



Many Applications: Weather and Atmospheric Systems.

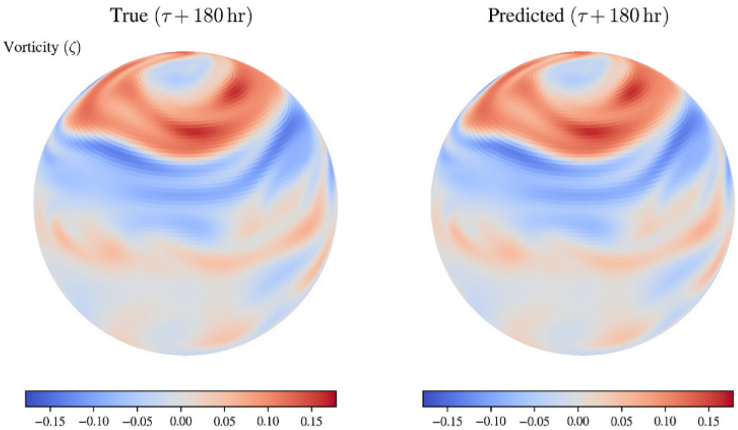


Figure: Minor, Seth, et al. "Learning physically interpretable atmospheric models from data with WSINDy." Journal of Geophysical Research: Machine Learning and Computation (2025)

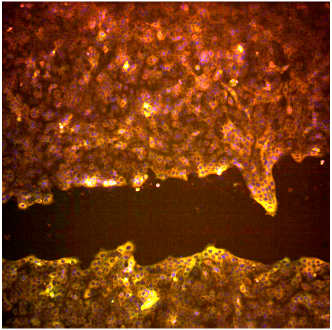
Many Applications:

- Mean field limits of interacting particles,
- Hybrid Dynamical Systems,
- Hidden compartment SIR-models,
- Latent variables,
- etc.

My Applications

Structured Population models:

$$\begin{cases} \partial_t n + \partial_x (g[n](x)n) = -d[n](x)n, \\ g[n](0)\, n(t,0) = \int_0^\infty \beta[n](x) n(t,x)\, \mathrm{d}x, \\ n(0,x) = n_0(x), \end{cases}$$



Cell migration:

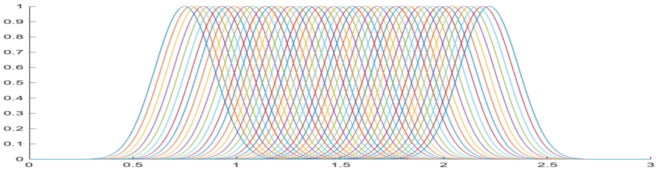
$$\begin{cases} \ddot{x}_i = \frac{1}{N}\sum_{j=1}^{N} f_{ar}(|x_j - x_i|, \theta_{ij})(x_i - x_j) \\ \qquad + \frac{1}{N}\sum_{j=1}^{N} f_{aln}(|x_j - x_i|, \theta_{ij})(v_i - v_j) \\ \qquad + \frac{1}{N}\sum_{j=1}^{N} f_d(|v_i|, \theta_{ij}) v_i. \end{cases}$$

Open ends

- Non-autonomous and heterogeneous dynamics?

$$\dot{u} = f(t, u) \quad \text{or} \quad \partial_t u = \mathcal{A}(x, t)u.$$

- Capturing boundary processes?



A general structured population model

We will assume the uncorrupted population density, $n^\star$ evolves according to a hyperbolic structured population model

$$\begin{cases} \partial_t n^\star(t,s) + \nabla_s \cdot (g^\star[n^\star](s)\, n^\star(t,s)) = f^\star[n^\star](s, n^\star), & (t,s) \in (0,T) \times \Omega, \\ g^\star[n^\star](s)\, n^\star(t,s) \cdot \vec{\eta}(s) = \int_\Omega \beta^\star[n^\star](\sigma) n^\star(t,\sigma)\, \mathrm{d}\sigma, & (t,s) \in [0,T] \times \partial\Omega^+, \\ g^\star[n^\star](s)\, n^\star(t,s) \cdot \vec{\eta}(s) = 0, & (t,s) \in [0,T] \times \partial\Omega^-, \\ n^\star(0,s) = n_0^\star(s), & s \in \Omega. \end{cases}$$

- $g^\star$ represents a transport rate (e.g., aging, growth),
- $f^\star$ represents some source/sink process (e.g., death, cell-division, coagulation-fragmentation),
- $\beta^\star$ represents some inflow rate (e.g., birth, shedding),
- $f[n^\star]$ represents some nonlocal dependence on the density (e.g., dependence on the total population).

A general structured population model

We will assume the uncorrupted population density, $n^\star$ evolves according to a hyperbolic structured population model

$$\begin{cases} \partial_t n^\star(t,s) + \nabla_s \cdot (g^\star[n^\star](s)\, n^\star(t,s)) = f^\star[n^\star](s, n^\star), & (t,s) \in (0,T) \times \Omega, \\ g^\star[n^\star](s)\, n^\star(t,s) \cdot \vec{\eta}(s) = \int_\Omega \beta^\star[n^\star](\sigma) n^\star(t,\sigma)\,\mathrm{d}\sigma, & (t,s) \in [0,T] \times \partial\Omega^+, \\ g^\star[n^\star](s)\, n^\star(t,s) \cdot \vec{\eta}(s) = 0, & (t,s) \in [0,T] \times \partial\Omega^-, \\ n^\star(0,s) = n_0^\star(s), & s \in \Omega. \end{cases}$$

- $g^\star$ represents a transport rate (e.g., aging, growth),
- $f^\star$ represents some source/sink process (e.g., death, cell-division, coagulation-fragmentation),
- $\beta^\star$ represents some inflow rate (e.g., birth, shedding),
- $f[n^\star]$ represents some nonlocal dependence on the density (e.g., dependence on the total population).

How can we recover $\beta^\star$?

A general structured population model

We will assume the uncorrupted population density, $n^\star$ evolves according to a hyperbolic structured population model

$$\begin{cases} \partial_t n^\star(t,s) + \nabla_s \cdot (g^\star[n^\star](s)\, n^\star(t,s)) = f^\star[n^\star](s, n^\star), & (t,s) \in (0,T) \times \Omega, \\ g^\star[n^\star](s)\, n^\star(t,s) \cdot \vec{\eta}(s) = \int_\Omega \beta^\star[n^\star](\sigma) n^\star(t,\sigma)\, \mathrm{d}\sigma, & (t,s) \in [0,T] \times \partial\Omega^+, \\ g^\star[n^\star](s)\, n^\star(t,s) \cdot \vec{\eta}(s) = 0, & (t,s) \in [0,T] \times \partial\Omega^-, \\ n^\star(0,s) = n_0^\star(s), & s \in \Omega. \end{cases}$$

Let $N^\star(t) = \int_\Omega n^\star(t,x)\,\mathrm{d}x$. Then

$$\frac{\mathrm{d}}{\mathrm{d}t} N^\star = \int_\Omega \beta^\star[n^\star](s) n^\star(t,s)\,\mathrm{d}s - \int_\Omega f^\star[n^\star](s) n^\star(t,s)\,\mathrm{d}s$$

Artificial Data

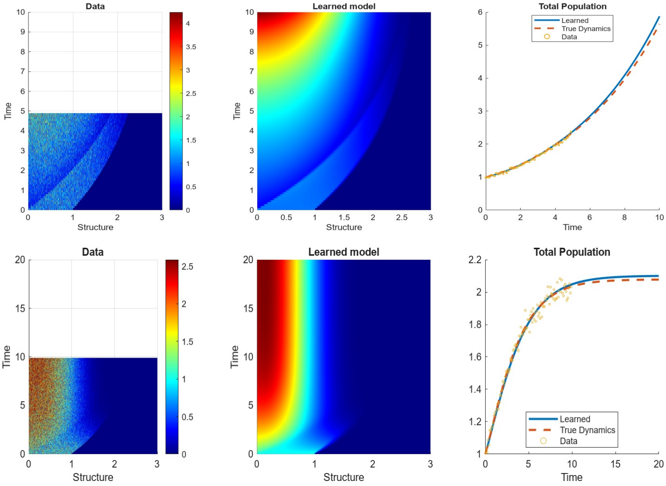
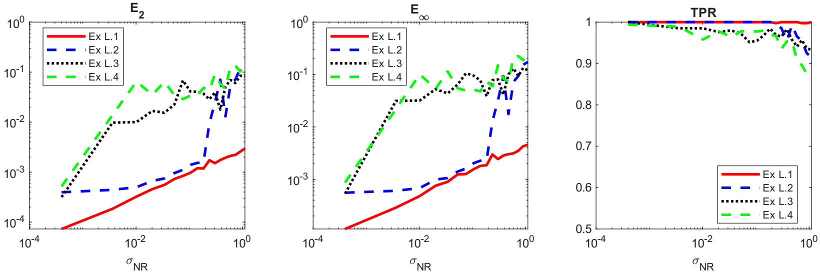


Figure: Typical results of using the WSINDy algorithm with $\sigma_{NR} \approx 0.66$.

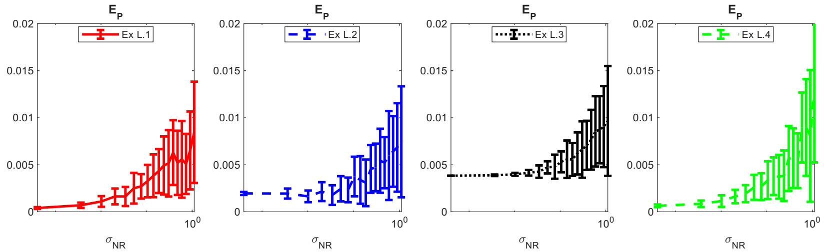
Effect of Noise



$$\mathbf{E}_2(\mathbf{w}) := \frac{\|\mathbf{w}-\mathbf{w}^\star\|_2}{\|\mathbf{w}^\star\|_2} \qquad \mathbf{E}_\infty(\mathbf{w}) := \max_{\{j:\mathbf{w}_j^\star\neq 0\}} \frac{|\mathbf{w}_j-\mathbf{w}_j^\star|}{|\mathbf{w}_j^\star|}$$

$$\mathbf{TPR}(\mathbf{w}) := \frac{TP}{TP+FP+FN} \qquad \mathbf{E}_p(\mathbf{w}) := \frac{\|\tilde{n}-n^\star\|_{L^2((T_{\text{test}},T)\times\Omega)}}{\|n^\star\|_{L^2((T_{\text{test}},T)\times\Omega)}}$$

Effect of Noise



$$\mathbf{E}_2(\mathbf{w}) := \frac{\|\mathbf{w}-\mathbf{w}^\star\|_2}{\|\mathbf{w}^\star\|_2} \qquad \mathbf{E}_\infty(\mathbf{w}) := \max_{\{j:\mathbf{w}_j^\star\neq 0\}} \frac{|\mathbf{w}_j-\mathbf{w}_j^\star|}{|\mathbf{w}_j^\star|}$$

$$\mathbf{TPR}(\mathbf{w}) := \frac{TP}{TP+FP+FN} \qquad \mathbf{E}_p(\mathbf{w}) := \frac{\|\tilde{n}-n^\star\|_{L^2((T_{\text{test}},T)\times\Omega)}}{\|n^\star\|_{L^2((T_{\text{test}},T)\times\Omega)}}$$

Effective Models vs Correct Models

Because these models are stable with respect to model ingredients, it is possible to learn incorrect, but effective model ingredients.

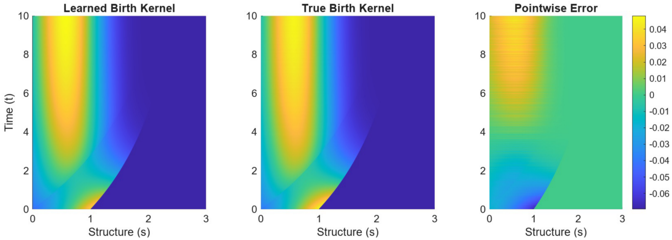


Figure: From left to right, we present the learned linear birth kernel $\beta(s)n^\star(t,s)$, the true nonlinear birth kernel $\beta^\star(s,N(t))n^\star(t,s)$, and their pointwise error over the training time.

Effective Models vs Correct Models

Because these models are stable with respect to model ingredients, it is possible to learn incorrect, but effective model ingredients.

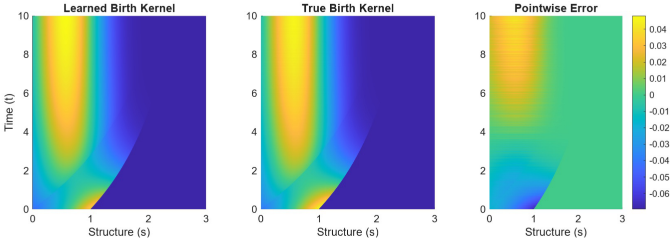


Figure: From left to right, we present the learned linear birth kernel $\beta(s)n^\star(t,s)$, the true nonlinear birth kernel $\beta^\star(s,N(t))n^\star(t,s)$, and their pointwise error over the training time.

Note: This problem tends to go away with more data.

Real Data

Journal of Animal Ecology

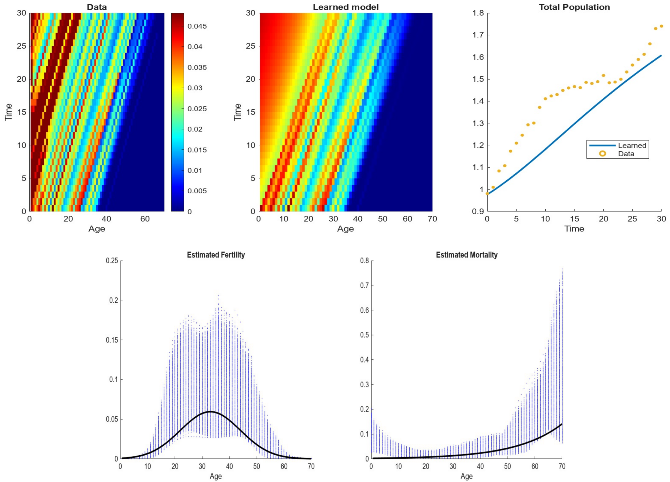
RESEARCH ARTICLE | Open Access

Changes in age-structure over four decades were a key determinant of population growth rate in a long-lived mammal

John Jackson, Khyne U. Mar, Win Htut, Dylan Z. Childs, Virpi Lummaa

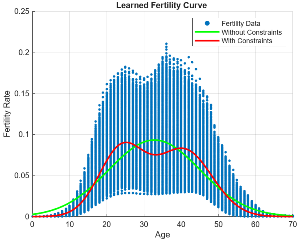
First published: 27 June 2020 | https://doi.org/10.1111/1365-2656.13290 | Citations: 8

Real Data

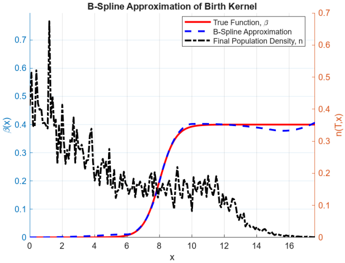


Data
Time
Age
Learned model
Time
Age
Total Population
Learned
Data
Time
Estimated Fertility
Age
Estimated Mortality
Age

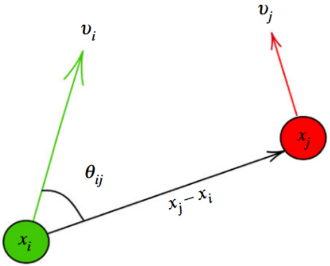
- Constraints?



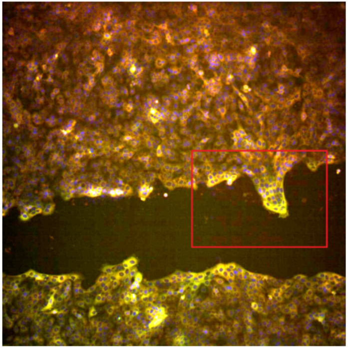
- Nonparametric?

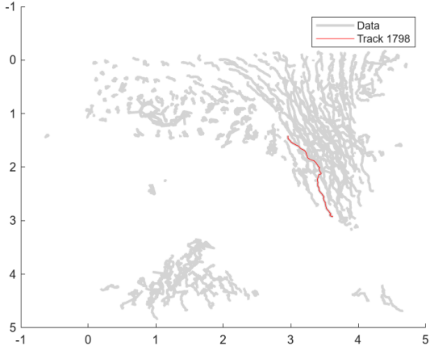


$$\ddot{x}_i = \frac{1}{N_{\text{tot}}} \sum_{j=1}^{N_{\text{tot}}} f_{\text{a-r}}(|x_i - x_j|, \theta_{ij})(x_i - x_j)$$
$$+ \frac{1}{N_{\text{tot}}} \sum_{j=1}^{N_{\text{tot}}} f_{\text{align}}(|x_i - x_j|, \theta_{ij})(v_i - v_j)$$
$$+ \frac{1}{N_{\text{tot}}} \sum_{j=1}^{N_{\text{tot}}} f_{\text{drag}}(|v_i|, \theta_{ij})v_i.$$

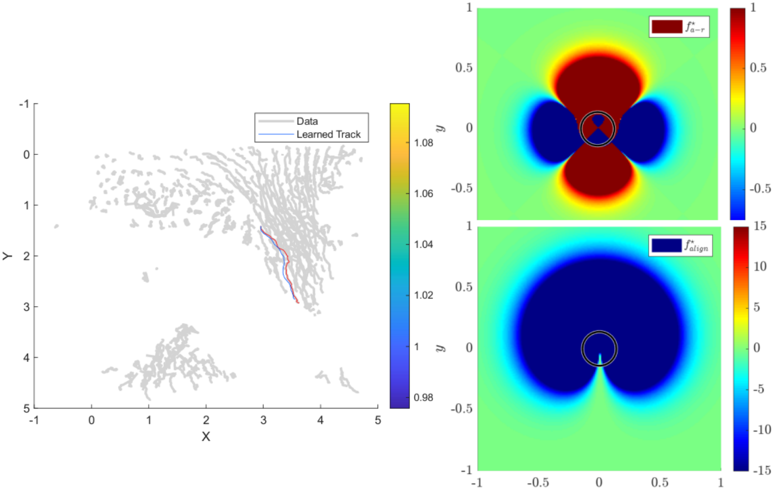


Results

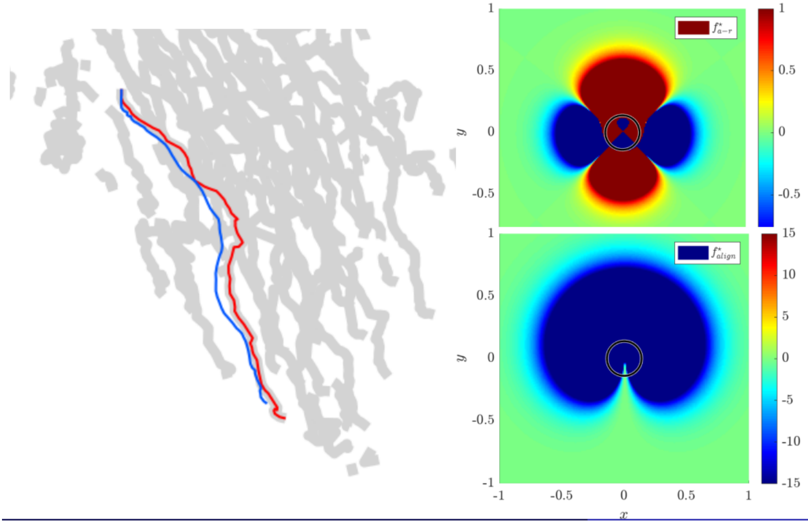




Results

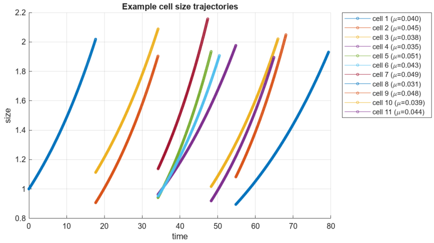

Data
Learned Track
Y
X
-1
0
1
2
3
4
5
1.08
1.06
1.04
1.02
1
0.98
y
x
0.5
-0.5
f^*_{a-r}
f^*_{align}
15
10
5
-5
-10
-15

Results

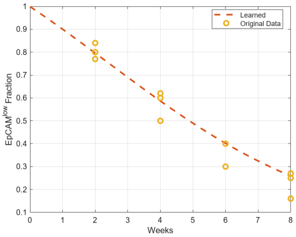


Current Work

- Individual Tracking



- Sparse data



Thanks!

Thank you!

I am grateful for the support of

- National Institute of General Medical Sciences of the National Institutes of Health under award number R35GM14933
- National Science Foundation under award number 2054085

Preprint: Structured Populations: PLOS Computational Biology
Cell migration: In prep.

Code: github.com/MathBioCU