Sparse additive model for ODEs

Zhe Gao

School of Mathematics
Sun Yat-sen University

Nov, 16 2022

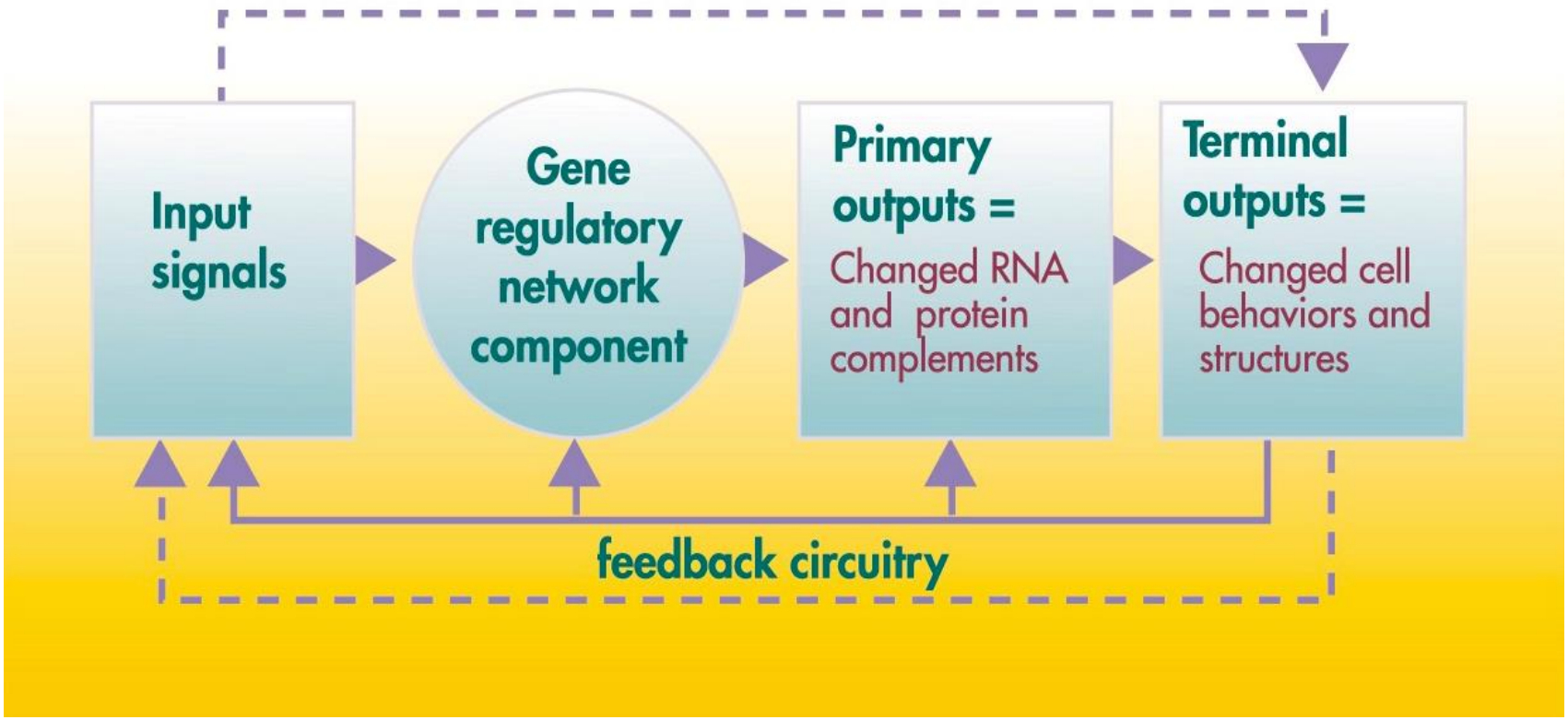
Table of Contents

Gene Regulatory Network

- A gene regulatory network (GRN) is a collection of molecular regulators that interact with each other and with other substances in the cell to govern the gene expression levels of mRNA and proteins which, in turn, determine the function of the cell.
- The regulator can be DNA, RNA, protein or any combination of two or more of these three that form a complex, such as a specific sequence of DNA and a transcription factor to activate that sequence. The interaction can be direct or indirect (through transcribed RNA or translated protein).
- Modeling techniques include differential equations (ODEs), Boolean networks, Petri nets, Bayesian networks, graphical Gaussian network models, Stochastic, and Process Calculi.

Gene Regulatory Network

GRN also play a central role in morphogenesis, the creation of body structures, which in turn is central to evolutionary developmental biology (evo-devo).



ODEs model

A general ODE model for GRNs can be written as

$$\boldsymbol{X}'(t) = \boldsymbol{F}(t, \boldsymbol{X}(t), \boldsymbol{\theta})$$

where $t \in [t_0, T]\,(0 \leq t_0 < T < \infty)$ is time, $\boldsymbol{X}(t) = (X_1(t), \ldots, X_p(t))^T$ is a vector representing the gene expression level of gene $1, \ldots, p$ at time t, and $\boldsymbol{X}'(t)$ is the first-order derivative of $\boldsymbol{X}(t)$. $\boldsymbol{F}$ serves as the link function that quantifies the regulatory effects of regulator genes on the expression change of a target gene, which depends on a vector of parameters $\boldsymbol{\theta}$.

ODEs model

In general, $\boldsymbol{F}$ can take any linear or nonlinear functional forms.

- **Linear ODEs:**

$$X_k'(t) = \sum_{j=1}^{p} \theta_{kj} X_j(t), \quad k = 1, 2, \ldots, p$$

where parameters $\boldsymbol{\theta} = \left\{\theta_{kj}\right\}_{k,j=1,\ldots,p}$ quantify the regulations and interactions among the genes in the network. SCAD is applied for variable selection.

ODEs model

- **General additive nonparametric ODEs:**

$$X_k'(t) = \mu_k + \sum_{j=1}^{p} f_{kj}\left(X_j(t)\right), \quad k = 1, 2, \ldots, p,$$

where μ_k is an intercept term and $f_{kj}(\cdot)$ is a smooth function to quantify the nonlinear relationship among related genes in the GRN.

ODEs model

- **Sparse additive ODE:**

$$X_k'(t) = \mu_k + \sum_{j=1}^{p} f_{kj}\left(X_j(t)\right), \quad k = 1, 2, \ldots, p,$$

Assume $f_{kj}(\cdot)$ is small for each of the p variables (genes), X_k, although the total number of variables (genes), p, in the network may be large. Also we assume that the measurements of gene expression for the k th gene are obtained at multiple time points, $t_i, i = 1, \ldots, n$,

$$Y_k\left(t_i\right) = X_k\left(t_i\right) + \varepsilon_k\left(t_i\right), \quad k = 1, 2, \ldots, p$$

where the measurement errors $\varepsilon_k\left(t_i\right)\left(i = 1, \ldots, n\right)$ are assumed to be iid with mean zero and variance $\sigma_k^2\left(0 < \sigma_k^2 < \infty\right)$.

Motivation

- The new high-throughput technologies such as DNA microarray and next generation RNA-Seq enable us to observe the dynamic features of gene expression profiles in a genome scale.
- High-dimensional nonlinear ODEs has not been addressed.
- The challenging question is how to perform model selection for the nonparametric SA-ODE model under the assumption of sparsity constraints on the index set $\left\{j : f_{kj}(\cdot) \neq 0\right\}$ of functions $f_{kj}(\cdot)$ that are not identically zero.

Strategy

A two-stage smoothing-based estimation method.

- **The first stage:** use a nonparametric smoothing approach to obtain the estimates of both the state variables and their derivatives from the observed data.
- **The second stage:** estimate the unknown parameters using a formulated pseudo-regression model.

Variable selection procedure

- **Step I:** Use the nonparametric smoothing approaches to estimate both the ODE state variables and their derivatives based on the measurement model.
- **Step II:** Use spline functions to approximate each of the nonparametric additive components in the SA-ODE model, and then form a "pseudo" sparse additive model.
- **Step III:** Apply the group LASSO approach to obtain an initial estimator and reduce the dimension of the problem.

Variable selection procedure

- **Step IV:** Apply the adaptive group LASSO approach for component selection,
- **Step V:** Use the regular LASSO again to the selected model from Step IV to further shrink some of the coefficients of B-spline basis to zero so that we can obtain a more parsimonious model at the end.

To summarise, estimate the state variables, solve SA-ODE, dimension reduction, variable selection.

Nonparametric smoothing

Use the penalized splines to estimate $\ddot{X}_k(t)$ and $\tilde{X}'_k(t)$, $k = 1, \ldots, p$.

$$X_k(t) \approx \sum_{j=-\nu}^{K_k} \delta_{k,j} N_{k,j,v+1}(t) = \mathbf{N}_{k,v+1}^T(t)\boldsymbol{\delta}_k,$$

where $\boldsymbol{\delta}_k = (\delta_{k,-v}, \ldots, \delta_{k,K_k})^T$ is the unknown coefficient vector, and $\mathbf{N}_{k,v+1}(t) = \{N_{k,-v,\nu+1}(t), \ldots, N_{k,K_k,v+1}(t)\}^T$ is the B-spline basis function vector of degree v and dimension $K_k + v + 1$ at a sequence of knots $t_0 = \tau_{k,-\nu} = \tau_{k,-\nu+1} = \cdots = \tau_{k,-1} = \tau_{k,0} < \tau_{k,1} < \cdots < \tau_{k,K_k} < \tau_{k,K_k+1} = \tau_{k,K_k+2} = \cdots = \tau_{k,K_k+\nu+1} = T$ on $[t_0, T]$.

Penalized spline

Define $n \times (K_k + v + 1)$ matrix $\mathbf{N}_k = \{\mathbf{N}_{k,v+1}(t_1), \ldots, \mathbf{N}_{k,v+1}(t_n)\}^T$, $\mathbf{Y}_k = (Y_k(t_1), \ldots, Y_k(t_n))^T$ and let $\mathbf{V}_k = \int_{t_0}^{T} \left[\mathbf{N}''_{k,v+1}(t)\right] \left[\mathbf{N}''_{k,v+1}(t)\right]^T dt$.

The penalized spline (P-spline) objective function is

$$L_k(\boldsymbol{\delta}_k; \lambda_k) = (\mathbf{Y}_k - \mathbf{N}_k\boldsymbol{\delta}_k)^T(\mathbf{Y}_k - \mathbf{N}_k\boldsymbol{\delta}_k) + \lambda_k\boldsymbol{\delta}_k^T\mathbf{V}_k\boldsymbol{\delta}_k.$$

The minimizer takes the form $\hat{\boldsymbol{\delta}}_k = \left(\mathbf{N}_k^T\mathbf{N}_k + \lambda_k\mathbf{V}_k\right)^{-1}\mathbf{N}_k^T\mathbf{Y}_k$. Then we have

$$\hat{X}_k(t) = \mathbf{N}_{k,v+1}^T(t)\hat{\boldsymbol{\delta}}_k, \quad \hat{X}'_k(t) = \left[\mathbf{N}'_{k,v+1}(t)\right]^T\hat{\boldsymbol{\delta}}_k.$$

Penalized spline

- The derivatives of spline functions can be simply expressed in terms of lower order spline functions.
- λ_k is obtained by the standard generalized cross-validation (GCV) method.

Pseudo-Sparse Additive Models

Recall the SA-ODE:

$$X_k'(t) = \mu_k + \sum_{j=1}^{p} f_{kj}\left(X_j(t)\right), \quad k = 1, 2, \ldots, p.$$

First, form the following PSA model based on the estimated state variables $\hat{X}_k(t)$ and their derivatives $\hat{X}_k'(t)$

$$H_{ki} = \mu_k + \sum_{j=1}^{p} f_{kj}\left(\hat{X}_{ji}\right) + \Upsilon_{ki}, \quad k = 1, 2, \ldots, p,$$
$$i = 1, 2, \ldots, n,$$

where $H_{ki} = \hat{X}_k'\left(t_i\right)$ and $\hat{X}_{ji} = \hat{X}_j\left(t_i\right), \Upsilon_{ki}$ is the sum of measurement errors and estimation errors of $\hat{X}_k'(t)$ and $\hat{X}_k(t)$ from Step I.

Pseudo-Sparse Additive Models

- The error terms are not iid, but dependent. Thus, this is not a standard sparse additive regression model.
- PSA model allows us to decouple the p-dimensional ODE model into p one-dimensional ODEs independently.

B-spline approximation

Apply truncated series expansions with B-spline bases to approximate the additive components in model. Then there exists a normalized B-spline basis $\{\phi_m, 1 \leq m \leq m_n\}$ on $[t_0, T]$ for $\mathcal{S}_n$, where $m_n \equiv K_n + l$ such that, for any $f_{kj}^* \in \mathcal{S}_n$, it can be expressed as

$$f_{kj}^*(x) = \sum_{m=1}^{m_n} \beta_{kjm}\phi_m(x), \quad k, j = 1, 2, \ldots, p,$$

where β_{kjm} are spline coefficients.

Pseudo-Sparse Additive Models

Replacing f_{kj} by its B-spline approximation, SPA model can be expressed as

$$H_{ki} = \mu_k + \sum_{j=1}^{p} \sum_{m=1}^{m_n} \beta_{kjm} \phi_m \left(\hat{X}_{ji} \right) + \Upsilon_{ki}^*, \quad k = 1, \ldots, p,$$
$$i = 1, \ldots, n,$$

where Υ_{ki}^* is the sum of Υ_{ki} and the approximation errors of the additive regression functions by splines.

Group LASSO

Let $\boldsymbol{\beta}_{kj} = \left(\beta_{kj1}, \ldots, \beta_{kjm_n}\right)^T, (k, j = 1, \ldots, p)$ and $\boldsymbol{\beta}_k = \left(\boldsymbol{\beta}_{k1}^T, \ldots, \boldsymbol{\beta}_{kp}^T\right)^T$. Then we have p groups of parameters and our purpose is to select nonzero groups, that is, nonzero $\boldsymbol{\beta}_{kj}$, $k, j = 1, \ldots, p$.

We impose the constraints $\sum_{i=1}^{n} \sum_{m=1}^{m_n} \beta_{kjm} \phi_m \left(\hat{X}_{ji}\right) = 0, 1 \leq j \leq p$ or use the centralization of the response and the basis functions to remove the restrictions.

A weight function with boundary restrictions should be imposed to achieve a better convergence rate for parameter estimation. Let $\mathbf{D}_{k1} = \operatorname{diag}\left\{d_{k1}\left(t_1\right), \ldots, d_{k1}\left(t_n\right)\right\}, \mathbf{D}_{k2} = \operatorname{diag}\left\{d_{k2}\left(t_1\right), \ldots, d_{k2}\left(t_n\right)\right\}$, and $\mathbf{D}_{k3} = \operatorname{diag}\left\{d_{k3}\left(t_1\right), \ldots, d_{k3}\left(t_n\right)\right\}$, where $d_{k1}(t), d_{k2}(t)$, and $d_{k3}(t)$ are prescribed nonnegative weight functions on $[t_0, T]$ with boundary conditions $d_{k1}\left(t_0\right) = d_{k1}(T) = 0, \quad d_{k2}\left(t_0\right) = d_{k2}(T) = 0$ and $d_{k3}\left(t_0\right) = d_{k3}(T) = 0$.

Group LASSO

The group LASSO estimator $\tilde{\boldsymbol{\beta}}_k$ by minimizing the following penalized weighted least-square criterion:

$$L_{k1}\left(\boldsymbol{\beta}_k; \lambda_{k1}\right) = \left(\mathbf{H}_k - \mathbf{Z}\boldsymbol{\beta}_k\right)^T \mathbf{D}_{k1}\left(\mathbf{H}_k - \mathbf{Z}\boldsymbol{\beta}_k\right) + \lambda_{k1} \sum_{j=1}^{p} \left\|\boldsymbol{\beta}_{kj}\right\|_2$$

where λ_{k1} is a penalty parameter, which can be determined by Bayesian information criterion (BIC) or extended Bayes information criterion (EBIC). Here, we have dropped μ_k in the arguments of L_{k1} with the centering $\hat{\mu}_k = \bar{H}_k$. Based on the group LASSO estimator $\tilde{\boldsymbol{\beta}}_k$, we can also obtain the estimates of the nonparametric functions, $\tilde{f}_{kj}(x) = \sum_{m=1}^{m_n} \tilde{\beta}_{kjm}\psi_m(x), 1 \leq j \leq p$.

Adaptive Group LASSO

We perform the adaptive group LASSO based on the results from Step III by setting $w_{kj} = \left\|\tilde{\beta}_{kj}\right\|_2^{-1}$ if $\left\|\tilde{\beta}_{kj}\right\|_2 > 0$, otherwise $w_{kj} = \infty$. Then we obtain the adaptive group LASSO estimator $\hat{\beta}_k$ by minimizing the penalized weighted least-square criterion,

$$L_{k2}\left(\beta_k; \lambda_{k2}\right) = \left(\mathbf{H}_k - \mathbf{Z}\beta_k\right)^T \mathbf{D}_{k2} \left(\mathbf{H}_k - \mathbf{Z}\beta_k\right) + \lambda_{k2} \sum_{j=1}^{p} w_{kj} \left\|\beta_{kj}\right\|_2$$

with a penalty parameter λ_{k2}, which can also be determined by BIC or EBIC. Then we obtain the adaptive group LASSO estimates of μ_k and $f_{kj}, \hat{\mu}_k = \bar{H}_k \equiv \frac{1}{n}\sum_{i=1}^{n} H_{ki}$ and $\hat{f}_{kj}(x) = \sum_{m=1}^{m_n} \hat{\beta}_{kjm}\psi_m(x), 1 \leq j \leq p$.

Regular LASSO for Shrinking Basis Coefficients

We reapply the regular LASSO or adaptive LASSO to the final model selected from the adaptive group LASSO in Step IV to shrink the coefficients of unnecessary basis functions into zero, so that we can obtain a final parsimonious model. The minimization criterion for the adaptive LASSO is

$$L_{k3}\left(\beta_k; \lambda_{k3}\right) = \left(\mathbf{H}_k^{(s)} - \mathbf{Z}^{(s)}\beta_k\right)^T \mathbf{D}_{k3} \left(\mathbf{H}_k^{(s)} - \mathbf{Z}^{(s)}\beta_k\right) + \lambda_{k3} \sum_{j=1}^{s} \sum_{m=1}^{m_n} w_{kjm} \left|\beta_{kjm}^{(s)}\right|,$$

where the superscript " (s) " stands for the corresponding quantities for groups picked up from Step IV and s is the total number of groups. The weight w_{kjm} is set as $\left|\hat{\beta}_{kjm}^{(s)}\right|^{-1}$ if $\left|\hat{\beta}_{kjm}^{(s)}\right| > 0$ and $w_{kjm} = \infty$, otherwise.

Application: identification of nonlinear dynamic gene regulatory networks

- The propose is used to identify a nonlinear dynamic gene regulatory network based on time course microarray data for T-cell activation.
- The central event in generation of an immune response is the activation of T-lymphocytes (T-cells). T-cell activation is initiated by the interaction between the T-cell receptor (TCR) complex and the antigenic peptide presented on the surface of an antigen-presenting cell.
- This event triggers a network of signaling molecules, including kinases, phosphatases, and adaptor proteins that couple the stimulatory signal received from the TCR to gene transcription events in the nucleus.

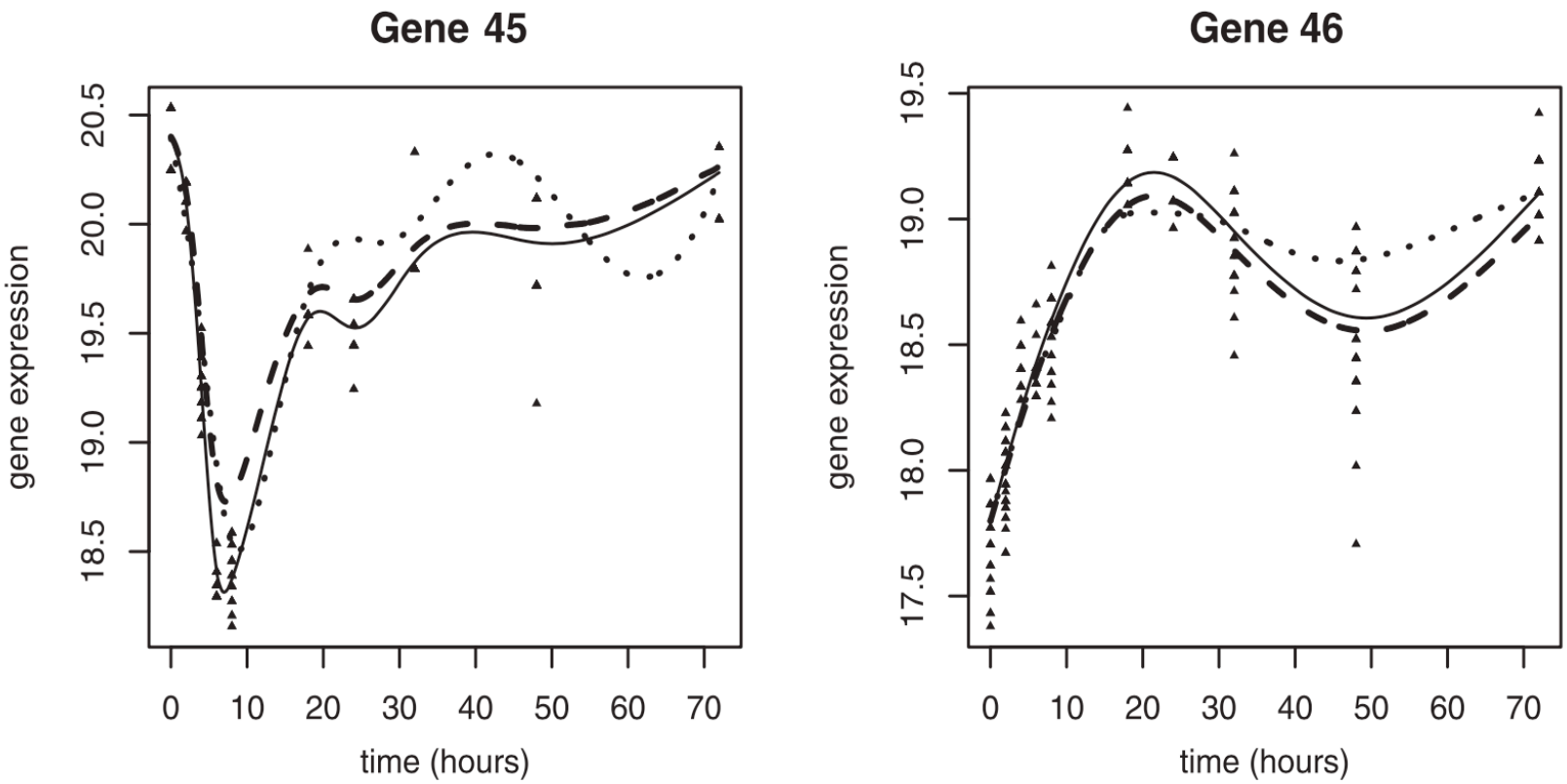
Application: identification of nonlinear dynamic gene regulatory networks

- We intend to apply the proposed SA-ODE model and the proposed variable selection method in the previous sections to establish a nonlinear dynamic regulatory network among 58 genes for the T-cell activation process.
- Each gene replicated 34 times at 10 time points.

Application: identification of nonlinear dynamic gene regulatory networks

- Using the non-parametric mixed-effects smoothing splines technique to obtain the estimates of the mean expression curves and their derivative curves for each gene, respectively.
- Form the PSA model.
- The penalized pseudo-least-square methods, the group LASSO, and adaptive group LASSO approaches are used to identify significant regulations (connections) among the 58 genes with potential nonlinear regulation effects.

Result

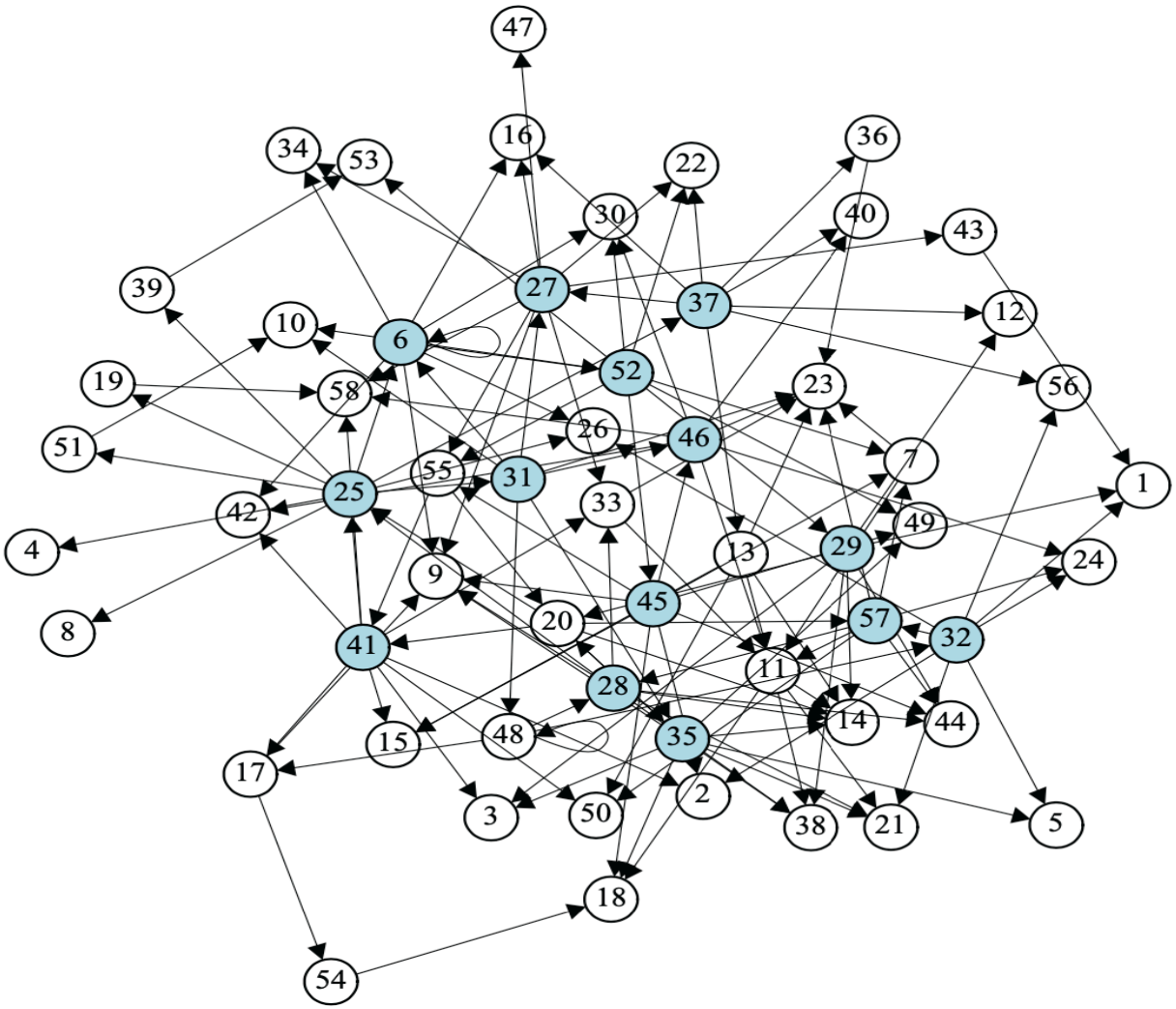


Five step expression curves (dashed lines), the raw data (dots) and the smoothed mean curves (solid lines) from Step I.

Result

- The identified GRN is that each of these 58 genes is regulated by only a few other genes (ranging from 1 to 8 genes), which reflects the fact of sparseness of the network connections.
- The important adaptor molecule in TCR signaling pathway, FYB (gene 45) is regarded as one of the genes having the highest number of outward connections.
- Three more FYB-regulated genes are identified, which carry functions in proliferation (gene 2), interference (gene 44), and apoptosis (gene 49).
- The direct regulation of integrin (gene 15) by FYB.
- Varying effects due to the expression level of the regulator genes.

Result


47
16
34
53
36
22
30
40
43
39
27
37
12
10
6
52
23
56
19
58
26
46
51
55
31
7
1
25
33
42
49
4
13
29
24
9
45
57
8
20
32
41
11
28
14
44
15
48
35
17
2
3
50
38
21
5
18
54

Connclusion

- We have proposed a sparse additive ordinary differential equation (SA-ODE) model for dynamic GRNs to capture the nonlinear regulation effects.
- The real data analysis results also show that the proposed SA-ODE model not only can capture complex nonlinear regulation effects, but also it can identify the varying-effects due to the expression levels of regulator genes.
- It is still a challenging prob- lem to perform variable selection for high-dimensional ODE models with partially observed state variables.

Table of Contents

ODEs model

A system of ODEs takes the form

$$X'(t;\theta) \equiv \begin{bmatrix} \frac{dX_1(t;\theta)}{dt} \\ \vdots \\ \frac{dX_p(t;\theta)}{dt} \end{bmatrix} = \begin{bmatrix} f_1(X(t;\theta),\theta) \\ \vdots \\ f_p(X(t;\theta),\theta) \end{bmatrix}$$
$$\equiv f(X(t;\theta),\theta); \quad t \in [0,1],$$

where $X(t;\theta) = (X_1(t;\theta),\ldots,X_p(t;\theta))^{\mathrm{T}}$ denotes a set of variables, and the form of the functions $f = (f_1,\ldots,f_p)^{\mathrm{T}}$ may be known or unknown. There is also an initial condition of the form $X(0;\theta) = C$, where C is a p-vector. Let $Y_i \in \mathbb{R}^p$ be the measurement of the system at time t_i such that

$$Y_i = X(t_i;\theta^*) + \epsilon_i, \quad i = 1,\ldots,n,$$

where θ^* denotes the true set of parameter values and the random p-vector ϵ_i represents independent measurement errors.

Existing methods-known f

- **Gold Standard Approach:** estimating the unknown parameter θ under a known form f.
- **Two-Step Collocation Methods:** first fitting a smoothing estimate to the observations Y_i, then solving the optimization problem to estimate θ.

$$\hat{\theta}^{\mathrm{TS}} = \arg\min_{\theta} \int_0^1 \left\| \hat{X}'(t;h) - f(\hat{X}(t;h),\theta) \right\|_2^2 dt$$

where

$$\hat{X}(\cdot;h) = \arg\min_{Z(\cdot)\in\mathcal{X}(h)} \sum_{i=1}^{n} \| Y_i - Z(t_i) \|_2^2 .$$

Existing methods-known f

- **The Generalized Profiling Method:** using a smoothing estimate that minimizes the weighted sum of a data-fitting loss and a model-fitting loss for any given θ.

$$\hat{\theta}_{\lambda}^{\mathrm{GP}} = \arg\min_{\theta} \sum_{i=1}^{n} \left\| Y_i - \check{X}(t_i; h, \theta) \right\|_2^2,$$

where

$$\check{X}(\cdot; h, \theta) = \arg\min_{Z(\cdot) \in \mathcal{X}(h)} \frac{1}{n} \sum_{i=1}^{n} \| Y_i - Z(t_i) \|_2^2 + \lambda \int_0^1 \left\| Z'(t) - f(Z(t), \theta) \right\|_2^2 dt.$$

Existing methods-unknown f

- Assume the state functions are additive,

$$X_j'(t) = \theta_{j0} + \sum_{k=1}^{p} f_{jk}\left(X_k(t)\right), \quad \theta_{j0} \in \mathbb{R}.$$

- Approximating the unknown f_{jk} with a truncated basis expansion. Consider a finite basis, $\psi(x) = (\psi_1(x), \ldots, \psi_M(x))^{\mathrm{T}}$, such that

$$f_{jk}\left(a_k\right) = \psi\left(a_k\right)^{\mathrm{T}} \theta_{jk} + \delta_{jk}\left(a_k\right), \quad \theta_{jk} \in \mathbb{R}^M,$$

where $\delta_{jk}\left(a_k\right)$ denotes the residual.

Existing methods-unknown f

- Solving optimization problems of the form

$$\hat{\theta}_j^{\mathrm{NP}} = \underset{\theta_{j0}\in\mathbb{R},\theta_{jk}\in\mathbb{R}^M}{\arg\min} \int_0^1 \left\| \hat{X}_j'(t;h) - \theta_{j0} - \sum_{k=1}^{p} \psi\left(\hat{X}_k(t;h)\right)^{\mathrm{T}} \theta_{jk} \right\|_2^2 dt$$
$$+ \lambda_n \sum_{k=1}^{p} \left[\int_0^1 \left\{ \psi\left(\hat{X}_k(t;h)\right)^{\mathrm{T}} \theta_{jk} \right\}^2 dt \right]^{1/2},$$

  for $j = 1, \ldots, p$, where

$$\hat{X}(\cdot;h) = \underset{Z(\cdot)\in\mathcal{X}(h)}{\arg\min} \sum_{i=1}^{n} \| Y_i - Z(t_i) \|_2^2 .$$

- A standardized group lasso penalty forces all elements in θ_{jk} to be either zero or nonzero when λ_n is large, thereby providing variable selection.

Methodology-ODE model

- Assume an additive ODE model

$$X_j'(t) = \theta_{j0} + \sum_{k=1}^{p} f_{jk}\left(X_k(t)\right), \quad \theta_{j0} \in \mathbb{R}.$$

- Using a finite basis $\psi(\cdot)$ to approximate the additive components f_{jk}:

$$f_{jk}\left(a_k\right) = \psi\left(a_k\right)^{\mathrm{T}} \theta_{jk} + \delta_{jk}\left(a_k\right), \quad \theta_{jk} \in \mathbb{R}^M,$$

where $\delta_{jk}\left(a_k\right)$ denotes the residual.

Methodology-approximating f

- Integrating both sides

$$X_j(t) = X_j(0) + \theta_{j0} t + \sum_{k=1}^{p} \Psi_k(t)^{\mathrm{T}} \theta_{jk} + \sum_{k=1}^{p} \int_0^t \delta_{jk} \left(X_k(u) \right) du,$$

where $\Psi_k(t)$ denotes the integrated basis such that

$$\begin{aligned} \Psi_k(t) &= \left(\Psi_{k1}(t), \ldots, \Psi_{kM}(t) \right)^{\mathrm{T}} \\ &= \int_0^t \psi \left(X_k(u) \right) du, k = 1, \ldots, p, \end{aligned}$$

and $\Psi_0(t) = t$.

Methodology-optimization problem

- Our method, called graph reconstruction via additive differential equations (GRADE), then solves the following problem for $j = 1, \ldots, p$:

$$\hat{\theta}_j = \underset{\substack{C_{j0}\in\mathbb{R},\theta_{j0}\in\mathbb{R},\\ \theta_{j1},\ldots,\theta_{jp}\in\mathbb{R}^M}}{\arg\min} \frac{1}{2n}\sum_{i=1}^{n}\left\{ Y_{ij} - C_{j0} - \theta_{j0}\hat{\Psi}_0(t_i) - \sum_{k=1}^{p}\theta_{jk}^{\mathrm{T}}\hat{\Psi}_k(t_i)\right\}^2$$
$$+ \lambda_{n,j}\sum_{k=1}^{p}\left[\frac{1}{n}\sum_{i=1}^{n}\left\{\theta_{jk}^{\mathrm{T}}\hat{\Psi}_k(t_i)\right\}^2\right]^{1/2},$$

where

$$\hat{X}(\cdot; h) = \underset{Z(\cdot)\in\mathcal{X}(h)}{\arg\min}\sum_{i=1}^{n}\| Y_i - Z(t_i)\|_2^2,$$

and

$$\hat{\Psi}_0(t) = t; \hat{\Psi}_k(t) = \int_0^t \psi\left(\hat{X}_k(u; h)\right) du, k = 1, \ldots, p.$$

Methodology-parameter

- $\lambda_{n,j}$ is a nonnegative sparsity-inducing tuning parameter. We may sometimes use $\lambda_{n,j} \equiv \lambda_n$ for $j = 1, \ldots, p$ for simplicity.
- Let $S_j \equiv \left\{ k : \left\| f_{jk}^* \right\|_2 \neq 0, k = 1, \ldots, p \right\}$ denote the set of true regulators. We let the estimated index set of regulators be $\hat{S}_j \equiv \left\{ k : \left\| \hat{\theta}_{jk} \right\|_2 \neq 0, k = 1, \ldots, p \right\}$. We then reconstruct the network using $\hat{S}_j, j = 1, \ldots, p$.

Methodology-algorithm

- Local polynomial regression is used to obtain the smoothing estimate.
- The smoothing tuning parameter h is selected by generalized cross-validation (GCV).
- We use BIC to select the number of bases M for ψ and $\hat{\Psi}$, and the sparsity tuning parameter λ_n.

Methodology-algorithm

In some studies, time-course data are collected from multiple samples, or experiments. Let R denote the total number of experiments, and $Y^{(r)}$ the observations in the r th experiment. Modify the estimation as

$$\hat{\theta}_j = \underset{\substack{C_{j0}^{(r)} \in \mathbb{R}, \theta_{j0} \in \mathbb{R}, \\ \theta_{j1}, \ldots, \theta_{jp} \in \mathbb{R}^M}}{\arg\min} \frac{1}{2Rn} \sum_{r=1}^{R} \sum_{i=1}^{n} \left\{ Y_{ij}^{(r)} - C_{j0}^{(r)} - \theta_{j0} \hat{\Psi}_0(t_i) - \sum_{k=1}^{p} \theta_{jk}^{\mathrm{T}} \hat{\Psi}_k^{(r)}(t_i) \right\}^2 \tag{2.1}$$

$$+ \lambda_n \sum_{k=1}^{p} \left[\frac{1}{Rn} \sum_{r=1}^{R} \sum_{i=1}^{n} \left\{ \theta_{jk}^{\mathrm{T}} \hat{\Psi}_k^{(r)}(t_i) \right\}^2 \right]^{1/2}, \tag{2.2}$$

where

$$\hat{X}^{(r)}(\cdot; h) = \underset{Z(\cdot) \in \mathcal{X}(h)}{\arg\min} \sum_{i=1}^{n} \left\| Y_i^{(r)} - Z(t_i) \right\|_2^2, r = 1, \ldots, R,$$

$$\hat{\Psi}_0(t) = t; \hat{\Psi}_k^{(r)}(t) = \int_0^t \psi\left(\hat{X}_k^{(r)}(u; h) \right) du, k = 1, \ldots, p.$$

Application to in Silico Gene Expression Data

- GeneNetWeaver (GNW) is based upon real gene regulatory networks of yeast and E. coli. It extracts sub-networks from the yeast or E. coli gene regulatory networks, and assigns a system of ODEs to the extracted network. This system of ODEs is nonadditive, and includes unobserved variables (Marbach et al. 2010). Therefore, the assumptions of GRADE are violated in the GNW data.

Data Generation

- We investigate 10 networks from GNW, of which five have 10 nodes and five have 100 nodes. For each network, GNW provides one set of noiseless gene expression data consisting of R perturbation experiments where the trajectories are measured at n $= 21$ evenly spaced time points in $[0, 1]$.
- Here $R = 10$ for the five 10-node networks and $R = 100$ for the five 100-node networks.
- We add independent $N(0, 0.025^2)$ measurement errors to the data at each timepoint.

Method

- We apply NeRDS as described in Henderson and Michailidis (2014). We apply GRADE using the formulation 2.2 to handle observations from multiple experiments, with the smoothing estimates $\hat{X}$ fit using smoothing splines with bandwidth chosen by GCV, and using cubic splines with two internal knots as the basis functions.
- The integral $\hat{\Psi}_k(t) = \int_0^t \psi\left(\hat{X}_k(u; h)\right) du$ is calculated numerically with step size 0.01.
- Finally, we apply an additional ℓ_2-type penalty to the θ_{jk} 's in 2.2 to match the setup of NeRDS. The tuning parameter for this penalty is set to be 0.1.

Result

Table 1. Area under ROC curves for NeRDS and GRADE.

	$p = 10$		$p = 100$	
	NeRDS	GRADE	NeRDS	GRADE
Ecoli1	0.450 (0.438, 0.462)	**0.545** (0.534, 0.557)	0.624 (0.622, 0.627)	**0.670** (0.667, 0.673)
Ecoli2	0.512 (0.502, 0.523)	**0.643** (0.634, 0.653)	0.637 (0.635, 0.640)	**0.653** (0.650, 0.656)
Yeast1	0.486 (0.476, 0.495)	**0.679** (0.666, 0.691)	0.610 (0.607, 0.612)	**0.636** (0.635, 0.638)
Yeast2	0.525 (0.518, 0.532)	**0.607** (0.600, 0.613)	0.568 (0.566, 0.569)	**0.584** (0.582, 0.585)
Yeast3	0.467 (0.460, 0.474)	**0.576** (0.566, 0.587)	**0.617** (0.616, 0.619)	0.567 (0.566, 0.568)

NOTES: The average area under the curves and 90% confidence intervals, over 100 simulated datasets. Networks and data-generating mechanisms are described in Section 6.1. Boldface indicates the method with larger AUC.

NeRDS outperforms GRADE in one network, while GRADE outperforms NeRDS in the other nine networks. This suggests that GRADE is a competitive exploratory tool for reconstructing gene regulatory networks.

Application to Calcium Imaging Recordings

Learning regulatory relationships among populations of neurons.

- We investigate the calcium imaging recording data from the Allen Brain Observatory project conducted by the Allen Institute for Brain Science.
- In this experiment, calcium fluorescence levels (a surrogate for neuronal activity) are recorded at 30 Hz on a region of the primary visual cortex while the subject mouse is shown 40 visual stimuli. The 40 visual stimuli are combinations of eight spatial orientations and five temporal frequencies. Each stimulus lasts for 2 sec and is repeated 15 times.
- The recorded videos are processed by the Allen Institute to identify individual neurons. In this particular experiment, there are 575 neurons. Each neuron's activity is defined as the average calcium fluorescence level of the pixels that it covers in the video.

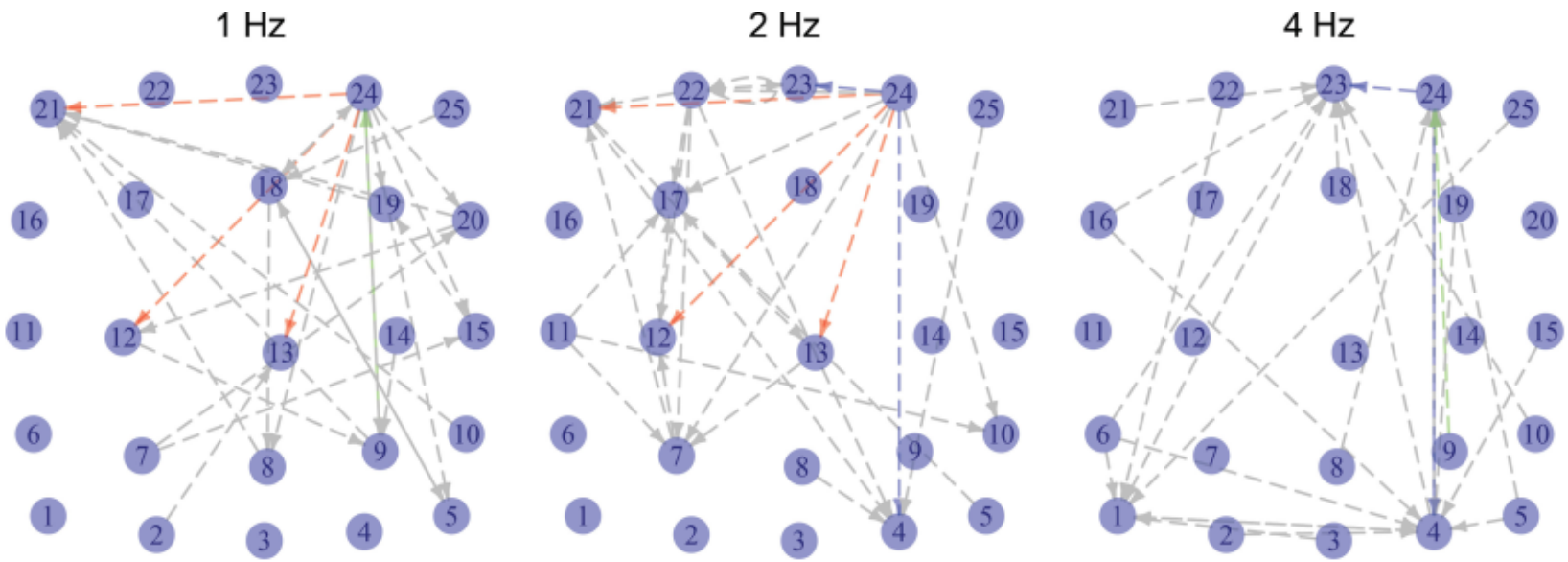
Method

- We define 25 neuronal populations by dividing the recording region into a 5×5 grid, where each population contains roughly 20 neurons.
- We use GRADE to capture the functional connectivity among the 25 neuronal populations.
- We estimate the functional connectivity corresponding to three different but related stimuli, consisting of frequencies of $1\ \mathrm{Hz}, 2\ \mathrm{Hz}$, and $4\ \mathrm{Hz}$, each at a spatial orientation of 90°.
- For each stimulus, we have calcium fluorescence levels of the $p = 25$ neuronal populations for each of $R = 15$ repetitions. There are 60 timepoints per repetition.

Method

- We apply GRADE using the formulation in 2.2 to reconstruct the functional connectivity under each of the three stimuli.
- Smoothing splines with bandwidth h selected with GCV are used to estimate $\hat{X}$.
- Cubic splines with four internal knots are the basis functions $\psi(\cdot)$.
- The sparsity parameter $\lambda_{j,n}$ for each nodewise regression is selected using BIC for each $j = 1, \ldots, 25$.
- For ease of visualization, we prefer a sparse network, and so we fit GRADE using tuning parameter values $\alpha\,(\lambda_{1,n}, \ldots, \lambda_{p,n})$, where the scalar α is selected so that each of the estimated networks contains approximately 25 edges.

Result



- In all three networks, the 24th neuronal population regulates many other neuronal populations, indicating that this region may contain neurons that are sensitive to this spatial orientation.
- The adjacent connectivity networksare somewhat similar to each other, whereas the networks at 1 Hz and 4 Hz have few similarities.

Summary

- We propose a new approach, GRADE, for estimating a system of high-dimensional additive ODEs.
- GRADE involves estimation of an integral rather than a derivative.
- Estimation for time-course data collected from multiple samples, or experiments.

Table of Contents

ODE model

A system of ODEs take the form,

$$\frac{dx(t)}{dt} = \begin{pmatrix} \frac{dx_1(t)}{dt} \\ \vdots \\ \frac{dx_p(t)}{dt} \end{pmatrix} = \begin{pmatrix} F_1(x(t)) \\ \vdots \\ F_p(x(t)) \end{pmatrix} = F(x(t)),$$

where $x(t) = (x_1(t), \ldots, x_p(t))^\top \in \mathbb{R}^p$ denotes the system of p variables of interest, $F = \{F_1, \ldots, F_p\}$ denotes the set of unknown functionals that characterize the regulatory relations among $x(t)$, and t indexes time in an interval standardized to $\mathcal{T} = [0, 1]$.

ODE assumption

The form of F:

- Linear

$$F_j(x(t)) = \theta_{j0} + \sum_{k=1}^{p} \theta_{jk} x_k(t) + \sum_{k \neq l, k=1}^{p} \sum_{l=1}^{p} \theta_{jkl} x_k(t) x_l(t).$$

- Generalized linear form

$$F_j(x(t)) = \theta_{j0} + \psi_j(x(t))^\top \theta_j,$$

where $\psi_j(x) = \left(\psi_{j1}(x), \ldots, \psi_{jd}(x)\right)^\top \in \mathbb{R}^d$ is a finite set of known basis functions.

- Generalized additive model

$$F_j(x(t)) = \theta_{j0} + \sum_{k=1}^{p} F_{jk}\left(x_k(t)\right) = \theta_{j0} + \sum_{k=1}^{p} \left\{ \psi\left(x_k(t)\right)^\top \theta_{jk} + \delta_{jk}\left(x_k(t)\right) \right\}.$$

where $\psi(x) = (\psi_1(x), \ldots, \psi_d(x))^\top \in \mathbb{R}^d$ is a finite set of common basis functions, and $\delta_{jk} \in \mathbb{R}$ is the residual function.

Problems

- How to measure interactions.
- How to conduct statistical inference.

In this article, we propose a novel approach of kernel ordinary differential equation (KODE) for estimation and inference of the ODE system.

- Allow pairwise interactions.
- Sparsity regularization to achieve selection of individual functionals.
- Confidence interval for the estimated signal trajectory

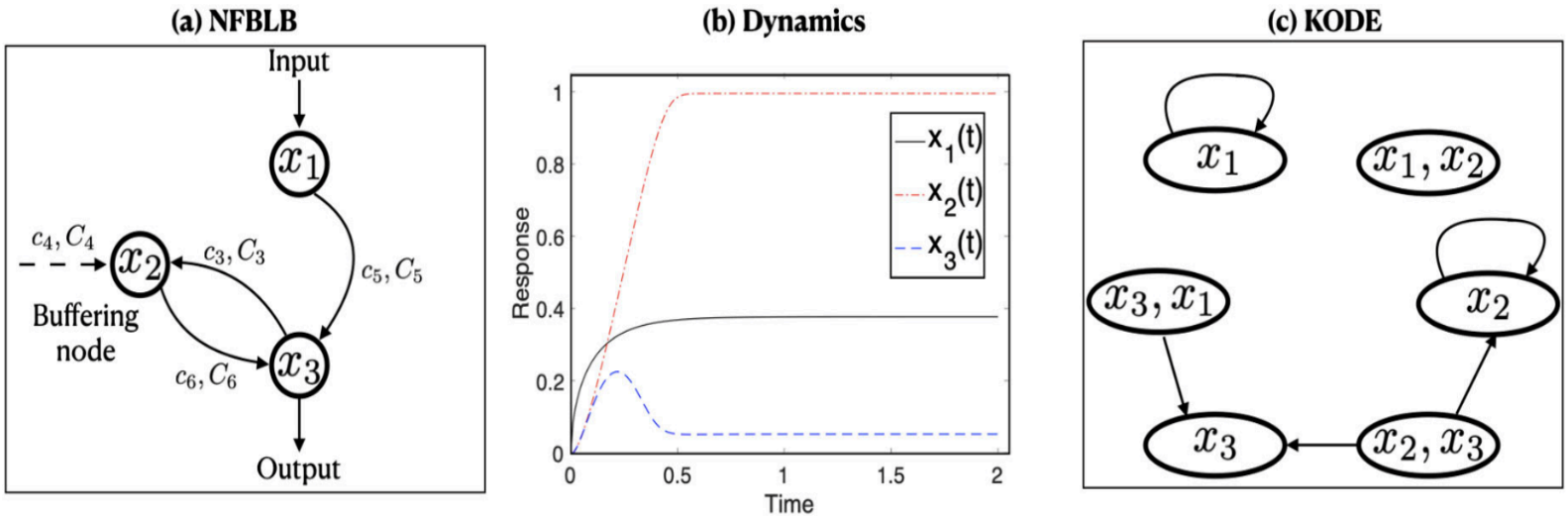
Motivating example: Enzymatic regulatory network

- All circuits of three-node enzyme network topologies that perform biochemical adaptation can be well approximated by two architectural classes: a negative feedback loop with a buffering node, and an incoherent feedforward loop with a proportioner node.
- The mechanism of the first class follows the Michaelis–Menten kinetic equations

$$\begin{aligned}
\frac{dx_1(t)}{dt} &= c_1 \frac{x_0\{1-x_1(t)\}}{\{1-x_1(t)\}+C_1} - \tilde{c}_1 c_2 \frac{x_1(t)}{x_1(t)+C_2} \\
\frac{dx_2(t)}{dt} &= c_3 \frac{\{1-x_2(t)\}x_3(t)}{\{1-x_2(t)\}+C_3} - \tilde{c}_2 c_4 \frac{x_2(t)}{x_2(t)+C_4}, \\
\frac{dx_3(t)}{dt} &= c_5 \frac{x_1(t)\{1-x_3(t)\}}{\{1-x_3(t)\}+C_5} - c_6 \frac{x_2(t)x_3(t)}{x_3(t)+C_6},
\end{aligned}$$

where $x_1(t), x_2(t), x_3(t)$ are three interacting nodes, such that $x_1(t)$ receives the input, $x_2(t)$ plays the diverse regulatory role, and $x_3(t)$ transmits the output.

Motivating example: Enzymatic regulatory network



- In this model, the functionals F_1, F_2, F_3 are all nonlinear, and both F_2 and F_3 involve two-way interactions.

Two-Step Collocation Estimation

- **First step:** Fit the smooth estimate

$$\widehat{x}_j(t) = \arg\min_{z_j \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \{y_{ij} - z_j(t_i)\}^2 + \lambda_{nj} J_1(z_j) \right\}, j = 1, \ldots, p$$

  where $J_1(\cdot)$ is a smoothness penalty in the function space $\mathcal{F}$, and z_j is a function in $\mathcal{F}$ that we minimize over.

- **Second step:** Solve an optimization problem to estimate the model parameters $\theta_{j0} \in \mathbb{R}$ and $\theta_j = (\theta_{j1}, \ldots, \theta_{jp})^\top \in \mathbb{R}^p$, for $j = 1, \ldots,$ p.

$$\min_{\theta_{j0}, \theta_j} \int_0^1 \left(\frac{d\widehat{x}_j(t)}{dt} - \theta_{j0} - \sum_{k=1}^{p} \theta_{jk} \widehat{x}_k(t) \right)^2 dt, \quad j = 1, \ldots, p.$$

Two-Step Collocation Estimation

- **Second step(alternative):** Use the integral $\int_0^t g_j(\widehat{x}(u))du$, rather than the derivative $d\widehat{x}_j(t)/dt$, and they estimated the model parameters $\theta_{j0} \in \mathbb{R}$ and $\theta_j = \left(\theta_{j1}, \ldots, \theta_{jd}\right)^\top \in \mathbb{R}^d$, for $j = 1, \ldots, p$,

$$\min_{\theta_j, \theta_j} \sum_{j=1}^{p} \int_0^1 \left\{ \widehat{x}_j(t) - \theta_{j0} - \theta_j^\top \int_0^t \psi_j(\widehat{x}(u))du \right\}^2 dt.$$

Kernel ODE

Let $\mathcal{H}_k$ denote a space of functions of $x_k(t) \in \mathcal{X}$ with zero marginal integral, where $\mathcal{X} \subset \mathbb{R}$ is a compact set. Let $\{1\}$ denote the space of constant functions. We construct the tensor product space as

$$\mathcal{H} = \{1\} \oplus \sum_{k=1}^{p} \mathcal{H}_k \oplus \sum_{k=1, k \neq l}^{p} \sum_{l=1}^{p} (\mathcal{H}_k \otimes \mathcal{H}_l) .$$

We assume the functionals $F_j, j = 1, \ldots, p$, in the ODE model are located in the space of $\mathcal{H}$. The identifiability is assured by the conditions specified through the averaging operators: $\int_{\mathcal{T}} F_{jk}(x_k(t))\, dt = 0$ for $k = 1, \ldots, p$.

Kernel ODE

Let $\|\cdot\|_{\mathcal{H}}$ denote the norm of $\mathcal{H}$, and $\mathcal{P}^k F_j$ and $\mathcal{P}^{kl} F_j$ denote the orthogonal projection of F_j onto $\mathcal{H}_k$ and $\mathcal{H}_k \otimes \mathcal{H}_l$, respectively. We consider a two-step collocation estimation method, by first obtaining a smoothing spline estimate $\widehat{x}(t) = (\widehat{x}_1(t), \ldots, \widehat{x}_p(t))^\top$, where

$$\widehat{x}_j(t) = \arg\min_{z_j \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \{y_{ij} - z_j(t_i)\}^2 + \lambda_{nj} \|z_j(t)\|_{\mathcal{F}}^2 \right\}, j = 1, \ldots, p,$$

then estimating $F_j \in \mathcal{H}$ and $\theta_{j0} \in \mathbb{R}$ by the following penalized optimization,

$$\min_{\theta_{j0}, F_j} \frac{1}{n} \sum_{i=1}^{n} \left\{ y_{ij} - \theta_{j0} - \int_0^{t_i} F_j(\widehat{x}(t)) dt \right\}^2 + \tau_{nj} \left(\sum_{k=1}^{p} \left\| \mathcal{P}^k F_j \right\|_{\mathcal{H}} + \sum_{k \neq l, k=1}^{p} \sum_{l=1}^{p} \left\| \mathcal{P}^{kl} F_j \right\|_{\mathcal{H}} \right).$$

Kernel ODE

- For the functionals, the formulation is highly flexible, nonlinear, and incorporates two-way interactions. Meanwhile, it naturally covers the linear ODE, and the additive ODE in as special cases.
- **Linear form:** If $\mathcal{H}$ is the linear functional space, $\mathcal{H} = \{1\} \oplus \sum_{k=1}^{p} \{x_k - 1/2\} \oplus \sum_{k \neq l} [\{x_k - 1/2\} \otimes \{x_l - 1/2\}]$ with the input space $\mathcal{X} = [0,1]^p$, then any F of the form in (4) belongs to $\mathcal{H}$.
- **Generalized linear form:** let $\mathcal{H}$ is spanned by some known generalized functions, $\mathcal{H} = \psi_{j1}(x) \oplus \cdots \oplus \psi_{jp}(x)$, then any F in (5) belongs to $\mathcal{H}$.

Kernel ODE

- The first penalty function J_1 is the squared RKHS norm corresponding to the RKHS $\{\mathcal{F}, \|\cdot\|_{\mathcal{F}}\}$.
- The second penalty function J_2 is a sum of RKHS norms on the main effects and pairwise interactions.

Theorem 1. Assume that the RKHS $\mathcal{H}$ can be decomposed as in (8). Then there exists a minimizer of (10) in $\mathcal{H}$ for any tuning parameter $\tau_{nj} \geq 0$. Moreover, the minimizer is in a finite-dimensional space.

Estimation

The estimation of the proposed kernel ODE system consists of two major steps.

- The smoothing spline estimation in ,

$$\widehat{x}_j(t) = \arg\min_{z_j \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \{y_{ij} - z_j(t_i)\}^2 + \lambda_{nj} \|z_j(t)\|_{\mathcal{F}}^2 \right\}, j = 1, \ldots, p,$$

  which is standard and the tuning of the smoothness parameter λ_{nj} is often done through generalized cross-validation.

- The second step is to solve

$$\min_{\theta_{j0}, F_j} \frac{1}{n} \sum_{i=1}^{n} \left\{ y_{ij} - \theta_{j0} - \int_0^{t_i} F_j(\widehat{x}(t)) dt \right\}^2$$
$$+ \tau_{nj} \left(\sum_{k=1}^{p} \left\| \mathcal{P}^k F_j \right\|_{\mathcal{H}} + \sum_{k \neq l, k=1}^{p} \sum_{l=1}^{p} \left\| \mathcal{P}^{kl} F_j \right\|_{\mathcal{H}} \right).$$

Equivalent problem

Consider an equivalent problem:

$$\min_{\theta_{j0},\theta_j,F_j} \frac{1}{n}\sum_{i=1}^{n}\left\{y_{ij}-\theta_{j0}-\int_0^{t_i}F_j(\widehat{x}(t))dt\right\}^2$$
$$+\eta_{nj}\left(\sum_{k=1}^{p}\theta_{jk}^{-1}\left\|\mathcal{P}^k F_j\right\|_{\mathcal{H}}^2+\theta_{jkl}^{-1}\sum_{k=1,k\neq l}^{p}\sum_{l=1}^{p}\left\|\mathcal{P}^{kl}F_j\right\|_{\mathcal{H}}^2\right)$$
$$+\kappa_{nj}\left(\sum_{k=1}^{p}\theta_{jk}+\sum_{k=1,k\neq l}^{p}\sum_{l=1}^{p}\theta_{jkl}\right),$$

subject to $\theta_k \geq 0, \theta_{kl} \geq 0, k, l = 1, \ldots, p, k \neq l$, where $\theta_j = \left(\theta_{j1}, \ldots, \theta_{jp}, \theta_{j12}, \ldots, \theta_{j1p}, \ldots, \theta_{jp1}, \ldots, \theta_{jp(p-1)}\right)^\top \in \mathbb{R}^{p^2}$ collects the parameters to estimate, and $\eta_{nj}, \kappa_{nj\geq 0}$ are the tuning parameters, $j = 1, \ldots, p$. The parameters θ_{jk} and θ_{jkl} to control the sparsity of the main effect and interaction terms in F_j.

Iterative algorithm

- **(1) Estimate θ_{j0} given fixed F_j and θ_j.**

$$\widehat{\theta}_{j0} = \bar{y}_j - \int_{\mathcal{T}} \bar{T}(t)\widehat{F}_j(\widehat{x}(t))dt,$$

where $T_i(t) = 1\{0 \le t \le t_i\}$, $\bar{T}(t) = \frac{1}{n}\sum_{i=1}^{n} T_i(t)$, and $\bar{y}_j = n^{-1}\sum_{i=1}^{n} y_{ij}$.

- **(2) Estimate the functional F_j given fixed θ_{j0} and θ_j.** The optimization problem becomes

$$\min_{F_j}\left\{\frac{1}{n}\sum_{i=1}^{n}\left[(y_{ij}-\bar{y}_j) - \int_{\mathcal{T}}\left\{T_i(t)-\bar{T}(t)\right\}F_j(\widehat{x}(t))dt\right]^2 + \eta_{nj}\left(\sum_{k=1}^{p}\widehat{\theta}_{jk}^{-1}\left\|\mathcal{P}^k F_j\right\|_{\mathcal{H}}^2 + \widehat{\theta}_{jkl}^{-1}\sum_{k=1,k\neq l}^{p}\sum_{l=1}^{p}\left\|\mathcal{P}^{kl}F_j\right\|_{\mathcal{H}}^2\right)\right\}.$$

Iterative algorithm

- **(2) Estimate the functional F_j given fixed θ_{j0} and θ_j(continue).** Let $K_j(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ denote the Mercer kernel generating the RKHS $\mathcal{H}_j, j = 1, \ldots, p$. Then $K_{kl} \equiv K_k K_l$ is the reproducing kernel of the RKHS $\mathcal{H}_k \otimes \mathcal{H}_l$. Let $K_{\theta_j} = \sum_{k=1}^{p} \widehat{\theta}_{jk} K_k + \sum_{k \neq l} \widehat{\theta}_{jkl} K_{kl}$. By the representer theorem, the solution $\widehat{F}_j$ to (12) is of the form,

$$\widehat{F}_j(\widehat{x}(t)) = b_j + \sum_{i=1}^{n} c_{ij} \int_{\mathcal{T}} K_{\theta_j}(\widehat{x}(t), \widehat{x}(s)) \left\{ T_i(s) - \bar{T}(s) \right\} ds$$

for some $b_j \in \mathbb{R}$ and $c_j = (c_{1j}, \ldots, c_{nj}) \in \mathbb{R}^n$.

Iterative algorithm

- **(2) Estimate the functional F_j given fixed θ_{j0} and θ_j(continue).** We obtain the following quadratic minimization problem in terms of $\{b_j, c_j\}$,

$$\min_{b_j, c_j} \frac{1}{n} \|(y_j - \bar{y}_j) - (Bb_j + \Sigma c_j)\|_2^2 + \eta_{nj} c_j^\top \Sigma c_j$$

which has a closed-form solution. Consider the QR decomposition $B = [Q_1 \quad Q_2]\,[R \quad 0]^\top$, where $Q_1 \in \mathbb{R}^{n\times 1}$, $Q_2 \in \mathbb{R}^{n\times(n-1)}$, and $[Q_1\, Q_2]$ is orthogonal such that $B^\top Q_2 = 0_{1\times(n-1)}$. Write $W_j = \Sigma + n\eta_{nj} I_n$, where I_n is the $n \times n$ identity matrix. Then the minimizers are,

$$c_j = Q_2 \left(Q_2^\top W_j Q_2\right)^{-1} Q_2^\top (y_j - \bar{y}_j),$$
$$b_j = R^{-1} Q_1^\top (y_j - \bar{y}_j - W_j c_j).$$

Iterative algorithm

- **(2) Estimate the functional F_j given fixed θ_{j0} and θ_j(continue).** We tune the parameter η_{nj} by minimizing the generalized crossvalidation criterion,

$$\mathrm{GCV} = \frac{\left\| A_j\left(\eta_{nj}\right)\left(y_j - \bar{y}_j\right) - \left(y_j - \bar{y}_j\right)\right\|^2}{\left[n^{-1}\,\mathrm{tr}\left\{I_n - A_j\left(\eta_{nj}\right)\right\}\right]^2},$$

where the smoothing matrix $A_j\left(\eta_{nj}\right) \in \mathbb{R}^{n\times n}$ is of the form,

$$A_j\left(\eta_{nj}\right) = I_n - n\eta_{nj} Q_2 \left(Q_2^\top W_j Q_2\right)^{-1} Q_2^\top$$

Iterative algorithm

- **(3) Estimate θ_j given fixed θ_{j0} and F_j.** Solve a usual ℓ_1-penalized regression problem,

$$\min_{\theta_j} \left\{ (z_j - G\theta_j)^\top (z_j - G\theta_j) + n\kappa_{nj} \left(\sum_{k=1}^{p} \theta_{jk} + \sum_{k \neq l, k=1}^{p} \sum_{l=1}^{p} \theta_{jkl} \right) \right\}$$

subject to $\theta_k \geq 0, \theta_{kl} \geq 0, k, l = 1, \ldots, p, k \neq l$. We employ Lasso in our implementation, and tune the parameter κ_{nj} using 10-fold crossvalidation, following the usual Lasso literature.

Algorithm

Algorithm 1 Iterative optimization algorithm for kernel ODE.

1: Initialization: the initial values for $\theta_{jk} = \theta_{jkl} = 1, j, k, l = 1, \ldots, p, k \neq l$, and the tuning parameters: (η_{nj}, κ_{nj}).
2: Fit smoothing spline model (9), and obtain $\widehat{x}_j(t)$, $j = 1, \ldots, p$.
3: **repeat**
4: Solve $\widehat{\theta}_{j0}$ given $\widehat{F}_j$ and $\widehat{\theta}_j, j = 1, \ldots, p$.
5: Solve $\widehat{F}_j$ in (12) given $\widehat{\theta}_{j0}$ and $\widehat{\theta}_j, j = 1, \ldots, p$.
6: Solve $\widehat{\theta}_j$ in (15) given $\widehat{\theta}_{j0}$ and $\widehat{F}_j, j = 1, \ldots, p$.
7: **until** the stopping criterion is met.

Confidence Intervals

Let $\widehat{\theta}_j$ denote the estimator of θ_j obtained from Algorithm 1 . Denote $M_j \subseteq \mathcal{M}$ as the index set of the nonzero entries of the sparse estimator $\widehat{\theta}_j$. Then the corresponding estimate of the functional F_j is,

$$\widehat{F}_{j,\widehat{\theta}_{M_j}}(\widehat{x}(t)) = b_j + \sum_{i=1}^{n} c_{ij} \int_{\mathcal{T}} K_{\widehat{\theta}_{M_j}}(\widehat{x}, \widehat{x}(s)) \left\{ T_i(s) - \bar{T}(s) \right\} ds.$$

For a nominal level $\alpha \in (0,1)$ and $i = 1, \ldots, n$, define $c_0\left(\widehat{x}_j(t_i)\right)$ as the smallest constant satisfying that,

$$\mathbb{P}_{n,F_j,\sigma_j}\left[\max_{M_j \subseteq \mathcal{M}} \sigma_j^{-1} \left| \left\{ \tilde{A}_{M_j} \right\}_{i\cdot} (y_j - \bar{y}_j) \right| \leq c_0\left(\widehat{x}_j(t_i)\right) \right] \geq 1 - \alpha,$$

where $\left\{ \tilde{A}_{M_j} \right\}_{i\cdot} = \left\{ A_{M_j} \right\}_{i\cdot} / \left\| \left\{ A_{M_j} \right\}_{i\cdot} \right\|_{l_2}$, $\left\{ A_{M_j} \right\}_{i\cdot}$ is the i th row of A_{M_j}, A_{M_j} is the smoothing matrix, and σ_j^2 is the variance of the error term ϵ_{ij}.

Confidence Intervals

We then construct the confidence interval CI $(\widehat{x}_j(t))$ for the prediction of true trajectory $x_j(t)$ following model selection as,

$$\mathrm{CI}\left(\widehat{x}_j\left(t_i\right)\right) = \int_{\mathcal{T}} \left\{ T_i(t) - \bar{T}(t) \right\} \widehat{F}_{j,\widehat{\theta}_{M_j}}(\widehat{x}(t))dt \pm c_0\left(\widehat{x}_j\left(t_i\right)\right)\sigma_j \left\| \left\{ A_{M_j} \right\}_{i\cdot} \right\|,$$

for any $i = 1, \ldots, n$ and $j = 1, \ldots, p$.

Cutoff value

Proposition 1. The value $c_0(\widehat{x}_j(t_i))$ in (16) is the same as the solution of $t \geq 0$ satisfying,

$$\mathbb{E}_U \mathbb{P}\left(\max_{M_j \subseteq \mathcal{M}} \left| \{\widetilde{A}_{M_j}\}_{i\cdot} V \right| \leq t/U \middle| U \right) = 1 - \alpha,$$

where V is uniformly distributed on the unit sphere in $\mathbb{R}^n$, and U is a nonnegative random variable such that U^2 follows a chi-squared distribution $\chi^2(n)$.

Cutoff value

- We first generate N iid copies of random vectors $V_1, \ldots, V_N$ uniformly distributed on the unit sphere in $\mathbb{R}^n$.
- We then calculate the quantity, $c_\nu = \max_{M_j \subseteq \mathcal{M}} \left| \left\{ \tilde{A}_{M_j} \right\}_{i.} V_\nu \right|$ for $\nu = 1, \ldots, N$. Let D_U denote the cumulative distribution function of U, and D_{χ^2} denote the cumulative distribution function of a $\chi^2(n)$ distribution. Then $D_U(t) = D_{\chi^2}\left(t^2\right)$.
- We next obtain $c_0\left(\widehat{x}_j\left(t_i\right)\right)$ by searching for c that solves $N^{-1} \sum_{i=1}^{N} D_U\left(c/c_i\right) = 1 - \alpha$, using, for example, a bisection searching method.
- Finally, we estimate the error variance σ_j^2 using the usual noise estimator in the context of RKHS; that is, $\widehat{\sigma}_j^2 = \left\| A_{M_j}\left(y_j - \bar{y}_j\right) - \left(y_j - \bar{y}_j\right) \right\|^2 / \operatorname{tr}\left(I - A_{M_j}\right)$.

Simulation: Setup

- For a given system of ODEs and the initial condition, we obtain the numerical solutions of the ODEs using the Euler method with step size 0.01.
- We fit the smoothing spline to estimate $x_j(t)$ in (9) using a Matérn kernel, $K_{\mathcal{F}}(x, x') = (1 + \sqrt{3}\|x - x'\|/\nu)\exp\left(-\sqrt{3}\|x - x'\|/\nu\right)$, where the smoothing parameter λ_{nj} is chosen by GCV, and the bandwidth ν is chosen by 10-fold cross-validation.
- We compute the integral $\int_0^{t_i} F_j(\widehat{x}(t))dt$ numerically with independent sets of 1000 Monte Carlo points.

Simulation: Enzymatic Regulatory Network

- A three-node enzyme regulatory network of a negative feedback loop with a buffering node.
- The ODE system is

$$\begin{aligned}
\frac{dx_1(t)}{dt} &= c_1 \frac{x_0\{1-x_1(t)\}}{\{1-x_1(t)\}+C_1} - \tilde{c}_1 c_2 \frac{x_1(t)}{x_1(t)+C_2} \\
\frac{dx_2(t)}{dt} &= c_3 \frac{\{1-x_2(t)\}x_3(t)}{\{1-x_2(t)\}+C_3} - \tilde{c}_2 c_4 \frac{x_2(t)}{x_2(t)+C_4} \\
\frac{dx_3(t)}{dt} &= c_5 \frac{x_1(t)\{1-x_3(t)\}}{\{1-x_3(t)\}+C_5} - c_6 \frac{x_2(t)x_3(t)}{x_3(t)+C_6}.
\end{aligned}$$

Results

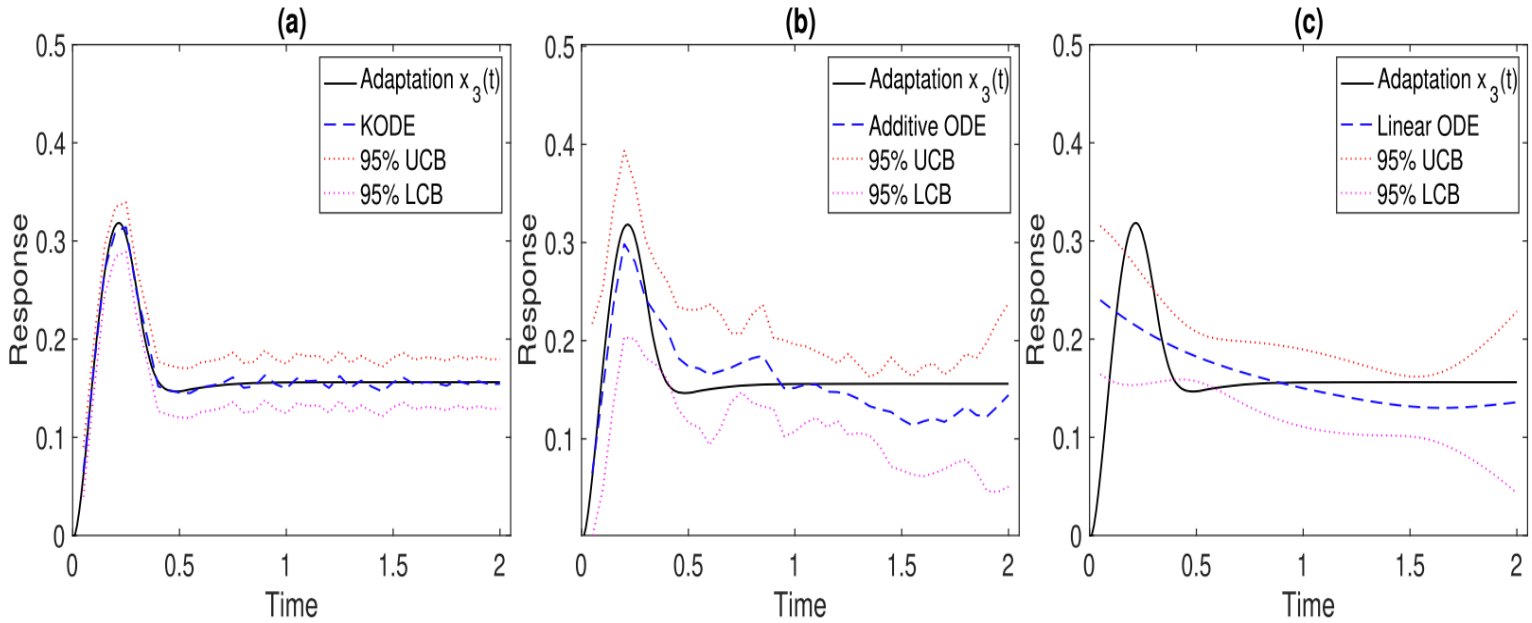


Figure 2. The true (black solid line) and the estimated (blue dashed line) trajectory of $x_3(t)$, with the 95% upper and lower confidence bounds (red dotted lines). The results are averaged over 500 data replications. (a) KODE; (b) additive ODE; (c) linear ODE.

Results

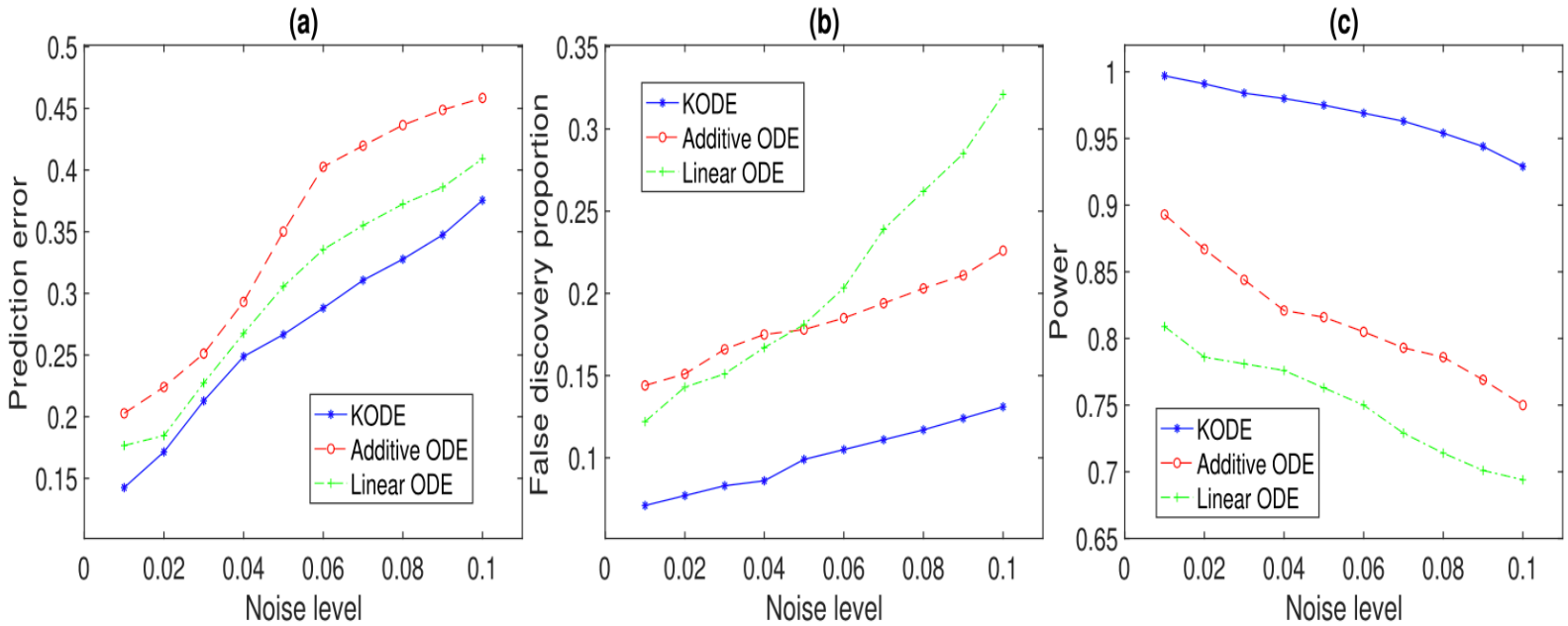


Figure 3. The prediction and selection performance of three ODE methods with varying noise level. The results are averaged over 500 data replications. (a) Prediction error; (b) false discovery proportion; (c) empirical power.

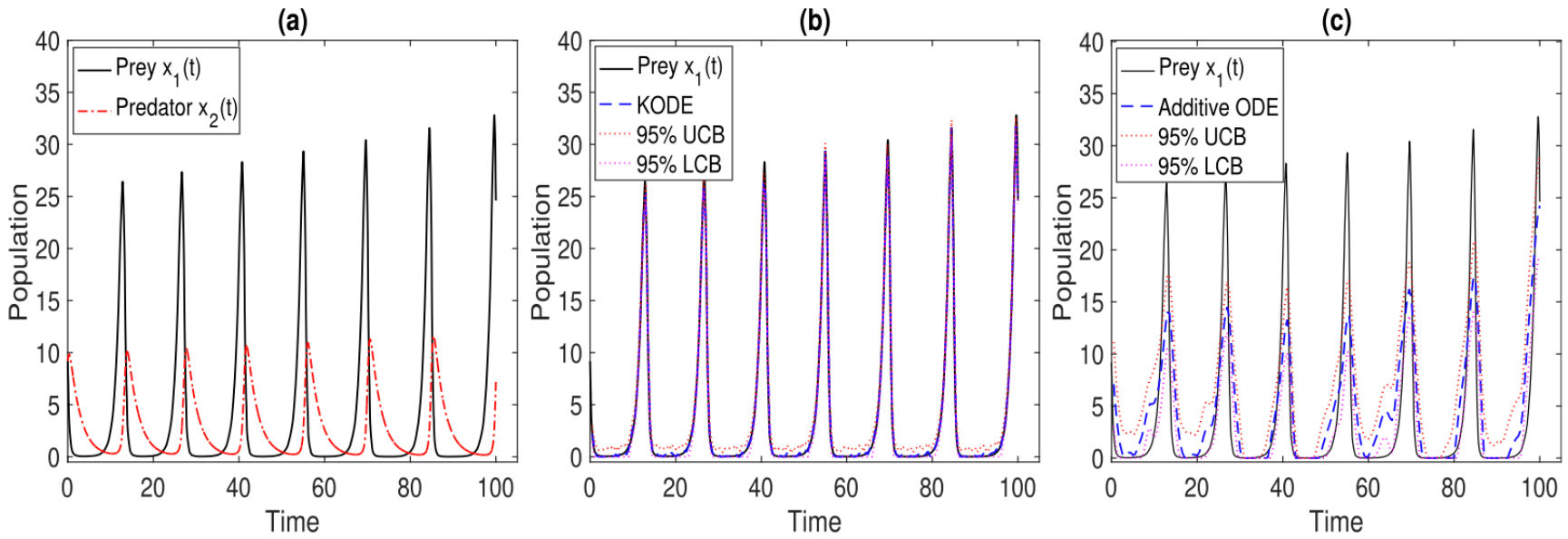
Simulation: Lotka-Volterra Equations

- Lotka-Volterra equations is high-dimensional, which are pairs of first-order nonlinear differential equations describing the dynamics of biological systems in which predators and prey interact.
- The ODE system is

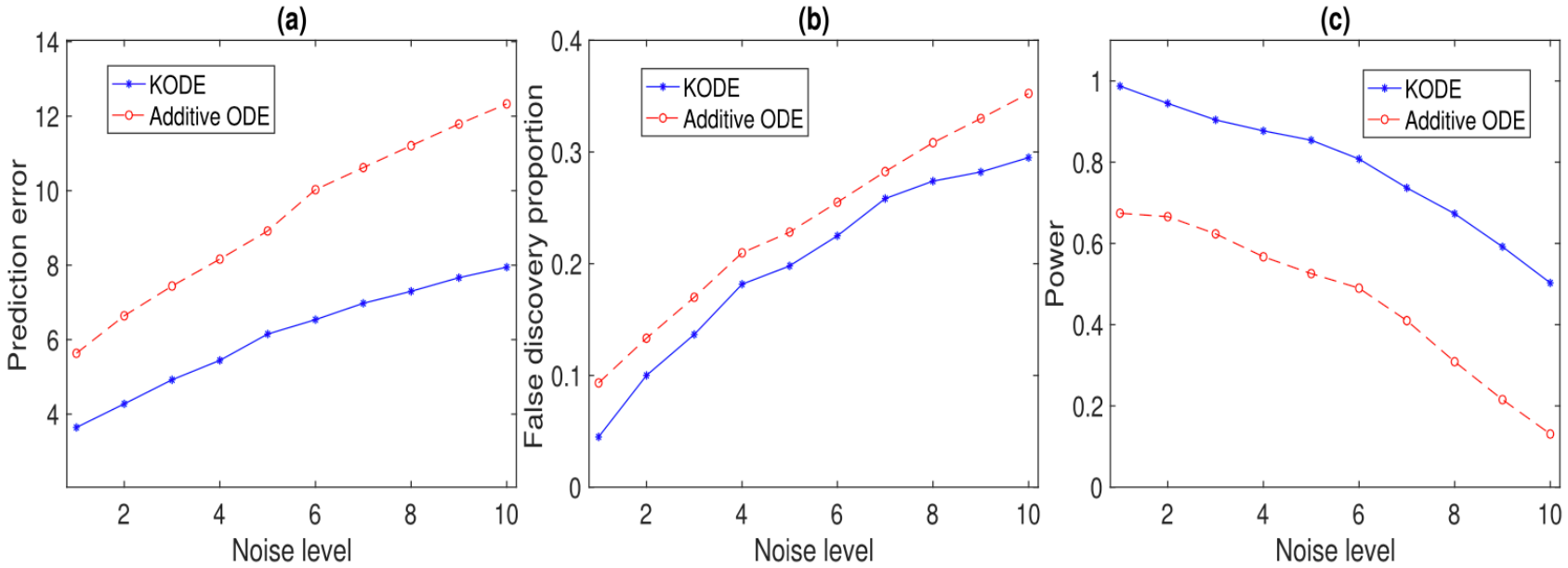
$$\frac{dx_{2j-1}(t)}{dt} = \alpha_{1,j}x_{2j-1}(t) - \alpha_{2,j}x_{2j-1}(t)x_{2j}(t)$$
$$\frac{dx_{2j}(t)}{dt} = \alpha_{3,j}x_{2j-1}(t)x_{2j}(t) - \alpha_{4,j}x_{2j}(t)$$

- The parameters $\alpha_{2,j}$ and $\alpha_{3,j}$ define the interaction between the two populations such that $dx_{2j-1}(t)/dt$ and $dx_{2j}(t)/dt$ are nonadditive functions of x_{2j-1} and x_{2j}, where x_{2j-1} is the prey and x_{2j} is the predator.

Results


(a)
Prey $x_1(t)$
Predator $x_2(t)$
Population
Time
(b)
Prey $x_1(t)$
KODE
95% UCB
95% LCB
Population
Time
(c)
Prey $x_1(t)$
Additive ODE
95% UCB
95% LCB
Population
Time

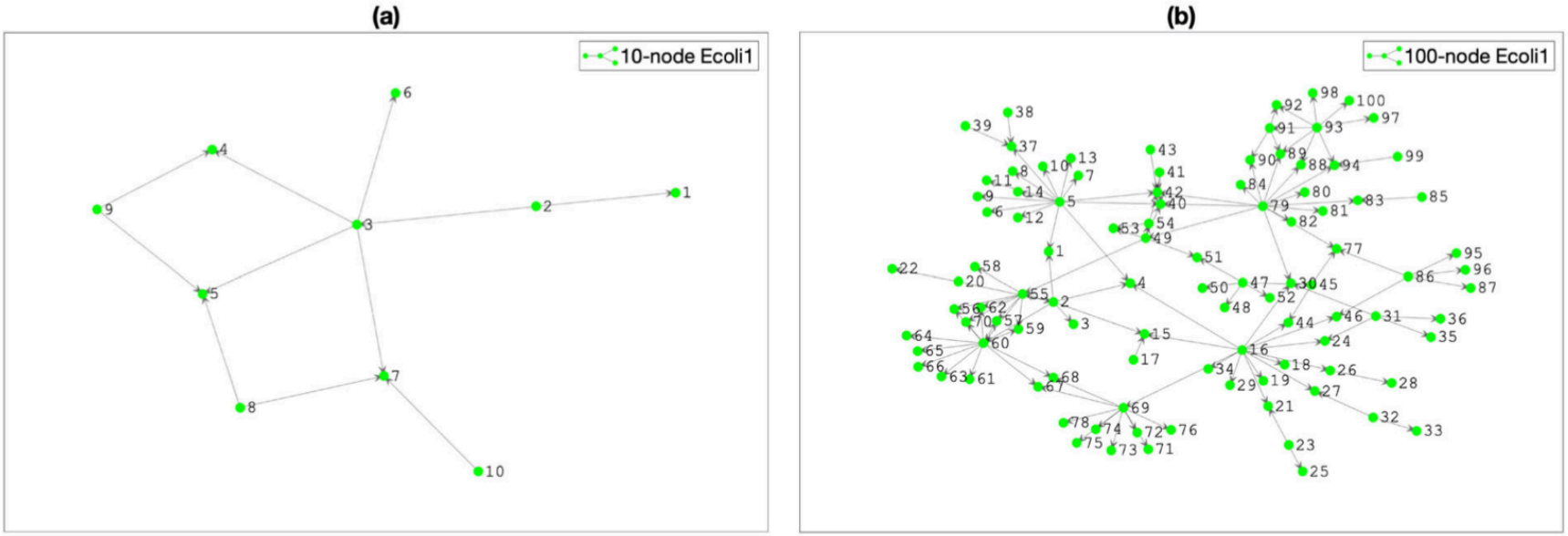
Results


(a)
Prediction error
Noise level
KODE
Additive ODE
(b)
False discovery proportion
Noise level
KODE
Additive ODE
(c)
Power
Noise level
KODE
Additive ODE

Application to Gene Regulatory Network

- GNW extracts two regulatory networks of E.coli (E.coli1, E.coli2), and three regulatory networks of yeast (yeast1, yeast2, yeast 3), each of which has two dimensions, $p = 10$ nodes and $p = 100$ nodes. This yields totally 10 combinations of network structures.
- The systems of ODEs for each extracted network are based on a thermodynamic approach, which leads to a nonadditive and nonlinear ODE structure.
- The network structures are sparse.

GRN data



GRN data

- For the 10-node network, GNW provides $R = 4$ perturbation experiments, and for the 100-node network, GNW provides $R = 46$ experiments. In each experiment, GNW generates the time-course data with different initial conditions of the ODE system to emulate the diversity of gene expression trajectories. All the trajectories are measured at n $= 21$ evenly spaced time points in [0, 1].
- We add independent measurement errors from a normal distribution with mean zero and standard deviation 0.025.

Method

We modify the KODE method, by seeking $F_j \in \mathcal{H}$ and $\theta_{j0} \in \mathbb{R}$ that minimize

$$\frac{1}{Rn}\sum_{r=1}^{R}\sum_{i=1}^{n}\left\{y_{ij}^{(r)} - \theta_{j0} - \int_0^{t_i} F_j\left(\widehat{x}^{(r)}(t)\right)dt\right\}^2$$
$$+\tau_{nj}\left(\sum_{k=1}^{p}\left\|\mathcal{P}^k F_j\right\|_{\mathcal{H}} + \sum_{k\neq l,k=1}^{p}\sum_{l=1}^{p}\left\|\mathcal{P}^{kl} F_j\right\|_{\mathcal{H}}\right)$$

where $\widehat{x}^{(r)}(t) = \left(\widehat{x}_1^{(r)}(t), \ldots, \widehat{x}_p^{(r)}(t)\right)^\top$ is the smoothing spline estimate obtained by,

$$\widehat{x}_j^{(r)}(t) = \arg\min_{z_j \in \mathcal{F}}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(y_{ij}^{(r)} - z_j(t_i)\right)^2 + \lambda_{nj}\|z_j(t)\|_{\mathcal{F}}^2\right\},$$
$$j = 1, \ldots, p, r = 1, \ldots, R.$$

Results

Table 1. The area under the ROC curve, and the 95% confidence interval, for 10 combinations of network structures from GNW.

	$p = 10$			$p = 100$		
	KODE	Additive ODE	Linear ODE	KODE	Additive ODE	Linear ODE
E.coli1	**0.582** (0.577, 0.587)	0.541 (0.535, 0.547)	0.460 (0.453, 0.467)	**0.711** (0.708, 0.714)	0.677 (0.672, 0.682)	0.640 (0.637, 0.643)
E.coli2	**0.662** (0.658, 0.666)	0.632 (0.625, 0.639)	0.562 (0.555, 0.569)	**0.685** (0.681, 0.689)	0.659 (0.652, 0.666)	0.533 (0.527, 0.539)
Yeast1	**0.603** (0.599, 0.607)	0.541 (0.536, 0.546)	0.436 (0.430, 0.442)	**0.619** (0.616, 0.622)	0.589 (0.581, 0.597)	0.569 (0.562, 0.576)
Yeast2	**0.599** (0.595, 0.603)	0.562 (0.555, 0.570)	0.536 (0.530, 0.542)	**0.606** (0.603, 0.609)	0.588 (0.582, 0.594)	0.541 (0.536, 0.546)
Yeast3	**0.612** (0.608, 0.616)	0.569 (0.564, 0.573)	0.487 (0.481, 0.493)	**0.621** (0.617, 0.625)	0.613 (0.607, 0.619)	0.609 (0.605, 0.613)

NOTE: The results are averaged over 100 data replications. Boldface indicates the method with larger AUC.

Conclusion

- In this article, we have developed a new reproducing kernel-based approach for a general family of ODE models that learn a dynamic system from noisy time-course data.
- We derive the post-selection confidence interval for the estimated signal trajectory.

Table of Contents

Sparse Identification of Nonlinear Dynamics (SINDy)

We consider dynamical systems (31) of the form

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

The vector $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the state of a system at time t, and the function $\mathbf{f}(\mathbf{x}(t))$ represents the dynamic constraints that define the equations of motion of the system, such as Newton's second law.

SINDy-State

Let $\dot{\mathbf{x}}(t)$ be the derivative of $\mathbf{x}(t)$. The data are sampled at several times $t_1, t_2, \cdots, t_m$ and arranged into two matrices:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \overset{\text{state}}{\overrightarrow{\begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix}}} \downarrow \text{time}$$

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

SINDy-Library

$\Theta(\mathbf{X})$ is a library, which may consist of candidate nonlinear functions of the columns of $\mathbf{X}$(constant, polynomial, and trigonometric terms). Here, higher polynomials are denoted as $\mathbf{X}^{P_2}, \mathbf{X}^{P_3}$, etc., where $\mathbf{X}^{P_2}$ denotes the quadratic nonlinearities in the state $\mathbf{x}$:

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

SINDy-Sparse regression

We believe that only a few of these nonlinearities are active in each row of $\mathbf{f}$, we may set up a sparse regression problem to determine the sparse vectors of coefficients $\Xi = [\xi_1 \xi_2 \cdots \xi_n]$ that determine which nonlinearities are active:

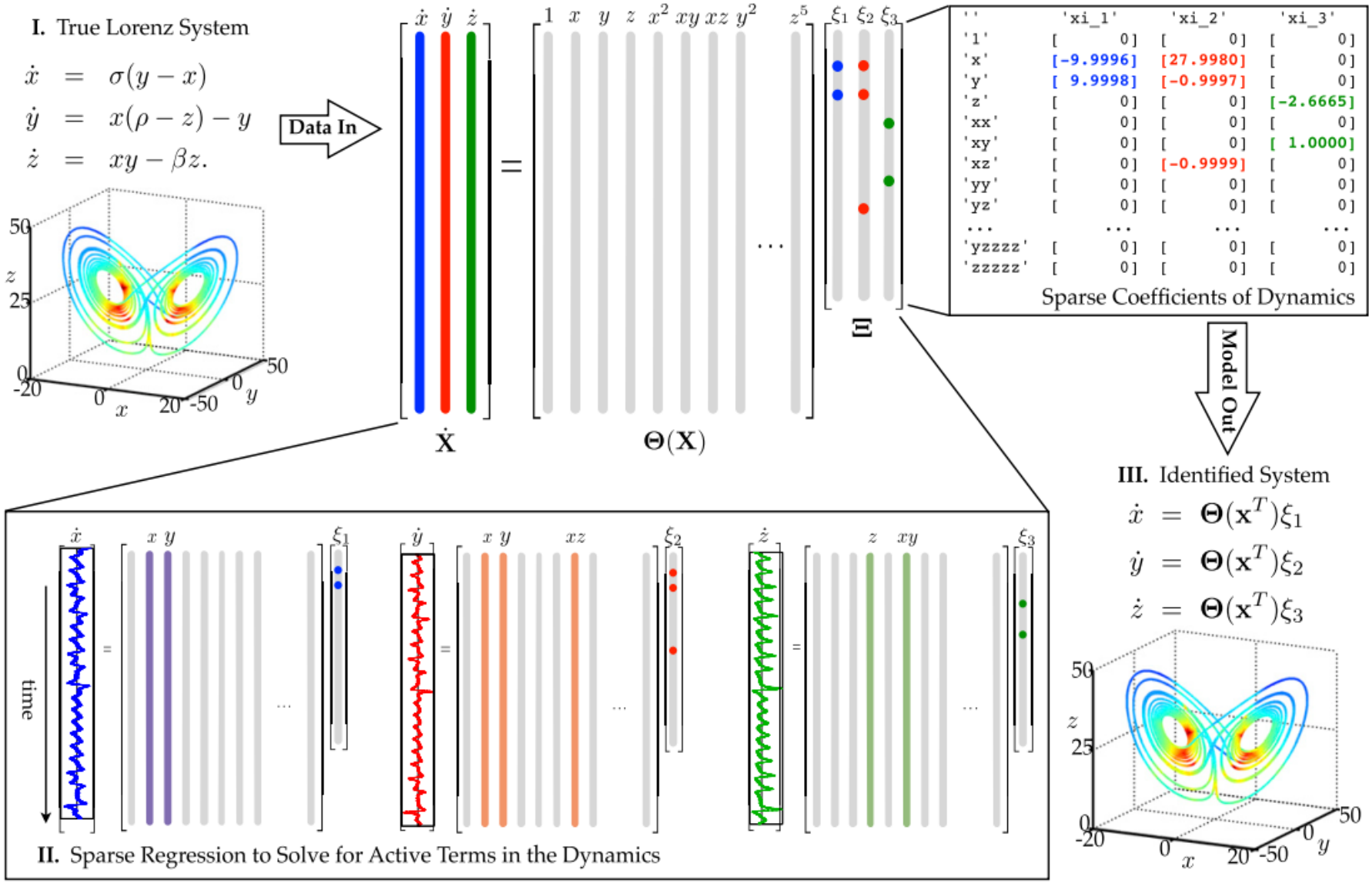
$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi$$

However, $\mathbf{X}$ and $\mathbf{X}$ are contaminated with noise,

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi + \eta\mathbf{Z},$$

where $\mathbf{Z}$ is modeled as a matrix of independent identically distributeld Gaussian entries with zero mean, and noise magnitude η.

SINDy



	xi_1	xi_2	xi_3
'1'	0	0	0
'x'	-9.9996	27.9980	0
'y'	9.9998	-0.9997	0
'z'	0	0	-2.6665
'xx'	0	0	0
'xy'	0	0	1.0000
'xz'	0	-0.9999	0
'yy'	0	0	0
'yz'	0	0	0
...	...	...	...
'yzzzz'	0	0	0
'zzzzz'	0	0	0

Algorithm

- The sequential thresholded least-squares algorithm is used to find Ξ, the tuning parameters are chosen by cross-validation.
- We use the total variation regularized derivative to de-noise the derivative.
- Obtaining a low-rank approximation using the singular value decomposition (SVD):

Chaotic Lorenz System

Consider a canonical model for chaotic dynamics, the Lorenz system:

$$\dot{x} = \sigma(y - x),$$
$$\dot{y} = x(\rho - z) - y,$$
$$\dot{z} = xy - \beta z.$$

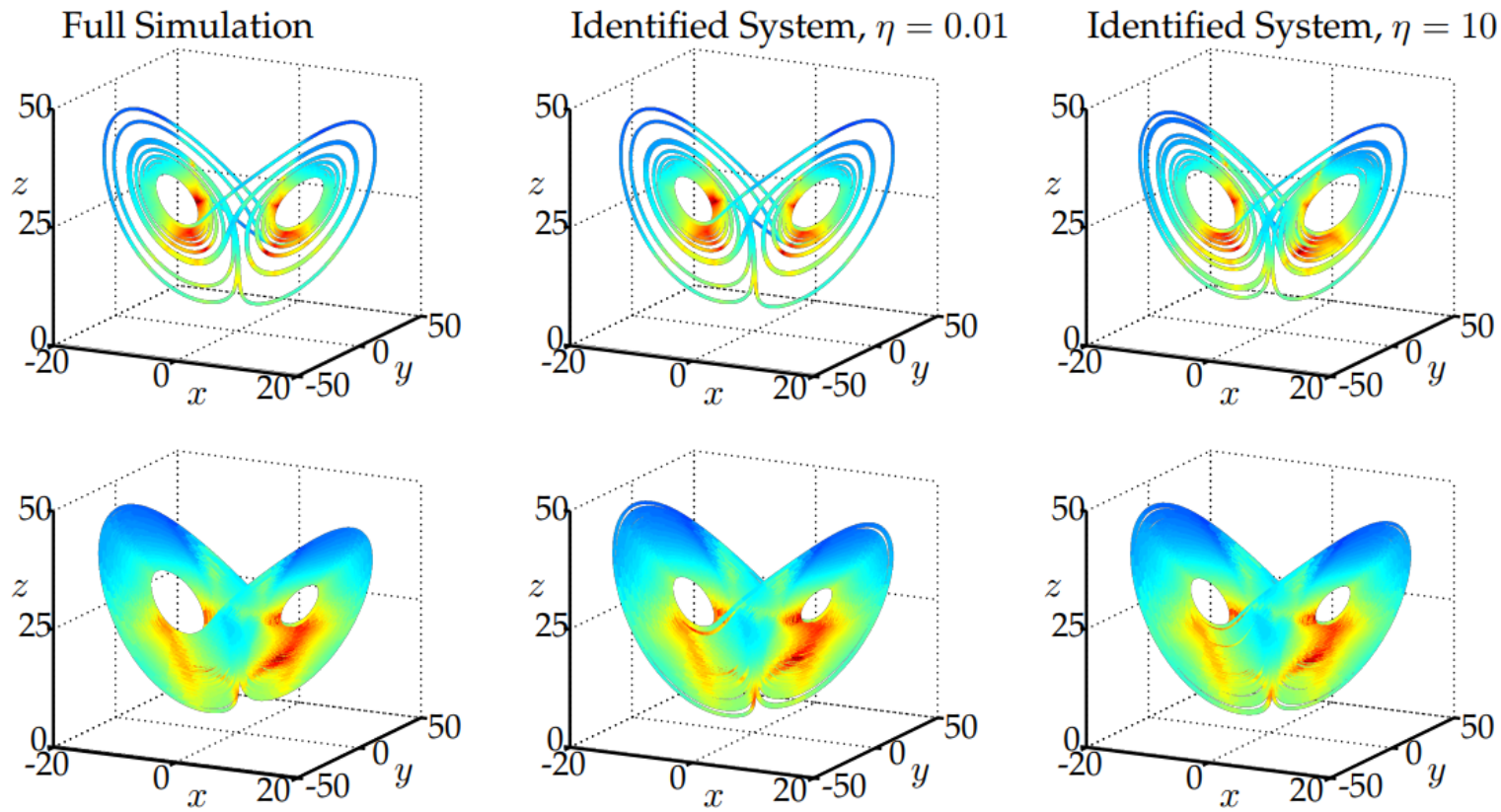
Chaotic Lorenz System

- Data are collected for the Lorenz system, and stacked into two large data matrices $\mathbf{X}$ and $\dot{\mathbf{X}}$, where each row of $\mathbf{X}$ is a snapshot of the state $\mathbf{x}$ in time, and each row of $\dot{\mathbf{X}}$ is a snapshot of the time derivative of the state $\dot{\mathbf{x}}$ in time.
- The dynamics are identified in the space of polynomials $\Theta(\mathbf{X})$ in (x, y, z) up to fifth order.

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & \cdots & | \\ x(t) & \mathbf{y}(t) & \mathbf{z}(t) & \mathbf{x}(t)^2 & \mathbf{x}(t)\mathbf{y}(t) & \cdots & \mathbf{z}(t)^5 \\ | & | & | & | & | & \cdots & | \end{bmatrix}.$$

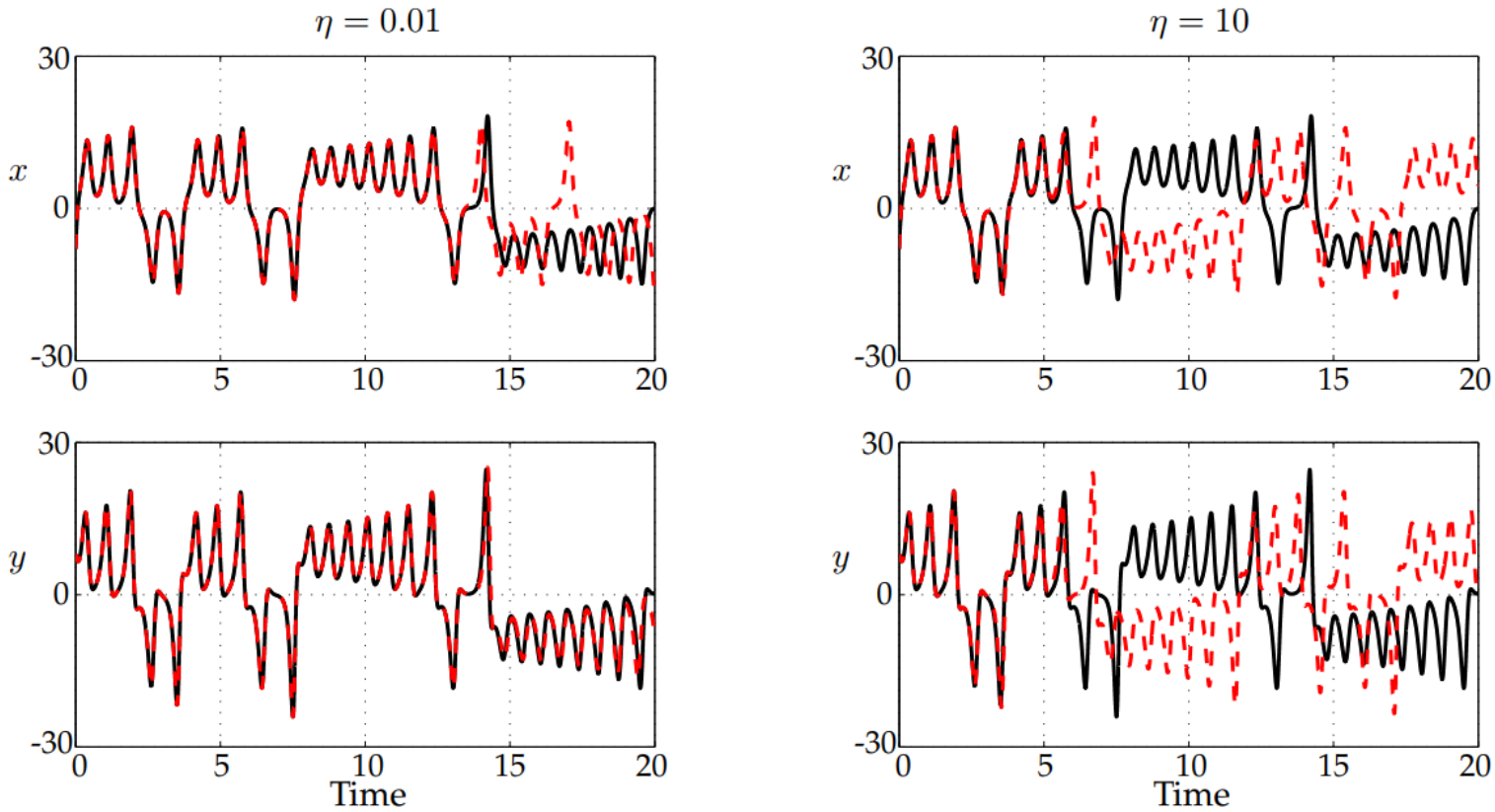
Chaotic Lorenz System

We add zero-mean Gaussian measurement noise with variance η to the exact derivatives.

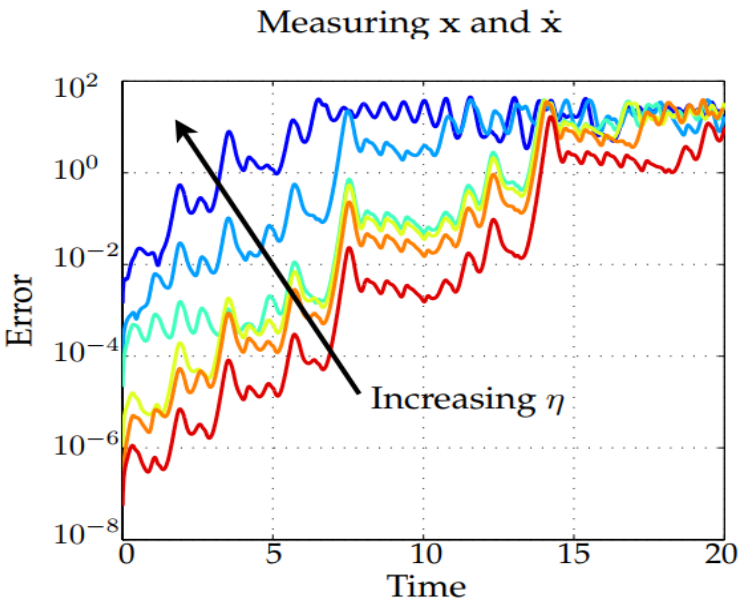


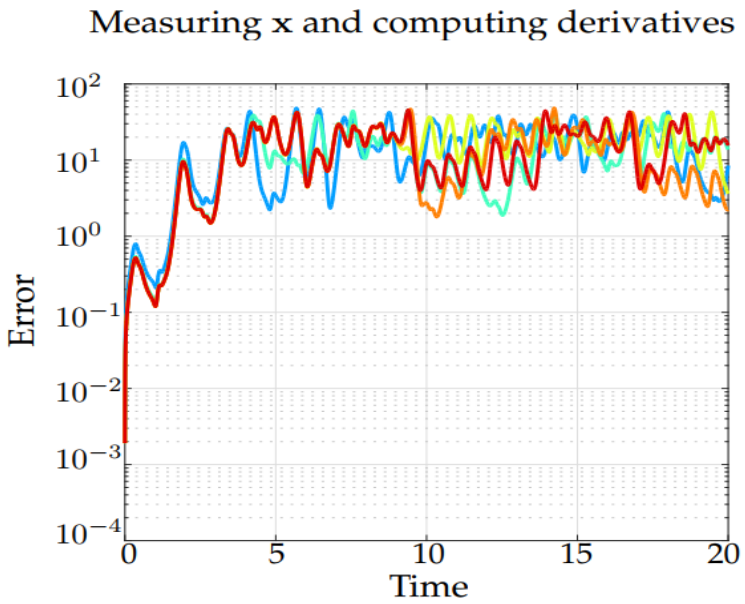
Dynamo view of trajectories of the Lorenz system

We add zero-mean Gaussian measurement noise with variance η to the exact derivatives.



Results





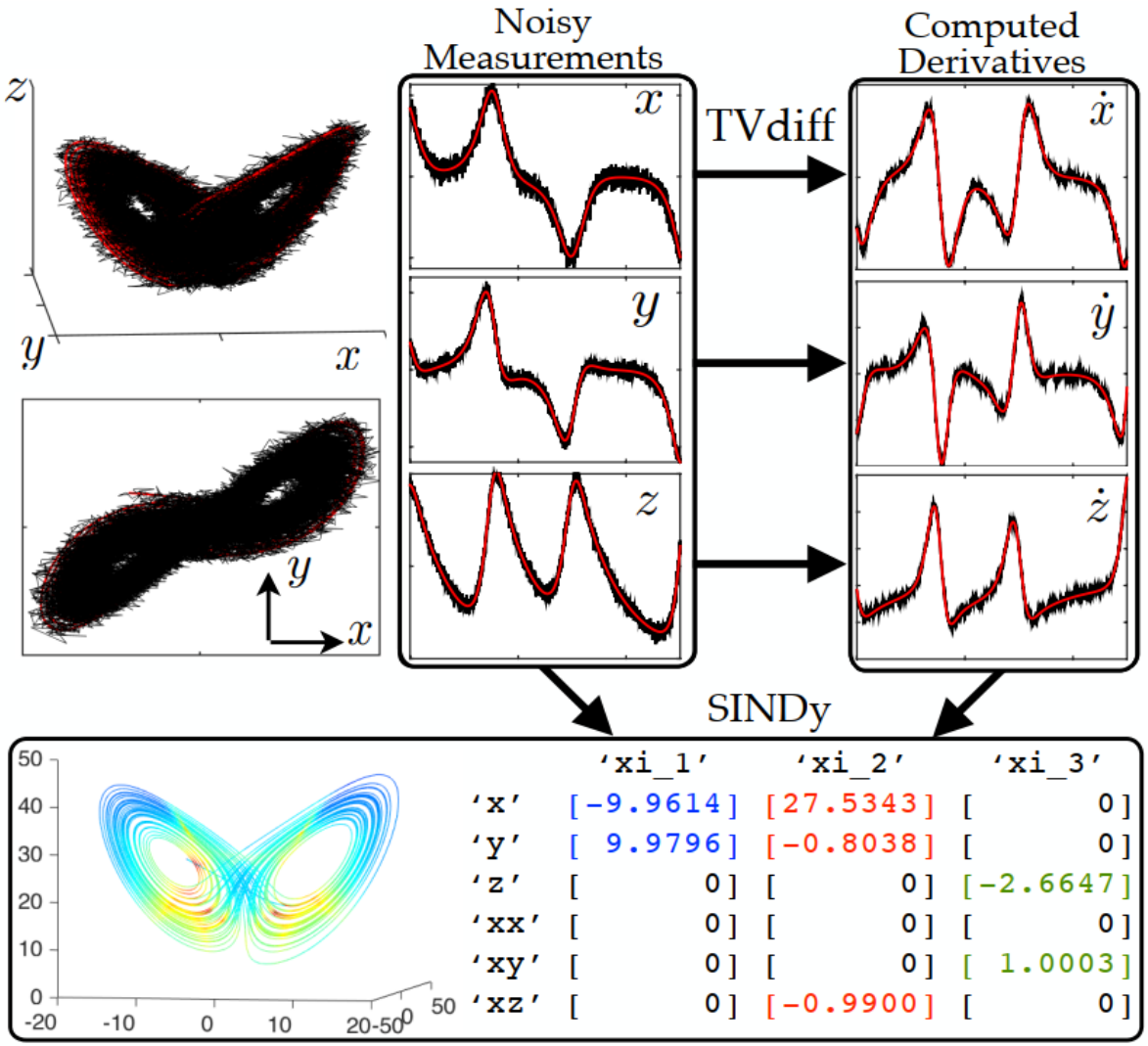
Although the ℓ_2 error increases for large noise values η, the form of the equations, and hence the attractor dynamics, are accurately captured. The system has a positive Lyapunov exponent, and small differences in model coefficients or initial conditions grow exponentially, until saturation, even though the attractor remains intact.

Chaotic Lorenz System

Next, we explore the SINDy algorithm on the Lorenz equation when only noisy measurements of the state $\mathbf{x}$ are available.

- Gaussian noise with variance η is added to the state $\mathbf{x}$, and derivatives $\dot{\mathbf{x}}$ are computed using the total-variation regularized derivative.
- This procedure is illustrated for a relatively large noise magnitude $\eta = 1.0$.
- Even for large noise magnitudes, the attractor dynamics are captured.

Results


Noisy
Measurements
Computed
Derivatives
TVdiff
SINDy
'xi_1' 'xi_2' 'xi_3'
'x' [-9.9614] [27.5343] [0]
'y' [9.9796] [-0.8038] [0]
'z' [0] [0] [-2.6647]
'xx' [0] [0] [0]
'xy' [0] [0] [1.0003]
'xz' [0] [-0.9900] [0]

Normal Forms, Bifurcations, and Parameterized Systems

In practice, many real-world systems depend on parameters, and dramatic changes, or bifurcations, may occur when the parameter is varied.

The normal forms associated with a bifurcation parameter μ:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \mu)$$
$$\dot{\mu} = 0.$$

It is then possible to identify the right hand side $\mathbf{f}(\mathbf{x}; \mu)$ as a sparse combination of functions of components in $\mathbf{x}$ as well as the bifurcation parameter μ.

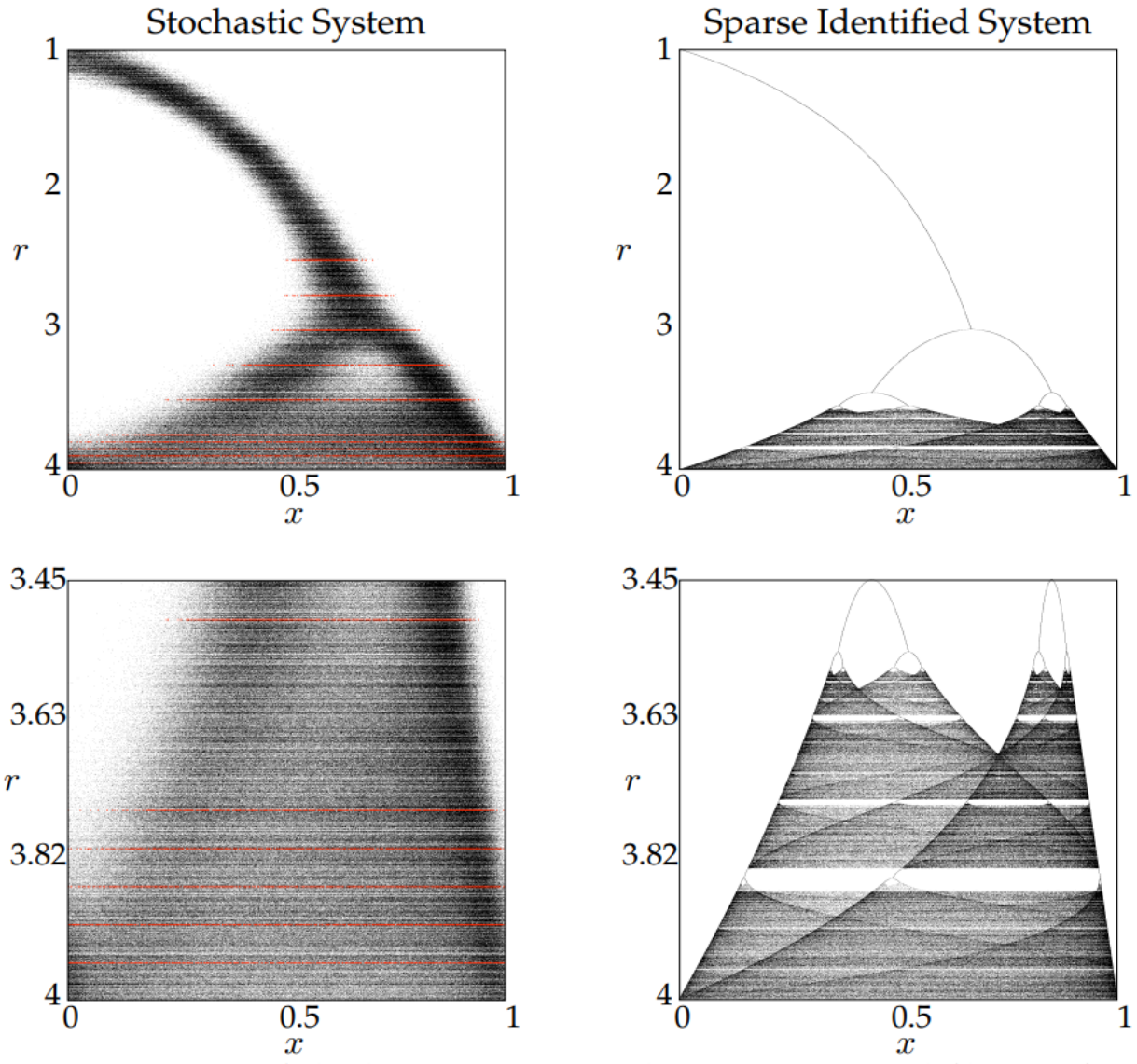
Logistic map

The logistic map is a classical model that exhibits a cascade of bifurcations, leading to chaotic trajectories. The dynamics with stochastic forcing η_k and parameter r are given by

$$x_{k+1} = rx_k(1 - x_k) + \eta_k.$$

Sampling the stochastic system at ten parameter values of r, the algorithm correctly identifies the underlying parameterized dynamics.

Results


Stochastic System
Sparse Identified System
1
2
3
4
r
0
0.5
1
x
3.45
3.63
3.82

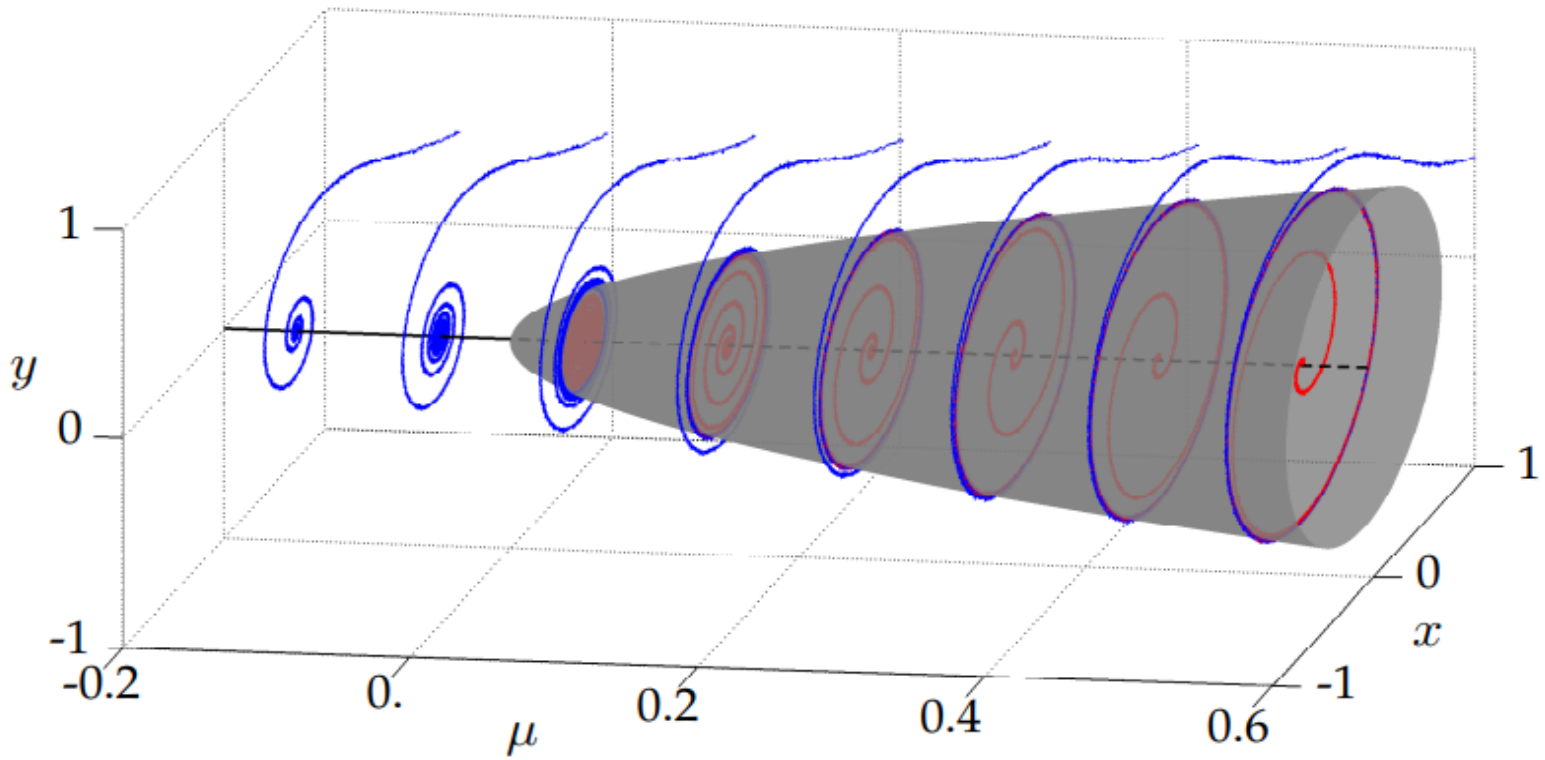
Hopf normal form

Noisy data is collected from the Hopf system

$$\dot{x} = \mu x + \omega y - Ax\left(x^2 + y^2\right)$$
$$\dot{y} = -\omega x + \mu y - Ay\left(x^2 + y^2\right)$$

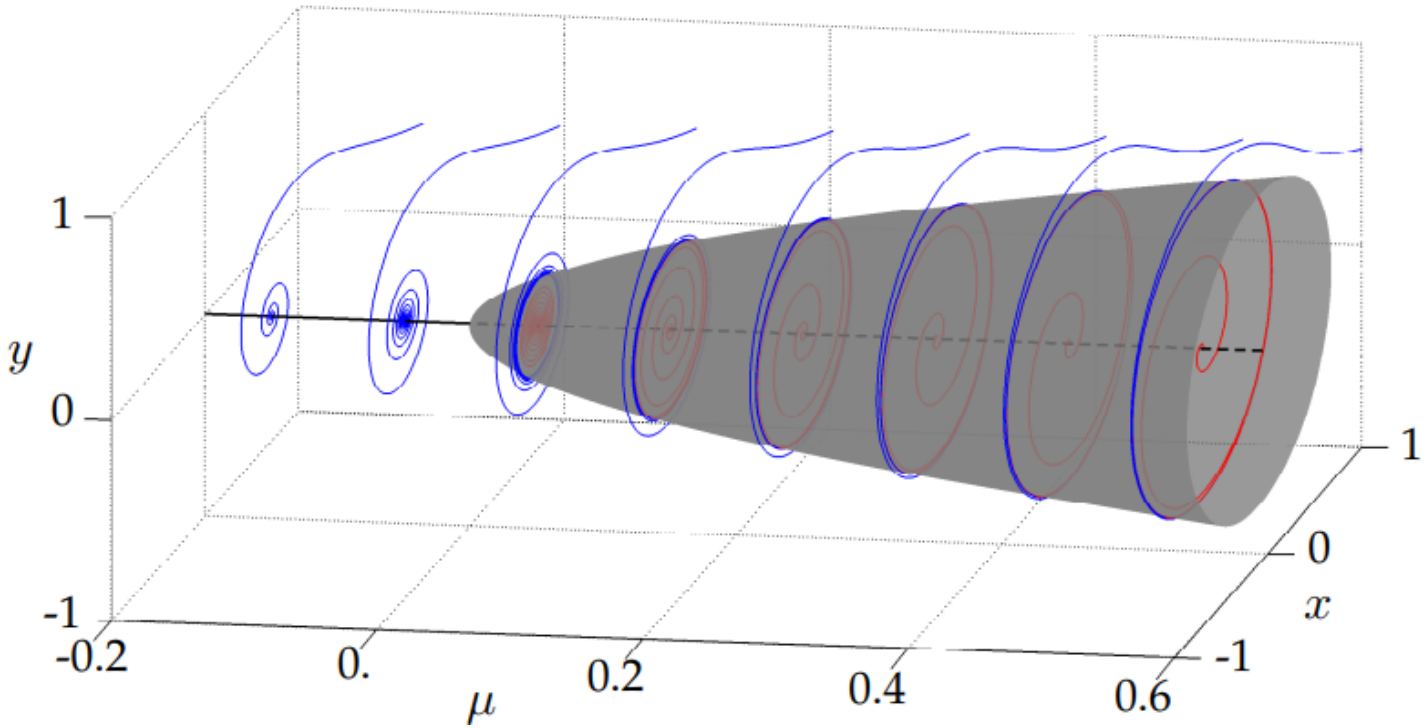
for various values of the parameter μ. The total variation derivative is used to de-noise the derivative for use in the algorithm.

Data



Result

Note that with noise in the training data, although the model terms are correctly identified, the actual values of the cubic terms are off by almost 8%. Collecting more training data or reducing the noise magnitude both improve the model agreement.



Conclusion

- We have demonstrated a powerful technique to identify nonlinear dynamical systems from data without assumptions on the form of the governing equations.
- We have demonstrated the robustness of the sparse dynamics algorithm to measurement noise and unavailability of derivative measurements in the Lorenz system, logistic map, and Hopf normal form examples.
- A significant outstanding issue in the above approach is the correct choice of measurement coordinates and the choice of sparsifying function basis for the dynamics.