Kernel Flows and Kernel Mode Decomposition for Learning Dynamical Systems from Data

Boumediene Hamzi,
Dept. of Computing and Mathematical Sciences, Caltech, CA, USA,
The Alan Turing Institute, London, UK.

with M. Darcy (Caltech), E. De Brouwer (KU Leuven), C. Kuehn (TUM), J. Lee (Caltech), R. Maulik (ANL), S. Mohamed (SUTD), H. Owhadi (Caltech), Y. Kevrekidis (JHU), L. Paillet (ENSTA), L. Yang & X. Sun & N. Xie (Nanjing Uni. of Aero. and Astro.).

Research Supported by the European Commission through the Marie Curie Fellowships Scheme, AFOSR (MURI), DoE, The Alan Turing Institute.

Epistemological context

How to analyze complex systems: Among the current approaches to analyze complex systems

- Theory of Dynamical Systems allows to analyze complex systems when the model is known. It offers nontrivial ways to analyze dynamical systems. It has the status of Theory. Currently, it is limited to low-dimensional models.
- Machine Learning is concerned with algorithms designed to accomplish a certain task, whose performance improves with the input of more data. It allows the analysis of some very high-dimensional complex systems on the basis of data when the model is not even known.
  Current limitations: Mostly a set of techniques and algorithms. No Methodologies. Theory still underdeveloped. It is not clear why the algorithms work and what is their domain of applicability.

$\rightsquigarrow$ It makes sense to combine Dynamical Systems and Machine Learning.

Goal

Goal: Fill the gap between Machine Learning and Dynamical Systems in the following directions

- Machine Learning for Dynamical Systems: how to analyze dynamical systems on the basis of observed data rather than attempt to study them analytically (it allows to extend the boundaries of the classical theory of dynamical systems).
- Dynamical Systems for Machine Learning: how to analyze algorithms of Machine Learning using tools from the theory of dynamical systems (allows to give solid foundations to the existing methods and understand their true potential and limits- identify the domain of applicability of the algorithms in ML).

Context

As pointed out by Steve Smale, the interaction between Dynamical Systems and Learning Theory is an important problem[1]:

> *"Some years ago, Felipe (Cucker) and I were trying to find something about the brain science and artificial intelligence starting from literature on neural nets. It was in this setting that we encountered the beautiful ideas and fast algorithms of learning theory. Eventually we were motivated to write on the mathematical foundations of this new area of science.*
> *I have found this arena to, with its new challenges and growing number of applications, be exciting. For example, the unification of dynamical systems and learning theory is a major problem. Another problem is to develop a comparative study of useful algorithms currently available and to give unity to these algorithms."*

[1] Felipe Cucker and Ding Xuan Zhou (2007), Learning Theory: An Approximation Theory Viewpoint.

Context

"Personal computing has developed to the point where in many cases it ought to be easier to simulate a dynamical system and analyze the empirical data, rather than attempt to study the system analytically. Indeed, for large classes of nonlinear systems, numerical analysis may be the only viable option. Yet the mathematical theory necessary to analyze dynamical systems on the basis of observed data is still largely underdeveloped."

J. Bouvrie and BH (2012), Empirical Estimators for Stochastically Forced Nonlinear Systems: Observability, Controllability and the Invariant Measure, `https://arxiv.org/pdf/1204.0563v1.pdf`

Outline

- Elements of Learning Theory and Function Approximation in RKHSs
- Probability Measures in RKHSs and the Maximum Mean Discrepancy
- Kernel Flows for Learning Chaotic Dynamical Systems: Parametric Kernel Flows, NonParametric Kernel Flows, Irregular Observations, Partial Observations, Sparse Kernel Flows, Hausdorff Metric based Kernel Flows.
- Learning and Detection of Critical Transitions for some Slow-Fast SDEs

Summary of the Approach

• We assume that there is a $\phi : \mathbb{R}^n \to \mathcal{H}; x \mapsto z$ where $\mathcal{H}$ is an RKHS such that we can perform an analysis (in general, but not necessarily, a linear analysis) in $\mathcal{H}$ then come back to $\mathbb{R}^n$.
• The transformation ϕ is obtained from the kernel that defines the RKHS (in general, it is not necessary to explicitly find ϕ). In practice, we will use $\phi(x) = [\phi_1(x) \cdots \phi_N(x)]^T$ with

$$\phi_i(x) = K(x, x(t_i))$$

where K is a reproducing kernel and $x(t_i)$ are measurements at time t_i, $i = 1, \cdots, N$ and $N \gg n$.
• Measurements/Data are used to construct the Hilbert Space where computations become "simpler".

Reproducing Kernel Hilbert Spaces

• Historical Context: Appeared in the 1930s as an answer to the question: when is it possible to embed a metric space into a Hilbert space ? (Schoenberg, 1937)
• Answer: If the metric satisfies certain conditions, it is possible to embed a metric space into a special type of Hilbert spaces called RKHSs.
• Properties of RKHSs have been further studied in the 1950s and later (Aronszajn, 1950; Schwartz, 1964; Wahba, 1990s; Smale, 2000s etc.)

Reproducing Kernel Hilbert Spaces

• Definition: A Hilbert Space is an inner product space that is complete and separable with respect to the norm defined by the inner product.

• Definition: For a compact $\mathcal{X} \subseteq \mathbb{R}^d$, and a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathbb{R}$, we say that $\mathcal{H}$ is a RKHS if there exists $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

i. k has the reproducing property, i.e. $\forall f \in \mathcal{H}$, $f(x) = \langle f(\cdot), k(\cdot, x)\rangle$ (k is called a reproducing kernel).

ii. k spans $\mathcal{H}$, i.e. $\mathcal{H} = \overline{\text{span}\{k(x, \cdot) | x \in \mathcal{X}\}}$.

• Definition: A Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space H with a reproducing kernel whose span is dense in H. Equivalently, a RKHS is a Hilbert space of functions where all evaluation functionals are bounded and linear.

Reproducing Kernel Hilbert Spaces

The important properties of reproducing kernels are
• The RKHS is unique.
• $\forall x, y \in \mathcal{X}$, $K(x, y) = K(y, x)$ (symmetry).
• $\sum_{i,j=1}^{m} \alpha_i \alpha_j K(x_i, x_j) \geq 0$ for $\alpha_i \in \mathbb{R}$ and $x_i \in \mathcal{X}$ (positive definitness).
• $\langle K(x, \cdot), K(y, \cdot)\rangle_{\mathcal{H}} = K(x, y)$. Using this property, one can immediately get the canonical feature map (Aronszajn's feature map): $\Phi_c(x) = K(x, \cdot)$.

• A Mercer kernel is a continuous positive definite kernel.
• The fact that Mercer kernels are positive definite and symmetric reminds us of similar properties of Gramians and covariance matrices. This is an essential fact that we are going to use in the following.
• Examples of kernels: $k(x, x') = \langle x, x'\rangle^d$, $k(x, x') = \exp\left(-\frac{||x-x'||_2^2}{2\sigma^2}\right)$, $k(x, x') = \tanh(\kappa\langle x, x'\rangle + \theta)$.

RKHS in Approximation Theory (aka Learning Theory)

• RKHS play an important role in learning theory whose objective is to find an unknown function $f \colon X \to Y$ from random samples $(x_i, y_i)|_{i=1}^m$.
• For instance, assume that the random probability measure that governs the random samples is ρ and is defined on $Z := X \times Y$. Let X be a compact subset of $\mathbb{R}^n$ and $Y = \mathbb{R}$. If we define the least square error of f as $\mathcal{E} = \int_{X\times Y}(f(x) - y)^2 d\rho$, then the function that minimizes the error is the regression function f_ρ defined as

$$f_\rho(x) = \int_{\mathbb{R}} y d\rho(y|x), \quad x \in X,$$

where $\rho(y|x)$ is the conditional probability measure on $\mathbb{R}$.

RKHS in Approximation Theory (aka Learning Theory)

• Since ρ is unknown, neither f_ρ nor $\mathcal{E}$ is computable. We only have the samples $\mathbf{s} := (x_i, y_i)|_{i=1}^m$. The error $\mathcal{E}$ is approximated by the empirical error $\mathcal{E}_{\mathbf{s}}(f)$ by

$$\mathcal{E}_{\mathbf{s}}(f) = \frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2 + \lambda||f||_{\mathcal{H}}^2,$$

for $\lambda \geq 0$, λ plays the role of a regularization parameter.

RKHS in Approximation Theory (aka Learning Theory)

• In learning theory, the minimization is taken over functions from a hypothesis space often taken to be a ball of a RKHS $\mathcal{H}_K$ associated to a kernel K, and the function $f_{\mathbf{s}}$ that minimizes the empirical error $\mathcal{E}_{\mathbf{s}}$ is

$$f_{\mathbf{s}}(x) = \sum_{j=1}^{m} c_j K(x, x_j) = \sum_{j=1}^{m} c_j \phi_j(x),$$

where the coefficients $(c_j)_{j=1}^{m}$ are obtained by solving the linear system

$$\lambda\, m\, c_i + \sum_{j=1}^{m} K(x_i, x_j) c_j = y_i, \quad i = 1, \cdots m,$$

and $f_{\mathbf{s}}$ is taken as an approximation of the regression function f_ρ.
• We call *learning* the process of approximating the unknown function f from random samples on Z.

RKHS in Change Point Detection

- We will consider a sequence of samples $x_1, x_2, \cdots, x_n$ from a domain $\mathcal{X}$.
- We are interested in detecting a possible change-point τ, such that before τ, the samples $x_i \sim P$ i.i.d for $i \leq \tau$, where P is the so-called background distribution, and after the change-point, the samples $x_i \sim Q$ i.i.d for $i \geq \tau + 1$, where Q is a post-change distribution.
- We map the dataset in an RKHS $\mathcal{H}$ then compute a measure of discrepancy Δ_n.
- Δ_n is small if $P = Q$ and large if P and Q are far apart.
- We will use the maximum mean discrepancy (MMD)

$$\text{MMD}[\mathcal{H}, P, Q] := \sup_{f \in \mathcal{H}, ||f|| \leq 1} \{\mathbb{E}_x[f(x)] - \mathbb{E}_y[f(y)]\},$$

as a measure of heteregoneity.

Probability Measures in RKHSes

• Let $\mathcal{H}$ be an RKHS on the separable metric space $\mathcal{X}$, with a continuous feature mapping $\phi : \mathcal{X} \to \mathcal{H}$. Assume that k is bounded, i.e. $\sup_{\mathcal{X}} k(x, x) < \infty$.
• Let $\mathcal{P}$ be the set of Borel probability measures on $\mathcal{X}$. We define the mapping to $\mathcal{H}$ of $P \in \mathcal{P}$ as the expectation of $\phi(x)$ with respect to P, i.e.

$$\begin{aligned} \mu_P : \mathcal{P} &\to \mathcal{H} \\ P &\mapsto \int_{\mathcal{X}} \phi(x) dP(x) =: \mu_k(P) \quad \text{(kernel mean embedding of P)} \end{aligned}$$

• The maximum mean discrepancy (MMD) between two probability measures P and Q is defined as the distance between two such mappings

$$MMD(P, Q) = ||\mu_k(P) - \mu_k(Q)||_{\mathcal{H}_k}$$

Probability Measures in RKHSes

• The maximum mean discrepancy (MMD) is defined as (Gretton et al., 2007)

$$\begin{aligned} \text{MMD}(P,Q) &:= ||\mu_P - \mu_Q||_{\mathcal{H}}, \\ &= \left(\mathbb{E}_{x,x'}(k(x,x')) + \mathbb{E}_{y,y'}(k(y,y')) - 2\mathbb{E}_{x,y}(k(x,y))\right)^{\frac{1}{2}} \end{aligned}$$

where x and x' are independent random variables drawn according to P, y and y' are independent random variables drawn according to Q, and x is independent of y.

• This quantity is a pseudo-metric on distributions, i.e. it satisfies all the qualities of a metric except $\text{MMD}(P,Q) = 0$ iff $P = Q$.

• For the MMD to be a metric, it is sufficient that the kernel is characteristic, i.e. the map $\mu_P : \mathcal{P} \to \mathcal{H}$ is injective. This is satisfied by the Gaussian kernel (both on compact domains and on $\mathbb{R}^d$) for example.

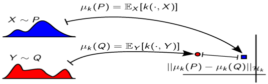
Probability Measures in RKHSes

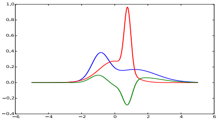
- **RKHS embedding**:

$$P \to \mu_k(P) = \mathbb{E}_{X\sim P}k(\cdot, X) \in \mathcal{H}_k$$

$$P \to [\mathbb{E}\varphi_1(X), \cdots, \mathbb{E}\varphi_s(X)] \in \mathbb{R}^s$$

- **Maximum Mean Discrepancy (MMD)** [Borgwardt et al, 2006; Gretton et al, 2007] between P and Q:





$$\text{MMD}_k(P, Q) = \|\mu_k(P) - \mu_k(Q)\|_{\mathcal{H}_k} = \sup_{f\in\mathcal{H}_k:\, \|f\|_{\mathcal{H}_k}\le 1} |\mathbb{E}f(X) - \mathbb{E}f(Y)|$$

Probability Measures in RKHSes

• For characteristic kernels, the MMD metrizes the weak- $\star$ topology on probability measures

$$\text{MMD}_k(P_n, P) \to 0 \Leftrightarrow P_n \rightsquigarrow P$$

• For characteristic kernels: convergence in distribution iff convergence in MMD.
• It is an Integral Probability Metric that can be computed directly from data without having to estimate the density as an intermediate step.
• Given two i.i.d samples $(x_1, \cdots, x_m)$ from P and $(y_1, \cdots, y_m)$ from Q, an unbiased estimate of the MMD is

$$\text{MMD}_u^2 := \frac{1}{m(m-1)} \sum_{i \neq j}^{m} [k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i)]$$

Kernel Flows for Learning Chaotic Dynamical Systems

Kernel Flows for Learning Chaotic Dynamical Systems

- Problem **P** : Given input/output data $(x_1, y_1), \cdots, (x_N, y_N) \in \mathcal{X} \times \mathbb{R}$, recover an unknown function u^* mapping $\mathcal{X}$ to $\mathbb{R}$ such that $u^*(x_i) = y_i$ for $i \in \{1, ..., N\}$.
- In the setting of optimal recovery, Problem **P** can be turned into a well posed problem by restricting candidates for u to belong to a Banach space of functions $\mathcal{B}$ endowed with a norm defined as

$$||u||^2 = \sup_{\phi \in \mathcal{B}^*} \frac{(\int \phi(x) u(x) dx)^2}{(\int \phi(x) K(x,y) \phi(y) dx dy)}$$

and identifying the optimal recovery as the minimizer of the relative error

$$\min_v \max_u \frac{||u - v||^2}{||u||^2},$$

where the max is taken over $u \in \mathcal{B}$ and the min is taken over candidates in $v \in \mathcal{B}$ such that $v(x_i) = u(x_i) = y_i$.

Kernel Flows for Learning Chaotic Dynamical Systems

• The method of KFs is based on the premise that *a kernel is good if there is no significant loss in accuracy in the prediction error if the number of data points is halved*. This led to the introduction of

$$\rho = \frac{||v^* - v^s||^2}{||v^*||^2}$$

which is the relative error between v^*, the optimal recovery of u^* based on the full dataset $X = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, and v^s the optimal recovery of both u^* and v^* based on half of the dataset $X^s = \{(x_i, y_i) \mid i \in \mathcal{S}\}$ $(\mathrm{Card}(\mathcal{S}) = N/2)$ which admits the representation

$$v^s = (y^s)^T A^s K(x^s, \cdot)$$

with $y^s = \{y_i \mid i \in \mathcal{S}\}$, $x^s = \{x_i \mid i \in \mathcal{S}\}$, $A^s = (\Theta^s)^{-1}$, $\Theta^s_{i,j} = K(x^s_i, x^s_j)$.

Kernel Flows for Learning Chaotic Dynamical Systems

Given a family of kernels $K_\theta(x, x')$ parameterized by θ, the KF algorithm can then be described as follows :

1. Select random subvectors X^b and Y^b of X and Y (through uniform sampling without replacement in the index set $\{1, \dots, N\}$)
2. Select random subvectors X^c and Y^c of X^b and Y^b (by selecting, at random, uniformly and without replacement, half of the indices defining X^b)
3. Let

$$\rho(\theta, X^b, Y^b, X^c, Y^c) := 1 - \frac{Y^{c,T} K_\theta(X^c, X^c)^{-1} Y_c}{Y^{f,T} K_\theta(X^b, X^b)^{-1} Y^b},$$

   be the squared relative error (in the RKHS norm $\|\cdot\|_{K_\theta}$ defined by K_θ) between the interpolants u^b and u^c obtained from the two nested subsets of the dataset and the kernel K_θ
4. Evolve θ in the gradient descent direction of ρ, i.e. $\theta \leftarrow \theta - \delta \nabla_\theta \rho$
5. Repeat.

Kernel Flows for Learning Chaotic Dynamical Systems

- Let $x_1, \ldots, x_k, \ldots$ be a time series in $\mathbb{R}^d$. Our goal is to forecast x_{n+1} given the observation of $x_1, \ldots, x_n$.
- We work under the assumption that this time series can be approximated by a solution of a dynamical system of the form

$$z_{k+1} = f^\dagger(z_k, \ldots, z_{k-\tau^\dagger+1}),$$

where $\tau^\dagger \in \mathbb{N}^*$ and $f^\dagger$ may be unknown.
- Given $\tau \in \mathbb{N}^*$, the approximation of the dynamical can then be recast as that of interpolating $f^\dagger$ from pointwise measurements

$$f^\dagger(X_k) = Y_k \text{ for } k = 1, \ldots, N$$

with $X_k := (x_{k+\tau-1}, \ldots, x_k)$, $Y_k := x_{k+\tau}$ and $N = n - \tau$.

Kernel Flows for Learning Chaotic Dynamical Systems

• Given a reproducing kernel Hilbert space of candidates for $f^\dagger$, and using the relative error in the RKHS norm $\|\cdot\|$ as a loss, the regression of the data (X_k, Y_k) with the kernel K associated with provides a minimax optimal approximation of $f^\dagger$ in . This interpolant (in the absence of measurement noise) is

$$f(x) = K(x, X)(K(X, X))^{-1} Y$$

where $X = (X_1, \dots, X_N)$, $Y = (Y_1, \dots, Y_N)$, $k(X, X)$ for the $N \times N$ matrix with entries $k(X_i, X_i)$, and $k(x, X)$ is the N vector with entries $k(x, X_i)$.
• Use different variants of Kernel Flows (KF) to learn the kernel K from the data (X_k, Y_k).

Kernel Flows for Learning Chaotic Dynamical Systems

Assume the kernel K to be parameterized by θ. To update θ in K_θ, we minimize one of the following metrics (different variants of KFs)

- Metric associated to the RKHS norm

$$\rho(\theta, X^b, Y^b, X^c, Y^c) := 1 - \frac{Y^{c,T} K_\theta(X^c, X^c)^{-1} Y_c}{Y^{f,T} K_\theta(X^b, X^b)^{-1} Y^b}$$

- Metric associated to Lyapunov exponents and minimize

$$\rho_L = |\lambda_{\max,N} - \lambda_{\max,N/2}|$$

- Metric associated to the Maximum Mean Discrepancy (MMD) and minimize

$$\rho_{\text{MMD}} = \text{MMD}(S_1, S_2)$$

between two different samples of the time series.

- Metric associated to the Hausdorff distance and minimize

$$\rho_{\text{HD}} = \text{HD}(\mathcal{A}_N, \mathcal{A}_{N/2})$$

- We use the kernel

$$
\begin{aligned}
k(x, y) &= \alpha_0 \max\{0, 1 - \frac{||x-y||_2^2|}{\sigma_0}\} + \alpha_1 \, e^{\frac{||x-y||_2^2}{\sigma_1^2}} + \alpha_2 e^{-\frac{||x-y||_2}{\sigma_2^2}} \\
&+ \alpha_3 e^{-\sigma_3 \sin^2(\sigma_4 \pi ||x-y||_2)} e^{-\frac{||x-y||_2^2}{\sigma_5^2}} + \alpha_4 ||x-y||_2^2
\end{aligned}
$$

Kernel Flows for Learning Chaotic Dynamical Systems

- Bernoulli map $x(k+1) = 2x(k) \bmod 1$

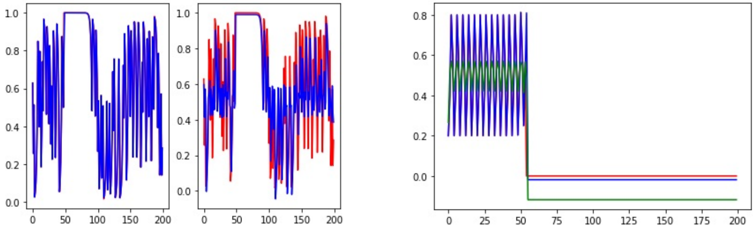


Figure: Time series generated by the true dynamics, approximation using the learned kernel and the kernel without learning for different initial conditions

Kernel Flows for Learning Chaotic Dynamical Systems

- Lorenz system

$$\begin{aligned} \frac{dx}{dt} &= s(y-x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= xy - bz \end{aligned}$$

with $s = 10,\ r = 28,\ b = 10/3.$

Kernel Flows for Learning Chaotic Dynamical Systems

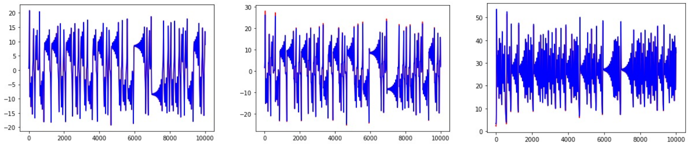


Figure: Time series generated by the true dynamics (red) and the approximation with the learned kernel (blue) - x component in the left figure, y component in the middle figure, z component in the right figure.

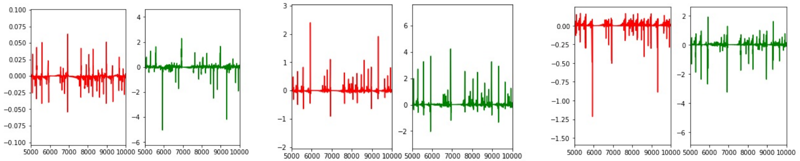


Figure: Difference between the true and the approximated dynamics with the learned kernel using ρ (red (first, third and fifth from the left)), with the initial kernel (green (second, fourth and sixth from the left)). x-component in the two figures at the left, y-component in the middle two figures, z-component in the right two figures.

Kernel Flows for Learning Chaotic Dynamical Systems

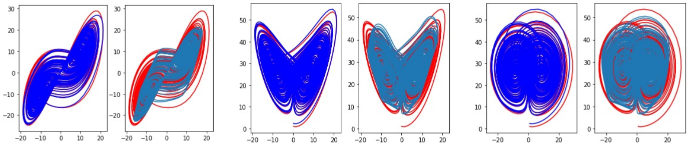


Figure: Projection of the true attractor and approximation of the attractor using a learned kernel on the XY,XZ and YZ axes (first, third and fifth from the left), Projection of the true attractor and approximation of the attractor using with initial kernel on the XY,XZ and YZ axes (second, fourth and sixth from the left)

Kernel Flows for Learning Chaotic Dynamical Systems

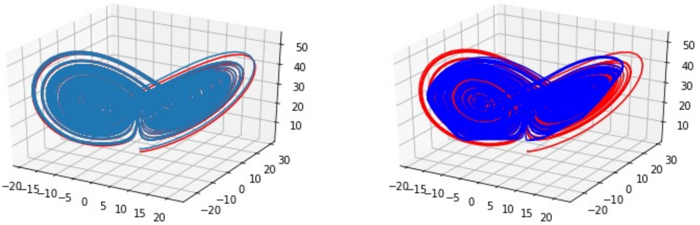
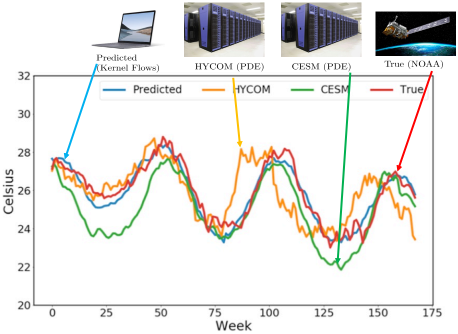


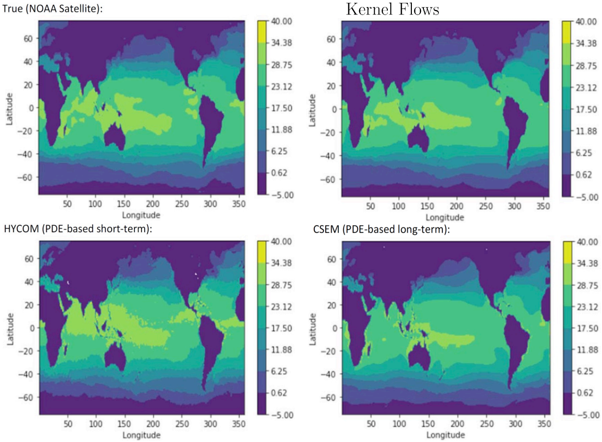
Figure: True attractor (blue) and approximation of the attractor using a learned kernel (red) [left], True attractor (blue) and approximation of the attractor using initial kernel (red) [right]

Kernel Flows for Learning Chaotic Dynamical Systems

- HYCOM: 800 core-hours per day of forecast on a Cray XC40 system
- CESM: 17 million core-hours on Yellowstone, NCAR's high-performance computing resource
- Architecture optimized LSTM: 3 hours of wall time on 128 compute nodes of the Theta supercomputer.
- Our method: 40 seconds to train on a single node machine (laptop) without acceleration



Kernel Flows for Learning Chaotic Dynamical Systems



Nonparametric Kernel Flows for Learning Chaotic Dynamical Systems

• Write $X := (X_1, \ldots, X_N)$ and $Y := (Y_1, \ldots, Y_N)$ for the input/output training data. Our goal is to learn a kernel of the form

$$K^{\phi}(x, x') = K(\phi(x, 1), \phi(x', 1)),$$

where K is a standard kernel and ϕ maps the input space into itself.

Nonparametric Kernel Flows for Learning Chaotic Dynamical Systems

- The warping of the input space ϕ satisfies the following ODE

$$\begin{cases} \dot{\phi}(x,t) = & v(\phi(x,t),t) \\ \phi(x,0) = & x \end{cases}$$

with

$$v(x,t) = \Gamma(x,q)\Gamma(q,q)^{-1}\dot{q}, \quad \text{and} \quad \dot{q} = -\nabla\left[\rho(q)\right],$$

where

- q corresponds to position variables in $\mathcal{X}^N$ starting from $q(0) = X = (X_1, \cdots, X_N)$.
- Γ is an operator/vector-valued kernel, $\Gamma(q,q)$ is an $N \times N$ matrix with entries $\Gamma(q_i, q_j)$.
- $\Gamma(x,q)$ is a $1 \times N$ vector with entries $\Gamma(x, q_i)$.
- ρ is the kernel flow loss associated with the input/output data (q, Y).

Nonparametric Kernel Flows for Learning Chaotic Dynamical Systems

• Using an explicit Euler scheme and regularizing with a nugget $\lambda > 0$ leads to an iteration of the form

$$\phi_{n+1}(x) = \phi_n(x) + \epsilon v_n(\phi_n(x)).$$

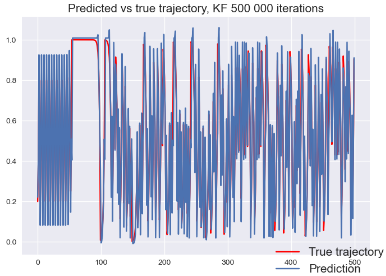
with $\phi_0(x) = x$.
• Writing $X = (X_1, \ldots, X_N)$ for the training points and
$q_n := \phi_n(X) := (\phi_n(X_1), \ldots, \phi_n(X_N))$, the discretized equations take the form

$$q_{n+1} = q_n - \epsilon \nabla \rho(q_n)$$

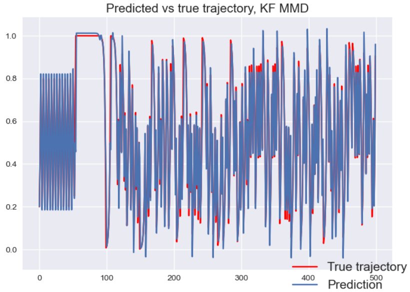
and

$$v_n(x) = \Gamma(x, q_n)\big(\Gamma(q_n, q_n) + \lambda I\big)^{-1}(q_{n+1} - q_n)/\epsilon$$

Nonparametric Kernel Flows for Learning Chaotic Dynamical Systems



(a) Time series (red) and the prediction (blue) by the learned kernel with ρ



(b) Time series (red) and the prediction (blue) by the learned kernel with ρ_{MMD}

Figure: Prediction results for the Bernoulli map

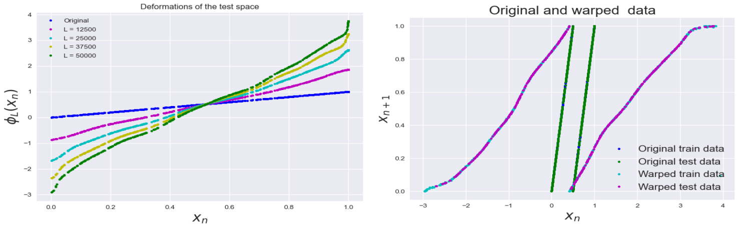


Figure: Deformation of input for different iterations of the flow function ϕ_L (left) and deformed final data (right).

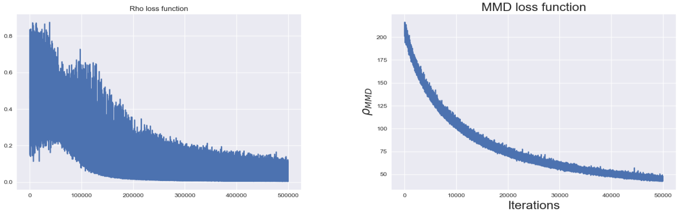


Figure: Convergence of the losses ρ and ρ_{MMD}.

Kernel Flows for Learning Irregularly-Sampled Time Series

• The above approach fails to be accurate for irregularly sampled series because it discards the information contained in the t_k.

• To address this issue, we consider the model

$$x_{k+1} = f^{\dagger}(x_k, \Delta_k, \dots, x_{k-\tau^{\dagger}+1}, \Delta_{k-\tau^{\dagger}+1}),$$

which incorporates the time differences $\Delta_k = t_{k+1} - t_k$ between observations.

• That is, we employ a time-aware time series representations by interleaving observations and time differences.

• The proposed strategy is then to construct a surrogate model by regressing $f^{\dagger}$ from past data and a kernel K_θ learned with Kernel Flows as described previously. Note that the past data takes are $X_k := (x_k, \Delta_k, \dots, x_{k+\tau-1}, \Delta_{k+\tau-1})$, $Y_k := x_{k+1}$ and $N = n - \tau$.

Kernel Flows for Learning Irregularly-Sampled Time Series

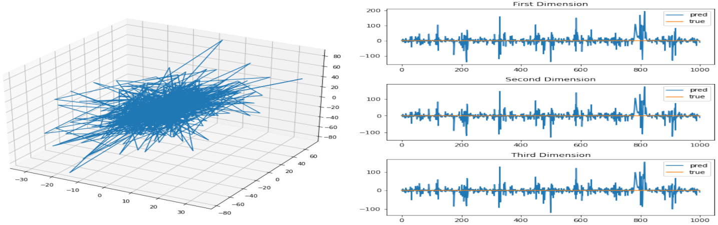


Figure: Attractor reconstruction (left), Time series reconstruction (right) without learning the kernel

Kernel Flows for Learning Irregularly-Sampled Time Series

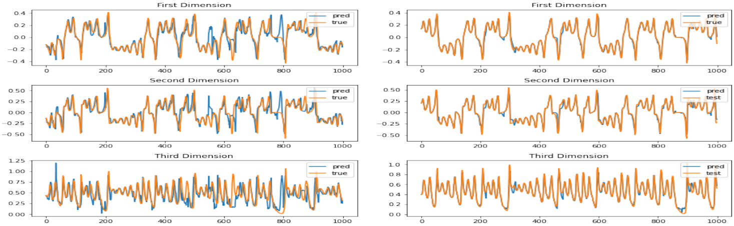


Figure: Reconstruction of the test time series of the Lorenz map with regular Kernel Flows (left) and irregular KFs (regular).

Kernel Flows for Learning Irregularly-Sampled Time Series

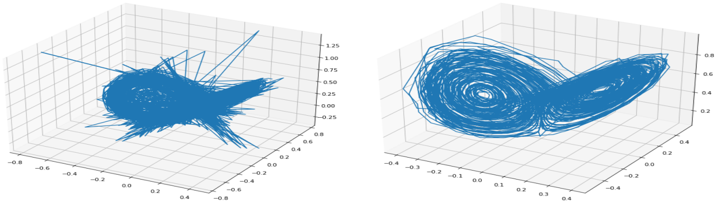


Figure: Approach with regular Kernel Flows (left), Approach with irregular Kernel Flows (right).

Kernel Flows for Learning Partially-Observed Dynamical Systems

- Consider the dynamical system

$$x(k+1) = f(x(k)) = \left[\begin{array}{c} f_n(x) \\ f_m(x) \end{array} \right]$$

where $f \in \mathcal{C}(\mathbb{R}^n \times \mathbb{R}^m, \mathbb{R}^{n+m})$.

- We assume that we have access to measurements from the first n components of the $x-$variable that we denote as x^n and that the remaining m components, that we denote as x^m, are not observed, i.e. we only observe $x^n(1), \dots, x^n(l)$. Our goal is to forecast $x(l+1)$ given the observation of $x^n(1), \dots, x^n(l)$.

Kernel Flows for Learning Partially-Observed Dynamical Systems

• This is equivalent to minimizing the following optimization problem w.r.t f_n, f_m and the the unknown $m-$variables required in the representer formula.

$$\min \mathcal{L} = ||f_n||^2_{\Gamma_1} + ||f_m||^2_{\Gamma_2} + \lambda \sum_{i=1}^{N} \Big((f_n(x_i^n, x_i^m) - x_{i+1}^n)^2 + (f_m(x_i^n, x_i^m) - x_{i+1}^m)^2 \Big),$$

• Let $A = (x_2^n, \cdots, x_{l+1}^n)$, $B = (x_2^m, \cdots, x_{l+1}^m)$, $C = (\ldots, (x_i^n, x_i^m), \ldots)$.
The minimizers of the loss $\mathcal{L}$ are $f_n(\cdot) = \Gamma_1(\cdot, C)(\Gamma_1(C, C) + \lambda^{-1} I_d)^{-1} A$, $f_m(\cdot) = \Gamma_2(\cdot, C)(\Gamma_2(C, C) + \lambda^{-1} I_d)^{-1} B$ which leads to the following reduced optimization problem

$$\min_B A^T(\Gamma_1(C, C) + \lambda^{-1} I_d)^{-1} A + B^T(\Gamma_2(C, C) + \lambda^{-1} I_d)^{-1} B$$

Kernel Flows for Learning Partially-Observed Dynamical Systems

Consider the Lorenz system

$$\begin{aligned} \dot{x} &= \sigma(y-x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z \end{aligned}$$

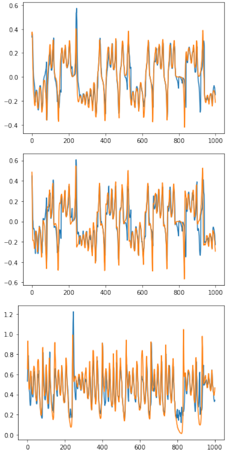
with $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$. First, we consider the case where we have access to the x and y variables but not z.
We follow the following steps: i.) find the auxiliary variable z_a, ii.) use kernel flows to learn the parameters of the kernel

$$K_\theta(x,y) = \theta_1^2 \exp\left(\frac{-\|x-y\|_2^2}{2\theta_0^2}\right) + \theta_3^2\left(x^\top y + \theta_2^2\right)^2 + \theta_6^2\left(\theta_4^2 + \theta_5^2\|x-y\|_2^2\right)^{-\frac{1}{2}} + \theta_9^2\left(\theta_8^2 + \|x-y\|_2^2\right)^{-\theta_7} +$$
$$\theta_{11}^2\left(1 + \frac{\|x-y\|_2^2}{\theta_{10}^2}\right)^{-1} + \theta_{12}^2 \max\left(0, 1 - \frac{\|x-y\|_2^2}{\theta_{13}^2}\right) + \theta_{14}^2 \exp\left(\frac{-\|x-y\|_2}{2\theta_{15}^2}\right) +$$
$$\theta_{16}^2 \exp\left(\frac{-\sin^2\left(\pi\|x-y\|_2^2/\theta_{17}\right)}{\theta_{18}^2}\right) \exp\left(-\frac{\|x-y\|_2^2}{\theta_{19}}\right) + \theta_{20}^2 \exp\left(\frac{-\sin^2\left(\pi\|x-y\|_2^2/\theta_{21}\right)}{\theta_{22}^2}\right)$$

Kernel Flows for Learning Partially-Observed Dynamical Systems

We generate 200 data points using initial conditions $x(0) = 0$, $y(0) = 0$, $z(0) = 0$, and sampling time $t_s = 0.01$, and we use gradient descent with step size $\eta = 10^{-7}$ to solve the optimization problem to find the auxiliary variable z_a.
For prediction, we started with a time delay $\tau^\dagger = 3$ but the results were poor. By increasing the time delay to $\tau^\dagger = 4$, the results improve and are in the figures below.

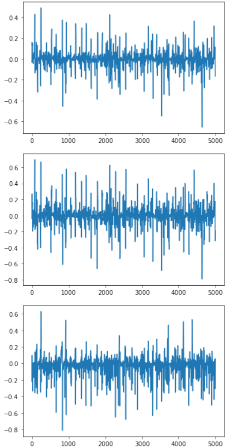
Kernel Flows for Learning Partially-Observed Dynamical Systems



True (blue) vs. Prediction (red) of the x variable (top), True (blue) vs. Prediction (red) of the y variable (middle), True (blue) vs. Prediction (red) of the y variable (bottom)

Kernel Flows for Learning Partially-Observed Dynamical Systems

The errors between the true and approximated values over longer simulation intervals are plotted in the figures below.



Kernel Flows for Learning Partially-Observed Dynamical Systems

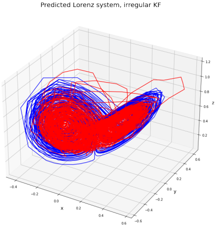


Figure: Reconstruction from true data (blue) vs. approximation (red) of the attractor.

Sparse Kernel Flows for Learning Chaotic Dynamics

- Consider a kernel of the form

$$K_{\beta,\theta}(x,y) = \sum_{i=1}^{m} \theta_i^2 k_i(x,y;\beta)$$

- Sparsify $K_{\beta,\theta}$ by L1 regularization

$$\mathcal{L}(\beta,\theta) = \arg\min_{\beta,\theta} 1 - \frac{y_c^\top K_{\beta,\theta}^{-1} y_c}{y_b^\top K_{\beta,\theta}^{-1} y_b} + \lambda \|\theta\|_1$$

- We apply it to a database of 131 chaotic dynamical systems.

We use the following kernel

$$
\begin{aligned}
K(x,y) =& \theta_1^2 \exp\left(\frac{-\|x-y\|_2^2}{2\beta_1^2}\right) + \theta_2^2\left(x^\top y + \beta_2^2\right)^2 + \theta_3^2\left(\beta_3^2 + \beta_4^2\|x-y\|_2^2\right)^{-\frac{1}{2}} + \theta_4^2\left(\beta_6^2 + \|x-y\|_2^2\right)^{-\beta_5} \\
&+ \theta_5^2\left(1 + \frac{\|x-y\|_2^2}{\beta_7^2}\right)^{-1} + \theta_6^2 \max\left(0, 1 - \frac{\|x-y\|_2^2}{\beta_8^2}\right) \\
&+ \theta_7^2 \exp\left(\frac{-\sin^2\left(\pi\|x-y\|_2^2/\beta_9\right)}{\beta_{10}^2}\right)\exp\left(-\frac{\|x-y\|_2^2}{\beta_{11}}\right) + \theta_8^2 \exp\left(\frac{-\sin^2\left(\pi\|x-y\|_2^2/\beta_{12}\right)}{\beta_{13}^2}\right) \\
&+ \theta_9^2 \exp\left(\frac{-\|x-y\|_2}{2\beta_{14}^2}\right) + \theta_{10}^2\left(\beta_{15}^2 + \beta_{16}^2\|x-y\|_2\right)^{-\frac{1}{2}} + \\
&\theta_{11}^2\left(\beta_{18}^2 + \|x-y\|_2\right)^{-\beta_{17}} + \theta_{12}^2\left(1 + \frac{\|x-y\|_2}{\beta_{19}^2}\right)^{-1} + \theta_{13}^2 \max\left(0, 1 - \frac{\|x-y\|_2}{\beta_{20}^2}\right) \\
&+ \theta_{14}^2 \exp\left(\frac{-\sin^2\left(\pi\|x-y\|_2/\beta_{21}\right)}{\beta_{22}^2}\right)\exp\left(-\frac{\|x-y\|_2}{\beta_{23}}\right) \\
&+ \theta_{15}^2 \exp\left(\frac{-\sin^2\left(\pi\|x-y\|_2/\beta_{24}\right)}{\beta_{25}^2}\right)
\end{aligned}
$$

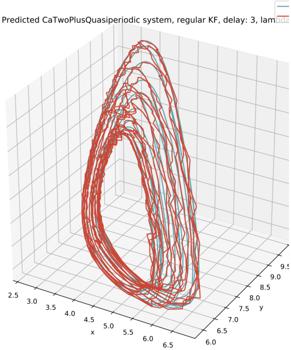
Sparse Kernel Flows for Learning Chaotic Dynamics

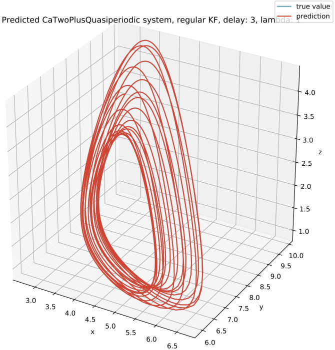
Example 1: Complex Ca^{2+} *oscillations*

$$\frac{d}{dt}z = V_{in} - V_2 + V_3 + k_f y - kz$$
$$\frac{d}{dt}y = V_2 - V_3 - k_f y$$
$$\frac{d}{dt}a = \beta V_4 - V_5 - \epsilon a$$

where $V_{in} = V_0 + V_1\beta$, $V_2 = V_{M2}\frac{z^2}{K_2^2+z^2}$, $V_3 = V_{M3}\frac{z^m}{K_z^m+z^m}\frac{y^2}{K_y^2+y^2}\frac{a^4}{K_a^4+a^4}$, $V_5 = V_{M5}\frac{a^p}{K_5^p+a^p}\frac{z^n}{K_d^n+z^n}$.

Sparse Kernel Flows for Learning Chaotic Dynamics





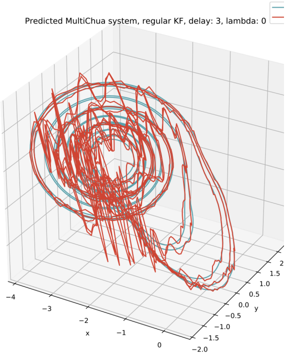
Example 2: Multiple interacting Chua electronic circuits
Equation:

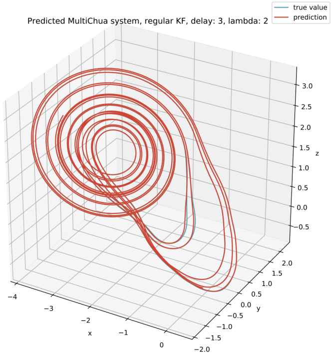
$$\frac{d}{dt}x = a(y - f(x))$$
$$\frac{d}{dt}y = x - y + z$$
$$\frac{d}{dt}z = -by$$

where

$$f(x) = m_7 x + \sum_{i=1}^{5} \frac{1}{2}(m_i - m_{i+1})\left(|x + c_{i+1}| - |x - c_{i+1}|\right)$$

Sparse Kernel Flows for Learning Chaotic Dynamics





Sparse Kernel Flows for Learning Chaotic Dynamics

Index	Name	CaTwoPlusQuasiperiodic		MultiChua	
		Regular KFs	Sparse KFs ($\lambda = 1$)	Regular KFs	Sparse KFs ($\lambda = 2$)
coefficients	θ_1	3.007	0.169	1.149	1.016
	θ_2	15.886	-3.287	1.558	1.834
	θ_3	2.260	0.495	1.131	0.965
	θ_4	3.290	0.166	1.152	0.974
	θ_5	3.297	0.113	1.152	0.965
	θ_6	4.735	0.009	0.731	0.853
	θ_7	5.063	**0**	1.516	0.852
	θ_8	0.947	0.769	0.162	**0**
	θ_9	3.055	0.294	1.378	1.013
	θ_{10}	2.404	0.505	1.307	0.962
	θ_{11}	3.892	0.204	1.575	1.017
	θ_{12}	3.895	0.133	1.578	1.019
	θ_{13}	6.611	**0**	1.294	0.941
	θ_{14}	8.462	-0.038	3.709	1.220
	θ_{15}	-2.451	7.375	0.538	0.232
error criterion	SMAPE	0.006	3.40×10^{-5}	0.069	0.004
	Hausdorff Distance	2.789	0.013	12.056	0.216

Hausdorff metric based Kernel Flows for Learning Chaotic Dynamics

- Consider a kernel of the form

$$K_{\beta,\theta}(x,y) = \sum_{i=1}^{m} \theta_i^2 k_i(x,y;\beta)$$

- Sparsify $K_{\beta,\theta}$ by L1 regularization and learn its parameters via cross-validation of the Hausdorff metric between the reconstruction of the attractor from N points and the reconstruction of the attractor from $N/2$ points

$$\mathcal{L}(\beta,\theta) = \arg\min_{\beta,\theta} HD(\mathcal{A}_N, \mathcal{A}_{N/2}) + \lambda\|\theta\|_1$$

- We apply it to a database of 131 chaotic dynamical systems.

Hausdorff metric based Kernel Flows for Learning Chaotic Dynamics

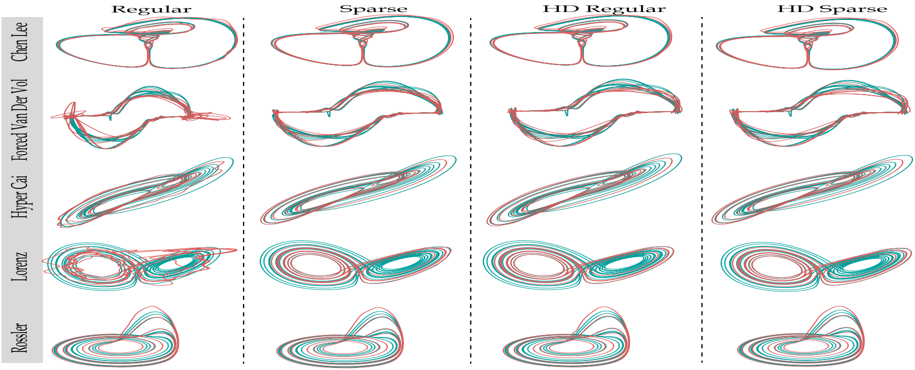


Figure: Comparison of four methods for the examples. In each plot, the green line presents true trajectory and the red line present predicted trajectory, respectively.

Hausdorff metric based Kernel Flows for Learning Chaotic Dynamics

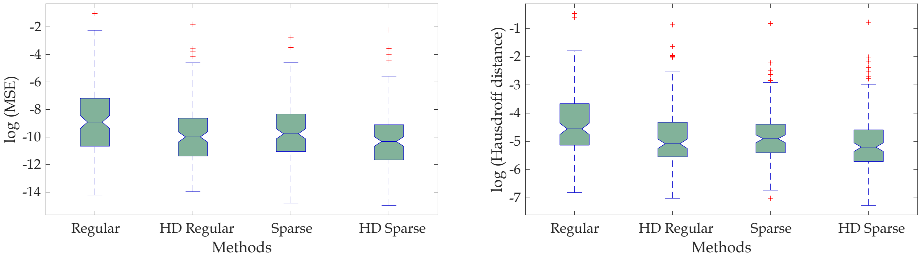


Figure: Distribution of forecasting errors for different methods for all 133 dynamical systems.

Detection of Critical Transitions for MultiScale Systems

Detection of Critical Transitions for MultiScale Systems

- Consider the fast-slow SDE

$$
\begin{aligned}
\dot{x}_1 &= \frac{1}{\epsilon} f_1(x_1, x_2) + \frac{\sigma_1}{\sqrt{\epsilon}} \eta_1(\tau), \\
\dot{x}_2 &= f_2(x_1, x_2) + \sigma_2 \eta_2(\tau)
\end{aligned}
$$

where $f_1 \in \mathcal{C}(\mathbb{R}^2; \mathbb{R})$ and $f_2 \in \mathcal{C}(\mathbb{R}^2; \mathbb{R})$ are Lipschitz and η_1, η_2 are independent white Gaussian noises.

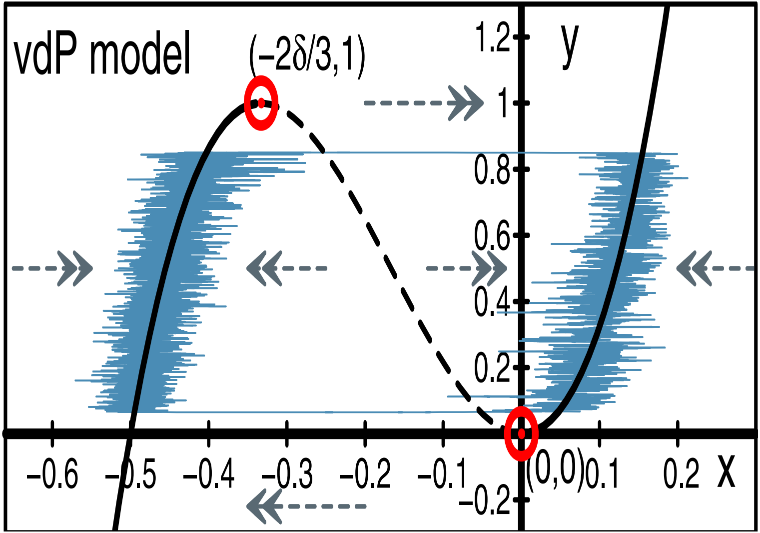
- x_1 is a fast variable in comparison to the slow variable x_2.
- The set $C_0 = \{(x_1, x_2) \in \mathbb{R}^2 : f_1(x_1, x_2) = 0\}$ is called the critical manifold.

MultiScale Systems

- The van der Pol model.
- The equations of the model are

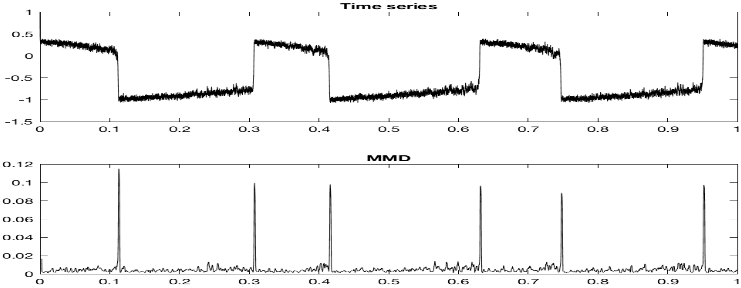
$$\begin{aligned} \dot{x}_1 &= \frac{1}{\epsilon}\left(x_2 - \frac{27}{4\,\delta^3} x_1^2 (x_1 + \delta)\right) + \frac{\sigma_1}{\sqrt{\epsilon}} \eta_1(t) \\ \dot{x}_2 &= -\frac{\delta}{2} - x_1 + \sigma_2 \eta_2(t) \end{aligned}$$

$\delta = 1, \sigma_1 = 0.1, \sigma_2 = 0.1, \varepsilon = 0.01.$


vdP model
(−2δ/3,1)
1.2
1
0.8
0.6
0.4
0.2
−0.2
y
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
(0,0)
0.1
0.2
x

MultiScale Systems

- Numerical Simulation



Detection of Critical Transitions for MultiScale Systems

• We'll use the following Gabor wavelet as basis to build the reproducing kernel :

$$G_{\tau,\omega,\theta}(t) := (\frac{2}{\pi^3})^{\frac{1}{4}}\sqrt{\frac{\omega}{\alpha}}\cos(\omega(t-\tau)+\theta)e^{-\frac{\omega^2(t-\tau)^2}{\alpha^2}}, \quad t,\tau,\theta \in \mathbb{R} \ \ \omega,\alpha > 0$$

This wavelet allows only to recognize modes of the form $t \to \cos(\omega(t-\tau)+\theta)$ "à la Fourier series".

• In our context, we extend these wavelets to detect signals of the form $t \to \mathrm{y}(\omega(t-\tau)+\theta)$ for 2π-periodic signal $y \in \mathrm{L}^2([0,2\pi])$. This can be done using

$$\chi_{y;\tau,\omega,\theta}(t) := (\frac{2}{\pi^3})^{\frac{1}{4}}\sqrt{\frac{\omega}{\alpha}}\mathrm{y}(\omega(t-\tau)+\theta)e^{-\frac{\omega^2(t-\tau)^2}{\alpha^2}}, \quad t,\tau,\theta \in \mathbb{R} \ \ \omega,\alpha > 0$$

Given χ, we construct the Gram matrix whose entries are

$$K_{y;\tau,\omega,\theta}(s,t) := \chi_{y;\tau,\omega,\theta}(s)\chi_{y;\tau,\omega,\theta}(t), \quad s,t \in [0,1]$$

Detection of Critical Transitions for MultiScale Systems

- The reproducing kernel K_y associated to y, we integrate $K_{y;\tau,\omega,\theta}(s,t)$ w.r.t τ,ω,θ over their domain of definition :

$$K_y(s,t) = \int_{\theta_{\min}}^{\theta_{\max}} \int_{\omega_{\min}}^{\omega_{\max}} \int_{\tau_{\min}}^{\tau_{\max}} K_{y;\tau,\omega,\theta}(s,t)d\tau\, d\omega\, d\theta, \quad s,t \in [0,1]$$

- For stochastic van der Pol, the function y and the corresponding kernel are

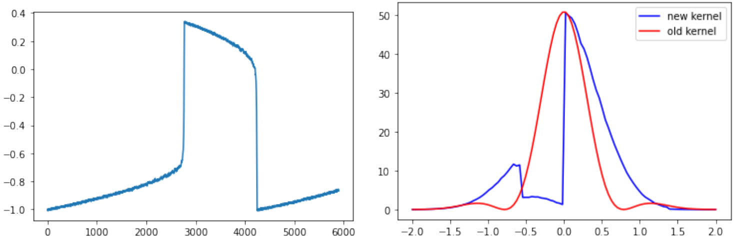


Figure: The function y used to build the kernel $k(s,t)$ (left), Projection on the $s-$axis of the plot of the kernel $K_G(s,t)$ from vs. kernel $K_\chi(s,t)$ (right)

Detection of Critical Transitions for MultiScale Systems

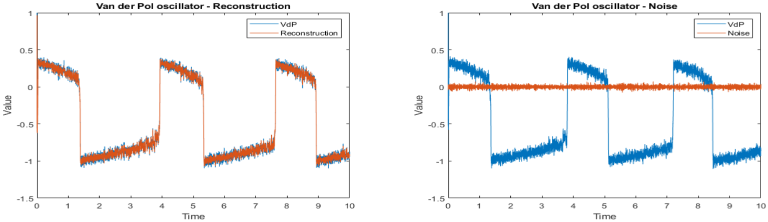


Figure: Reconstruction and noise for stochastic Van der Pol

Detection of Critical Transitions for MultiScale Systems

- We define the energy of a sliding window $W_i = [i\tau, (i+1)\tau]$ of width τ as

$$\mathcal{E}_i = v_i^T K_{\mathcal{T}}^{-1} K_{\omega_i} K_{\mathcal{T}}^{-1} v_i$$

where $K_{\mathcal{T}}(s,t) = \sum_i K_{w_i}(s,t) + \sigma^2 I_d$ with σ large and I_d the identity matrix, v_i is the signal in the interval $[i\tau, (i+1)\tau]$, $K_{w_i}(s,t) = K(x(s), x(t))$ with $s, t \in W_i$, and $K_{w_i}(s,t) = 0$ otherwise.

Detection of Critical Transitions for MultiScale Systems

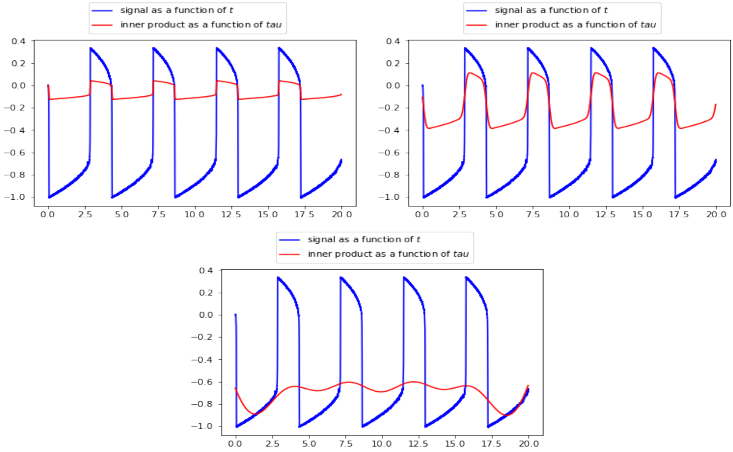


Figure: Energy $\mathcal{E}$ for $\alpha = 0.01$ (top left) and $\alpha = 0.1$ (top right), $\alpha = 2.0$ (bottom)

Conclusions

• We used different variants of kernel flows to approximate chaotic dynamical systems.
• We used the maximum mean discrepancy and extended kernel mode decomposition to detect critical transitions.

References (all online on Arxiv and Researchgate)

1. Lu Yang, Xiuwen Sun, Boumediene Hamzi, Yannis Kevrekidis, Houman Owhadi, Naiming Xie (2023). Learning dynamical systems from data: A simple cross-validation perspective, part VI: Hausdorff metric based training of kernels to learn attractors with application to 133 chaotic dynamical systems.
2. Lu Yang, Xiuwen Sun, Boumediene Hamzi, Houman Owhadi, Naiming Xie (2023). Learning dynamical systems from data: A simple cross-validation perspective, part V: Sparse Kernel Flows for 131 chaotic dynamical systems.
3. Boumediene Hamzi, Houman Owhadi, and Yannis Kevrekidis (2022). Learning dynamical systems from data: A simple cross-validation perspective, part IV: Case with Partial Observations.
4. Boumediene Hamzi, AmirHossein Jafarian, Houman Owhadi, Léo Paillet (2022). A Note on Microlocal Kernel Design for Some Slow-Fast Stochastic Differential Equations with Critical Transitions and Application to EEG Signals.
5. M. Darcy, B. Hamzi, G. Livieri, H. Owhadi, P. Tavallali (2022). One-Shot Learning of Stochastic Differential Equations with Computational Graph Completion.
6. Jonghyeon Lee, Edward De Brouwer, Boumediene Hamzi, Houman Owhadi (2021). Learning dynamical systems from data: A simple cross-validation perspective, part III: Irregularly-Sampled Time Series.
7. M Darcy, B Hamzi, J Susiluoto, A Braverman, H Owhadi (2021). Learning dynamical systems from data: a simple cross-validation perspective, part II: nonparametric kernel flows.
8. B. Hamzi, R. Maulik, H. Owhadi (2021), Data-driven geophysical forecasting: Simple, low-cost, and accurate baselines with kernel methods.
9. B. Hamzi and H. Owhadi (2020), Learning dynamical systems from data: A simple cross-validation perspective, part I: Parametric kernel flows.
10. B. Haasdonk, B. Hamzi, G. Santin and D. Wittwar (2020), Kernel methods for center manifold approximation and a data-based version of the Center Manifold Theorem.
11. Stefan Klus, Feliks Nüske, Boumediene Hamzi (2020), Kernel-based approximation of the Koopman generator and the Schrödinger operator.
12. Andreas Bittracher, Stefan Klus, Boumediene Hamzi, Peter Koltai and Christof Schütte (2020), Dimensionality Reduction of Complex Metastable Systems via Kernel Embeddings of Transition Manifolds.
13. B. Hamzi, C. Kuehn, S. Mohamed (2019), A Note on Kernel Methods for Multiscale Systems with Critical Transitions, Mathematical Methods in the Applied Sciences.
14. B. Haasdonk, B. Hamzi, G. Santin and D. Wittwar (2018), Greedy Kernel Methods for Center Manifold Approximation, Proc. of ICOSAHOM.
15. J. Bouvrie and B. Hamzi (2017), Kernel Methods for the Approximation of Some Key Quantities of Nonlinear Systems, J. of Comp. Dyn.
16. J. Bouvrie and B. Hamzi (2017), Kernel Methods for the Approximation of Nonlinear Systems, SIAM J. Control & Optimization, vol. 55, no. 4.
17. P. Giesl, B. Hamzi, M. Rasmussen, K. Webster (2016), Approximation of Lyapunov Functions from Noisy Data, J. of Comp. Dyn.
18. J. Bouvrie and B. Hamzi (2012), Empirical Estimators for the Controllability Energy and Invariant Measure of Stochastically Forced Nonlinear Systems, in Proc. of the 2012 American Control Conference.
19. J. Bouvrie and B. Hamzi (2010), Balanced Reduction of Nonlinear Control Systems in Reproducing Kernel Hilbert Spaces, Proc. 48th Annual Allerton Conference on Communication, Control, and Computing, pp. 294-301.

Other Activities on MLDS

- Special Issue on "Machine Learning and Dynamical Systems" in Physica D.
- Machine Learning and Dynamical Systems Seminar, hosted by the Alan Turing Institute (London, UK), cf. https://sites.google.com/site/boumedienehamzi/machine-learning-and-dynamical-systems-seminar to join mailinglist.
- Possibly 4th Symposium on MLDS at the Fields Institute in Toronto in 2024.