Physica D 421 (2021) 132817

Contents lists available at ScienceDirect

Physica D

journal homepage: www.elsevier.com/locate/physd

Learning dynamical systems from data: A simple cross-validation perspective, part I: Parametric kernel flows

Boumediene Hamzi [a,*], Houman Owhadi [b]

[a] *Department of Mathematics, Imperial College London, United Kingdom*
[b] *Department of Computing and Mathematical Sciences, Caltech, CA, USA*

ARTICLE INFO

Article history:
Received 4 July 2020
Received in revised form 3 December 2020
Accepted 5 December 2020
Available online 5 February 2021

Keywords:
Learning dynamical systems
Data
Kernel methods
Kernel flows

ABSTRACT

Regressing the vector field of a dynamical system from a finite number of observed states is a natural way to learn surrogate models for such systems. We present variants of cross-validation (Kernel Flows (Owhadi and Yoo, 2019) and its variants based on Maximum Mean Discrepancy and Lyapunov exponents) as simple approaches for learning the kernel used in these emulators.

1. Introduction

Linear stochastic models (autoregressive (AR), moving average (MA), ARMA models) and chaotic dynamical systems are natural predictive models for time series [1–5].

The prediction of chaotic systems from time-series (initially investigated in [6]) has been investigated from the regression perspectives of support vector machines [7,8], reservoir computing [9,10], deep feed-forward artificial neural networks (ANN), and recurrent neural networks with long short-term memory (RNN-LSTM) [11–14]. Reservoir computing was observed to be efficient for predictions but not very accurate for estimating Lyapunov exponents. On the other hand, RNN-LSTM were observed to be accurate for estimating Lyapunov exponents but not as good as reservoir computing for predictions (see [15] for a survey). Although Reproducing Kernel Hilbert Spaces (RKHS) [16] has provided strong mathematical foundations for analyzing dynamical systems [17–27], the accuracy of these emulators depends on the kernel and the problem of selecting a good kernel has received less attention.

We investigate Kernel Flows [28] (KF) as a generic tool for selecting the kernel used to learn chaotic dynamical systems. The KF strategy is to induce an ordering (quantifying the quality of a kernel) in a space of kernels and use gradient descent to identify a good kernel. KF is an efficient method of learning kernels with predictive capabilities using random projections that guarantees good performance while reducing computational cost. KF is also a variant of cross-validation (see discussion in [29]) in the sense that it operates under the premise that a kernel must be good if the number of points used to interpolate the data can be halved without significant loss in accuracy, i.e., the method presented in [28] uses the regression relative error between two interpolants (measured in the RKHS norm of the kernel) as the quantity to minimize.

In this paper, we use this metric along two new ones to learn the parameters of the kernel. The first one is the difference between two estimations of the maximal Lyapunov exponent (the second estimator using a random half of the data points of the first). The second metric is the Maximum Mean Discrepancy (MMD) [30] computed from two different samples of a time series or between a sample and a subsample of half length. Our paper is numerical in nature and we refer to [29] for a rigorous analysis of KF (and comparisons with Empirical Bayes for learning PDEs) and to [31] for its applications to training neural networks.

The main contributions of this paper are as follows.

- We show that combining KF with the kriging of the vector field significantly improves the accuracy of (1) the prediction of chaotic time series (2) the reconstruction of attractors (3) the reconstruction of the dynamics from lower dimensional projections of the state space.
- We show that Kernel Mode Decomposition can recover time delays in the reconstruction of the dynamics.
- We introduce Lyapunov exponents and MMD as two new cross validation metrics for kriging vector fields.

[*] Corresponding author.
E-mail addresses: boumediene.hamzi@gmail.com (B. Hamzi), owhadi@caltech.edu (H. Owhadi).

https://doi.org/10.1016/j.physd.2020.132817
0167-2789/© 2020 Elsevier B.V. All rights reserved.

The remainder of the manuscript is structured as follows. We describe the problem in Section 2 and propose three cross-validation metrics to learn the parameters of the kernel used for approximating the vector field of the dynamical system. In Section 3, we investigate the performance of these methods for the Bernoulli map, the logistic map, the Hénon map and the Lorenz system. In the Appendix, we recall optimal recovery theoretical foundations of KF.

2. The problem and its proposed cross-validation solutions

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

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

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

$$f^{\dagger}(X_k) = Y_k \text{ for } k = 1, \dots, N, \tag{2}$$

with $X_k := (x_{k+\tau-1}, \dots, x_k)$, $Y_k := x_{k+\tau}$ and $N = n - \tau$. Given a reproducing kernel Hilbert space[1] of candidates $\mathscr{H}$ for $f^{\dagger}$, and using the relative error in the RKHS norm $\|\cdot\|_{\mathscr{H}}$ as a loss, the regression of the data (X_k, Y_k) with the kernel K associated with $\mathscr{H}$ provides a minimax optimal approximation [32] of $f^{\dagger}$ in $\mathscr{H}$. This interpolant (in the absence of measurement noise) is

$$f(x) = K(x, X)(K(X, X))^{-1}Y, \tag{3}$$

where $X = (X_1, \dots, X_N)$, $Y = (Y_1, \dots, Y_N)$, $k(X, X)$ for the $N \times N$ matrix with entries $k(X_i, X_j)$, and $k(x, X)$ is the N vector with entries $k(x, X_i)$. This interpolation has also a natural interpretation in the setting of Gaussian process (GP) regression: (i) (3) is the conditional mean of the centered GP $\xi \sim \mathcal{N}(0, K)$ with covariance function K conditioned on $\xi(X_k) = Y_k$, and (ii) the interpolation error between $f^{\dagger}$ and f is bounded by the conditional standard deviation of the GP ξ, i.e.

$$|f^{\dagger}(x) - f(x)| \le \sigma(x)\|f^{\dagger}\|_{\mathscr{H}}, \tag{4}$$

with

$$\sigma^2(x) = K(x, x) - K(x, X)(K(X, X))^{-1}K(x, X)^T. \tag{5}$$

Evidently the accuracy of the proposed approach depends on the kernel K and one of our goals is to also learn that kernel from the data (X_k, Y_k) with Kernel Flows (KF) [28].

Given a family of kernels $K_\theta(x, x')$ parameterized by θ, the KF algorithm can then be described as follows [28,31]:

i. Select random subvectors X^b and Y^b of X and Y (through uniform sampling without replacement in the index set $\{1, \dots, N\}$)

ii. 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)

iii. Let[2]

$$\rho(\theta, X^b, Y^b, X^c, Y^c) := 1 - \frac{Y^{c,T} K_\theta(X^c, X^c)^{-1} Y_c}{Y^{f,T} K_\theta(X^b, X^b)^{-1} Y^b}, \tag{6}$$

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

iv. Evolve θ in the gradient descent direction of ρ, i.e. $\theta \leftarrow \theta - \delta \nabla_\theta \rho$

v. Repeat.

Since the motivation of learning a dynamical system from data is not necessarily about only making prediction but also about emulating the qualitative behavior of the dynamical systems, we also consider different metrics in step 3 of the algorithm described above. The first new metric is by considering, in the case of chaotic systems, that a kernel is good if the estimate of the Lyapunov exponent obtained from the kernel approximation of the dynamics does not change if half of the data is used. So we will minimize[3]

$$\rho_L = |\lambda_{\mathrm{max},N} - \lambda_{\mathrm{max},N/2}|, \tag{7}$$

instead of (6) with $\lambda_{\mathrm{max},N}$ is the estimate of the maximal Lyapunov exponent from the kernel approximation of the dynamics with N sample points and $\lambda_{\mathrm{max},N/2}$ is the estimate of the maximal Lyapunov exponent from the kernel approximation of the dynamics with $N/2$ sample points. We use the algorithm of Eckmann et al. [33] to estimate the Lyapunov exponents from data by considering the kernel approximation of the dynamics. We use the Python implementation in [34] to estimate the Lyapunov exponents from data.

The second new metric is based on the Maximum Mean Discrepancy (MMD) [30] that is a distance on the space of probability measures with a representer theorem for empirical distributions which we recall in the Appendix. Our strategy for learning the kernel K will then simply be to minimize the MMD

$$\rho_{\mathrm{MMD}} = \mathrm{MMD}(S_1, S_2), \tag{8}$$

between two different samples,[4] $S_1 = x_{\sigma_1}, \dots, x_{\sigma_m}$ and $S_2 = x_{\mu_1}, \dots, x_{\mu_m}$, of the time series.

	$[\alpha_0, \sigma_0, \alpha_1, \sigma_1]$	No. of iterations	R_1	R_2
ρ	[1.31, 1.01, 0.99, 0.99]	100	0.019	0.015
ρ_{MMD}	[0.830, 2.780, 0.562, 2.926]	1000	0.027	0.011
No learning	[0, 1, 1, 1]	0	0.182	0.118

3. Numerical experiments

We now numerically investigate the efficacy of the cross-validation approaches described in the previous section in learning chaotic dynamical systems.

3.1. Bernoulli map

We first use the Bernoulli map

$$x(k+1) = 2x(k) \bmod 1, \tag{9}$$

which is a prototypical chaotic dynamical system [35]. We initialize (9) from an (irrational) initial condition $x(0) = \pi/3$ and use 200 points to train the kernel and for interpolation. We use a parameterized family of kernels of the form

$$k(x, y) = \alpha_0 \max\{0, 1 - \frac{\|x-y\|_2^2|}{\sigma_0}\} + \alpha_1 e^{\frac{\|x-y\|_2^2}{\sigma_1^2}}. \tag{10}$$

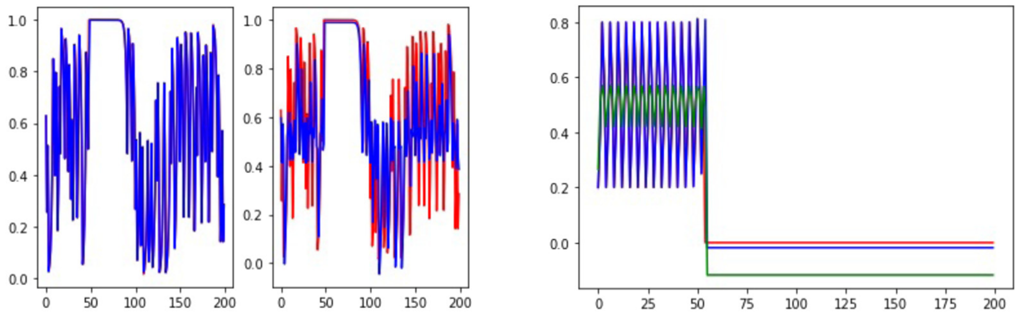
We set the initial kernel to be the Gaussian kernel and initialize the parameters with $(\alpha_0, \sigma_0, \alpha_1, \sigma_1) = (0, 1, 1, 1)$. The parameters of the kernel after training with ρ and ρ_{MMD} and the Root Mean

[1] A brief overview of RKHSs is given in the Appendix.

[2] $\rho := \|u^b - u^c\|_{K_\theta}^2 / \|u^b\|_{K_\theta}^2$, with $u^b(x) = K_\theta(x, X^b)K_\theta(X^b, X^b)^{-1}Y^b$ and $u^c(x) = K_\theta(x, X^c)K_\theta(X^c, X^c)^{-1}Y^c$, and ρ admits the representation (6) enabling its computation.

[3] One could also look at a metric that involves estimates of all Lyapunov exponents instead of just the maximal one.

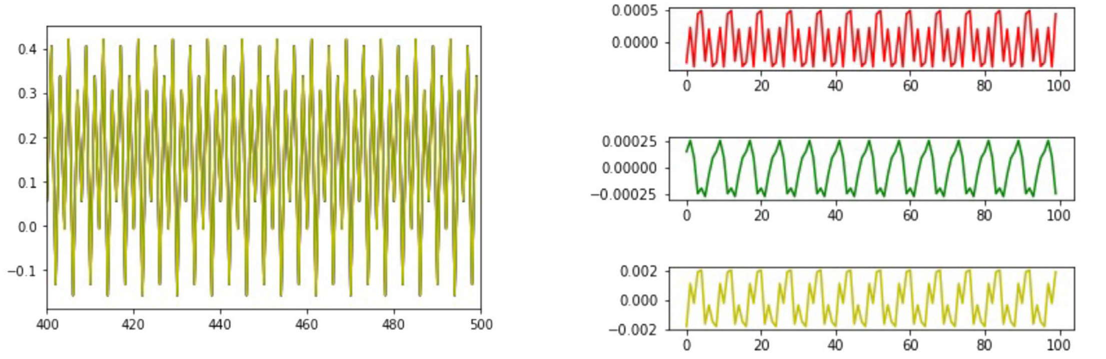
[4] One could also consider the MMD between a sample S_1 of size m and a subsample of S_1 of size $m/2$.



(a)

(b)

Fig. 1. (a) Time series generated by the true dynamics (red) and the approximation (blue) with the learned kernel (left) and the initial kernel (right), for an irrational initial condition $\pi/10$, (b) Time series generated by the true dynamics (red), the approximation with the learned kernel (blue), the kernel approximation without learning the kernel (green), for a rational initial condition 0.1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Time series generated by the true dynamics (red) and the approximation with the learned kernel using ρ (blue), the approximation with the learned kernel using ρ_L (green), approximation without learning (yellow)

(b) Difference between the true and the approximated dynamics with the learned kernel using ρ (top), with the learned kernel using ρ_L (middle), with the initial kernel (bottom), for an initial condition $x_0 = 0.3$

Fig. 2. Prediction results for the logistic map.

Square Errors[5] (RMSEs) with 5000 points are summarized in the following table with R_1 being the RMSE for $x(0) = \pi/10$ and R_2 the RMSE for $x(0) = 0.1$.

Fig. 1 shows results for an irrational initial condition $x(0) = \pi/10$ and 5000 points and a rational initial condition $x(0) = 0.1$.

We also consider a parameterized family of kernels of the form

$$k(x,y) = \alpha_0 \max\{0, 1 - \frac{\|x-y\|_2^2|}{\sigma_0}\} + \alpha_1 e^{\frac{\|x-y\|_2^2}{\sigma_1^2}} + \alpha_2 e^{-\frac{\|x-y\|_2}{\sigma_2^2}} + \alpha_3 e^{-\sigma_3 \sin^2(\sigma_4 \pi \|x-y\|_2)} e^{-\frac{\|x-y\|_2^2}{\sigma_5^2}} + \alpha_4 \|x-y\|_2^2. \tag{11}$$

Results are summarized in Table 1.

3.2. Example 2 (Logistic map)

Consider the logistic map $x(k+1) = 4x(k)(1-x(k))$. To approximate this map, we use an initial condition $x(0) = 0.1$ and use 200 points to train the kernel and for interpolation. We use a kernel of the form

$$k(x,y) = \alpha_0 e^{-\sigma_1 \sin^2(\pi\sigma_2\|x-y\|_2^2)} e^{-\|x-y\|_2^2/\sigma_3^2},$$

and initialize with the set of parameters $(\alpha_0, \sigma_1, \sigma_2, \sigma_3) = (1, 1, 1, 1)$. Let R_1 be the RMSE for an initial condition $x(0) = 0.4$, R_2 for $x(0) = 0.97$ with 5000 points.

	$[\alpha_0, \sigma_1, \sigma_2, \sigma_3]$	No. of it.	R_1	R_2
ρ	[0.95, 0.98, 1.20, 0.62]	100	0.0004	0.002
ρ_L	[0.6, 1.8, 2.3, 1.4]	1000	0.001	0.001
No learning	[1, 1, 1, 1]	0	0.004	0.004

Fig. 2.a shows the results for an initial condition $x(0) = 0.3$ and 5000 points. Fig. 2.b shows the prediction errors for the case of an approximation with a learned kernel using ρ, ρ_L and a kernel without learning. Fig. 3 shows the plot of error interval for $f^\dagger(x)$ given by $\Delta(f(x))$ in (28).

We also consider a parameterized family of kernels of the form

$$k(x,y) = \alpha_0^2 \max\{0, 1 - \frac{\|x-y\|_2^2|}{\sigma_0}\} + \alpha_1^2 e^{\frac{\|x-y\|_2^2}{\sigma_1^2}} + \alpha_2^2 e^{-\frac{\|x-y\|_2}{\sigma_2^2}} + \alpha_3^2 e^{-\sigma_3 \sin^2(\sigma_4 \pi \|x-y\|_2^2)} e^{-\frac{\|x-y\|_2^2}{\sigma_5^2}} + \alpha_4^2 \|x-y\|_2^2. \tag{12}$$

We initialize with a Gaussian kernel. The results are summarized in the following table where R_1 corresponds to the RMSE with $x(0) = 0.4$ and R_2 corresponds to the RMSE with $x(0) = 0.97$ (see Table 2).

[5] The Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data. Formally it is defined as RMSE $= \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$ with $\hat{y}_1, \ldots, \hat{y}_n$ are predicted values, $y_1, \ldots, y_n$ are observed values and n is the number of observations.

Table 1

	$[\alpha_0, \sigma_0, \alpha_1, \sigma_1, \alpha_2, \sigma_2, \alpha_3, \sigma_3, \sigma_4, \sigma_5, \alpha_4]$	No. of it.	R_1	R_2
ρ	[23.98, 1.13, 1.13, 0.83, 32.73, 0.72, 32.09, 0.29, 4.47, 0.20, 0.10]	500	0.016	0.014
No learning	[0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0]	0	0.182	0.118

Table 2

	$[\alpha_0, \sigma_0, \alpha_1, \sigma_1, \alpha_2, \sigma_2, \alpha_3, \sigma_3, \sigma_4, \sigma_5, \alpha_4]$	No. of it.	R_1	R_2
ρ	$[0.15, 0.96, 0.99, 1.02, 0.08, 0.98, -3.96\ 10^{-05}, 0.99, 0.99, 0.99, 0.98]$	500	0.0003	0.0004
No learning	[0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0]	0	0.004	0.004

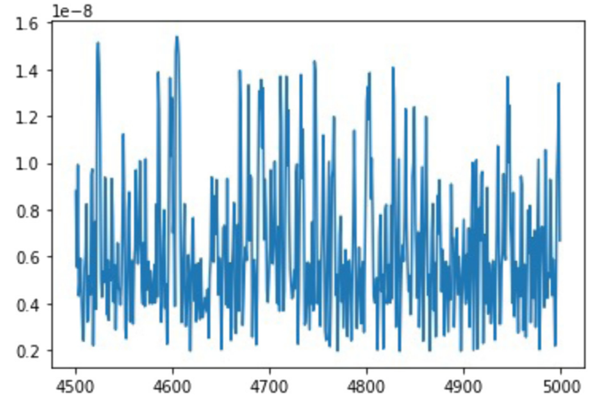


Fig. 3. Uncertainty $\Delta(f(x))$ in formula (27) for an initial condition $x_0 = \pi/4$.

3.3. Example 3 (Hénon map)

Consider the Hénon map

$$x(k+1) = 1 - ax(k)^2 + y(k),$$
$$y(k+1) = bx(k),$$

with $a = 1.4$ and $b = 0.3$. To learn this map, we generate 100 points with initial conditions $(x(0), y(0)) = (0.9, -0.9)$ to learn two kernels

$$k_i(x, y) = \alpha_i + (\beta_i + \|x - y\|_2^{\kappa_i})^{\sigma_i} + \delta_i e^{-\|x-y\|_2^2/\mu_i^2},$$

$(i = 1, 2)$ corresponding to the two maps $\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} \mapsto x(k+1)$ and $\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} \mapsto y(k+1)$. We initialize with a Gaussian kernel and after 1000 iterations, we get[6]

	$\begin{bmatrix} \alpha_1 & \beta_1 & \kappa_1 & \sigma_1 & \delta_1 & \mu_1 \\ \alpha_2 & \beta_2 & \kappa_2 & \sigma_2 & \delta_2 & \mu_2 \end{bmatrix}$	No. of it.	R_1
ρ	$\begin{bmatrix} 0.99 & 1.12 & 0.74 & 2.21 & 0.98 & 0.89 \\ 1.00 & 1.01 & 3.35 & 0.008 & 0.95 & 1.35 \end{bmatrix}$	1000	$\begin{bmatrix} 0.04 \\ 0.01 \end{bmatrix}$
No learning	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 1.0 \end{bmatrix}$	0	$\begin{bmatrix} 0.07 \\ 0.01 \end{bmatrix}$

We generate a time series for the initial conditions $(x(0), y(0)) = (-0.1, 0.1)$ and simulate for 5000 points. Fig. 4 shows the true and approximated dynamics as well as the difference between the true and approximated dynamics using the learned kernel and without learning the kernel.

We also consider a parameterized family of kernels of the form

$$k(x, y) = \alpha_{0,i}^2 \max\{0, 1 - \frac{\|x - y\|_2^2|}{\sigma_{0,i}}\} + \alpha_{1,i}^2 e^{\frac{\|x-y\|_2^2}{\sigma_{1,i}^2}} + \alpha_{2,i}^2 e^{-\frac{\|x-y\|_2}{\sigma_{2,i}^2}}$$
$$+ \alpha_{3,i}^2 e^{-\sigma_{3,i} \sin^2(\sigma_{4,i}\pi \|x-y\|_2^2)} e^{-\frac{\|x-y\|_2^2}{\sigma_{5,i}^2}} + \alpha_{4,i}^2 \|x - y\|_2^2. \quad (13)$$

We initialize with a Gaussian kernel. The results are summarized in the following table where R_1 corresponds to the RMSE with $x(0) = 0.4$ and R_2 corresponds to the RMSE with $x(0) = 0.97$ and 5000 points (see Table 3).

3.3.1. Finding τ

Now, we consider the scalar dimensional version of the Hénon map as $x(k+1) = 1 - ax(k)^2 + bx(k-1)$. We aim at learning the kernel and finding the optimal time delay τ. We start with an initial condition $(x(0), y(0)) = (0.8, -0.9)$ and generate 100 points for learning. We use a kernel of the form

$$k(x, y) = \alpha_0 + (\beta_0 + \|x - y\|_2^{\gamma_0})^{\sigma_0}.$$

We generate 100 points for different values of τ from 0 to 6. Fig. 5 shows the root mean square error (RMSE) for prediction with 5000 points and initial condition $(x(0), y(0)) = (0.1, -0.1)$. It shows that $\tau = 1$ is where the RMSE starts stabilizing and can be viewed as an optimal embedding delay.

Another method for finding the embedding delay is the Kernel Mode Decomposition (KMD) [36] of the time series. We consider a representation of the time series as

$$v(t+1) = \sum_{j=0}^{N} \alpha_j K(V_{\tau^\dagger}(t), V_{\tau^\dagger}(j)), \quad (14)$$

with $V_{\tau^\dagger}(t) = [v(t) \cdots v(t - \tau^\dagger)]$. Following [36], we define the model alignment energy $\mathscr{E}_i$ associated to the time-shift $\tau = i$, $i = 0, \ldots, \tau_{\max}$ as

$$\mathscr{E}_i = v^T K^{-1} K_i K^{-1} v, \quad (15)$$

with

$$K(x, y) = \sum_{i=0}^{\tau_{\max}} K_i(x, y), \quad (16)$$

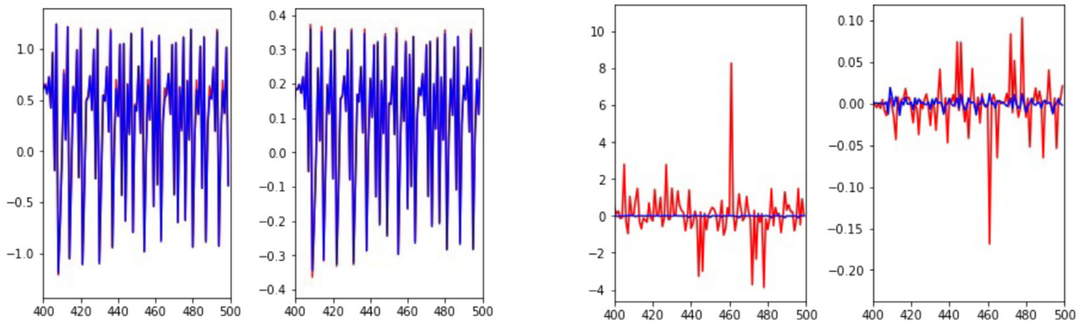
and $K_i(x, y) = K(\mathscr{S}_i x, \mathscr{S}_i y)$ with $\mathscr{S}_i$ the time-truncation operator that truncates time-series at the i-th element: given a time series $Y = \{Y_t : t \in \mathbb{T}\}$, where $\mathbb{T}$ is the index set, $\mathscr{S}_i Y = \{[y(t-i) \cdots y(t)] : t \in \mathbb{T}\}$.

We use the embedding delay $\tau^\dagger$ that maximizes $\mathscr{E}_i$. We apply this method to $x(k+1) = 1 - ax(k)^2 + bx(k-1)$. We use $K(x, y) = 1 + e^{-\|x-y\|_2^2}$ to compute the energies of the embedding

[6] We notice that the algorithm converges to non-integer powers. Terms of the form $\|x-y\|_2^\alpha$ can be represented as $e^{\alpha \log \|x-y\|_2}$ which could be a reproducing kernel.

Table 3

	$\begin{bmatrix} \alpha_{0,1} & \sigma_{0,1} & \alpha_{1,1} & \sigma_{1,1} & \alpha_{2,1} & \sigma_{2,1} & \alpha_{3,1} & \sigma_{3,1} & \sigma_{4,1} & \sigma_{5,1} & \alpha_{4,1} \\ \alpha_{0,2} & \sigma_{0,2} & \alpha_{1,2} & \sigma_{1,2} & \alpha_{2,2} & \sigma_{2,2} & \alpha_{3,2} & \sigma_{3,2} & \sigma_{4,2} & \sigma_{5,2} & \alpha_{4,2} \end{bmatrix}$	N	R_1
ρ	$\begin{bmatrix} 4.48\,10^{-08} & 1.00 & 2.25 & 2.41 & 0.0 & 1.01 & 0.17 & 1.07 & 1.17 & 1.21 & 0.60 \\ 0.18 & 0.96 & 1.09 & 2.30 & 0.20 & 1.00 & 0.26 & 1.03 & 1.11 & 0.84 & 1.65\,10^{-14} \end{bmatrix}$	5000	$\begin{bmatrix} 0.05 \\ 0.008 \end{bmatrix}$
No learning	$\begin{bmatrix} 0.0 & 1.0 & 1.0 & 1.0 & 0.0 & 1.0 & 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 1.0 & 1.0 & 0.0 & 1.0 & 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \end{bmatrix}$	0	$\begin{bmatrix} 0.08 \\ 0.01 \end{bmatrix}$



(a) True (blue) and approximated dynamics with the learned kernel (red) ($x-$ component on the left, $y-$ component on the right)

(b) Difference between the true and the approximated dynamics with the learned kernel (blue), with the initial kernel (red) ($x-$ component on the left, $y-$ component on the right)

Fig. 4. Prediction results for the Hénon map. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

delays and get that $\mathscr{E}_1$ is the maximal value and we deduce that the optimal embedding delay is 1 which agrees with the model.

Considering the Hénon map in the y-variable, we get $y(k+2) = b - \frac{a}{b}y^2(k+1) + by(k)$. We compute the energy $\mathscr{E}_i$ of the embedding delay i, observe that $\mathscr{E}_1$ is the maximal value and deduce that the optimal embedding delay is 1 which agrees with the model.

Fig. 5 shows the values of the energies of the time-delays for both the x-dynamics and y-dynamics.

3.3.2. Using partial information to approximate the dynamics

In order to learn the dynamics with partial information using measurements from x only, we use the kernel

$$k_i(x,y) = \alpha_{1,i}^2 \max(0, 1 - \frac{\|x-y\|^2}{\sigma_{1,i}}) + \alpha_{2,i}^2 e^{-\frac{\|x-y\|^2}{\sigma_{2,i}^2}} + \alpha_{3,i}^2\|x-y\|^2 + \alpha_{4,i}^2 e^{-\frac{\|x-y\|}{\sigma_{4,i}}},$$

and $\tau = 1$, i.e. we learn kernels for the mappings $\begin{pmatrix} x(k) \\ x(k-1) \end{pmatrix} \mapsto x(k+1)$ and $\begin{pmatrix} x(k) \\ x(k-1) \end{pmatrix} \mapsto y(k+1)$. We use 50 points with initial condition $x(0), x(1) = (0.9, -0.9)$ for training and the parameters of the learned kernel are summarized in the following table. Fig. 6 shows the results for initial conditions $(x(0), x(1)) = (-0.83, 0.57)$ with RMSE R_1 (see Table 4).

3.4. Example 4 (The Lorenz system)

Consider the Lorenz system

$$\frac{dx}{dt} = s(y-x), \tag{17}$$

$$\frac{dy}{dt} = rx - y - xz, \tag{18}$$

$$\frac{dz}{dt} = xy - bz, \tag{19}$$

with $s = 10$, $r = 28$, $b = 10/3$. We use the initial condition $(x(0), y(0), z(0)) = (0., 1., 1.05)$ and generate 10,000 (training) points with a time step $h = 0.01$.

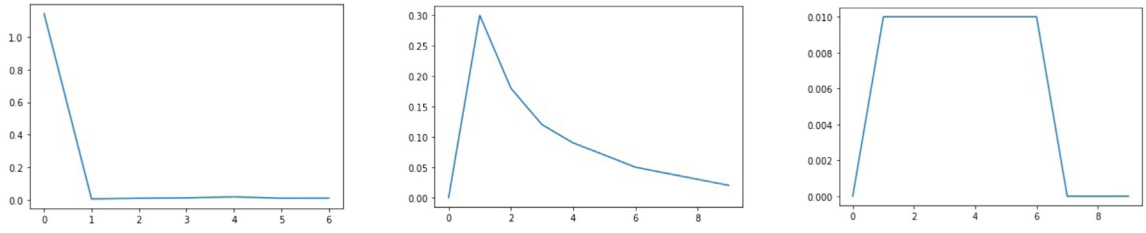
We randomly pick $N = 100$ points out of the original 10,000 points to train the kernel at each iteration (i.e. at each iteration we use 100 randomly selected points to compute the gradient of ρ and move the parameters in the gradient descent direction by one small step) and use the last random selection of $N = 100$ points for interpolation (prediction). We use a kernel of the form

$$K_i(x,y) = \alpha_{0,i} + (\alpha_{1,i} + \|x-y\|_2)^{\beta_i} + \alpha_{2,i} e^{(-\|x-y\|_2^2/\sigma_i^2)},$$

for $i = 1, 2, 3$. The table below summarizes the results for training using ρ and ρ_L as well as the RMSE for an initial condition $(x(0), y(0), z(0)) = (0.5, 1.5, 2.5)$ and 50,000 points

	$\begin{bmatrix} \alpha_{0,1} & \alpha_{1,1} & \beta_1 & \alpha_{2,1} & \sigma_1 \\ \alpha_{0,2} & \alpha_{1,2} & \beta_2 & \alpha_{2,2} & \sigma_2 \\ \alpha_{0,3} & \alpha_{1,3} & \beta_3 & \alpha_{2,3} & \sigma_3 \end{bmatrix}$	No. of iterations	R_1
ρ	$\begin{bmatrix} 1.00 & 0.95 & 2.02 & 0.94 & 1.08 \\ 1.00 & 1.02 & 1.79 & 0.98 & 1.00 \\ 1.00 & 0.99 & 1.90 & 0.99 & 1.00 \end{bmatrix}$	1000	$\begin{bmatrix} 0.0003 \\ 0.04 \\ 0.01 \end{bmatrix}$
ρ_L	$\begin{bmatrix} 0.55 & 2.5 & 0.6 & 0.55 & 0.95 \\ 0.55 & 2.5 & 0.6 & 0.55 & 0.95 \\ 0.55 & 2.5 & 0.6 & 0.55 & 0.95 \end{bmatrix}$	10,000	$\begin{bmatrix} 0.39 \\ 0.31 \\ 0.43 \end{bmatrix}$
No learning	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 1.0 \end{bmatrix}$	0	$\begin{bmatrix} 55.55 \\ 68.42 \\ 50.19 \end{bmatrix}$

Fig. 7 shows the results for an initial condition $(x(0), y(0), z(0)) = (0.5, 1.5, 2.5)$ and 10,000 points. Fig. 8 shows the prediction errors for the case of an approximation with a learned kernel and a kernel without learning. Fig. 9 shows the projection of the attractor and its approximation with a learned kernel and a kernel without learning. Fig. 10 shows the attractor with a learned kernel and a kernel without learning.



(a) The RMSE as a function of τ

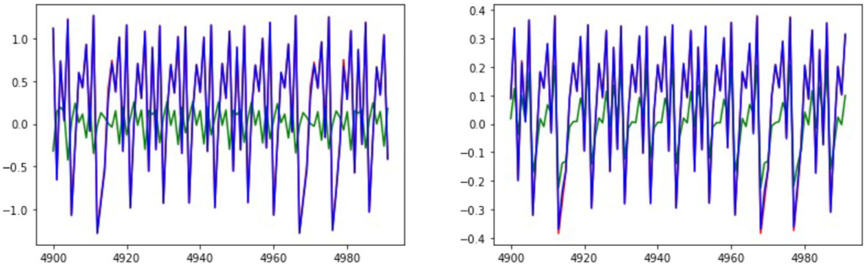
(b) Energy of the time-delays using KMD for the $x-$dynamics

(c) Energy of the time-delays using KMD for the $y-$dynamics

Fig. 5. Energy of the time-delays using RMSE and KMD.

Table 4

	$\begin{bmatrix} \alpha_{1,1} & \sigma_{1,1} & \alpha_{2,1} & \sigma_{2,1} & \alpha_{3,1} & \sigma_{3,1} & \alpha_{4,1} \\ \alpha_{1,2} & \sigma_{1,2} & \alpha_{2,2} & \sigma_{2,2} & \alpha_{3,2} & \sigma_{3,2} & \alpha_{4,2} \end{bmatrix}$	No. of it.	R_1
ρ	$\begin{bmatrix} 1.5\ 10^{-15} & 1.0 & 7.02 & -2.94 & -6.75 & 4.9\ 10^{-47} & 0.07 \\ 0.21 & 0.75 & 1.70 & 3.54 & 3.7\ 10^{-27} & 0.13 & 0.91 \end{bmatrix}$	5000	$\begin{bmatrix} 0.019 \\ 0.005 \end{bmatrix}$
No learning	$\begin{bmatrix} 0.0 & 1.0 & 1.0 & 1.0 & 0.0 & 1.0 & 1.0 \\ 0.0 & 1.0 & 1.0 & 1.0 & 0.0 & 1.0 & 1.0 \end{bmatrix}$	0	$\begin{bmatrix} 0.87 \\ 0.14 \end{bmatrix}$



(a) $x-$component

(b) $y-$component

Fig. 6. True dynamics (red), approximated dynamics with the learned kernel (blue), with the kernel without learning (green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

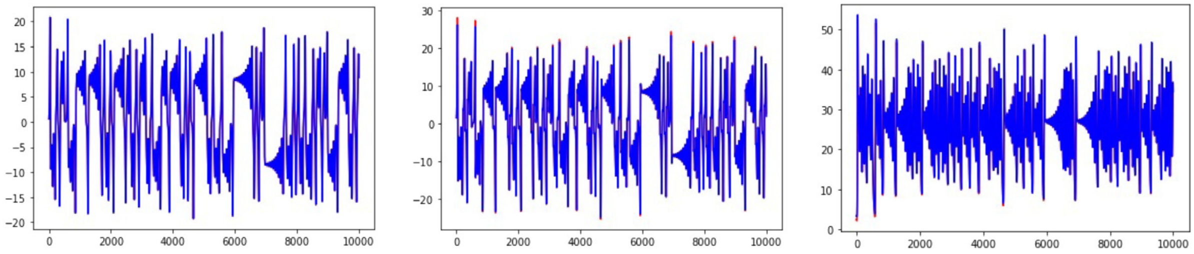


Fig. 7. Time series generated by the true dynamics (red) and the approximation with the learned kernel (blue) - x component in the left figure, y component in the middle figure, z component in the right figure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

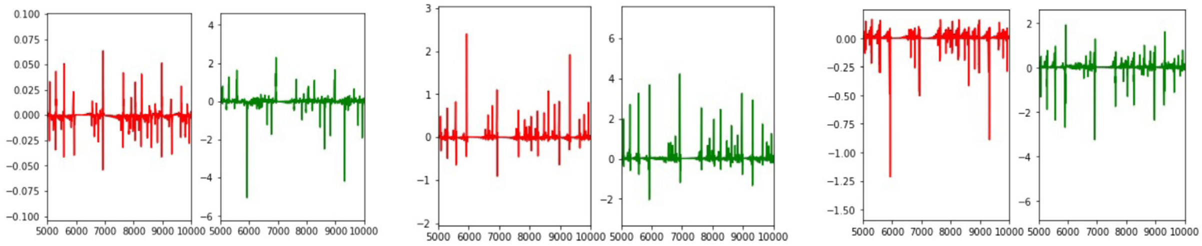


Fig. 8. Difference between the true and the approximated dynamics with the learned kernel using ρ (red (first, third and fifth from the left)), with the initial kernel (green (second, fourth and sixth from the left)). x-component in the two figures at the left, y-component in the middle two figures, z-component in the right two figures.

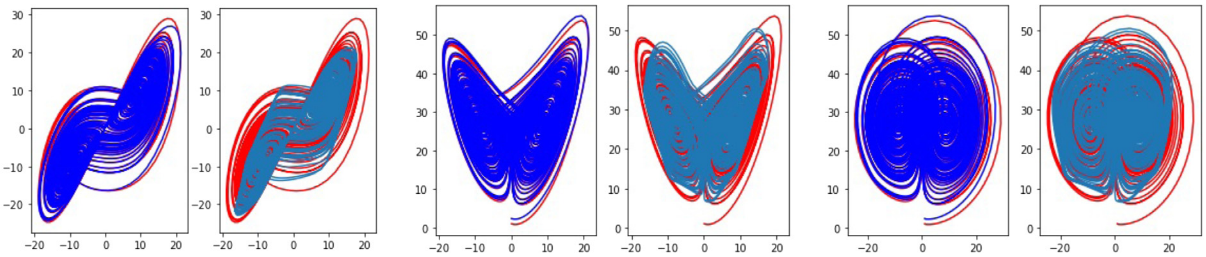


Fig. 9. Projection of the true attractor and approximation of the attractor using a learned kernel on the XY, XZ and YZ axes (first, third and fifth from the left), Projection of the true attractor and approximation of the attractor using with initial kernel on the XY, XZ and YZ axes (second, fourth and sixth from the left).

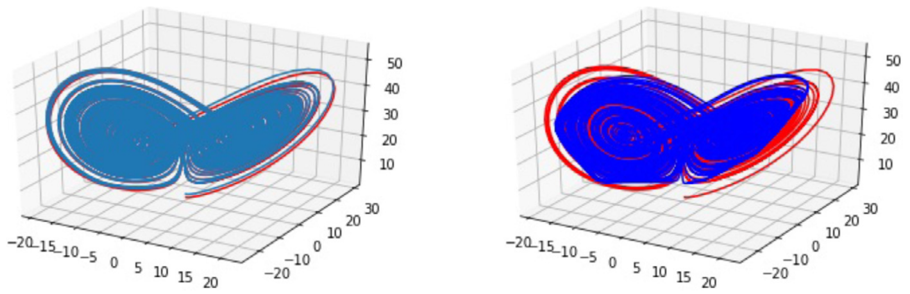


Fig. 10. True attractor (blue) and approximation of the attractor using a learned kernel (red) [left], True attractor (blue) and approximation of the attractor using initial kernel (red) [right]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We also consider a parameterized family of kernels of the form

$$K_i(x, y) = \alpha_{0,i}^2 \max\{0, 1 - \frac{\|x-y\|_2^2|}{\sigma_{0,i}}\} + \alpha_{1,i}^2 e^{\frac{\|x-y\|_2^2}{\sigma_{1,i}^2}} + \alpha_2^2 e^{-\frac{\|x-y\|_2}{\sigma_{2,i}^2}}$$
$$+ \alpha_{3,i}^2 e^{-\sigma_{3,i}\sin^2(\sigma_{4,i}\pi\|x-y\|_2^2)} e^{-\frac{\|x-y\|_2^2}{\sigma_{5,i}^2}} + \alpha_{4,i}^2\|x-y\|_2^2. \quad (20)$$

The training and prediction results are shown in the following table with R_1 the RMSE corresponding to 50,000 points with initial conditions $(0.5, 1.5, 2.5)$ (see Table 5).

Remarks.

i. Convergence results that characterize the error estimates of the difference between a dynamical system and its approximation from data using kernel methods can be found in [22,23].
ii. In the case of very large datasets, it is possible to reduce the number of points during training by considering greedy techniques as in [37,38].
iii. It is possible to include new measurements when approximating the dynamics from data without repeating the learning process. This can be done by working in Newton basis as in [39].
iv. During the numerical experiments, we noticed a tradeoff between accuracy and robustness in the choice of family of kernels, i.e. a richer family kernels can lead to more accurate results but seems to be less robust to perturbations originating from the optimization algorithm. This is consistent with the Bayesian interpretation of Gaussian process regression and the extreme lack of robustness of Bayesian inference with respect to the selection of the prior [40–43].

4. Conclusion

Our experiments suggest that using cross-validation (with KF and variants) to learn the kernel used to approximate the vector field of a dynamical system, and thereby its dynamics, significantly improves the accuracy of such approximations. Although our paper is entirely numerical, the simplicity of the proposed approach and the diversity of the experiments raise the question of the existence of a general and fundamental convergence theorem for cross-validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

B.H. thanks the European Commission for funding through the Marie Curie fellowship STALDYS-792919 (Statistical Learning for Dynamical Systems). H.O. gratefully acknowledges support by the Air Force Office of Scientific Research, USA under award number FA9550-18-1-0271 (Games for Computation and Learning). We thank Deniz Eroğlu, Yoshito Hirata, Jeroen Lamb, Edmilson Roque, Gabriele Santin and Yuzuru Sato for useful comments.

Appendix. Reproducing kernel hilbert spaces

We give a brief overview of reproducing kernel Hilbert spaces as used in statistical learning theory [16]. Early work developing the theory of RKHS was undertaken by N. Aronszajn [44].

Definition A.1. Let $\mathscr{H}$ be a Hilbert space of functions on a set $\mathscr{X}$. Denote by $\langle f, g\rangle$ the inner product on $\mathscr{H}$ and let $\|f\| = \langle f, f\rangle^{1/2}$ be the norm in $\mathscr{H}$, for f and $g \in \mathscr{H}$. We say that $\mathscr{H}$ is a reproducing kernel Hilbert space (RKHS) if there exists a function $K : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ such that
i. $K_x := K(x, \cdot) \in \mathscr{H}$ for all $x \in \mathscr{X}$.
ii. K spans $\mathscr{H}$: $\mathscr{H} = \overline{\mathrm{span}\{K_x \mid x \in \mathscr{X}\}}$.
iii. K has the *reproducing property*: $\forall f \in \mathscr{H}, f(x) = \langle f, K_x\rangle$.

K will be called a reproducing kernel of $\mathscr{H}$. $\mathscr{H}_K$ will denote the RKHS $\mathscr{H}$ with reproducing kernel K where it is convenient to explicitly note this dependence.

Table 5

	$\begin{bmatrix} \alpha_{0,1} & \sigma_{0,1} & \alpha_{1,1} & \sigma_{1,1} & \alpha_{2,1} & \sigma_{2,1} & \alpha_{3,1} & \sigma_{3,1} & \sigma_{4,1} & \sigma_{5,1} & \alpha_{4,1} \\ \alpha_{0,2} & \sigma_{0,2} & \alpha_{1,2} & \sigma_{1,2} & \alpha_{2,2} & \sigma_{2,2} & \alpha_{3,2} & \sigma_{3,2} & \sigma_{4,2} & \sigma_{5,2} & \alpha_{4,2} \\ \alpha_{0,3} & \sigma_{0,3} & \alpha_{1,3} & \sigma_{1,3} & \alpha_{2,3} & \sigma_{2,3} & \alpha_{3,3} & \sigma_{3,3} & \sigma_{4,3} & \sigma_{5,3} & \alpha_{4,3} \end{bmatrix}$	n	R_1
ρ	$\begin{bmatrix} 0.16 & 0.99 & 1.59 & 0.98 & 0.15 & 0.99 & 0.16 & 1.00 & 1.00 & 0.99 & -31.28 \\ -1.03 & 0.99 & -10.96 & 0.10 & -1.18 & 0.97 & -1.07 & 1.00 & 1.00 & 0.99 & 60.87 \\ 0.07 & 0.99 & 0.68 & 0.89 & 0.07 & 1.00 & 0.07 & 1.00 & 0.99 & 0.99 & 0.79 \end{bmatrix}$	1000	$\begin{bmatrix} 1.0\,10^{-11} \\ 0.24 \\ 0.17 \end{bmatrix}$
	$\begin{bmatrix} 0.0 & 1.0 & 1.0 & 1.0 & 0.0 & 1.0 & 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 1.0 & 1.0 & 0.0 & 1.0 & 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 1.0 & 1.0 & 0.0 & 1.0 & 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \end{bmatrix}$	0	$\begin{bmatrix} 54.25 \\ 70.21 \\ 674.92 \end{bmatrix}$

The important properties of reproducing kernels are summarized in the following proposition.

Proposition A.1. *If K is a reproducing kernel of a Hilbert space $\mathscr{H}$, then*
i. $K(x,y)$ is unique.
ii. $\forall x, y \in \mathscr{X}$, $K(x,y) = K(y,x)$ (symmetry).
iii. $\sum_{i,j=1}^{q} \alpha_i \alpha_j K(x_i, x_j) \geq 0$ for $\alpha_i \in \mathbb{R}$, $x_i \in \mathscr{X}$ and $q \in \mathbb{N}_+$ (positive definiteness).
iv. $\langle K(x,\cdot), K(y,\cdot)\rangle = K(x,y)$.

Common examples of reproducing kernels defined on a compact domain $\mathscr{X} \subset \mathbb{R}^n$ are the (1) constant kernel: $K(x,y) = k > 0$ (2) linear kernel: $K(x,y) = x \cdot y$ (3) polynomial kernel: $K(x,y) = (1 + x \cdot y)^d$ for $d \in \mathbb{N}_+$ (4) Laplace kernel: $K(x,y) = e^{-\|x-y\|_2/\sigma^2}$, with $\sigma > 0$ (5) Gaussian kernel: $K(x,y) = e^{-\|x-y\|_2^2/\sigma^2}$, with $\sigma > 0$ (6) triangular kernel: $K(x,y) = \max\{0, 1 - \frac{\|x-y\|_2^2}{\sigma}\}$, with $\sigma > 0$. (7) locally periodic kernel: $K(x,y) = \sigma^2 e^{-2\frac{\sin^2(\pi\|x-y\|_2/p)}{\ell^2}} e^{-\frac{\|x-y\|_2^2}{2\ell^2}}$, with $\sigma, \ell, p > 0$.

Theorem A.1. *Let $K : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ be a symmetric and positive definite function. Then there exists a Hilbert space of functions $\mathscr{H}$ defined on $\mathscr{X}$ admitting K as a reproducing Kernel. Conversely, let $\mathscr{H}$ be a Hilbert space of functions $f : \mathscr{X} \to \mathbb{R}$ satisfying $\forall x \in \mathscr{X}, \exists \kappa_x > 0$, such that $|f(x)| \leq \kappa_x \|f\|_{\mathscr{H}}, \quad \forall f \in \mathscr{H}$. Then $\mathscr{H}$ has a reproducing kernel K.*

Theorem A.2. *Let $K(x,y)$ be a positive definite kernel on a compact domain or a manifold X. Then there exist a Hilbert space $\mathscr{F}$ and a function $\Phi : X \to \mathscr{F}$ such that*

$$K(x,y) = \langle \Phi(x), \Phi(y) \rangle_{\mathscr{F}} \quad \text{for} \quad x, y \in X.$$

Φ is called a feature map, and $\mathscr{F}$ a feature space.[7]

A.1. Function approximation in RKHSs: An optimal recovery viewpoint

In this section we review function approximation in RKHSs from the point of view of optimal recovery as discussed in [32].

Problem. **P**: Given input/output data $(x_1, y_1), \ldots, (x_N, y_N) \in \mathscr{X} \times \mathbb{R}$, recover an unknown function u^* mapping $\mathscr{X}$ to $\mathbb{R}$ such that $u^*(x_i) = y_i$ for $i \in \{1, \ldots, N\}$.

In the setting of optimal recovery [32] Problem **P** can be turned into a well posed problem by restricting candidates for u to belong to a Banach space of functions $\mathscr{B}$ endowed with a norm $\|\cdot\|$ and identifying the optimal recovery as the minimizer of the relative error

$$\min_v \max_u \frac{\|u - v\|^2}{\|u\|^2}, \tag{21}$$

where the max is taken over $u \in \mathscr{B}$ and the min is taken over candidates in $v \in \mathscr{B}$ such that $v(x_i) = u(x_i) = y_i$. For the validity of the constraints $u(x_i) = y_i$, $\mathscr{B}^*$, the dual space of $\mathscr{B}$, must contain delta Dirac functions $\phi_i(\cdot) = \delta(\cdot - x_i)$. This problem can be stated as a game between Players I and II and can then be represented as

$$\begin{tikzcd} \text{(Player I)} \;\; u \in \mathscr{B} \arrow[dr, "\text{max}"'] & & v \in L(\Phi, \mathscr{B}) \;\; \text{(Player II)} \arrow[dl, "\text{min}"] \\ & \frac{\|u - v(u)\|}{\|u\|}\,. & \end{tikzcd} \tag{22}$$

If $\|\cdot\|$ is quadratic, i.e. $\|u\|^2 = [Q^{-1}u, u]$ where $[\phi, u]$ stands for the duality product between $\phi \in \mathscr{B}^*$ and $u \in \mathscr{B}$ and $Q : \mathscr{B}^* \to \mathscr{B}$ is a positive symmetric linear bijection (i.e. such that $[\phi, Q\phi] \geq 0$ and $[\psi, Q\phi] = [\phi, Q\psi]$ for $\phi, \psi \in \mathscr{B}^*$). In that case the optimal solution of (21) has the explicit form

$$v^* = \sum_{i,j=1}^{N} u(x_i) A_{i,j} Q\phi_j, \tag{23}$$

where $A = \Theta^{-1}$ and $\Theta \in \mathbb{R}^{N \times N}$ is a Gram matrix with entries $\Theta_{i,j} = [\phi_i, Q\phi_j]$.

To recover the classical representer theorem, one defines the reproducing kernel K as

$$K(x,y) = [\delta(\cdot - x), Q\delta(\cdot - y)]$$

In this case, $(\mathscr{B}, \|\cdot\|)$ can be seen as an RKHS endowed with the norm

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

and (23) corresponds to the classical representer theorem

$$v^*(\cdot) = y^T A K(x, \cdot), \tag{24}$$

using the vectorial notation $y^T A K(x,\cdot) = \sum_{i,j=1}^{N} y_i A_{i,j} K(x_j, \cdot)$ with $y_i = u(x_i)$, $A = \Theta^{-1}$ and $\Theta_{i,j} = K(x_i, x_j)$.

Now, let us consider the problem of learning the kernel from data. As introduced in [28], 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

$$\rho = \frac{\|v^* - v^s\|^2}{\|v^*\|^2} \tag{25}$$

which is the relative error between v^*, the optimal recovery (24) 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 $X^s = \{(x_i, y_i) \mid i \in \mathscr{S}\}$ ($\text{Card}(\mathscr{S}) = N/2$) which admits the representation

$$v^s = (y^s)^T A^s K(x^s, \cdot) \tag{26}$$

[7] The dimension of the feature space can be infinite, for example in the case of the Gaussian kernel.

with $y^s = \{y_i \mid i \in \mathscr{S}\}$, $x^s = \{x_i \mid i \in \mathscr{S}\}$, $A^s = (\Theta^s)^{-1}$, $\Theta^s_{i,j} = K(x^s_i, x^s_j)$. This quantity ρ is directly related to the game in (22) where one is minimizing the relative error of v^* versus v^s. Instead of using the entire the dataset X one may use random subsets X^{s_1} (of X) for v^* and random subsets X^{s_2} (of X^{s_1}) for v^s.

Replacing $\|u^*\|_{\mathscr{H}}$ by the RKHS norm of the interpolant of v^* (with both testing and training points) in (4) gives an error interval for $v^*(x)$ in (24) as

$$v^*(x) \pm \Delta(v^*(x)), \tag{27}$$

with

$$\Delta(v^*(x)) = \sigma(x)\sqrt{Y^{f,T} K(X^f, X^f)^{-1} Y^f}, \tag{28}$$

and where (X^f, Y^f) corresponds to the concatenation of the training and testing points. Local error estimates such as (27) are classical in Kriging [45] (see also [46][Thm. 5.1] for applications to PDEs).

A.2. The maximum mean discrepancy

Let $\mathscr{P}$ be the set of Borel probability measures on $\mathscr{X}$. Given a probability distribution P we define its kernel mean embedding (with respect to a kernel k with RKHS $\mathscr{H}$) as

$$\begin{array}{rcl} \mu_P : \mathscr{P} & \rightarrow & \mathscr{H} \\ P & \mapsto & \int_{\mathscr{X}} k(x,y)dP(y) =: \mu_k(P) \end{array}$$

The maximum mean discrepancy (MMD) between two probability measures P and Q is then defined as the distance between two such embeddings and can be expressed as

$$\begin{array}{rcl} \mathrm{MMD}(P,Q) & := & \|\mu_P - \mu_Q\|_{\mathscr{H}}, \\ & = & \Big(\mathbb{E}_{x,x'}(k(x,x')) + \mathbb{E}_{y,y'}(k(y,y')) \\ & & -2\mathbb{E}_{x,y}(k(x,y)) \Big)^{\frac{1}{2}} \end{array}$$

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.

Given i.i.d. samples from $X := \{x_1, \dots, x_m\}$ and $Y := \{y_1, \dots, y_n\}$, from P and Q respectively, recall that the MMD in RKHSs is defined as the difference between the kernel mean embeddings defined as follows. Given i.i.d, samples $(x_1, \dots, x_m)$ from P and $(y_1, \dots, y_n)$ from Q, the MMD between the empirical distributions $(\delta_{x_1} + \cdots + \delta_{x_m})/m$ and $(\delta_{y_1} + \cdots + \delta_{y_n})/n$ is an unbiased estimate of $\mathrm{MMD}(P,Q)$ with the representation

$$\mathrm{MMD}^2_u := \frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i,x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(y_i,y_j) - \frac{2}{nm}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i,y_j) \tag{29}$$

References

[1] George.E.P. Box, Gwilym M. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, 1976.
[2] H. Abarbanel, Analysis of Observed Chaotic Data, in: Institute for Nonlinear Science, Springer, New York, 2012.
[3] Holger Kantz, Thomas Schreiber, Nonlinear Time Series Analysis, Cambridge University Press, USA, 1997.
[4] A. Nielsen, Practical Time Series Analysis: Prediction with Statistics and Machine Learning, O'Reilly Media, 2019.
[5] R.H. Shumway, D.S. Stoffer, Time Series Analysis and Its Applications: with R Examples, in: Springer Texts in Statistics, Springer, New York, 2010.
[6] Martin Casdagli, Nonlinear prediction of chaotic time series, Physica D 35 (3) (1989) 335–356.
[7] Klaus-Robert Müller, Alex J. Smola, Gunnar Rätsch, Bernhard Schölkopf, Jens Kohlmorgen, Vladimir Vapnik, Predicting time series with support vector machines, in: Proceedings of the 7th International Conference on Artificial Neural Networks, ICANN '97, Springer-Verlag, Berlin, Heidelberg, 1997, pp. 999–1004.
[8] S. Mukherjee, E. Osuna, F. Girosi, Nonlinear prediction of chaotic time series using support vector machines, in: Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop, 1997, pp. 511–520.
[9] Jaideep Pathak, Zhixin Lu, Brian R. Hunt, Michelle Girvan, Edward Ott, Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data, Chaos 27 (12) (2017) 121102.
[10] Zhixin Lu, Brian R. Hunt, Edward Ott, Attractor reconstruction by machine learning, Chaos 28 (6) (2018) 061104.
[11] Steven L. Brunton, Joshua L. Proctor, J. Nathan Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (15) (2016) 3932–3937.
[12] M. Budišić, R. Mohr, I. Mezić, The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications, Springer, 2020.
[13] Steven L. Brunton, W. Bingni, Joshua L. Proctor, Eurika Kaiser, J. Nathan Kutz, Chaos as an intermittently forced linear system, Nature Commun. 8 (1) (2017) 19.
[14] Liva Ralaivola, Florence d'Alché Buc, Dynamical modeling with kernels for nonlinear time series prediction, in: S. Thrun, L.K. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems, Vol. 16, MIT Press, 2004, pp. 129–136.
[15] Ashesh Chattopadhyay, Pedram Hassanzadeh, Krishna V. Palem, Devika Subramanian, Data-driven prediction of a multi-scale lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ann, and RNN-LSTM, 2019, CoRR, abs/1906.08829.
[16] Felipe Cucker, Steve Smale, On the mathematical foundations of learning, Bull. Amer. Math. Soc. 39 (2002) 1–49.
[17] J. Bouvrie, B. Hamzi, Balanced reduction of nonlinear control systems in reproducing kernel hilbert space, in: Proc. 48th Annual Allerton Conference on Communication, Control, and Computing, 2010, pp. 294–301, http://arxiv.org/abs/1011.2952.
[18] J. Bouvrie, B. Hamzi, Balanced reduction of nonlinear control systems in reproducing kernel hilbert space, in: Proc. of the 2012 American Control Conference, 2012, pp. 294–301, http://arxiv.org/abs/1204.0563.
[19] J. Bouvrie, B. Hamzi, Kernel methods for the approximation of some key quantities of nonlinear systems, J. Comput. Dyn. (1) (2017) http://arxiv.org/abs/1204.0563.
[20] B. Haasdonk, B. Hamzi, G. Santin, D. Wittwar, Greedy kernel methods for center manifold approximation, in: Proc. of ICOSAHOM 2018, International Conference on Spectral and High Order Methods, (1), 2018, https://arxiv.org/abs/1810.11329.
[21] Stefan Klus, Feliks Nüske, Boumediene Hamzi, Kernel-based approximation of the koopman generator and schrödinger operator, Entropy (2020) https://arxiv.org/abs/2005.13231.
[22] J. Bouvrie, B. Hamzi, Kernel methods for the approximation of nonlinear systems, SIAM J. Control Optim. (2017) https://arxiv.org/abs/1108.2903.
[23] P. Giesl, B. Hamzi, M. Rasmussen, K. Webster, Approximation of Lyapunov functions from noisy data, J. Comput. Dyn. (2019) https://arxiv.org/abs/1601.01568.
[24] Andreas Bittracher, Stefan Klus, Boumediene Hamzi, Péter Koltai, Christof Schütte, Dimensionality reduction of complex metastable systems via kernel embeddings of transition manifolds, 2019, https://arxiv.org/abs/1904.08622.
[25] Stefan Klus, Feliks Nuske, Boumediene Hamzi, Kernel-based approximation of the koopman generator and schrödinger operator, Entropy 22 (2020) https://www.mdpi.com/1099-4300/22/7/722.
[26] Stefan Klus, Feliks Nüske, Sebastian Peitz, Jan-Hendrik Niemann, Cecilia Clementi, Christof Schütte, Data-driven approximation of the koopman generator: Model reduction, system identification, and control, Physica D 406 (2020) 132416.
[27] Romeo Alexander, Dimitrios Giannakis, Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques, Physica D 409 (2020) 132520.
[28] H. Owhadi, G.R. Yoo, Kernel flows: From learning kernels from data into the abyss, J. Comput. Phys. 389 (2019) 22–47.
[29] Yifan Chen, Houman Owhadi, Andrew M. Stuart, Consistency of empirical bayes and kernel flow for hierarchical parameter estimation, 2020, https://arxiv.org/abs/2005.11375.
[30] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, Alexander Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (25) (2012) 723–773.
[31] Gene Ryan Yoo, Houman Owhadi, Deep regularization and direct training of the inner layers of neural networks with kernel flows, 2020, https://arxiv.org/abs/2002.08335.
[32] Houman Owhadi, Clint Scovel, Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization: From a Game Theoretic Approach to Numerical Approximation and Algorithm Design, in: Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2019.
[33] J.P. Eckmann, S. Oliffson Kamphorst, D. Ruelle, S. Ciliberto, Liapunov exponents from time series, Phys. Rev. A 34 (1986) 4971–4979.

[34] Christopher Schölzel, Nonlinear measures for dynamical systems, 2019, https://doi.org/10.5281/zenodo.3814723.

[35] W. Melo, S. Strien, One-Dimensional Dynamics, Springer, 1993.

[36] Houman Owhadi, Clint Scovel, Gene Ryan Yoo, Kernel mode decomposition and programmable/interpretable regression networks, 2019, https://arxiv.org/abs/1907.08592.

[37] G. Santin, B. Haasdonk, Kernel methods for surrogate modelling, 2019, ArXiv e-prints arXiv:1907.10556, https://arxiv.org/abs/1907.10556.

[38] Florian Schäfer, Matthias Katzfuss, Houman Owhadi, Sparse cholesky factorization by kullback-leibler minimization, 2020, https://arxiv.org/abs/2004.14455.

[39] Maryam Pazouki, Robert Schaback, Bases for kernel-based spaces, J. Comput. Appl. Math. 236 (4) (2011) 575–588, International Workshop on Multivariate Approximation and Interpolation with Applications (MAIA 2010).

[40] Houman Owhadi, Clint Scovel, Tim Sullivan, On the brittleness of bayesian inference, SIAM Rev. 57 (4) (2015) 566–582.

[41] Houman Owhadi, Clint Scovel, Tim Sullivan, et al., Brittleness of bayesian inference under finite information in a continuous world, Electron. J. Stat. 9 (1) (2015) 1–79.

[42] Houman Owhadi, Clint Scovel, Qualitative robustness in bayesian inference, ESAIM Probab. Stat. 21 (2017) 251–274.

[43] H. Owhadi, C. Scovel, Brittleness of Bayesian inference and new Selberg formulas, Commun. Math. Sci. 14 (2016) 83–145, arXiv:1304.7046.

[44] N. Aronszajn, Theory of reproducing kernels, Trans. Amer. Math. Soc. 68 (3) (1950) 337–404.

[45] Zong min Wu, Robert Schaback, Local error estimates for radial basis function interpolation of scattered data, IMA J. Numer. Anal. 13 (1992) 13–27.

[46] Houman Owhadi, Bayesian numerical homogenization, Multiscale Model. Simul. 13 (3) (2015) 812–828.