Prediction of High-Dimensional Time Series with Exogenous Variables Using Generalized Koopman Operator Framework in Reproducing Kernel Hilbert Space

Jia-Chen Hua[0000−0002−6591−1417]1⋆, Farzad Noorian[2], Philip H.W. Leong[2], Gemunu Gunaratne[3], and Jorge Gonçalves[1]

[1] Luxembourg Centre for Systems Biomedicine,
University of Luxembourg, 4367 Belvaux, Luxembourg
[2] School of Electrical and Information Engineering,
University of Sydney, NSW 2006, Australia
[3] Department of Physics,
University of Houston, Houston, Texas 77204, USA

Abstract. We propose a novel methodology to predict high-dimensional time series with exogenous variables using Koopman operator framework, by assuming that the time series are generated by some underlying unknown dynamical system with input as exogenous variables. In order to do that, we first generalize the definition of the original Koopman operator to allow for input to the underlying dynamical system. We then obtain a formulation of the generalized Koopman operator in reproducing kernel Hilbert space (RKHS) and a new derivation of its numerical approximation methods, namely, Extended Dynamic Mode Decomposition (EDMD) and its kernel-based version. We also obtain a statistical interpretation of kernel-based EDMD developed for deterministic Koopman operator by utilizing the connection between RKHS and Gaussian processes regression, and relate it to the stochastic Koopman and Perron-Frobenius operator. In applications, we found that the prediction performance of this methodology is promising in forecasting real world high-dimensional time series with exogenous variables, including financial markets data. We believe that this methodology will be of interest to the community of scientists and engineers working on quantitative finance, econometrics, system biology, neurosciences, meteorology, oceanography, system identification and control, data mining, machine learning, computational intelligence, and many other fields involving high-dimensional time series and spatio-temporal data.

Keywords: High-dimensional time-series, Spatio-temporal dynamics, Complex system, Koopman operator, Perron-Frobenius operator, Dynamical system, Reproducing kernel Hilbert space, Gaussian Processes, Machine learning, Data mining, Econophysics, Financial markets modeling, Energy forecasting, Collective behavior

⋆ The corresponding author would like to thank Dr. Alexandre Mauroy for insightful discussions on generalizing Koopman operator to systems with input.

1 Introduction

In many application fields, a high-dimensional time series $\{\boldsymbol{x}_t\}$ may be considered as being generated by or sampled from some underlying dynamical system $(\mathbb{R}^N, t, \boldsymbol{F}^t)$ that is often nonlinear and stochastic, where $\boldsymbol{x}_t \mapsto \boldsymbol{F}^{\tau}(\boldsymbol{x}_t) = \boldsymbol{x}_{t+\tau} \in \mathbb{R}^N$ is the N-dimensional state vector evolved by the flow $\boldsymbol{F}$. However, these high-dimensional state vectors are usually extrinsic measurements or outputs of the underlying lower dimensional true state's dynamics $\boldsymbol{z}_{t+\tau} = \hat{\boldsymbol{F}}^{\tau}(\boldsymbol{z}_t)$. When the task is to predict each component of the output $\boldsymbol{x}_t$, using some machine learning techniques to learn its superficial dynamics $\boldsymbol{F}^t$ without identifying the true state dynamics $\boldsymbol{z}_{t+\tau} = \hat{\boldsymbol{F}}^{\tau}(\boldsymbol{z}_t)$ may be computationally heavy and not optimal. Nevertheless, if there is no need for identification other than prediction of outputs, it is often favorable to find some intrinsic feature maps $\{\varphi_i(\boldsymbol{x}_t)\}_{i=1}^{M}$ (where $M < N$ and the M-dimensional feature vector $\boldsymbol{\varphi}(\boldsymbol{x}_t)$ is not necessarily the same as the underlying true state $\boldsymbol{z}_t$) to embed the high-dimensional output to lower dimensional intrinsic manifold, or in other words, to learn both the geometry *and* dynamics for *simultaneous* dimensionality reduction and prediction, as shown in the schematic Figure 1. Therefore, the question is how to find such $\{\varphi_i(\boldsymbol{x}_t)\}_{i=1}^{M}$ that given the latest output $\boldsymbol{x}_t$, one can predict feature maps' future values $\boldsymbol{\varphi}(\boldsymbol{x}_{t+\tau})$ and transform back to the pre-image $\boldsymbol{x}_{t+\tau}$, and how is $\boldsymbol{\varphi}(\boldsymbol{x}_{t+\tau})$ related to $\boldsymbol{\varphi}(\boldsymbol{x}_t)$ on the intrinsic manifold. The key to these questions is the Koopman operator of dynamical systems [1–3], whose eigenfunctions can serve as the desired intrinsic feature maps $\{\varphi_i(\boldsymbol{x}_t)\}$. The Koopman operator is a linear operator that enables investigation of a nonlinear dynamical system using linear theories and techniques, and since it has been developed as a data-driven framework [4, 5], most of its applications up to now are dealing with high-dimensional time series. There have been several major numerical methods developed to extract the spectral properties of Koopman operator from time series data, and utilizing these properties for time series prediction has several major advantages [6].

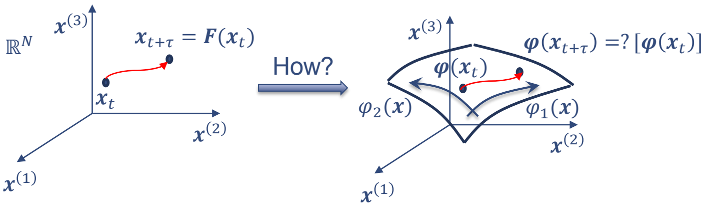
$\mathbb{R}^N$
$\boldsymbol{x}^{(3)}$
$\boldsymbol{x}_{t+\tau} = \boldsymbol{F}(\boldsymbol{x}_t)$
$\boldsymbol{x}_t$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}^{(1)}$
How?
$\boldsymbol{x}^{(3)}$
$\boldsymbol{\varphi}(\boldsymbol{x}_{t+\tau}) =? [\boldsymbol{\varphi}(\boldsymbol{x}_t)]$
$\boldsymbol{\varphi}(\boldsymbol{x}_t)$
$\varphi_2(\boldsymbol{x})$
$\varphi_1(\boldsymbol{x})$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}^{(1)}$

Fig. 1 Manifold learning that is capable of simultaneous dimensionality reduction and prediction.

In this paper, we generalize the Koopman operator framework to systems with inputs as exogenous variables. By taking the simplest generalization approach [7], we found that the techniques and methods that we developed for

Kernel KMR [6] methodology can be utilized almost directly with minimal modification. Hence we can generalize Kernel KMR to Kernel GKMRX (Kernel-based Generalized Koopman Mode Regression with eXogenous variables) to predict high dimensional time series with exogenous variables. In the theory part of this paper, we formulate the Koopman operator in reproducing kernel Hilbert space (RKHS), which is the most important function space in modern machine learning, and we obtain a new derivation of the Extended Dynamic Mode Decomposition (EDMD) algorithm [5] and its kernel-based extension [8] by exploiting the Dirac bra–ket notation [9]. Moreover, we obtain a statistical interpretation of these numerical methods developed for deterministic Koopman operator by exploiting the connection between RKHS and Gaussian processes regression, and relate it to the stochastic Koopman and Perron-Frobenius operator. In the application part, we test our new prediction methodology for various types of data from different fields and obtain promising initial results.

2 Theory

2.1 Koopman operator of dynamical system and its generalization to systems with input

Consider a high dimensional time series $\{\boldsymbol{x}_n\}$ sampled from an underlying dynamical system $(\mathcal{M}, n, \boldsymbol{F})$, where $n \in \mathbb{Z}$ is discrete time, $\mathcal{M} \subset \mathbb{R}^N$ is the N-dimensional state space containing the $\{\boldsymbol{x}_n\}$, and $\boldsymbol{x}_i \mapsto \boldsymbol{F}(\boldsymbol{x}_i) = \boldsymbol{x}_{i+1}$ defines the evolution law. For continuous-time dynamical system $(\mathcal{M}, t, \boldsymbol{F}^t)$, the flow $\boldsymbol{F}^t$ evolves the system state as $\boldsymbol{x}_0 \mapsto \boldsymbol{F}^t(\boldsymbol{x}_0) = \boldsymbol{x}_t$. Since time series data are often sampled with a fixed time gap τ, the adjacent two snapshots of the system are related by $\boldsymbol{F}^\tau(\boldsymbol{x}_t) = \boldsymbol{x}_{t+\tau}$. When the context is clear, we will drop the τ in $\boldsymbol{F}^\tau$ to denote either the discrete time map or continuous time flow of a fixed time gap τ. Here we restrict to stationary time series, or at least locally stationary time series, which can be considered as being sampled from autonomous dynamical systems. We will generalize the Koopman operator to systems with input later.

The (deterministic) Koopman operator $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ is defined as $(\mathcal{K}\phi)(\boldsymbol{x}) = (\phi \circ \boldsymbol{F})(\boldsymbol{x}) = \phi(\boldsymbol{F}(\boldsymbol{x}))$, where $\circ$ denotes the composition of ϕ with $\boldsymbol{F}$, and $\mathcal{F}$ is the "feature space" consisting of scalar observables or functions of state space $\phi : \mathcal{M} \to \mathbb{C}$. Since $\mathcal{K}\phi$ is another element in $\mathcal{F}$, the Koopman operator defines a new dynamical system $(\mathcal{F}, n, \mathcal{K})$ where $\mathcal{K}$ evolves an observable or feature $\phi \in \mathcal{F}$ to a new function $\mathcal{K}\phi$ that gives the value of ϕ at "one step in the future". Unlike $\boldsymbol{F}$ which is finite dimensional, $\mathcal{K}$ is infinite dimensional because it acts on function space $\mathcal{F}$. However, it is also linear even when $\boldsymbol{F}$ is nonlinear, and hence one can investigates its spectral properties, *i.e.,* eigenvalues and eigenfunctions, which we refer to as Koopman eigenvalues $\{\mu_k\}$ and eigenfunctions $\{\varphi_k\}$.

Technical results showed that $\mathcal{K}$ fully characterizes $\boldsymbol{F}$ under very general conditions [2, 3], so dynamical systems $(\mathcal{M}, n, \boldsymbol{F})$ and $(\mathcal{F}, n, \mathcal{K})$ are two different representations of the same evolution, as shown in the schematic Figure 2. The link between these two representations is the "full state observable" $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$, where $\boldsymbol{x} \mapsto \boldsymbol{F}(\boldsymbol{x})$, and $g_i \mapsto (\mathcal{K}g_i) = g_i \circ \boldsymbol{F}$ where $g_i \in \mathcal{F}$ is the i-th component of the *vector-valued observable* $\boldsymbol{g} : \mathcal{M} \to \mathbb{R}^N$. Assuming g_i is in the span of a set of K Koopman eigenfunctions $\{\varphi_k\}_{k=1}^K$, where K could (and often will) be infinite, then it can be projected as $g_i = \sum_{k=1}^K \xi_{ik}\varphi_k$ with $\xi_{ik} \in \mathbb{C}$. Hence $\boldsymbol{g}$ can be obtained by "stacking" these weights into vectors (i.e., $\boldsymbol{\xi}_j = [\xi_{1j}, \xi_{2j}, \ldots, \xi_{Nj}]^T$). As a result,

$$\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{x}) = \sum_{k=1}^{K} \boldsymbol{\xi}_k \varphi_k(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{\xi}_k$ is the k-th *Koopman mode* corresponding to the eigenfunction φ_k. To make prediction or arrive at the system state of "one step in the future", one can either evolve $\boldsymbol{x}$ through $\boldsymbol{F}$ directly, or evolve the full state observable $\boldsymbol{g}(\boldsymbol{x})$ through the Koopman operator $\mathcal{K}$ as $\boldsymbol{g}(\boldsymbol{F}(\boldsymbol{x})) = (\mathcal{K}\boldsymbol{g})(\boldsymbol{x}) = \sum_{k=1}^K \boldsymbol{\xi}_k(\mathcal{K}\varphi_k)(\boldsymbol{x})$ $= \sum_{k=1}^K \mu_k \boldsymbol{\xi}_k \varphi_k(\boldsymbol{x})$. Similarly, for continuous time case, we have $\boldsymbol{x}_{t+\tau} = \boldsymbol{F}^\tau(\boldsymbol{x}_t)$ $= \boldsymbol{g}(\boldsymbol{F}^\tau(\boldsymbol{x}_t)) = (\mathcal{K}_\tau \boldsymbol{g})(\boldsymbol{x}_t) = \sum_{k=1}^K \mathrm{e}^{\lambda_k \tau} \boldsymbol{\xi}_k \varphi_k(\boldsymbol{x}_t)$, where λ_k and φ_k are the k-th eigenvalue and eigenfunction of the infinitesimal generator $\hat{\mathcal{K}} \triangleq \frac{\mathrm{d}}{\mathrm{d}t}$ of the semi-group of Koopman operators $\{\mathcal{K}_t\}_{t\in\mathbb{R}^+}$, and $\mu_k = \mathrm{e}^{\lambda_k \tau}$ is the k-th eigenvalue of finite-time Koopman operator $\mathcal{K}_\tau = \mathrm{e}^{\tau\hat{\mathcal{K}}}$.

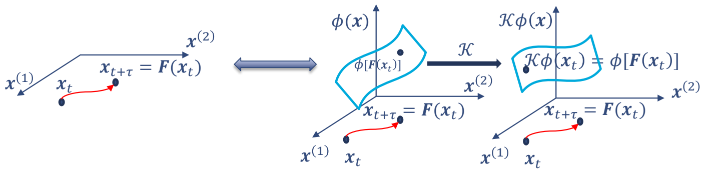
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}_t$
$\boldsymbol{x}_{t+\tau} = \boldsymbol{F}(\boldsymbol{x}_t)$
$\phi(\boldsymbol{x})$
$\phi[\boldsymbol{F}(\boldsymbol{x}_t)]$
$\mathcal{K}$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}_{t+\tau} = \boldsymbol{F}(\boldsymbol{x}_t)$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}_t$
$\mathcal{K}\phi(\boldsymbol{x})$
$\mathcal{K}\phi(\boldsymbol{x}_t) = \phi[\boldsymbol{F}(\boldsymbol{x}_t)]$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}_{t+\tau} = \boldsymbol{F}(\boldsymbol{x}_t)$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}_t$

Fig. 2 Koopman operator is the "pullback" of $\boldsymbol{F}$: when the system evolves from $\boldsymbol{x}_t$ to $\boldsymbol{F}(\boldsymbol{x}_t)$ in state space, a function ϕ defined on the state space is evolved by $\mathcal{K}$ such that the new function $\mathcal{K}\phi$ evaluated at old state $\boldsymbol{x}_t$ is taking the value of old function ϕ evaluated at new state $\boldsymbol{F}(\boldsymbol{x}_t)$.

In order to compute $\{(\mu_k, \varphi_k, \boldsymbol{\xi}_k)\}_{k=1}^K$ of Koopman eigenvalues, eigenfunctions, and modes from data, one has to find a matrix representation of $\mathcal{K}$ by projecting it into some subspace of $\mathcal{F}$ spanned by a basis $\{\psi_k(\boldsymbol{x})\}_{k=1}^K$. For computational feasibility and convenience, we usually require $\psi_k(\cdot) \in L^2(\mathcal{M})$, such that we can compute inner products using training data $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_M, \boldsymbol{y}_M)\}$ where $\boldsymbol{y}_i = \boldsymbol{F}(\boldsymbol{x}_i)$, in order to require $\{\psi_k(\boldsymbol{x})\}_{k=1}^K$ to be orthonormal by computing the Moore–Penrose pseudoinverse of the data matrix $\boldsymbol{\Psi}_x^+$, where $[\boldsymbol{\Psi}_x]_{ij} = \psi_j(\boldsymbol{x}_i)$. Exploiting Dirac's bra–ket notation [9] to write functions, functionals, inner products, and linear operators in a compact way, we denote the i-th row of $\boldsymbol{\Psi}_x^+$ as $\langle\psi_i|$ such that the inner product $\langle\psi_i|\psi_j\rangle_{L^2} = \delta_{ij}$, where δ_{ij} is the Kronecker delta. Hence in this "feature space" $\mathcal{F}_K \triangleq \operatorname{span}\{\psi_k(\cdot)\}_{k=1}^K$, the identity operator can be written as $\mathbb{1}_{\mathcal{F}_K} = \sum_{k=1}^K |\psi_k\rangle\langle\psi_k|$, and $\mathcal{K}$ projected to $\mathcal{F}_K$ can be written as $\mathcal{K} = \mathcal{K}\mathbb{1}_{\mathcal{F}_K} = \mathcal{K}\sum_{k=1}^K |\psi_k\rangle\langle\psi_k| = \sum_{k=1}^K |\psi_k \circ \boldsymbol{F}\rangle\langle\psi_k|$. Therefore, the elements of matrix representation $\boldsymbol{K}$ of $\mathcal{K}$ is $\boldsymbol{K}_{ij} = \langle\psi_i|\mathcal{K}|\psi_j\rangle_{L^2} =$

$\langle\psi_i|\psi_j \circ \boldsymbol{F}\rangle_{L^2}$, and $\boldsymbol{K} = \boldsymbol{\Psi}_x^+ \boldsymbol{\Psi}_y$, where the j-th column of $\boldsymbol{\Psi}_y$ is $|\psi_j \circ \boldsymbol{F}\rangle$ and $[\boldsymbol{\Psi}_y]_{ij} = \psi_j \circ \boldsymbol{F}(\boldsymbol{x}_i) = \psi_j(\boldsymbol{y}_i)$. Eigenvalue problem $\mathcal{K}|\varphi_k\rangle = \mu_k|\varphi_k\rangle$ becomes eigenvalue equation of $\boldsymbol{K}$ as $\boldsymbol{K}\boldsymbol{v}_k = \mu_k \boldsymbol{v}_k$, where the i-th component of $\boldsymbol{v}_k$ is $(\boldsymbol{v}_k)_i = \langle\psi_i|\varphi_k\rangle_{L^2}$, so the eigenfunction $|\varphi_k\rangle = \sum_{i=1}^{K} |\psi_i\rangle(\boldsymbol{v}_k)_i$, or $\boldsymbol{\Phi}_x = \boldsymbol{\Psi}_x \boldsymbol{V}$ in matrix notation, where $[\boldsymbol{\Phi}_x]_{ij} = \varphi_j(\boldsymbol{x}_i)$ and columns of $\boldsymbol{V}$ are $\{\boldsymbol{v}_k\}$. The *continuous-time eigenvalue* can be computed as $\lambda_k \triangleq \log(\mu_k)/\tau$, and according to Eq. (1), Koopman modes $\{\boldsymbol{\xi}_k\}$ can be computed by projecting $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$ onto $\{\varphi_k(\boldsymbol{x})\}$ as $\boldsymbol{\Xi} = \boldsymbol{\Phi}_x^+ \boldsymbol{X}$, where the i-th rows of $\boldsymbol{\Xi}$ and $\boldsymbol{X}$ are $\boldsymbol{\xi}_i^T$ and $\boldsymbol{x}_i^T$, respectively. This procedure is called extended Dynamic Mode Decomposition (EDMD)[5] and it has become one of the most widely adopted numerical methods for data-driven Koopman spectral analysis, even outside the fluid dynamics community where the Koopman operator's spectral properties was throughly investigated for the first time [2].

Furthermore, there are several ways to generalize Koopman operator to systems with input [7]. One of the simplest ways is to augment the system state $\boldsymbol{x}_t$ with the current input $\boldsymbol{u}_t \in \mathbb{R}^{N'}$, such that the dimension of the extended system state $\tilde{\boldsymbol{x}}$ will be $N + N'$. The time evolution of the system will be extended as $\tilde{\boldsymbol{x}}_{t+\tau} = \tilde{\boldsymbol{F}}^\tau(\tilde{\boldsymbol{x}}_t) = \tilde{\boldsymbol{F}}^\tau(\boldsymbol{x}_t, \boldsymbol{u}_t)$, where the first N components of $\tilde{\boldsymbol{F}}$ and $\tilde{\boldsymbol{x}}$ are $\boldsymbol{x}_{t+\tau} = \boldsymbol{F}^\tau(\boldsymbol{x}_t, \boldsymbol{u}_t)$, and we assume that there is a purely formal map or flow that "shifts" the input as $\boldsymbol{u}_{t+\tau} = \boldsymbol{S}^\tau(\boldsymbol{x}_t, \boldsymbol{u}_t)$, since there is not necessarily any "dynamics" of the input. The generalized Koopman operator can be defined on this extended system as before $\mathcal{K}\phi(\tilde{\boldsymbol{x}}_t) = \phi \circ \tilde{\boldsymbol{F}}^\tau(\tilde{\boldsymbol{x}}_t)$. For prediction purposes, we are only interested in the original system state $\boldsymbol{x}$, so there is no need to project N' dimensional full state observable of input $\boldsymbol{g}_u(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{u}_t$ on Koopman eigenfunctions in order to compute the corresponding Koopman modes for input. Except for this trivial difference, all the available numerical procedures for Koopman spectral analysis and prediction can be applied with very little modification. Notice that this augmentation trick can be also applied to previous state and input, such that one can investigate a system with finite amount of memory in the same way as investigating a system without memory. For simplicity, we only consider memoryless system in this paper, and this topic will be left for future investigation.

2.2 Reproducing kernel Hilbert space and Gaussian processes regression

In this subsection, we briefly summarize the basic theory of reproducing kernel Hilbert space (RKHS) and its relation to Gaussian processes regression. For a more complete exposition of this topic with technical details, we refer the readers to Refs. [10, 11].

The RKHS is a Hilbert space of functions equipped with inner product $\langle\cdot|\cdot\rangle_{\mathcal{H}_k}$ satisfying: (1) $\forall \boldsymbol{x}$ fixed, $k(\boldsymbol{x}, \boldsymbol{y}) = k_{\boldsymbol{x}}(\cdot) \in \mathcal{H}_k$ is a function of $\boldsymbol{y}$; (2) $k(\cdot,\cdot)$ has the "reproducing" property: $\forall f \in \mathcal{H}_k$, $\langle f(\cdot)|k_{\boldsymbol{x}}(\cdot)\rangle_{\mathcal{H}_k} = f(\boldsymbol{x})$. It follows from (2) that $\langle k_{\boldsymbol{y}}(\cdot)|k_{\boldsymbol{x}}(\cdot)\rangle_{\mathcal{H}_k} = k_{\boldsymbol{y}}(\boldsymbol{x}) = k_{\boldsymbol{x}}(\boldsymbol{y}) = k(\boldsymbol{y}, \boldsymbol{x})$. Each RKHS has a unique k, and according to Moore-Aronszajn theorem, given any symmetric positive definite function $k(\boldsymbol{y}, \boldsymbol{x})$, there is a unique RKHS such that $k(\boldsymbol{y}, \boldsymbol{x})$ is the reproducing kernel. In fact, this theorem showed that this unique RKHS $\{f \in \mathcal{H}_k | f(\cdot) = \sum_{i=1}^{M\to\infty} \alpha_i k(\cdot, \boldsymbol{x}_i)\}$ can be built from defining the inner product $\langle f|g\rangle_{\mathcal{H}_k} = \sum_{j=1}^{M'\to\infty} \sum_{i=1}^{M\to\infty} \alpha_i \beta_j k(\boldsymbol{y}_j, \boldsymbol{x}_i)$, where $g(\cdot) = \sum_{j=1}^{M'\to\infty} \beta_j k(\cdot, \boldsymbol{y}_j)$. It satisfies the reproducing property $\langle f(\cdot)|k_{\boldsymbol{x}}(\cdot)\rangle_{\mathcal{H}_k} = \sum_{i=1}^{M\to\infty} \langle \alpha_i k(\cdot, \boldsymbol{x}_i)|k_{\boldsymbol{x}}(\cdot)\rangle_{\mathcal{H}_k}$ $= \sum_{i=1}^{M\to\infty} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) = f(\boldsymbol{x})$. The reproducing kernels can be considered as a basis of this RKHS, and they are also called “point evaluation functionals”.

Another representation of RKHS is from Mercer’s theorem, which states that a positive (semi-)definite function can be eigen-decomposed as $k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{\infty} \sigma_i q_i(\boldsymbol{x}) q_i(\boldsymbol{x}')$, where $\{q_i(\cdot)\}$ are orthonormal in L^2, and $\{\sigma_i\}_{i=1}^{M\to\infty}$ is a non-increasing sequence of eigenvalues with $\sigma_M \to 0$ when $M \to \infty$. It follows from this theorem that the unique RKHS associated to this $k(\boldsymbol{x}, \boldsymbol{x}')$ is $\{f \in L^2 | \sum_{i=1}^{\infty} \frac{\langle q_i|f\rangle_{L^2}^2}{\sigma_i} < \infty\}$, and the inner product is given by $\langle f|g\rangle_{\mathcal{H}_k} = \sum_{i=1}^{\infty} \langle f|q_i\rangle_{L^2} \frac{1}{\sigma_i} \langle q_i|g\rangle_{L^2}$. One consequence of this inner product is that the induced norm is $\|f\|_{\mathcal{H}_k}^2 = \langle f|f\rangle_{\mathcal{H}_k} = \sum_{i=1}^{\infty} \frac{\langle q_i|f\rangle_{L^2}^2}{\sigma_i}$, and in order to be bounded, the components $f_i = \langle q_i|f\rangle_{L^2}$ must decay quickly when i increases, which effectively imposes a smoothness requirement on L^2 in order for it to become a RKHS. Another consequence of this inner product is that one can define $\{p_i(\cdot) = \sqrt{\sigma_i} q_i(\cdot)\}$ such that it is an orthonormal basis of this unique RKHS, and as an analogue to the Dirac delta which can be represented by $\delta_{\boldsymbol{x}}(\cdot) = \sum_{i=1}^{\infty} q_i(\boldsymbol{x}) q_i(\cdot)$, the reproducing kernel functions can be written as $k_{\boldsymbol{x}}(\cdot) = \sum_{i=1}^{\infty} p_i(\boldsymbol{x}) p_i(\cdot)$.

For a regularized optimization problem $J[f] = \frac{1}{2\lambda_M^2} \sum_{i=1}^{M} (y_i - f(\boldsymbol{x}_i))^2 + \frac{1}{2} \|f\|_{\mathcal{H}_k}^2$ given some training data or observations $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_M, y_M)\}$, where $\boldsymbol{x}_i \in \mathbb{R}^N$ and $y_i \in \mathbb{R}$, the representer theorem [12] asserts that the minimizer $\hat{f}(\cdot) = \sum_{i=1}^{M} \alpha_i k(\cdot, \boldsymbol{x}_i)$, such that one can effectively minimize $J[\alpha_i]$ by setting the derivatives with respect to α_i equal to zeros, and then the α_i’s can be solved as a column vector $\boldsymbol{\alpha} = (\boldsymbol{G} + \lambda_M^2 \boldsymbol{I})^{-1} \boldsymbol{y}$, where $\boldsymbol{y} = [y_1, \cdots, y_M]^T$ are the training outputs, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{G}$ is the kernel Gramian matrix where $\boldsymbol{G}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Given a new test data $\boldsymbol{x}_*$, the predicted function output is $\hat{f}(\boldsymbol{x}_*) = \boldsymbol{k}(\boldsymbol{x}_*)^T (\boldsymbol{G} + \lambda_M^2 \boldsymbol{I})^{-1} \boldsymbol{y}$, where $\boldsymbol{k}(\boldsymbol{x}_*)^T = [k(\boldsymbol{x}_*, \boldsymbol{x}_1), \cdots, k(\boldsymbol{x}_*, \boldsymbol{x}_M)]$. This is the same as the posterior mean of Gaussian processes regression with i.i.d. noise variance λ_M^2.

A more heuristic view of predicting the function output given a new test data is from the point evaluation at this new data. As an analogue to the point evaluation in L^2 using Dirac delta $f(\boldsymbol{x}_*) = \langle \delta_{\boldsymbol{x}_*}|f\rangle_{L^2} = \int f(\boldsymbol{x}) \delta(\boldsymbol{x} - \boldsymbol{x}_*) d\boldsymbol{x}$ (which is computationally infeasible using training data), one can work in the RKHS using the reproducing kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ as: $f(\boldsymbol{x}_*) = \langle k_{\boldsymbol{x}_*}|f\rangle_{\mathcal{H}_k} = \sum_{i=1}^{M} \langle k_{\boldsymbol{x}_*}|q_i\rangle_{L^2} \frac{1}{\sigma_i} \langle q_i|f\rangle_{L^2}$, where the inner products in L^2 can be approximated by summation using training data as $\langle g|f\rangle_{L^2} = \int g(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} \approx \sum_{i=1}^{M} g(\boldsymbol{x}_i) f(\boldsymbol{x}_i)$ $= \sum_{i=1}^{M} \langle g|k_{\boldsymbol{x}_i}\rangle_{\mathcal{H}_k} \langle k_{\boldsymbol{x}_i}|f\rangle_{\mathcal{H}_k}$. Hence one can obtain $\sum_{i=1}^{M} \langle k_{\boldsymbol{x}_*}|q_i\rangle_{L^2} \frac{1}{\sigma_i} \langle q_i|f\rangle_{L^2} \approx \sum_{ijl} \langle k_{\boldsymbol{x}_*}|k_{\boldsymbol{x}_j}\rangle_{\mathcal{H}_k} \langle k_{\boldsymbol{x}_j}|q_i\rangle_{\mathcal{H}_k} \frac{1}{\sigma_i} \langle q_i|k_{\boldsymbol{x}_l}\rangle_{\mathcal{H}_k} \langle k_{\boldsymbol{x}_l}|f\rangle_{\mathcal{H}_k}$. Notice that the kernel Gramian matrix has eigen-decomposition $\boldsymbol{G} = \boldsymbol{Q}\boldsymbol{\Sigma}^2\boldsymbol{Q}^T$, where $\boldsymbol{Q}_{ij} = q_j(\boldsymbol{x}_i) = \langle k_{\boldsymbol{x}_i}|q_j\rangle_{\mathcal{H}_k}$ and $\boldsymbol{\Sigma}$ is diagonal with $\boldsymbol{\Sigma}_{ii} = \sqrt{\sigma_i}$. Hence $\boldsymbol{G}^{-1} = \boldsymbol{Q}\boldsymbol{\Sigma}^{-2}\boldsymbol{Q}^T$ and $(\boldsymbol{G}^{-1})_{ij}$

$= \sum_{l=1}^{M} \langle k_{\boldsymbol{x}_i} | q_l \rangle_{\mathcal{H}_k} \frac{1}{\sigma_l} \langle q_l | k_{\boldsymbol{x}_j} \rangle_{\mathcal{H}_k}$. Finally one arrives at

$$f(\boldsymbol{x}_*) = \langle k_{\boldsymbol{x}_*} | f \rangle_{\mathcal{H}_k} = \boldsymbol{k}(\boldsymbol{x}_*)^T \boldsymbol{G}^{-1} [f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_M)]^T, \tag{2}$$

which is the same as the posterior mean in noiseless Gaussian processes regression. Replacing $\boldsymbol{G}^{-1}$ by the Moore–Penrose pseudoinverse $\boldsymbol{G}^{+}$ will be equivalent to regularization, or adding noise in Gaussian processes regression. A typical way of regularization using $\boldsymbol{G}^{+}$ is to truncate out some small eigenvalues σ_i's and the corresponding eigenvectors $q_i(\boldsymbol{x})$'s, although a more sophisticated way to perform this truncation is using a smooth cutoff developed in Ref. [6]. A useful result following the above derivation is that the inner product in RKHS can be approximated using training data as $\langle g | f \rangle_{\mathcal{H}_k} \approx \sum_{ij} \langle g | k_{\boldsymbol{x}_i} \rangle_{\mathcal{H}_k} [\boldsymbol{G}^{-1}]_{ij} \langle k_{\boldsymbol{x}_j} | f \rangle_{\mathcal{H}_k}$, which means that the "resolution of the identity" or the projection operator into this RKHS can be approximated by training data as $\mathbb{1}_{\mathcal{H}_k} = \sum_{i=1}^{M} |p_i\rangle\langle p_i| \approx \sum_{ij} |k_{\boldsymbol{x}_i}\rangle_{\mathcal{H}_k} [\boldsymbol{G}^{-1}]_{ij} \, {}_{\mathcal{H}_k}\langle k_{\boldsymbol{x}_j}|$.

In summary, deterministic approximation of a function in RKHS, or point evaluation of a function on new data can have a statistical interpretation via Gaussian processes regression. Moreover, since $\boldsymbol{k}(\boldsymbol{x}_*)^T \boldsymbol{G}^{-1}$ is a row vector of weights on the training outputs $[f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_M)]^T$, and if it sums up to 1 and if the amount of training data is sufficiently large, it may be considered as a density estimation for the posterior distribution of Gaussian processes, which will induce a density on the training data $[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M]^T$. A special case is the point evaluation on training data $f(\boldsymbol{x}_i) = \boldsymbol{k}(\boldsymbol{x}_i)^T \boldsymbol{G}^{-1} [f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_M)]^T$, where $\boldsymbol{k}(\boldsymbol{x}_i)^T \boldsymbol{G}^{-1}$ will become a row vector with every element equal to zero except for the i-th one equal to 1, which is a probability mass function concentrated on $\boldsymbol{x}_i$ that approximates the Dirac delta distribution $\delta_{\boldsymbol{x}_i}(\cdot)$. Again, replacing $\boldsymbol{G}^{-1}$ by the Moore–Penrose pseudoinverse $\boldsymbol{G}^{+}$ effectively corresponds to Gaussian processes with additive noise such that the Dirac delta will become a narrow Gaussian centered at the training data.

2.3 Koopman operator in reproducing kernel Hilbert space

To interpret Koopman operator in RKHS, first notice that it can be also defined as an integral operator [13–16], which enables a better and uniform formulation of both deterministic and stochastic Koopman operator, and its Hermitian adjoint, namely the Perron-Frobenius operator $\mathcal{L} = \mathcal{K}^{\dagger}$, where the † denotes Hermitian adjoint. Again, consider the dynamical system $(\mathcal{M}, t, \boldsymbol{F}^t)$. When $\boldsymbol{F}^t$ is highly nonlinear and/or stochastic, starting from an initial point on $\mathcal{M}$ and keeping track of its single trajectory along the time evolution will become meaningless, as any finite initial difference will blow up exponentially. Instead, a better strategy is to investigate the statistical behavior of a swarm of points' time evolution, which leads to the investigation of (probability) measure/density on $\mathcal{M}$ and its time evolution induced by $\boldsymbol{F}^t$. Consider a probability density function ρ defined on $\mathcal{M}$, and for computational convenience, we require $\rho \in \mathcal{F} \subseteq L^2(\mathcal{M})$. When $\boldsymbol{F}$ evolves an arbitrary swarm of points of system states on $\mathcal{M}$, *i.e.,* evolves the pre-image $\boldsymbol{F}^{-1}(\mathbb{A})$ of any measurable domain $\mathbb{A} \subseteq \mathcal{M}$ to $\mathbb{A}$ at time τ later, the density ρ on $\boldsymbol{F}^{-1}(\mathbb{A})$ will be evolved by a linear operator to a new density on $\mathbb{A}$ as $\int_{\mathbb{A}}(\mathcal{L}_\tau\rho)(\boldsymbol{y})d\boldsymbol{y} = \int_{\boldsymbol{F}^{-1}(\mathbb{A})}\rho(\boldsymbol{x})d\boldsymbol{x}$, such that the probability measure in conserved, where the $\mathcal{L}_\tau$ is the Perron-Frobenius operator that evolves probability densities. If $\boldsymbol{F}$ is stochastic, which means that $\boldsymbol{F}(\boldsymbol{x})$ follows a *transition probability density* $p_\tau(\boldsymbol{y}|\boldsymbol{x})$, the Perron-Frobenius operator can be also defined as

$$(\mathcal{L}_\tau\rho)(\boldsymbol{y}) = \int_{\boldsymbol{F}^{-1}(\mathbb{A})}\rho(\boldsymbol{x})p_\tau(\boldsymbol{y}|\boldsymbol{x})d\boldsymbol{x}. \tag{3}$$

A special case is the deterministic system, where $p_\tau(\boldsymbol{y}|\boldsymbol{x})$ will become a Dirac delta distribution $\delta_{\boldsymbol{F}(\boldsymbol{x})}(\boldsymbol{y}) = \delta(\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{x}))$, such that the center of an initial Dirac delta distribution $\delta_{\boldsymbol{x}}$ will be moved in consistence with the dynamics as $\mathcal{L}_\tau\delta_{\boldsymbol{x}}(\boldsymbol{y}) = \int_{\boldsymbol{F}^{-1}(\mathbb{A})}\delta(\boldsymbol{x}-\boldsymbol{x}')\delta(\boldsymbol{y}-\boldsymbol{F}(\boldsymbol{x}'))d\boldsymbol{x}' = \delta_{\boldsymbol{F}(\boldsymbol{x})}(\boldsymbol{y})$. Analogous to this, notice that Koopman operator for deterministic system is defined as $(\mathcal{K}_\tau h)(\boldsymbol{x}) = (h\circ\boldsymbol{F})(\boldsymbol{x}) = h(\boldsymbol{F}(\boldsymbol{x}))$, it can be also written as $(\mathcal{K}_\tau h)(\boldsymbol{x}) = \int_{\mathbb{A}} h(\boldsymbol{y})\delta(\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{x}))d\boldsymbol{y}$, and following this idea, the Koopman operator for stochastic system should be defined as

$$(\mathcal{K}_\tau h)(\boldsymbol{x}) = \int_{\mathbb{A}} h(\boldsymbol{y})p_\tau(\boldsymbol{y}|\boldsymbol{x})d\boldsymbol{y} = \mathbb{E}[h(\boldsymbol{F}(\boldsymbol{x}))|\boldsymbol{x}], \tag{4}$$

which is the conditional expectation of observable h's value at time τ later. Using these definitions, one can check that the Koopman operator and Perron-Frobenius operator are adjoint to each other for both deterministic and stochastic systems, by considering how the expectation value of an observable over some region evolves in time:

$$\begin{aligned}\mathbb{E}[h(\boldsymbol{y})] &= \int_{\mathbb{A}}(\mathcal{L}_\tau\rho)(\boldsymbol{y})h(\boldsymbol{y})d\boldsymbol{y} = \langle\mathcal{L}_\tau\rho|h\rangle_{L^2} = \int_{\mathbb{A}}\int_{\boldsymbol{F}^{-1}(\mathbb{A})}\rho(\boldsymbol{x})p_\tau(\boldsymbol{y}|\boldsymbol{x})d\boldsymbol{x}h(\boldsymbol{y})d\boldsymbol{y}\\ &= \int_{\boldsymbol{F}^{-1}(\mathbb{A})}\mathbb{E}[h(\boldsymbol{F}(\boldsymbol{x}))|\boldsymbol{x}]\rho(\boldsymbol{x})d\boldsymbol{x} = \int_{\boldsymbol{F}^{-1}(\mathbb{A})}(\mathcal{K}_\tau h)(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} = \langle\rho|\mathcal{K}_\tau h\rangle_{L^2},\end{aligned} \tag{5}$$

where $\mathcal{K}_\tau$ acting to the left on $\langle\rho|$ is $\langle\rho|\mathcal{K}_\tau|h\rangle_{L^2} = \langle\mathcal{K}_\tau^\dagger\rho|h\rangle_{L^2} = \langle\mathcal{L}_\tau\rho|h\rangle_{L^2}$. This formulation enables us to predict the expectation of a function's value at a later time when tracking and predicting a single trajectory is not meaningful due to high nonlinearity and/or stochasticity of $\boldsymbol{F}$, and we can relate this formulation to Koopman and Perron-Frobenius operators in RKHS as follow:

Recall from Eq. (2) that point evaluation in RKHS is the same as computing some expectation value such as the posterior mean of Gaussian processes, for example, $\langle k_{\boldsymbol{x}_i}|f\rangle_{\mathcal{H}_k} = \boldsymbol{k}(\boldsymbol{x}_i)^T\boldsymbol{G}^{-1}[f(\boldsymbol{x}_1),\cdots,f(\boldsymbol{x}_M)]^T$, where $\boldsymbol{k}(\boldsymbol{x}_i)^T\boldsymbol{G}^{-1}$ is a row vector with all zero elements except for the i-th equal to 1, which may be considered as discrete approximation to Dirac delta distribution $\delta_{\boldsymbol{x}_i}$. Replacing $\boldsymbol{G}^{-1}$ by pseudo-inverse $\boldsymbol{G}^+$ will be equivalent to regularization or adding noise to the Gaussian processes, such that $\boldsymbol{k}(\boldsymbol{x}_i)^T\boldsymbol{G}^+$ can approximate some narrow Gaussian centered at $\boldsymbol{x}_i$. Similarly, consider the projection of Koopman operator in RKHS by point evaluation of a function $|h\rangle$ evolved by $\mathcal{K}$ at a new state $\boldsymbol{x}_*$ as $\langle k_{\boldsymbol{x}_*}|\mathcal{K}|h\rangle_{\mathcal{H}_k} = \boldsymbol{k}(\boldsymbol{x}_*)^T\boldsymbol{G}^{-1}[\mathcal{K}h(\boldsymbol{x}_1),\cdots,\mathcal{K}h(\boldsymbol{x}_M)]^T$, where the $\boldsymbol{k}(\boldsymbol{x}_*)^T\boldsymbol{G}^{-1}$ is expected to approximate the initial density $\rho(\boldsymbol{x})$ before time evolution in Eq. (3), in the limit of infinite amount of training data, *i.e.,* $M \to \infty$.

On the other hand, recall that the identity operator in RKHS $\mathbb{1}_{\mathcal{H}_k} = \sum_i |p_i\rangle\langle p_i| \approx \sum_{ij} |k_{\boldsymbol{x}_i}\rangle_{\mathcal{H}_k} \boldsymbol{G}^{-1}\, {}_{\mathcal{H}_k}\langle k_{\boldsymbol{x}_j}|$, and inner product can also be approximated as $\langle g|\mathbb{1}_{\mathcal{H}_k}|f\rangle_{\mathcal{H}_k} \approx \sum_{ij}\langle g|k_{\boldsymbol{x}_i}\rangle_{\mathcal{H}_k}[\boldsymbol{Q}\boldsymbol{\Sigma}^{-2}\boldsymbol{Q}_y^T]_{ij}\langle k_{\boldsymbol{y}_j}|f\rangle_{\mathcal{H}_k}$, where $[\boldsymbol{Q}_y^T]_{ij} = \langle q_i|k_{\boldsymbol{y}_j}\rangle_{\mathcal{H}_k} = \sum_l \langle q_i|q_l\rangle_{L^2}\frac{1}{\sigma_l}\langle q_l|k_{\boldsymbol{y}_j}\rangle_{L^2} \approx \sum_l \frac{1}{\sigma_i}\langle q_i|k_{\boldsymbol{x}_l}\rangle_{\mathcal{H}_k}\langle k_{\boldsymbol{x}_l}|k_{\boldsymbol{y}_j}\rangle_{\mathcal{H}_k} = [\boldsymbol{\Sigma}^{-2}\boldsymbol{Q}^T\boldsymbol{K}^T]_{ij}$, and $\boldsymbol{K}_{ij} = \langle k_{\boldsymbol{x}_i}|\mathcal{K}|k_{\boldsymbol{x}_j}\rangle_{\mathcal{H}_k} = k_{\boldsymbol{x}_j}(\boldsymbol{F}(\boldsymbol{x}_i)) = k(\boldsymbol{y}_i, \boldsymbol{x}_j) = \langle k_{\boldsymbol{y}_i}|k_{\boldsymbol{x}_j}\rangle_{\mathcal{H}_k}$. It follows that $\boldsymbol{Q}\boldsymbol{\Sigma}^{-2}\boldsymbol{Q}_y^T = \boldsymbol{Q}\boldsymbol{\Sigma}^{-2}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{\Sigma}^{-2}\boldsymbol{Q}^T\boldsymbol{K}^T = \boldsymbol{G}^{-2}\boldsymbol{K}^T$, and hence $\mathbb{1}_{\mathcal{H}_k}$ can also be approximated by $\mathbb{1}_{\mathcal{H}_k} \approx \sum_{ij} |k_{\boldsymbol{x}_i}\rangle_{\mathcal{H}_k}[\boldsymbol{G}^{-2}\boldsymbol{K}^T]_{ij}\, {}_{\mathcal{H}_k}\langle k_{\boldsymbol{y}_j}|$. After substituting the $\mathbb{1}_{\mathcal{H}_k}$'s in $\langle k_{\boldsymbol{x}_*}|\mathbb{1}_{\mathcal{H}_k}\mathcal{K}\mathbb{1}_{\mathcal{H}_k}|h\rangle_{\mathcal{H}_k}$ with appropriate approximations, one can obtain

$$\langle k_{\boldsymbol{x}_*}|\mathcal{K}|h\rangle_{\mathcal{H}_k} \approx \boldsymbol{k}(\boldsymbol{x}_*)^T\boldsymbol{G}^{-1}\boldsymbol{K}\boldsymbol{G}^{-2}\boldsymbol{K}^T[h(\boldsymbol{y}_1), \cdots, h(\boldsymbol{y}_M)]^T. \tag{6}$$

When the number of training snapshots pairs $M \to \infty$, we would expect that $\boldsymbol{k}(\boldsymbol{x}_*)^T\boldsymbol{G}^{-1}$ approximates $\rho(\boldsymbol{x})$, and $\boldsymbol{K}\boldsymbol{G}^{-2}\boldsymbol{K}^T$ approximates the transition density $p_\tau(\boldsymbol{y}|\boldsymbol{x})$, such that the matrix multiplication between $\boldsymbol{G}^{-1}$ and $\boldsymbol{K}$ in Eq. (6) approximates the integral over $\boldsymbol{x}$ in Eq. (3), and the matrix multiplication between $\boldsymbol{K}^T$ and $[h(\boldsymbol{y}_1), \cdots, h(\boldsymbol{y}_M)]^T$ in Eq. (6) approximates the integral over $\boldsymbol{y}$ in Eq. (4). Finally, we can consider Eq. (6) as an appropriate discrete approximation of Eq. (5) using training data, and the point evaluation of a function h evolved by Koopman operator in RKHS at a new data point $\langle k_{\boldsymbol{x}_*}|\mathcal{K}|h\rangle_{\mathcal{H}_k}$ is equivalent to predicting its expectation value $\mathbb{E}[h(\boldsymbol{y})]$ over training data at a later time. Notice that during the derivation of Eq. (6), *we did not use the definition of stochastic Koopman operator*, but by using training data, we can indeed approximate $(\mathcal{K}_\tau h)(\boldsymbol{x}) = \int_{\mathbb{A}} h(\boldsymbol{y})p_\tau(\boldsymbol{y}|\boldsymbol{x})d\boldsymbol{y} = \mathbb{E}[h(\boldsymbol{F}(\boldsymbol{x}))|\boldsymbol{x}]$ by the rows of $\boldsymbol{K}\boldsymbol{G}^{-2}\boldsymbol{K}^T[h(\boldsymbol{y}_1), \cdots, h(\boldsymbol{y}_M)]^T$, and approximate $(\mathcal{L}_\tau\rho)(\boldsymbol{y}) = \int_{\boldsymbol{F}^{-1}(\mathbb{A})} \rho(\boldsymbol{x})p_\tau(\boldsymbol{y}|\boldsymbol{x})d\boldsymbol{x}$ by the columns of $\boldsymbol{k}(\boldsymbol{x}_*)^T\boldsymbol{G}^{-1}\boldsymbol{K}\boldsymbol{G}^{-2}\boldsymbol{K}^T$. These nice relations are induced by the connection between deterministic approximation of a function in RKHS and Gaussian processes regression, and replacing $\boldsymbol{G}^{-1}$ by $\boldsymbol{G}^{+}$ will turn these almost singular densities to narrow Gaussians, which have even better statistical interpretation and correspond to regularized optimization in RKHS and noisy Gaussian processes regression that usually have better prediction accuracy.

In order to predict the future state of the system using the spectral properties of Koopman operator in RKHS, we first need to obtain a matrix representation of $\mathcal{K}$ projected in this space. Following the derivation of EDMD procedure in previous section, one can write $\mathbb{1}_{\mathcal{H}_k} = \sum_i |p_i\rangle\langle p_i| = \sum_i |q_i\rangle_{L^2}\frac{1}{\sigma_i}\, {}_{L^2}\langle q_i| = \sum_{ij} |q_j\rangle_{L^2}\frac{1}{\sigma_j}\langle q_j|p_i\rangle_{L^2}\langle p_i| \approx \sum_{il} |k_{\boldsymbol{x}_l}\rangle_{\mathcal{H}_k}\langle k_{\boldsymbol{x}_l}|q_i\rangle_{\mathcal{H}_k}\frac{1}{\sqrt{\sigma_i}}\langle p_i| = \sum_{il} |k_{\boldsymbol{x}_l}\rangle_{\mathcal{H}_k}[\boldsymbol{Q}\boldsymbol{\Sigma}^{+}]_{li}\langle p_i| = \sum_{il} |p_i\rangle_{\mathcal{H}_k}[\boldsymbol{\Sigma}^{+}\boldsymbol{Q}^T]_{il}\, {}_{\mathcal{H}_k}\langle k_{\boldsymbol{x}_l}|$, where $|p_i\rangle = \sqrt{\sigma_i}|q_i\rangle$ (in some literature they are called canonical features or Mercer's features due to Mercer's theorem). Then $\mathcal{K}$ can be written as $\mathcal{K}\mathbb{1}_{\mathcal{H}_k} = \sum_k |p_k \circ \boldsymbol{F}\rangle\langle p_k|$, and its matrix representation is $\hat{\boldsymbol{K}}_{ij} = \langle p_i|\mathcal{K}|p_j\rangle_{\mathcal{H}_k} = \langle p_i|\mathbb{1}_{\mathcal{H}_k}\mathcal{K}\mathbb{1}_{\mathcal{H}_k}|p_j\rangle_{\mathcal{H}_k} = [\boldsymbol{\Sigma}^{+}\boldsymbol{Q}^T\boldsymbol{K}\boldsymbol{Q}\boldsymbol{\Sigma}^{+}]_{ij}$, where we plugged in the last two expressions of $\mathbb{1}_{\mathcal{H}_k}$ above, and $\boldsymbol{K}_{ij} = \langle k_{\boldsymbol{x}_i}|\mathcal{K}|k_{\boldsymbol{x}_j}\rangle_{\mathcal{H}_k} = k_{\boldsymbol{x}_j}(\boldsymbol{F}(\boldsymbol{x}_i)) = k(\boldsymbol{y}_i, \boldsymbol{x}_j) = \langle k_{\boldsymbol{y}_i}|k_{\boldsymbol{x}_j}\rangle_{\mathcal{H}_k}$ can be computed directly on training data. Similarly, the eigenvalue problem can be solved by computing eigenvalues and eigenvectors of $\hat{\boldsymbol{K}}$, where the i-th component of eigenvector $\boldsymbol{v}_j$ is $(\boldsymbol{v}_j)_i = \langle p_i|\varphi_j\rangle_{\mathcal{H}_k}$, so the eigenfunction $|\varphi_j\rangle = \sum_i |p_i\rangle(\boldsymbol{v}_j)_i$. The point evaluation of an eigenfunction on training data is $\langle k_{\boldsymbol{x}_i}|\varphi_j\rangle_{\mathcal{H}_k} = \varphi_j(\boldsymbol{x}_i) = \sum_l \langle k_{\boldsymbol{x}_i}|p_l\rangle_{\mathcal{H}_k}(\boldsymbol{v}_j)_l = \sum_{nl}\langle k_{\boldsymbol{x}_i}|k_{\boldsymbol{x}_n}\rangle_{\mathcal{H}_k}\langle k_{\boldsymbol{x}_n}|q_l\rangle_{\mathcal{H}_k}\frac{1}{\sqrt{\sigma_l}}(\boldsymbol{v}_j)_l = [\boldsymbol{G}\boldsymbol{Q}\boldsymbol{\Sigma}^+\boldsymbol{V}]_{ij}$, where columns of $\boldsymbol{V}$ are $\{\boldsymbol{v}_j\}$. By defining $[\boldsymbol{\Phi}_x]_{ij} = \langle k_{\boldsymbol{x}_i}|\varphi_j\rangle_{\mathcal{H}_k}$ and $[\boldsymbol{\Phi}_y]_{ij} = \langle k_{\boldsymbol{y}_i}|\varphi_j\rangle_{\mathcal{H}_k}$, we can write the matrix of eigenfunctions evaluated on training data in a compact form as $\boldsymbol{\Phi}_x = \boldsymbol{G}\boldsymbol{Q}\boldsymbol{\Sigma}^+\boldsymbol{V}$ and $\boldsymbol{\Phi}_y = \boldsymbol{K}\boldsymbol{Q}\boldsymbol{\Sigma}^+\boldsymbol{V}$. Following the same convention and notation in derivation of EDMD, the matrix of Koopman modes can be solved as $\boldsymbol{\Xi} = \boldsymbol{\Phi}_x^+\boldsymbol{X} = \boldsymbol{\Phi}_y^+\boldsymbol{Y} = [\mathrm{diag}(\mathrm{e}^{\lambda\tau})]^+\boldsymbol{\Phi}_x^+\boldsymbol{Y}$, where rows in $\boldsymbol{Y}$ are $\{\boldsymbol{y}^T\}$ and $[\mathrm{diag}(\mathrm{e}^{\lambda\tau})]$ is the diagonal matrix containing the finite time eigenvalues $\mu_i = \mathrm{e}^{\lambda_i\tau}$. This procedure is called kernel-based Koopman spectral analysis [8] and it is currently being adopted as a better approach for other applications [6]. Finally, given a new system state $\boldsymbol{x}_*$, the prediction of the l-th component of system state $F_l(\boldsymbol{x}_*)$ will be a point evaluation of the $\mathcal{K}$-evolved observable g_l at $\boldsymbol{x}_*$ as

$$\begin{aligned}\langle k_{\boldsymbol{x}_*}|F_l\rangle_{\mathcal{H}_k} = \langle k_{\boldsymbol{x}_*}|\mathcal{K}|g_l\rangle_{\mathcal{H}_k} &= \sum_{i=1}^{M}\langle k_{\boldsymbol{x}_*}|\mathcal{K}|\varphi_i\rangle_{\mathcal{H}_k}\boldsymbol{\Xi}_{il} = \sum_{i=1}^{M}\langle k_{\boldsymbol{x}_*}|\varphi_i\rangle_{\mathcal{H}_k}\mathrm{e}^{\lambda_i\tau}\boldsymbol{\Xi}_{il} \\ &= \sum_{i=1}^{M} k(\boldsymbol{x}_*, \boldsymbol{x}_i)[\boldsymbol{Q}\boldsymbol{\Sigma}^+\boldsymbol{V}[\mathrm{diag}(\mathrm{e}^{\lambda\tau})]\boldsymbol{\Xi}]_{il},\end{aligned} \tag{7}$$

where $\boldsymbol{\Xi}_{il}$ is the Koopman mode associated with the i-th eigenfunction when projecting $g_l(\boldsymbol{x})$ on $\boldsymbol{\Phi}_x$.

Another benefit of working in RKHS is that when properly choosing and/or designing the kernel functions (*e.g.,* Gaussian RBF kernel), the unique associated RKHS is dense in the space of continuous bounded functions, which means that these kernel functions are universal approximators to any function in this very large and general function space, and hence they should achieve better approximation and prediction in most cases, especially in computing Koopman eigenfunctions via point evaluation $\varphi_j(\boldsymbol{x}_i) = \langle k_{\boldsymbol{x}_i}|\varphi_j\rangle_{\mathcal{H}_k}$.

3 Numerical algorithm

Recall Eq. (7), if one needs to predict all state variables at a future time, one can simply compute

$$\boldsymbol{F}(\boldsymbol{x}_*) = \boldsymbol{k}(\boldsymbol{x}_*)^T\boldsymbol{Q}\boldsymbol{\Sigma}^+\boldsymbol{V}[\mathrm{diag}(\mathrm{e}^{\lambda\tau})]\boldsymbol{\Xi}, \tag{8}$$

where $\boldsymbol{k}(\boldsymbol{x}_*)^T = [k(\boldsymbol{x}_*, \boldsymbol{x}_1), \cdots, k(\boldsymbol{x}_*, \boldsymbol{x}_M)]$. Notice that for system with input, all the $\boldsymbol{x}_*$, $\boldsymbol{x}_i$, and $\boldsymbol{y}_i$ are extended states with input, but the Koopman modes $\boldsymbol{\Xi}$ will only contain N columns corresponding to the first N components of the extended state, which eliminates meaningless prediction on input. Another observation is that if we substitute $\boldsymbol{\Xi}$ in Eq. (8) with $\boldsymbol{\Xi} = [\mathrm{diag}(\mathrm{e}^{\lambda\tau})]^+\boldsymbol{\Phi}_x^+\boldsymbol{Y}$, after some simplification, we will get $\boldsymbol{k}(\boldsymbol{x}_*)^T\boldsymbol{G}^+\boldsymbol{Y}$, which is exactly the regularized optimization in RKHS or Gaussian processes regression on each state variable one-by-one. As we elaborated in Ref. [6], one of the major advantages of utilizing the spectral properties of Koopman operator is to linearly decompose the system dynamics as a summation over individual modes, such that it is possible to regularize, sort, perform more "physical" cross-validation, and optimize these modes in order to generate an ensemble of predictors to achieve better prediction. When investigating time series with exogenous variables as a dynamical system with input, since the only major change on the numerical procedure is to neglect the Koopman modes associated with input, one can simply work with the remaining Koopman modes and all techniques and methods developed for Kernel-based Koopman modes regression (Kernel KMR)[6] can be employed almost unchanged. Hence we achieved a simple yet useful extension of Kernel KMR, which we refer to as *Kernel-based Generalized Koopman Mode Regression with eXogenous variables* (Kernel GKMRX). For more details on the techniques and methods constituting the Kernel KMR, we suggest referring to Ref. [6].

4 Numerical examples and applications

We tested this new methodology by predicting high-dimensional stock prices' log returns while considering trading volumes of these stocks as exogenous variables. Due to the page limit rule, we refer the readers to Ref. [6] for detailed description of the stock markets data that we used. Compared to Kernel KMR, the Kernel GKRMX can achieve about 0.1% improvement on both root-mean-squared error (RMSE) and mean absolute error (MAE). This insignificant improvement is due to the fact that stock returns time series are very close to random walk, and trading volume as extra information will not change this fact to improve the prediction significantly. However, when applying to electricity generation and consumption time series with weather condition as exogenous variables, we expect some major improvement over Kernel KMR as reported in Ref. [6], and we are currently testing and summarizing these results.

5 Conclusion and outlook

In this paper, we generalized our previously developed Kernel KMR methodology to Kernel GKMRX (Kernel-based Generalized Koopman Mode Regression with eXogenous variables) for prediction of high dimensional time series with exogenous variables, by utilizing a simple yet useful generalization of Koopman operator to dynamical systems with input that generates the time series. We found that the techniques and methods that we developed for Kernel KMR can be employed in Kernel GKMRX with minimal modification. By formulating Koopman operator in reproducing kernel Hilbert space, we obtained a new derivation of the kernel-based EDMD and the original EDMD algorithms by using Dirac bra–ket notation. Moreover, we obtained a statistical interpretation of these numerical methods developed for deterministic Koopman operator by exploiting the connection between RKHS and Gaussian processes regression, and relate them to the stochastic Koopman and Perron-Frobenius operators. This connection and statistical interpretation are crucial to justify the application of existing data-driven deterministic Koopman spectral analysis to non-deterministic dynamical systems, and account for the advantage of kernel-based EDMD over original EDMD which relies on explicit choice of basis functions spanning the space where the Koopman operator is projected and approximated. In applications, we found that the prediction performance of this methodology is promising in forecasting real world high-dimensional time series with exogenous variables, e.g., stock returns time series with trading volume as exogenous variables.

This generalization of Koopman operator to systems with input is not unique, and we are keen to investigate other generalization for prediction purposes. Moreover, even the very simple trick in this generalization that we used in this paper can be developed further to investigate system with memory in the same way as for memoryless systems. These will be left for future work. Another possible improvement, which is still an open question, is the design of kernel functions. When utilizing Gaussian RBF kernels, it should be possible to optimize the kernel widths as hyper-parameters by some other more sophisticated techniques in machine learning. This, again, will be left for future investigation.

References

1. B.O. Koopman, PNAS **17**(5), 315 (1931)
2. I. Mezić, Nonlinear Dyn **41**(1-3), 309 (2005). DOI 10.1007/s11071-005-2824-x
3. M. Budišić, R.M. Mohr, I. Mezić, Chaos: An Interdisciplinary Journal of Nonlinear Science **22**(4), 047510 (2012). DOI 10.1063/1.4772195
4. C.W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, D.S. Henningson, J. Fluid Mech. **641**, 115 (2009). DOI 10.1017/S0022112009992059
5. M.O. Williams, I.G. Kevrekidis, C.W. Rowley, Journal of Nonlinear Science pp. 1–40 (2015). DOI 10.1007/s00332-015-9258-5
6. J.C. Hua, F. Noorian, D. Moss, P.H.W. Leong, G.H. Gunaratne, Nonlinear Dynamics **90**(3), 1785 (2017). DOI 10.1007/s11071-017-3764-y
7. J.L. Proctor, S.L. Brunton, J.N. Kutz, arXiv:1602.07647 [math] (2016)
8. M.O. Williams, C.W. Rowley, I.G. Kevrekidis, arXiv:1411.2260 [math] (2014)
9. P.A.M. Dirac, Mathematical Proceedings of the Cambridge Philosophical Society **35**(03), 416 (1939). DOI 10.1017/S0305004100021162
10. T. Hofmann, B. Schölkopf, A.J. Smola, The Annals of Statistics **36**(3), 1171 (2008). DOI 10.1214/009053607000000677
11. C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning (MIT Press, Cambridge, MA, 2006)
12. B. Schölkopf, R. Herbrich, A.J. Smola, in *Computational Learning Theory* (Springer, Berlin, Heidelberg, 2001), Lecture Notes in Computer Science, pp. 416–426. DOI 10.1007/3-540-44581-1_27
13. A. Lasota, M.C. Mackey, *Chaos, Fractals, and Noise*, *Applied Mathematical Sciences*, vol. 97 (Springer, New York, NY, 1994). DOI 10.1007/978-1-4612-4286-4
14. P. Cvitanović, R. Artuso, R. Mainieri, G. Tanner, G. Vattay, *Chaos: Classical and Quantum* (Niels Bohr Inst., Copenhagen, 2016)
15. S. Klus, P. Koltai, C. Schütte, Journal of Computational Dynamics **3**(1), 1 (2016). DOI 10.3934/jcd.2016003
16. S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, F. Noé, arXiv:1703.10112 [math] (2017)