Evolving Gaussian Processes for Learning and Control

Tutorial with examples

Harshal Maske, Hassan Kingravi, Joshua Whitman, Girish Chowdhary,
POC email: girishc@illinois.edu

Introduction

Modeling of large-scale stochastic phenomena with both spatial and temporal (spatiotemporal) evolution is a fundamental problem in the applied sciences. While modeling spatiotemporal phenomena has traditionally been the province of the field of geostatistics, it has in recent years gained more attention in the machine learning community [1]. The data-driven models developed through machine learning techniques provide a way to capture complex spatiotemporal phenomena which are not easily modeled by first-principles alone.

In the machine learning community, kernel methods represent a class of extremely well-studied and powerful methods for inference in spatial domains; in these techniques, correlations between the input variables are encoded through a covariance kernel, and the model is formed through a linear weighted combination of the kernels [2]–[4]. In recent years, kernel methods have been applied to spatiotemporal modeling with varying degrees of success [1], [2]. Many recent techniques in spatiotemporal modeling have focused on nonstationary covariance kernel design and associated hyperparameter learning algorithms [5]–[7]. These methods, which focus on the careful design of covariance kernels, have been proposed as an alternative to the naive approach of simply including time as an additional input variable in the kernel [8]. The careful design/optimization of covariance kernel avoids an explosion in the number of parameters (kernels utilized) of the model which would be inevitable in a model that simply adds time as an additional input variable, and has been shown to better account for spatiotemporal couplings. However, there are two key challenges with existing kernel based approaches: The first is ensuring the scalability of the model to large scale phenomena, which manifests due to the fact that the problem of optimizing the covariance kernel (known as hyperparameter optimization in the ML community) is not convex in general, leading to methods that are difficult to implement especially in online settings, susceptible to getting stuck at local minima, and highly computationally demanding for large datasets. The second key challenge is in using existing kernel-based machine learning models for analysis and synthesis of observers and controllers for the large

scale spatiotemporal phenoemena. While the first challenge can be addressed with increasing computational power for large datasets, addressing the latter (and vastly more fundamental) challenge is particularly important in the design of reliable engineering systems, such as distributed sensor/actuator networks intended for monitoring physical phenomena, autonomous soft-robots, or other physical systems with distributed sensing and actuation.

In this paper, we present a new perspective to solving the spatiotemporal monitoring problem that brings together kernel-based modeling, systems theory, and Bayesian filtering. We define the monitoring problem as follows: *Given an approximate predictive model of the spatiotemporal phenomena learned using historic data, estimate the current latent state of the phenomena in the presence of uncertainty using as few sensors as possible*. In particular, we argue that when it comes to predictive inference over spatiotemporal phenomena, a Kalman-filter type approach of predicting and correcting with feedback from a set of minimal sensors is a robust way of dealing with real-world uncertainties and inherent modeling errors. In the context of this specific problem, our *main contributions are two-fold*: first, we demonstrate that spatiotemporal functional evolution can be modeled using stationary kernels with a linear dynamical systems layer on their mixing weights. In particular, in contrast with existing work, this approach does not necessarily require the design of complex spatiotemporal kernels, and can accommodate positive-definite kernels on any domain on which it is possible to define them, which includes non-Euclidean domains such as Riemannian manifolds, strings, graphs and images [9]. Second, we show that such a model can be utilized to determine sensing locations that guarantee that the hidden states of functional evolution can be estimated using a Bayesian state-estimator (Kalman filter) with very few measurements. We provide sufficient conditions on the number and location of sensor measurements required and prove non-conservative lower bounds on the minimum number of sampling locations by developing fundamental results on observability of kernel based models. The validity of the presented model and sensing techniques is corroborated using synthetic and large real datasets.

The fundamental idea of building observers and controllers introduced in this paper is generalizable beyond the particular application of spatiotemporal monitoring. Indeed, the significance of the contributions of this paper are in fusing machine learning theory with systems theory to provide a pathway to address major challenges in spatiotemporal monitoring and control. The problem of state estimation of a temporally evolving finite-dimensional state-space system has been extensively studied in the dynamical systems and feedback-control community [10]. Here, fundamental results in observability/controllability provide sufficient conditions on the structure of the state transition and measurement matrix such that the latent state can be estimated in the presence of measurement and process noise. Such filters can be naively extended to the functional domain (e.g. [11]), but have not typically been studied in context of the spatiotemporal

monitoring problem studied here.

The marriage of systems theory with machine learning pursued in this paper is exciting because it can provide a formal way of answering fundamental questions about complex systems, such as: What is the least number of sensors required to observe a distributed system? Where to place sensors/actuators to guarantee observability/controllability of the system? And how does random sensor placement affect observability/controllability? We expect that follow on work will exploit the framework presented in this paper of utilizing linear models in feature spaces of machine learning models to enable practical and analyzable data-driven engineering systems. To facilitate the development of the theory, we have focused this paper on the problem of monitoring spatiotemporal phenomena. However, the idea can be generalized to any distributed cyber-physical system that is changing with space and time.

Elements of the work presented in this paper first appeared in the premier machine learning conference Neural Information Processing Systems (NIPS 2016) ([12]), IEEE CDC 2015 conference [13], and IEEE ACC 2018 conference [14]. This paper presents a comprehensive set of results in a single journal publication, and introduces results on observability in the presence of random sensor placement. As such, we have focused in this article mostly on the fundamental theory and practical algorithms for modeling, estimation, and control, while the excruciating details of how to optimally implement the presented algorithms are omitted[1]. Section summarizes some related work in machine learning in this area. Section formulates the problem, introduces kernel observers, and develops the main theoretical and algorithmic results. Section presents a result on the expected number of randomly placed sensors required to monitor a spatiotemporal process in the context of our model. Section **??** demonstrates the efficacy of the algorithms on several challenging and large real-world datasets. The paper is concluded in Section **??** and proofs of main results are provided in the Appendix.

Related Work

There is a large body of literature on spatiotemporal modeling in geostatistics where specific process dependent kernels can be used [1], [15]. From the machine learning perspective, a naive approach is to utilize both spatial and temporal variables as inputs to a Mercer kernel [16]. However, this technique leads to an ever-growing kernel dictionary. Furthermore, constraining the dictionary size or utilizing a moving window will occlude learning of long-term patterns. Periodic or nonstationary covariance functions and nonlinear transformations have been proposed

[1]Instead an open-source code-base is made available in MATLAB on http://daslab.illinois.edu/software.html and in Python on GitHub https://github.com/hkingravi/funcobspy?files=1

to address this issue [2], [6]. Work focusing on nonseparable and nonstationary covariance kernels seeks to design kernels optimized for environment-specific dynamics, and to tune their hyperparameters in local regions of the input space. Seminal work in [17] proposes a process convolution approach for space-time modeling. This model captures nonstationary structure by allowing the convolution kernel to vary across the input space. This approach can be extended to a class of nonstationary covariance functions, thereby allowing the use of a Gaussian process (GP) framework, as shown in [18]. However, since this model's hyperparameters are inferred using MCMC integration, its application has been limited to smaller datasets. To overcome this limitation, [7] proposes to use the mean estimates of a second isotropic GP (defined over latent length scales) to parameterize the nonstationary covariances. Finally, [5] considers nonistropic variation across different dimension of input space for the second GP as opposed to isotropic variation by [7]. Issues with this line of approach include the nonconvexity of the hyperparameter optimization problem and the fact that selection of an appropriate nonstationary covariance function for the task at hand is a nontrivial design decision (as noted in [19]).

Apart from directly modeling the covariance function using additional latent GPs, there exist several other approaches for specifying nonstationary GP models. One approach maps the nonstationary spatial process into a latent space, in which the problem becomes approximately stationary [20]. Along similar lines, [21] extends the input space by adding latent variables, which allows the model to capture nonstationarity in original space. Both these approaches require MCMC sampling for inference, and as such are subject to the limitations mentioned in the preceding paragraph.

A geostatistics approach that finds dynamical transition models on the linear combination of weights of a parameterized model [1], [11] is advantageous when the spatial and temporal dynamics are hierarchically separated, leading to a convex learning problem. As a result complex nonstationary kernels are often not necessary (although they can be accommodated). The approach presented in this paper aligns closely with this vein of work. A systems-theoretic study of this viewpoint enables the fundamental contributions of the paper, which are 1) allowing for inference on more general domains with a larger class of basis functions than those typically considered in the geostatistics community, and 2) quantifying the minimum number of measurements required to estimate the state of functional evolution.

Lastly, sensor placement optimization is also a well-studied area. Examples include, but are not limited to 1) geometric approaches, which seek to provide a covering of the operating space without making assumptions about the spatiotemporal dynamics [22], and 2) information-theoretic approaches, which place their focus on sensor placement optimizing strategies based on mutual information and information entropy for Gaussian process models [23]. It should be

noted that the contribution of the paper concerning sensor placement is to provide *sufficient conditions* for monitoring rather than optimization of the placement locations, and therefore a comparison with these approaches is not considered in the experiments.

Kernel Observers

This section outlines our modeling framework and presents theoretical results associated with the number of sampling locations required for monitoring functional evolution.

Problem Formulation

We focus on predictive inference of a time-varying stochastic process, whose mean f evolves temporally as $f_{\tau+1} \sim \mathbb{F}(f_\tau, \eta_\tau)$, where $\mathbb{F}$ is a distribution varying with time τ and exogenous inputs η. Our approach builds on the fact that in several cases, temporal evolution can be hierarchically separated from spatial functional evolution. A classical and quite general example of this is the *abstract evolution equation* (AEO), which can be defined as the evolution of a function u embedded in a Banach space $\mathcal{B}$: $\dot{u}(t) = \mathcal{L}u(t)$, subject to $u(0) = u_0$, and $\mathcal{L} : \mathcal{B} \to \mathcal{B}$ determines spatiotemporal transitions of $u \in \mathcal{B}$ [24]. This model of spatiotemporal evolution is very general (AEOs, for example, model many PDEs), but working in Banach spaces can be computationally taxing. A simple way to make the approach computationally realizable is to place restrictions on $\mathcal{B}$: in particular, we restrict the sequence f_τ to lie in a reproducing kernel Hilbert space (RKHS), the theory of which provides powerful tools for generating flexible classes of functions with relative ease [2]. In a kernel-based model, $k : \Omega \times \Omega \to \mathbb{R}$ is a positive-definite Mercer kernel on a domain Ω that models the covariance between any two points in the input space, and implies the existence of a smooth map $\psi : \Omega \to \mathcal{H}$, where $\mathcal{H}$ is an RKHS with the property $k(x, y) = \langle \psi(x), \psi(y) \rangle_{\mathcal{H}}$. The key insight behind the proposed model is that spatiotemporal evolution in the input domain corresponds to temporal evolution of the mixing weights of a kernel model alone in the functional domain. Therefore, f_τ can be modeled by tracing the evolution of its mean embedded in a RKHS using switched ordinary differential equations (ODE) when the evolution is continuous, or switched difference equations when it is discrete (Figure 1). The advantage of this approach is that it allows us to utilize powerful ideas from systems theory for deriving necessary and sufficient conditions for spatiotemporal monitoring.

In this paper, we restrict our attention to the class of functional evolutions $\mathbb{F}$ defined by linear Markovian transitions in an RKHS. While extension to the nonlinear case is possible (and non-trivial), it is not pursued in this paper to help ease the exposition of the key ideas. The class of linear transitions in RKHS is rich enough to approximately model many real-world datasets, as suggested by our experiments.

Let $y \in \mathbb{R}^N$ be the measurements of the function available from N sensors, $\mathcal{A} : \mathcal{H} \to \mathcal{H}$ be a linear transition operator in the RKHS $\mathcal{H}$, and $\mathcal{K} : \mathcal{H} \to \mathbb{R}^N$ be a linear measurement operator. The model for the functional evolution and measurement studied in this paper is:

$$f_{\tau+1} = \mathcal{A} f_\tau + \eta_\tau, \quad y_\tau = \mathcal{K}_\tau f_\tau + \zeta_\tau, \tag{1}$$

where η_τ is a zero-mean stochastic process in $\mathcal{H}$, and ζ_τ is a Wiener process in $\mathbb{R}^N$. Classical treatments of kernel methods emphasize that for most kernels, the feature map ψ is unknown, and possibly infinite-dimensional; this forces practioners to work in the dual space of $\mathcal{H}$, whose dimensionality is the number of samples in the dataset being modeled. This conventional wisdom precludes the use of kernel methods for most tasks involving modern datasets, which may have millions and sometimes billions of samples [25]. An alternative is to work with a feature map $\widehat{\psi}(x) := \left[\, \widehat{\psi}_1(x) \; \cdots \; \widehat{\psi}_M(x) \,\right]^T$ to an approximate feature space $\widehat{\mathcal{H}}$, with the property that for every element $f \in \mathcal{H}$, $\exists \widehat{f} \in \widehat{\mathcal{H}}$ and an $\epsilon > 0$ s.t. $\|f - \widehat{f}\| < \epsilon$ for an appropriate function norm. A few such approximations are listed below.

Dictionary of atoms: Let Ω be compact. Given points $\mathcal{C} = \{c_1, \ldots, c_M\}$, $c_i \in \Omega$, we have a dictionary of atoms $\mathcal{F}^{\mathcal{C}} = \{\psi(c_1), \cdots, \psi(c_M)\}$, $\psi(c_i) \in \mathcal{H}$, the span of which is a strict subspace $\widehat{\mathcal{H}}$ of the RKHS $\mathcal{H}$ generated by the kernel. Here,

$$\widehat{\psi}_i(x) := \langle \psi(x), \psi(c_i) \rangle_{\mathcal{H}} = k(x, c_i). \tag{2}$$

Low-rank approximations: Let Ω be compact, let $\mathcal{C} = \{c_1, \ldots, c_M\}$, $c_i \in \Omega$, and let $K \in \mathbb{R}^{M \times M}$, $K_{ij} := k(c_i, c_j)$ be the Gram matrix computed from $\mathcal{C}$. This matrix can be diagonalized to compute approximations $(\widehat{\lambda}_i, \widehat{\phi}_i(x))$ of the eigenvalues and eigenfunctions $(\lambda_i, \phi_i(x))$ of the kernel [26]. These spectral quantities can then be used to compute $\widehat{\psi}_i(x) := \sqrt{\widehat{\lambda}_i} \widehat{\phi}_i(x)$.

Random Fourier features: Let $\Omega \subset \mathbb{R}^n$ be compact, and let $k(x, y) = e^{-\|x-y\|^2/2\sigma^2}$ be the Gaussian RBF kernel. Then random Fourier features approximate the kernel feature map as $\widehat{\psi}_\omega : \Omega \to \widehat{\mathcal{H}}$, where ω is a sample from the Fourier transform of $k(x, y)$, with the property that $k(x, y) = \mathbb{E}_\omega[\langle \widehat{\psi}_\omega(x), \widehat{\psi}_\omega(y) \rangle_{\widehat{\mathcal{H}}}]$ [25]. In this case, if $V \in \mathbb{R}^{M/2 \times n}$ is a random matrix representing the sample ω, then $\widehat{\psi}_i(x) := \left[\, \frac{1}{\sqrt{M}} \sin([Vx]_i), \frac{1}{\sqrt{M}} \cos([Vx]_i) \,\right]$. Similar approximations exist for other radially symmetric kernels, as well as dot-product kernels.

In the approximate space case, we replace the transition operator $\mathcal{A} : \mathcal{H} \to \mathcal{H}$ in (1) by $\widehat{\mathcal{A}} : \widehat{\mathcal{H}} \to \widehat{\mathcal{H}}$. This approximate regime, which combines the flexibility of a truly nonparametric approach with computational realizability, still allows for the representation of rich phenomena, as will be seen in the sequel, and in Figure 3. The finite-dimensional evolution equations approximating (1) in dual form are

$$w_{\tau+1} = \widehat{A} w_\tau + \eta_\tau, \quad y_\tau = K w_\tau + \zeta_\tau, \tag{3}$$

where we have matrices $\widehat{A} \in \mathbb{R}^{M\times M}$, $K \in \mathbb{R}^{N\times M}$, the vectors $w_\tau \in \mathbb{R}^M$, and where we have slightly abused notation to let y_τ, η_τ and ζ_τ denote their $\widehat{\mathcal{H}}$ counterparts. Here K is the matrix whose rows are of the form $K_{(i)} = \widehat{\Psi}(x_i) = \left[\,\widehat{\psi}_1(x_i)\ \widehat{\psi}_2(x_i)\ \cdots\ \widehat{\psi}_M(x_i)\,\right]$. In systems-theoretic language, each row of K corresponds to a *measurement* at a particular location, and the matrix itself acts as a measurement operator.

The equations (1) suggest an immediate extension to functional control problems. Pick another basis for $\mathcal{H}$ as $\widetilde{\psi}(x) := \begin{bmatrix}\widetilde{\psi}_1(x) & \cdots & \widetilde{\psi}_{\ell'}(x)\end{bmatrix}^T$, where the functions $\widetilde{\psi}_j(x)$ are used to approximate the RKHS $\mathcal{H}$ generated by the kernel. We denote the span of these functions as $\widetilde{\mathcal{H}}$. In the dictionary of atoms case, an example would be another set of atoms $\mathcal{F}_D = \begin{bmatrix}\psi(d_1) & \cdots & \psi(d_{\ell'})\end{bmatrix}$, $\psi(d_j) \in \mathcal{H}$, $d_j \in \Omega$, with $\widetilde{\mathcal{H}}$ being a strict subspace of the RKHS $\mathcal{H}$ generated by the kernel. The functional evolution equation is then as follows:

$$f_{\tau+1} = \mathcal{A}f_\tau + \mathcal{B}\delta_\tau + \eta_\tau, \quad y_\tau = \mathcal{K}_\tau f_\tau + \zeta_\tau, \tag{4}$$

where the control functions δ_τ evolve in $\widetilde{\mathcal{H}}$, and $\mathcal{B} : \widetilde{\mathcal{H}} \to \widehat{\mathcal{H}}$. To derive the finite-dimensional equivalent of $\mathcal{B}$, we have to work out the structure of the matrix B: since $\widehat{\mathcal{H}}$ is not, in general, isomorphic to $\widetilde{\mathcal{H}}$, this imposes strict restrictions on B. We can derive B using least squares using the inner product of $\mathcal{H}$. An instructive example is where both $\widehat{\mathcal{H}}$ and $\widetilde{\mathcal{H}}$ are generated by dictionaries of atoms; recall that in this case, $\mathcal{F}^{\mathcal{C}} = \begin{bmatrix}\psi(c_1) & \cdots & \psi(c_M)\end{bmatrix}$ is the basis for $\widehat{\mathcal{H}}$, and let $\delta = \sum_{j=1}^{\ell'} \acute{w}_j \psi(d_j)$, and let $\mathcal{F}^{\mathcal{C}} = \begin{bmatrix}\psi(c_1) & \cdots & \psi(c_M)\end{bmatrix}$ be the basis for $\mathcal{H}^{\mathcal{C}}$. Then the projection of δ onto $\widehat{\mathcal{H}}$ can be derived as

$$\begin{bmatrix}\langle\delta,\psi(c_1)\rangle_{\mathcal{H}}\\ \vdots \\ \langle\delta,\psi(c_M)\rangle_{\mathcal{H}}\end{bmatrix} = \underbrace{\begin{bmatrix}\langle\psi(d_1),\psi(c_1)\rangle_{\mathcal{H}} & \cdots & \langle\psi(d_{\ell'}),\psi(c_1)\rangle_{\mathcal{H}}\\ \vdots & \ddots & \vdots \\ \langle\psi(d_1),\psi(c_M)\rangle_{\mathcal{H}} & \cdots & \langle\psi(d_{\ell'}),\psi(c_M)\rangle_{\mathcal{H}}\end{bmatrix}}_{K_{CD}} \begin{bmatrix}\acute{w}_1\\ \vdots \\ \acute{w}_{\ell'}\end{bmatrix}.$$

Note that in the dictionary of atoms case, the entries of K_{CD} can be computed in closed form as $K_{CDij} := k(d_i, c_j)$, using the reproducing property. This derivation shows that the operator B is simply $K_{CD} \in \mathbb{R}^{M\times\ell'}$, the kernel matrix between the data C generating the atoms $\mathcal{F}^{\mathcal{C}}$ of $\widehat{\mathcal{H}}$ and the data D generating the atoms $\mathcal{F}_D$ of $\widetilde{\mathcal{H}}$. Thus, the finite-dimensional evolution equations equivalent to (4) are

$$w_\tau = \widehat{A}w_\tau + K_{CD}\acute{w}_\tau, \quad y_\tau = K_\tau w_\tau. \tag{5}$$

We define the *generalized observability matrix* [27] as $\mathcal{O}_\Upsilon = \begin{bmatrix}K\widehat{A}^{\tau_1}\\ \cdots \\ K\widehat{A}^{\tau_L}\end{bmatrix}$ where $\Upsilon = \{\tau_1, \ldots, \tau_L\}$ are the set of instances τ_i when we apply the operator K. A linear system is said to be *observable* if $\mathcal{O}_\Upsilon$ has full column rank (i.e. $\mathrm{Rank}(\mathcal{O}_\Upsilon) = M$) for $\Upsilon = \{0, 1, \ldots, M-1\}$ [27]. Observability guarantees two critical facts: firstly, it guarantees that the state w_0 can be recovered exactly from a

finite series of measurements $\{y_{\tau_1}, y_{\tau_2}, \ldots, y_{\tau_L}\}$; in particular, defining $y_\Upsilon = \left[y_{\tau_1}^T, y_{\tau_2}^T, \cdots, y_{\tau_L^T}\right]^T$, we have that $y_\Upsilon = \mathcal{O}_\Upsilon w_0$. Secondly, it guarantees that a feedback based *observer* can be designed such that the estimate of w_τ, denoted by $\widehat{w}_\tau$, converges exponentially fast to w_τ in the limit of samples. Note that all our theoretical results assume $\widehat{A}$ is available: while we perform system identification in the experiments (Section **??**), it is not the focus of the paper.

We are now in a position to formally state the spatiotemporal modeling, control, and inference problems being considered: given a spatiotemporally evolving system modeled using (3), choose a set of N sensing locations such that even with $N \ll M$, the functional evolution of the spatiotemporal model can be estimated (which corresponds to *monitoring*), can be predicted robustly (which corresponds to *Bayesian filtering*), and which can be controlled (which corresponds to *functional control*). Our approach to solve the monitoring and prediction problem relies on the design of the measurement operator K so that the pair $(K, \widehat{A})$ is observable: any Bayesian state estimator (e.g. a Kalman filter) utilizing this pair is denoted as a **kernel observer** [2]. In the controls case, given a spatiotemporally evolving system modeled using (5), we need to choose a set of N sensing locations and ℓ' control locations, such that even with $N \ll M,\ \ell' \ll M$, the functional evolution of the spatiotemporal model can be controlled; in this case, we must design both a measurement operator K and a control operator K_{CD} such that the pair $(K_{CD}, \widehat{A})$ is controllable: a controls system utilizing this pair and the measurement operator K is denoted as a **kernel controller**.

Preliminaries on Rational Canonical Structures

We take a geometric approach towards the choice of sampling locations for inferring w_τ in (3); the extension for control is similar. We use the notation $\mathcal{V}$, with $\dim(\mathcal{V}) = M$, to emphasize the fact that these theorems hold for any finite-dimensional vector space. Consider the linear operator $\mathcal{A} : \mathcal{V} \to \mathcal{V}$, and recall that the definition of observability requires the construction of a linear operator $\mathcal{K} : \mathcal{V} \to \mathcal{U}$, with $\dim(\mathcal{U}) = N$, such that $\operatorname{rank}\left[(\mathcal{K})^T \cdots \left(\mathcal{K}\mathcal{A}^{M-1}\right)^T\right]^T = M$. In most applications, if $N \geq M$, and $\operatorname{rank}\mathcal{K} = N$, it is reasonable to expect that observability may be achieved. However, for our purposes, N must be *significantly* less than M. Therefore, we must design $\mathcal{K}$ with as small a rank as possible. To do so, we require a series of vectors v_i that, under repeated iterations of $\mathcal{A}$, can generate a basis for $\mathcal{V}$. For this task, we will use a fundamental decomposition result from the theory of modules, known as the *rational canonical structure* of $\mathcal{A}$ [28]. The intuition here is that if the sequence $\{v_i\}_i$ can generate this basis, it can be directly used to construct $\mathcal{K}$.

[2] In the case where no measurements are taken, for the sake of consistency, we denote the state estimator as an **autonomous kernel observer**, despite this being somewhat of an oxymoron.

The linear operator $\mathcal{A} : \mathcal{V} \to \mathcal{V}$ has a characteristic polynomial $\pi(\lambda)$ such that $\pi(\mathcal{A}) = 0$ by the Cayley-Hamilton theorem. The minimal polynomial (MP) of $\mathcal{A}$ is the monic polynomial $\alpha(\cdot)$ of least degree (denoted by $\deg(\cdot)$) given as $\alpha(\lambda) = a_0 + a_1\lambda + \cdots + \lambda^{\deg(\alpha)} = 0$, such that $\alpha(\mathcal{A}) = a_0 I + a_1\mathcal{A} + \cdots + \mathcal{A}^{\deg(\alpha)} = \mathbf{0}$. The MP is unique and divides $\pi(\lambda)$, so that $\deg(\alpha) \leq \deg(\pi)$. The MP of a vector $v \in \mathcal{V}$ *relative to* $\mathcal{A}$ is the unique monic polynomial ξ_v of least degree such that $\xi_v(\mathcal{A})v = a_0 v + a_1\mathcal{A}v + \cdots + \mathcal{A}^{\deg(\alpha)}v = 0$. If $\deg(\alpha) = M$, then $\mathcal{A}$ is *cyclic* and $\exists v \in \mathcal{V}$, such that the vectors $\{v, \mathcal{A}v, \ldots, \mathcal{A}^{M-1}v\}$ form a basis for $\mathcal{V}$; this is the same as saying that the pair $(v^T, \mathcal{A}^T)$ is observable. A subspace $\mathcal{V}_\mathcal{S} \subset \mathcal{V}$ s.t. $\mathcal{A}\mathcal{V}_\mathcal{S} \subset \mathcal{V}_\mathcal{S}$ is *$\mathcal{A}$-cyclic* if $\mathcal{A}_{|\mathcal{V}_\mathcal{S}}$, the restriction of $\mathcal{A}$ to the subspace $\mathcal{V}_\mathcal{S}$, is cyclic. If $\alpha(\lambda)$ is the minimal polynomial of $\mathcal{A}$ and $\deg(\alpha) = m < M$, $\exists\ v \in \mathcal{V}$ such that $\{v, \mathcal{A}v, \ldots, \mathcal{A}^{m-1}v\}$ span an m-dimensional $\mathcal{A}$-cyclic subspace $\mathcal{V}_\mathcal{S}$, with v being the *cyclic generator* of $\mathcal{V}_\mathcal{S}$. The subspace $\mathcal{V}_\mathcal{S}$ decomposes $\mathcal{V}$ relative to $\mathcal{A}$. By the rational canonical structure theorem (Theorem 0.1 of [28]), $\mathcal{A}$ can be successively decomposed into subspaces $\mathcal{V}_i \subset \mathcal{V}$, $i \in \{1, \ldots, \ell\}$, s.t. $\mathcal{V} = \mathcal{V}_1 \oplus \ldots \oplus \mathcal{V}_\ell$, $\mathcal{A}\mathcal{V}_i \subset \mathcal{V}_i$, and $\mathcal{A}_{|\mathcal{V}_i}, i \in \{1, \ldots, \ell\}$, are cyclic[3]. The integer ℓ is unique and is called the *cyclic index of* $\mathcal{A}$.

One of our main results is to show that the cyclic index is a lower bound on the number of measurements required to reconstruct w_τ (see Prop. 3 and Alg. **??**). The matrix transform associated to this theorem is known as the *Frobenius normal form* (denoted by $C \in \mathbb{R}^{M\times M}$): for $\mathcal{A} \in \mathbb{R}^{M\times M}$, $\exists Q \in \mathbb{R}^{M\times M}$ invertible such that $\mathcal{A} = QCQ^{-1}$. We will also use the *Jordan decomposition*, where for $\mathcal{A} \in \mathbb{R}^{M\times M}$, $\exists P \in \mathbb{R}^{M\times M}$ invertible such that $\mathcal{A} = P\Lambda P^{-1}$, where Λ is a unique block diagonal matrix with Jordan blocks with λ_i along the diagonal. If all the eigenvalues λ_i are nonzero and real, we say the matrix has a *full-rank Jordan decomposition*.

Main Results

In this section, we prove results concerning the observability of spatiotemporally varying functions modeled by the functional evolution and measurement equations (3) formulated in Section . In particular, observability of the system states implies that we can recover the current state of the spatiotemporally varying function using a small number of sampling locations N, which allows us to 1) track the function, and 2) predict its evolution forward in time. We work with the approximation $\widehat{\mathcal{H}} \approx \mathcal{H}$: given M basis functions, this implies that the dual space of $\widehat{\mathcal{H}}$ is $\mathbb{R}^M$. Proposition 1 shows that if $\widehat{A}$ has a full-rank Jordan decomposition, the observation matrix K meeting a condition called *shadedness* (Definition 1) is sufficient for the system to be observable. Proposition 2 provides a lower bound on the number of sampling locations required for observability which holds for any $\widehat{A}$. Proposition 3 constructively shows the existence of an abstract measurement map $\widetilde{K}$ achieving this lower bound. Since the measurement map does not have the structure of a kernel matrix, a slightly weaker sufficient condition for the observability

[3]In general, the subspaces $\mathcal{V}_i$ are not unique for a fixed $\mathcal{A}$.

of any $\widehat{A}$ is in Theorem 1. Finally, since both K and K_{CD} are kernel matrices generated from a shared kernel, these observability results translate directly into controllability results. Proofs of all claims are in the appendix.

Definition 1: **(Shaded Observation Matrix)** Given $k : \Omega \times \Omega \to \mathbb{R}$ positive-definite on a domain Ω, let $\{\widehat{\psi}_1(x), \ldots, \widehat{\psi}_M(x)\}$ be the set of bases generating an approximate feature map $\widehat{\psi} : \Omega \to \widehat{\mathcal{H}}$, and let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be the set of sampling (or sensing) locations, with each $x_i \in \Omega$. Let $K \in \mathbb{R}^{N \times M}$ be the observation matrix, where $K_{ij} := \widehat{\psi}_j(x_i)$. For each row $K_{(i)} := [\, \widehat{\psi}_1(x_i) \cdots \widehat{\psi}_M(x_i) \,]$, define the set $\mathcal{I}_{(i)} := \{\iota_1^{(i)}, \iota_2^{(i)}, \ldots, \iota_{M_i}^{(i)}\}$ to be the indices in the observation matrix row i which are nonzero. Then if $\bigcup_{i \in \{1,\ldots,N\}} \mathcal{I}^{(i)} = \{1, 2, \ldots, M\}$, we denote K as a *shaded observation matrix* (see Figure 2a).

This definition seems quite abstract, so the following remark considers a more concrete example.

Remark 1: let $\widehat{\psi}$ be generated by the dictionary given by $\mathcal{C} = \{c_1, \ldots, c_M\}$, $c_i \in \Omega$. Note that since $\widehat{\psi}_j(x_i) = \langle \psi(x_i), \psi(c_j) \rangle_{\mathcal{H}} = k(x_i, c_j)$, K is the kernel matrix between $\mathcal{X}$ and $\mathcal{C}$. For the kernel matrix to be shaded thus implies that there does not exist an atom $\psi(c_j)$ such that the projections $\langle \psi(x_i), \psi(c_j) \rangle_{\mathcal{H}}$ vanish for all x_i, $1 \leq i \leq N$. Intuitively, the shadedness property requires that the sensor locations x_i are privy to information propagating from every c_j. As an example, note that, in principle, for the Gaussian kernel, a single row generates a shaded kernel matrix[4].

Proposition 1: Given $k : \Omega \times \Omega \to \mathbb{R}$ positive-definite on a domain Ω, let $\{\widehat{\psi}_1(x), \ldots, \widehat{\psi}_M(x)\}$ be the set of bases generating an approximate feature map $\widehat{\psi} : \Omega \to \widehat{\mathcal{H}}$, and let $\mathcal{X} = \{x_1, \ldots, x_N\}$, $x_i \in \Omega$. Consider the discrete linear system on $\widehat{\mathcal{H}}$ given by the evolution and measurement equations (3). Suppose that a full-rank Jordan decomposition of $\widehat{A} \in \mathbb{R}^{M \times M}$ of the form $\widehat{A} = P \Lambda P^{-1}$ exists, where $\Lambda = [\, \Lambda_1 \cdots \Lambda_O \,]$, and there are no repeated eigenvalues. Then, given a set of time instances $\Upsilon = \{\tau_1, \tau_2, \ldots, \tau_L\}$, and a set of sampling locations $\mathcal{X} = \{x_1, \ldots, x_N\}$, the system (3) is observable if the observation matrix K_{ij} is shaded according to Definition 1, Υ has distinct values, and $|\Upsilon| \geq M$.

When the eigenvalues of the system matrix are repeated, it is not enough for K to be shaded. In the next proposition, we take a geometric approach and utilize the rational canonical form of $\widehat{A}$ to obtain a lower bound on the number of sampling locations required. Let r be the number of unique eigenvalues of $\widehat{A}$, and let γ_{λ_i} denote the geometric multiplicity of eigenvalue λ_i. Then

[4]However, in this case, the matrix can have many entries that are extremely close to zero, and will probably be very ill-conditioned.

the *cyclic index* of $\widehat{A}$ is defined as $\ell = \max_{1 \leq i \leq r} \gamma_{\lambda_i}$ [28] (see preliminary section for details).

Proposition 2: Suppose that the conditions in Proposition 1 hold, with the relaxation that the Jordan blocks $\begin{bmatrix} \Lambda_1 & \cdots & \Lambda_O \end{bmatrix}$ may have repeated eigenvalues (i.e. $\exists \Lambda_i$ and Λ_j s.t. $\lambda_i = \lambda_j$). Then there exist kernels $k(x, y)$ such that the lower bound ℓ on the number of sampling locations N is given by the cyclic index of $\widehat{A}$. In other words, the system in (3) is observable if $N \geq \ell$.

Section gives a concrete example to build intuition regarding this lower bound. We now show how to construct a matrix $\widetilde{K}$ corresponding to the lower bound ℓ.

Proposition 3: Given the conditions stated in Proposition 2, it is possible to construct a measurement map $\widetilde{K} \in \mathbb{R}^{\ell \times M}$ for the system given by (3), such that the pair $(\widetilde{K}, \widehat{A})$ is observable.

The construction provided in the proof of Proposition 3 is utilized in Algorithm **??**, which uses the rational canonical structure of $\widehat{A}$ to generate a series of vectors $v_i \in \mathbb{R}^M$, whose iterations $\{v_1, \ldots, \widehat{A}^{m_1 - 1} v_1, \ldots, v_\ell, \ldots, \widehat{A}^{m_\ell - 1} v_\ell\}$ generate a basis for $\mathbb{R}^M$. Unfortunately, the measurement map $\widetilde{K}$, being an abstract construction unrelated to the kernel, does not directly select $\mathcal{X}$. We will show how to use the measurement map to guide a search for $\mathcal{X}$ in Remark 2 (in Appendix). For now, we state a sufficient condition for observability of a general system.

Theorem 1: Suppose that the conditions in Proposition 1 hold, with the relaxation that the Jordan blocks $\begin{bmatrix} \Lambda_1 & \cdots & \Lambda_O \end{bmatrix}$ may have repeated eigenvalues. Let ℓ be the cyclic index of $\widehat{A}$. Define

$$\mathbf{K} = \begin{bmatrix} K^{(1)^T} & \ldots & K^{(\ell)^T} \end{bmatrix}^T \tag{6}$$

as the *ℓ-shaded matrix* (see Figure 2b) which consists of ℓ shaded matrices with the property that any subset of ℓ columns in the matrix are linearly independent from each other. Then system (3) is observable if Υ has distinct values, and $|\Upsilon| \geq M$.

While Theorem 1 is a quite general result, the condition that any ℓ columns of $\mathbf{K}$ be linearly independent is a very stringent condition. One scenario where this condition can be met with minimal measurements is in the case when the feature map $\widehat{\psi}(x)$ is generated by a dictionary of atoms with the Gaussian RBF kernel evaluated at sampling locations $\{x_1, \ldots, x_N\}$ according to (2), where $x_i \in \Omega \subset \mathbb{R}^d$, and x_i are sampled from a non-degenerate probability distribution on Ω such as the uniform distribution. For a semi-deterministic approach, when the dynamics matrix $\widehat{A}$ is block-diagonal, we can utilize a simple heuristic:

Remark 2: Let Ω be compact, $\mathcal{C} = \{c_1, \ldots, c_M\}$, $c_i \in \Omega$, and let the approximate feature map be defined by (2). Consider the system (3) with $\widehat{A} = \Lambda$, and let $\Upsilon = \{0, 1, \ldots, M - 1\}$. Then the measurement map $\widetilde{K}$'s values lie in $\{0, 1\}$; in particular, each row $\widetilde{K}^{(j)}$, $j \in \{1, \ldots, \ell\}$,

corresponds to a subspace $\widehat{\mathcal{H}}_j$, generated by a subset of centers $\mathcal{C}^{(j)} \subset \mathcal{C}$. Generate samples $x_i^{(j)}$ to create a kernel matrix $K^{(j)}$ that is shaded only with respect to centers $\mathcal{C}^{(j)}$. Once this is done, move on to the next subspace $\widehat{\mathcal{H}}_{j+1}$. When all ℓ rows of $\widetilde{K}$ are accounted for, construct the matrix $\mathbf{K}$ as in (6). Then the resulting system $(\mathbf{K}, \widehat{A})$ is observable.

This heuristic is formalized in Algorithm 2 in the supplementary. Note that in practice, the matrix $\widehat{A}$ needs to be inferred from measurements of the process f_τ. If no assumptions are placed on $\widehat{A}$, it's clear that at least M sensors are required for the system identification phase. Future work will study the precise conditions under which system identification is possible with less than M sensors.

Discussion of Theoretical Results

The systems-theoretic approach taken in this paper reveals something rather surprising: functions with complex dynamics (with a small cyclic index) can be recovered with less sensor placements than functions with simpler dynamics. Although seemingly counterintuitive, it becomes clear that this is because complex dynamics, which are characterized by a lower geometric multiplicity of the eigenvalues, ensure that the orbit $\mathbf{\Theta} := \{\widehat{A}w_\tau\}_{\tau\in\Upsilon}$ traverses a greater portion of $\mathbb{R}^M \equiv \widehat{\mathcal{H}}$ and thus that fewer sensors can recover more geometric information. On the other hand, in 'simpler' functional evolution, $\mathbf{\Theta}$ evolves along strict subspaces of $\mathbb{R}^M$, and so more independent sensors are required to infer the same amount of information.

In the case described in Remark 2, we have a set of centers $\mathcal{C} = \{c_1, \ldots, c_M\}$, which generate the bases $\mathcal{F}^{\mathcal{C}} = \{\psi(c_1), \cdots, \psi(c_M)\}$. Let the cyclic index be ℓ: this implies that there exist ℓ subsets $\Psi^{(i)}$ of $\mathcal{F}^{\mathcal{C}}$ with at least one element $\psi(c_j)$ each, leading to $\binom{M}{\ell}$ possible choices: Figure 6 represents these choices as hyperplanes separating the subsets. The measurement map described in Alg. **??** induces this *decomposition of bases* $\mathcal{F}^{\mathcal{C}} = \{\Psi^{(1)}, \ldots, \Psi^{(\ell)}\}$ in polynomial time. Further, each subset $\Psi^{(i)}$ is directly associated to a subset of centers $\mathcal{C}^{(i)} \subset \mathcal{C}$, which allows us to pick targeted sensor locations $x_i \in \Omega$. In particular, for radially symmetric kernels such as the Gaussian, the centroid of the convex hull of $\mathcal{C}^{(i)}$ is sufficient for generating a sensor placement. The measurement map is a significant theoretical insight into sensor placement for dynamically changing environments, because it directly takes into account the dynamics of the process. Of course, in practice, this may be too expensive for approximate feature spaces with M very large, so one can use random sampling to generate the sensor locations instead, at the cost of N being larger than ℓ. The advantage here though is that since random sampling is computationally inexpensive, different choices of sensor placements can be generated and evaluated relatively quickly.

Another point to note is that since the collection of bases $\{\widehat{\psi}_i(x)\}_{i=1}^M$ determines the

richness of the function space $\widehat{\mathcal{H}} \approx \mathcal{H}$ we operate in, it determines the fidelity of the model approximation to the true time-varying function. As a consequence, observability of the system in $\widehat{\mathcal{H}}$ refers to the best possible approximation in $\widehat{\mathcal{H}}$. The greater the number of bases, the higher the dimensionality, which results in greater model fidelity, but which may require a much greater number of measurements for state recovery. This is where the lower bounds presented in the paper are particularly useful, because they show that for functional evolutions corresponding to certain $\widehat{A}$, *the number of sensor placements are essentially independent of the dimensionality* M, but depend rather on the cyclic index of $\widehat{A}$.

Figure 5 gives an overall picture on the process of generating a kernel observer, while Figure 6 gives two approaches to sensor selection in our framework. The measurement map approach can generate a smaller set of sensors than the random placement approach, but comes at an additional computational cost.

Random Sensor Placement

We now elaborate on how the challenging problem of sensor placement can be tackled through random selection. This process of random selection is a product of the kernel observer model described in the section . We present the theoretical background required to prove Theorem 2, which states the expected number of randomly placed sensors required to monitor a given spatiotemporal process, and Theorem 3, which determines the probability with which optimal sensor placement is ensured given that, N number of sensors have been placed.

As discussed earlier, we work with an approximate feature space $\widehat{\mathcal{H}}$, with the corresponding transition operator $\widehat{A} : \widehat{\mathcal{H}} \to \widehat{\mathcal{H}}$, representing finite-dimensional functional evolution. To achieve observability for the pair $(\widehat{A}, K)$, row vectors of the corresponding observability matrix, $\mathcal{O}$, should form the basis for the $\mathbb{R}^M$-dimensional space $\widehat{\mathcal{H}}$. According to the rational canonical structure Theorem [28], $\widehat{A}$ can successively decompose the dual space $\mathbb{R}^M$ into subspaces, $\mathcal{V}_i \subset \mathcal{V}$, $i \in \{1, \ldots, \ell\}$, with properties, i) $\mathcal{V} = \mathcal{V}_1 \oplus ... \oplus \mathcal{V}_\ell$, ii) $\widehat{A}\mathcal{V}_i \subset \mathcal{V}_i$, and iii) $\widehat{A}|\mathcal{V}_i, i \in \{1, \ldots, \ell\}$, are cyclic. The integer ℓ is unique and is called the *cyclic index of* $\widehat{A}$. Each of these properties contribute towards the theorem on the number of random samples required to achieve observability. The first property shows that the space $\mathbb{R}^M$ can be decomposed into ℓ independent subspaces. The second property shows that the vector $v_i \in \mathcal{V}_i$ stays in $\mathcal{V}_i$ even when operated upon by $\widehat{A}$. Thus, to generate bases for $\mathbb{R}^M$, one needs at least ℓ vectors, say, $v_1, \ldots, v_\ell$, with respect to each subspace $\mathcal{V}_1, \ldots, \mathcal{V}_\ell$. This holds due to the third property, but requires that the vectors $v_1, \ldots, v_\ell$, are the cyclic generators of their corresponding subspaces. Our analysis is based on whether a randomly selected sensor can generate a cyclic generator. To examine this, recall that a row vector $K_{(i)}$ generated by a randomly selected sensor location x_i takes the form,

$$K_{(i)} = \Big[k(x_i, c_1), \ldots, k(x_i, c_M)\Big]. \tag{7}$$

Here, for radial kernels for example, the entries corresponding to the centers closer to x_i tend to be non-zero, whereas the others tend to be zero. The rows $K_{(i)}$ from random sensor placement must be able to generate a basis for a subspace $\mathcal{V}_i$, and thus must be cyclic generators. We will derive the expected number of random sensor placements sufficient for observability for the case where $\widehat{A} = \Lambda$ and then attempt to generalize the result for any $\widehat{A}$. Note Λ is a block diagonal Jordan form. In this case, the cyclic generator for each subspace $\mathcal{V}_i$, is a vector v_i with non-zero entries corresponding to the leading entry of the Jordan blocks of $\mathcal{V}_i$.

Overall, our construction is as follows: for each subspace $\mathcal{V}_i$, let $\mathcal{C}_{\mathcal{V}_i} \subset \mathcal{C}$ be the centers corresponding to those leading entry of Jordan blocks: then the minimum number of random samples required to generate the bases for $\mathcal{V}_i$ is equal to the number of Jordan blocks comprising $\mathcal{V}_i$. Altogether, the minimum number of random samples required to generate a basis for $\mathbb{R}^M$ is equal to the total number of Jordan blocks in $\widehat{A}$. Let ς be the total number of Jordan blocks in $\widehat{A}$, then

$$\varsigma = \sum_{\lambda \in \sigma(\widehat{A})} \gamma_{\widehat{A}}(\lambda) \tag{8}$$

where $\sigma(\widehat{A})$ represents the spectrum of $\widehat{A}$, whose elements are the eigenvalues of $\widehat{A}$, and $\gamma_{\widehat{A}}(\lambda)$ is the geometric multiplicity corresponding to the eigenvalue λ, which is also equal to the total number of Jordan blocks corresponding to the eigenvalue λ. Define a set of centers $\mathcal{C}_\varsigma$ with elements $\{c_1, c_2, \ldots, c_\varsigma\}$, to be the centers corresponding to the leading entries of the Jordan blocks. For sensor location $x \in \Omega$, and $\epsilon > 0$, let $k(x, c_j) > \epsilon$, denote the region $\Omega_j \subset \Omega$, such that the kernel evaluation with respect to center c_j is greater than ϵ, that is $\Omega_j \equiv \{x \in \Omega : k(x, c_j) > \epsilon\}$. We define p_ϵ as

$$p_\epsilon = \min_{c_j \in \mathcal{C}_\varsigma} \frac{\nu(k(x, c_j) > \epsilon)}{\nu(\Omega)}, \tag{9}$$

where ν is a measure in the real analysis sense. Hence, p_ϵ corresponds to a lower bound on the probability that a random sample lies within the $\epsilon-$shaded region of a particular center c_j. With all of this in place, we can prove the following theorem.

Theorem 2: Given the spatiotemporal function $f(x, \tau)$ with $x \in \Omega \subseteq \mathbb{R}^D, \tau \in \mathbb{Z}^+$ its kernel observer model (3), and a tolerance parameter $\epsilon > 0$, the expected number of randomly placed sensor locations required to achieve observability for the pair $(K, \widehat{A})$ is ς/p_ϵ where ς is the summation over geometric multiplicities of each $\lambda \in \sigma(\widehat{A})$ given by Equation (8).

Theorem 3: Given the spatiotemporal function $f(x, \tau)$ with $x \in \Omega \subseteq \mathbb{R}^D, \tau \in \mathbb{Z}^+$, its kernel observer model (3), a tolerance parameter $\epsilon > 0$, summation over geometric multiplicities

of each $\lambda \in \sigma(\widehat{A})$ denoted by ς as in Equation (8), and a constant $\delta \in (0, 1]$, the probability that pair $(K, \widehat{A})$ is unobservable after the selection of N random sensors is at most $e^{\frac{-1}{2}(Np_\epsilon - 2\varsigma)}$, where p_ϵ is given by Equation (9) and $N \geq \varsigma/p_\epsilon$.

For the case when $\widehat{A} \neq \Lambda$, a change of basis can be used to obtain $\Lambda = P^{-1}\widehat{A}P$, where P is the projection map. There are two challenges in performing the above analysis for Λ so obtained: first, the leading entries of Jordan blocks do not directly correspond to the centers $\{c_1, \ldots, c_M\}$ which was the case for $\widehat{A} = \Lambda$. Second, although we can obtain the transformation of the row vector (Equation (7)) using the projection map P, we can no longer arrive at the definition of the probability p_ϵ as in Equation (9). The existence of the similarity transform hints that the results in Theorems 2-3 should hold for any $\widehat{A}$, but the mathematical tools utilized in the paper seem to be insufficient to prove them. However, we present some empirical evidence for these claims for when $\widehat{A} \neq \Lambda$ in Section **??**.

Generalizing Across Similar Spatiotemporally Evolving Systems

Building on the Kernel Observers method, let us introduce Evolving Gaussian Processes (E-GP). The primary novelty in this method of generating a model is learning an $\widehat{A}$ matrix for *multiple* systems. The ultimate goal of this research would be to generate highly efficient machine learning models that can be used instead of the costly numerical simulations for design and autonomy purposes. This would be a major success for the design and control of complex physical systems, such as soft robotics, as they would significantly reduce the cost and resources required in simulations. The ability to generalize across different physical situations, is critical. This is a difficult problem, as it requires that the model have the capability to actually learn the underlying physics and not just input-output relationships. For example, in the context of fluid flows, these models must be able to predict fluid dynamics at different conditions (e.g. Reynolds number) than the training data. E-GP, as far as the authors know, was the first machine learning method to generalize across spatiotemporally evolving systems of such complexity using end-to-end data.

We found that the class of functional evolutions $\mathbb{F}$ defined by linear Markovian transitions in a RKHS is still sufficient to model the nonlinear Navier Stokes equations which govern fluid dynamics, since the unknown map ψ allows us to model highly nonlinear dynamics in the input space. However, we do expect that phenomena such as bifurcation or turbulence will require nonlinear mappings $\mathcal{H}$. There are three steps to generate an E-GP model:

1) After picking the kernel and estimating the bandwidth hyperparameter σ (we utilize the maximum likelihood approach, although other approaches can be used), find an optimal basis vector set $\mathcal{C}$ using the algorithm in [29].

2) Use Gaussian process inference to find weight vectors for each time-step in the training set(s), generating the sequence $w_\tau, \tau = 1, \ldots, T$ for each system. A uniform time-step makes next step easier but can be worked around for non-uniform data sets
3) Using the weight trajectory, use matrix least-squares with the equation $\widehat{A}[w_1, w_2, ..., w_{T-1}] = [w_2, w_3, ..., w_T]$ to solve for $\widehat{A}$.
4) To generate a multi-system model, concatenate the weight trajectories from each similar system in the least-squares computation of $\widehat{A}$. That is, let $W_\theta = [w_1^{(\theta)}, w_2^{(\theta)}, ..., w_{n-1}^{(\theta)}]$ and $W'_\theta = [w_2^{(\theta)}, w_3^{(\theta)}, ..., w_n^{(\theta)}]$ be the weight trajectory and next weight trajectory for some parameter . Then we solve the least-squares problem $\widehat{A} = [W_{\theta_1}, \ldots, W_{\theta_n}] = [W'_\theta, \ldots, W'_{\theta_n}]$

For the sake of defining when it is appropriate to expect this method to be able to generalize across different spatiotemporally evolving systems, we shall define what it means for two fluid flows to be *similar*. In configuring a fluid dynamics simulation, a set of quantifiable parameters are defined. Two dynamical fluid systems S_1 and S_2 are considered *similar* if they have the same configuration of parameters and differ only in the value of at most one parameter. Furthermore, we require that the parameter be continuously variable, and that any observable quantity in the domain of the system vary smoothly as that parameter varies from its value in S_1 to its value in S_2. For example, for fluids flowing past identical cylinders, the Reynolds number associated with the free stream velocity may be varied to produce similar systems. However, to replace the system's cylinder with a triangle would be to qualitatively change the configuration of the system parameters, and thus would produce a non-similar system.

Unlike neural networks, the weights in an E-GP do not exist in some abstract, difficult-to-comprehend space, but are associated with kernel centers in specific locations in the domain. We refer to this attribute of E-GPs as the *spatial encoding* property. This property is an extremely valuable tool for gaining insight into the learned model works:

1) By plotting which kernel centers are associated with which invariant subspaces in the transition matrix, one can visualize where the eigenfunctions are found and how the dynamic modes are separated spatially
2) By plotting arrows from center c_j to c_i for each of the largest elements $\hat{a}_{ij}$ of $\widehat{A}$, one can visualize how different areas of the domain influence each other's evolution.
3) By performing an eigendecomposition of the $\widehat{A}$ matrix, and transforming the eigenvectors back from the weight space to the function space, one can obtain the Koopman modes (and associated eigenvalues) of the system.

Conclusion

This tutorial introduced Gaussian Processes (GP) and Gaussian Process Regression (GPR) from the perspective of Bayesian nonparametric inference for use in control and reinforcement learning problems. It emphasized the key features of GPRs, including: *1)* how they allow for automatic data-driven feature selection, thereby not requiring practitioners to predefine feature numbers and locations; *2)* the way in which GPs offer uncertainty measures of predictions; *3)* how Bayesian inference allows for natural optimization and selection of GP hyperparameters; and *4)* how GPs offer a principled way for performing budgeted online inference. The tutorial provided concrete examples demonstrating how GPR for adaptive control and off-policy reinforcement learning, and also discussed the application of GPs to model value functions in optimal control and planning problems, and to model reward functions in inverse optimal control problems. Moreover, it showed how GPs can be used as the basis of a probabilistic framework capable of modeling system dynamics and measurement functions, with use in model predictive control and state estimation and filtering problems. In addition, the tutorial elaborated the connections between GPR and other widely-used regression techniques, so that readers and practitioners are able to understand the unique features of GPs and its generality. Finally, the tutorial discussed a few limitations of the basic GPR approach and provided a brief overview of several advanced GP models that overcome such limitations. The discussions presented here are by no means exhaustive; our intention is merely to highlight the key features of Gaussian Processes, through illustrative examples, in order to help interested practitioners to more efficiently explore the relevant literature.

References

[1] Noel Cressie and Christopher K Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2011.

[2] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, December 2005.

[3] K.-R. Müller, S. Mika, G. Rätsch, S. Tsuda, and B Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, 2001.

[4] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[5] Sahil Garg, Amarjeet Singh, and Fabio Ramos. Learning non-stationary space-time models for environmental monitoring. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012.

[6] Chunsheng Ma. Nonstationary covariance functions that model space–time interactions. *Statistics & Probability Letters*, 61(4):411–419, 2003.

[7] Christian Plagemann, Kristian Kersting, and Wolfram Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. In *Machine learning and knowledge discovery in databases*, pages 204–219. Springer, 2008.

[8] G. Chowdhary, H. Kingravi, J.P. How, and P. Vela. Nonparametric adaptive control using gaussian processes. In *IEEE Conference on Decision and Control (CDC)*. IEEE, 2013.

[9] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel Methods on Riemannian Manifolds with Gaussian RBF Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.

[10] A. Gelb. *Applied Optimal Estimation*. MIT press (QA402.A5), Cambridge, MA, 1974.

[11] Kanti V Mardia, Colin Goodall, Edwin J Redfern, and Francisco J Alonso. The kriged kalman filter. *Test*, 7(2):217–282, 1998.

[12] Hassan Kingravi, Harshal Maske, and Girish Chowdhary. Kernel observers: Systems theoretic modeling and inference of spatiotemporally varying processes. In *Advances in Neural Information Processing Systems (NIPS)*, Barcelona, Spain, 2016.

[13] Hassan A. Kingravi, Harshal Maske, and Girish Chowdhary. Kernel controllers: A systems-theoretic approach for data-driven modeling and control of spatiotemporally evolving processes. *arXiv*, abs/1508.02086, 2015.

[14] Harshal Maske, Hassan Kingravi, and Girish Chowdhary. Sensor selection via observability analysis in feature space. In *American Control Conference*, 2018. accepted.

[15] Christopher K Wikle. A kernel-based spectral model for non-gaussian spatio-temporal processes. *Statistical Modelling*, 2(4):299–314, 2002.

[16] Fernando Pérez-Cruz, Steven Van Vaerenbergh, Juan José Murillo-Fuentes, Miguel Lázaro-

Gredilla, and Ignacio Santamaria. Gaussian processes for nonlinear signal processing: An overview of recent advances. *Signal Processing Magazine, IEEE*, 30(4):40–50, 2013.

[17] David Higdon. A process-convolution approach to modelling temperatures in the north atlantic ocean. *Environmental and Ecological Statistics*, 5(2):173–190, 1998.

[18] C Paciorek and M Schervish. Nonstationary covariance functions for gaussian process regression. *Advances in neural information processing systems*, 16:273–280, 2004.

[19] Amarjeet Singh, Fabio Ramos, H Durrant-Whyte, and William J Kaiser. Modeling and decision making in spatio-temporal processes for environmental surveillance. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5490–5497. IEEE, 2010.

[20] Alexandra M Schmidt and Anthony O'Hagan. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):743–758, 2003.

[21] Tobias Pfingsten, Malte Kuss, and Carl Edward Rasmussen. Nonstationary gaussian process regression using a latent extension of the input space. *URL http://www. kyb. mpg. de/˜ tpfingst*, 2006.

[22] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[23] C. Guestrin, A. Krause, and A. Singh. Near-optimal sensor placements in gaussian processes. In *International Conference on Machine Learning*, 2005.

[24] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[25] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

[26] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *NIPS*, pages 682–688, 2001.

[27] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ, 1996.

[28] W Murray Wonham. *Linear multivariable control*. Springer, 1974.

[29] Lehel Csatö and Manfred Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.

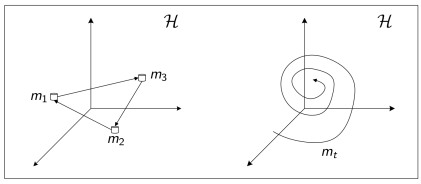
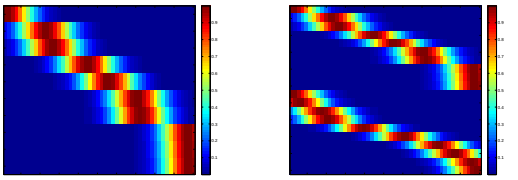
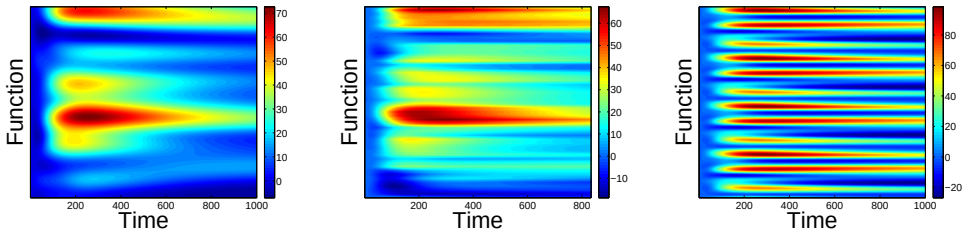


Figure 1: Two types of Hilbert space evolutions. Left: discrete switches in RKHS $\mathcal{H}$; Right: smooth evolution in $\mathcal{H}$.



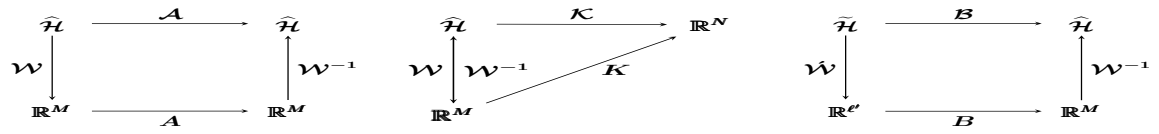
(a) 1-shaded (Def. 1) (b) 2-shaded (Eq. (6))

Figure 2: Shaded observation matrices for dictionary of atoms. Each row represents a sensing location with the color map indicating the evaluation of kernel function w.r.t the others points in the domain.



(a) Gaussian (b) Laplacian (c) Periodic

Figure 3: One-dimensional function evolution over a fixed transition matrix A, initial condition w_0 and centers $\mathcal{C}$, but with different kernels $k(x, y)$. Each y-vector at a given value of x represents the output of the function, which evolves from left to right. As seen, changing the kernel creates quite different dynamic behaviors.



(a) Relationship between $\mathcal{A}$ and A (b) Relationship between $\mathcal{K}$ and K (c) Relationship between $\mathcal{B}$ and B

Figure 4: Commutative diagrams between primal and dual spaces

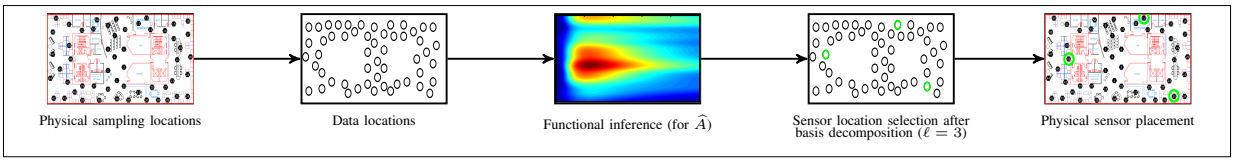


Figure 5: Overall description of how the kernel observer fits in the sensing framework. Physical locations are mapped to data locations, over which historical data is collected as a time series. Functional inference is performed over $\widehat{\mathcal{H}}$ to solve for $\widehat{A}$. The measurement operator K is then computed (see Figure 6), leading to sensor placement.

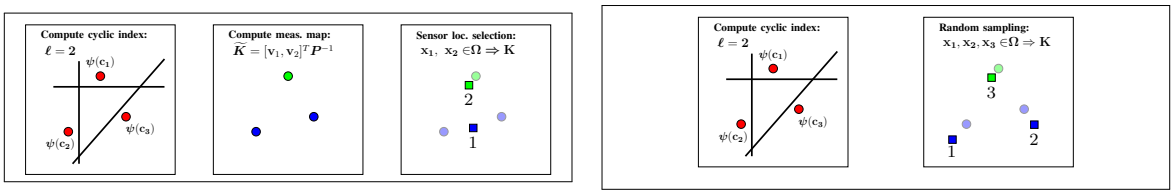


Figure 6: Diagram demonstrating sensor placement using the measurement map or random sampling approaches. The circles represent data locations associated to bases (e.g. $c_j \Leftrightarrow \psi(c_j)$) and the squares represent sensor locations (e.g. $x_i \Leftrightarrow \psi(x_i)$) . The cyclic index ($\ell = 2$) indicates how many possible couplings of bases exist, which can be represented as a choice of $\binom{M}{\ell}$ hyperplanes in Ω. If the measurement map is computed (left), the correct couplings are chosen (green vs. blue), and a smaller number of sensors (2) can be placed. Alternatively, random sampling (right) is more computationally efficient, but generally requires more sensors (3).

Sidebar: Limitations of Parametric Models – A Flight Control Example

Sidebar: Feature Spaces in Machine Learning

Suppose we have data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, where $x_i \in \Omega_I$ and $y_i \in \Omega_O$. Here, Ω_I is the input domain, and is generally a subset of $\mathbb{R}^D$, although more general sets such as discrete spaces, graphs, or text documents can be considered. Similarly, Ω_O, the output domain, can be just as general as Ω_I. We wish to solve for functions f in some space of functions $\mathcal{J}$ such that $f(x_i) = y_i \ \forall i$. Generally, to restrict the complexity of the space $\mathcal{J}$, a *loss function* $L(f, \mathcal{D}) \mapsto \mathbb{R}$ is chosen, which measures the error between a prediction $f(x_i)$ given a datapoint x_i, and y_i, averaged over the entire dataset $\mathcal{D}$, and the optimization problem becomes

$$f^* = \arg\min_{\mathcal{J}} L(f, \mathcal{D}) + \lambda g(f), \tag{S1}$$

where $\lambda \in \mathbb{R}$, and $g(f)$ represent some constraints on the function f, such as smoothnes. Control theorists are most likely familiar with input-output pairs where $x_i \in \mathbb{R}^N$ and y_i is either in $\mathbb{R}$ or $\mathbb{R}^M$ (*regression*). In machine learning, the most common task is when the y_i are discrete (*classification*). Different combinations of task, loss functions, and spaces $\mathcal{J}$ result in different algorithms to solve these problems, which can sometimes form entire subfields of machine learning.

The choice of the function space $\mathcal{J}$ can be critical for the task we want to perform, similar to how the choice of the state space is in control theory. Let's consider a simple example. Suppose we have data from two *classes* $\mathcal{D}_A = \{(x_1^A, y_1^A), \ldots, (x_N^A, y_N^A)\}$ and $\mathcal{D}_B = \{(x_1^B, y_1^B), \ldots, (x_N^B, y_N^B)\}$, where $x_i^{\{A,B\}} \in \mathbb{R}^D$, and $y_i^{\{A,B\}} \in \{-1, +1\}$, shown in Figure S1a. Let f be chosen from the class of linear algorithms, i.e. $f = w^T x + b$, where $w \in \mathbb{R}^D, b \in \mathbb{R}$. We pick a loss L that returns a loss of zero when the prediction is the correct class, and if the prediction is the incorrect class, returns a higher value for misclassifications that are closer to the boundary. A classical example of such a loss is that used by the *perceptron algorithm*, which can be written as

$$L(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \max(0, -y_i w^T x). \tag{S2}$$

This loss measures how accurate the prediction of the perceptron is on average. The general algorithm is as follows:

1) Initialize $w \in \mathbb{R}^D$ to all zeros.
2) For a fixed number of iterations, or until some stopping criterion is met:
    a) For each training example (x_i, y_i),
        i) Let $\hat{y}_i = \text{sgn}(w^T x_i)$.
        ii) If $y_i \neq \hat{y}_i$, update $w \leftarrow w + y_i x_i$.

The perceptron was one of the first machine learning models, and the genesis of modern neural networks [**?**]. Figure S1 shows a visual representation of where the perceptron algorithm can solve for the decision boundary with zero error. However, if the structure of the data has some nonlinearities, no solution will be found, as seen in Figure S2. In this case, the original space the data resides in is, in some sense, not a rich enough representation. If we could construct a mapping of the data to a different space which gives a learning algorithm more degrees of freedom to work with, linear algorithms can still be deployed. If we map the same data using a nonlinear map $\phi(x, y) := (x^2, y^2, 2xy)$, the perceptron now finds a solution in 3 dimensions, as seen in Figure S3. This example shows why so much of the work in machine learning focuses on learning the right representation for the data, for the right representation makes the classification task easy. Two major threads of research in the arena of feature maps over the last 40 years are kernel methods, and neural networks, the latter of which has gained remarkable notoriety in the last 10 years. These lines of research represent distinctly different strategies for generating feature maps from data.

In Figure S2, the data was mapped using an explicit feature map. Kernel methods utilize an elegant strategy for generating feature maps from data, using a remarkably simple trick called *the kernel trick*. Given a positive-definite kernel function $k(x, y) : \Omega \times \Omega \to \mathbb{R}$, Mercer's theorem guarantees the existence of a feature map $\psi : \Omega \to \mathcal{H}$, where $\mathcal{H}$ is a *reproducing kernel Hilbert space (RKHS)*, and the map ψ obeys the property

$$k(x, y) = \langle \psi(x), \psi(y) \rangle_{\mathcal{H}}. \tag{S3}$$

Recall that since $\mathcal{H}$ is an RKHS, given $c \in \Omega$, $k(x, c) = \langle \psi(x), \psi(c) \rangle_{\mathcal{H}}$, and $k(x, c) := \psi_c \in \mathcal{H}$. Furthermore, $\mathrm{span}\{\psi_x\}_{x \in \Omega}$ is dense in $\mathcal{H}$. There exist kernels that generate $\mathcal{H}$s that are extremely high dimensional: for example, the RBF kernel $k(x, y) = e^{-\gamma \|x-y\|^2}$ is infinite-dimensional. This high degree of freedom enables the design of powerful learning algorithms that are linear in $\mathcal{H}$, but nonlinear in the input domain Ω. The canonical example of this is the support vector machine (SVM) [**?**], but a more instructive example for us is the perceptron algorithm.

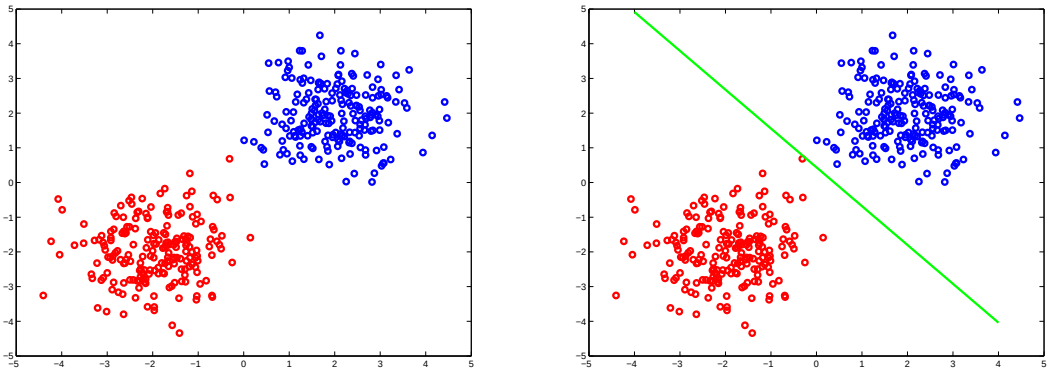
Suppose we trained a percetron using $\mathcal{X} = \{x_1, \ldots, x_N\}$, $x_i \in \mathbb{R}^D$ for some D, $y_i \in \{-1, 1\}$. The prediction of the perceptron is $\hat{y} = \mathrm{sgn}(w^T x)$, where $w \in \mathbb{R}^D$. It can be shown that $w = \sum_{i=1} \alpha_i y_i x_i$, where α_i is the number of times x_i was misclassified. This allows us to derive the dual version of this algorithm, because

$$\hat{y} = \mathrm{sgn}(w^T x) = \mathrm{sgn} \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle_{\mathbb{R}^D}.$$

The dot product $\langle x_i, x \rangle_{\mathbb{R}^D}$ can be replaced with the kernel, leading to the *kernel perceptron algorithm*:

1) Initialize $\alpha \in \mathbb{R}^N$ to all zeros.
2) For a fixed number of iterations, or until some stopping criterion is met:
    a) For each training example (x_j, y_j),
        i) Let $\hat{y}_i = \operatorname{sgn} \sum_{i=1}^{n} \alpha_i y_i k(x_i, x_j)$.
        ii) If $y_i \neq \hat{y}_i$, update $\alpha_j \leftarrow \alpha_j + 1$.

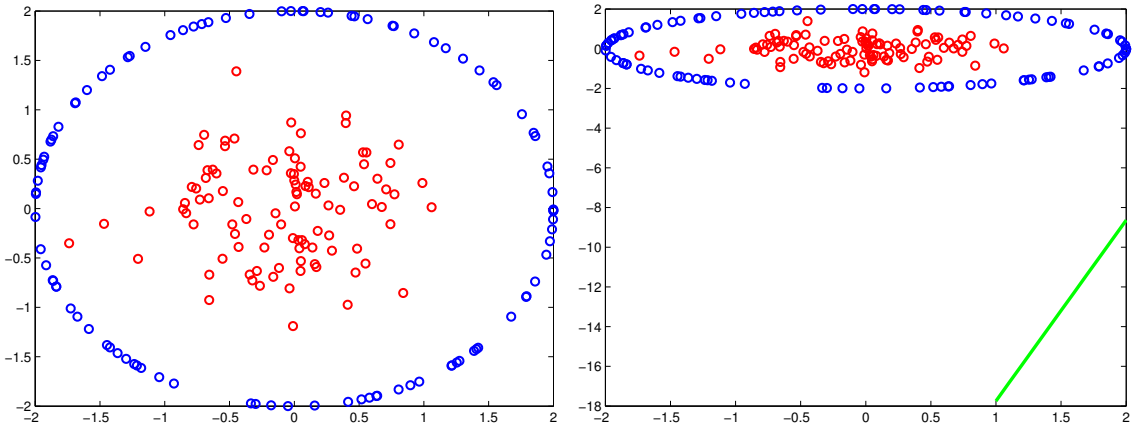
This algorithm is nonlinear in the input domain, but linear in the feature space $\mathcal{H}$, which led the deep learning community to, somewhat pejoratively, label this as an example of a *shallow learning architecture*. Kernel methods can also be used in a more direct fashion: if we have a subspace $\mathcal{H}' \subset \mathcal{H}$ with a basis generated from $\mathcal{C} = \{c_1, \ldots, c_M\}$, i.e. $\mathcal{H}' = \operatorname{span}\{\psi_{c_1}, \ldots, \psi(c_M)\}$, a linear model in $\mathcal{H}'$ is again given by a vector $w \in \mathbb{R}^M$. Suppose this weight vector represents a boundary in $\mathcal{H}'$: to compute which side of this boundary a point x would lie on in $\mathcal{H}'$, we simply compute $\operatorname{sgn}(\sum_{i=1}^{M} w_i \langle \psi(x), \psi(c_i) \rangle_{\mathcal{H}}) = \operatorname{sgn}(\sum_{i=1}^{M} w_i k(x, c_i))$. The choice of the kernel and its parameters depends on the dataset and the loss function. The kernel and the data together form the feature space. Because kernel methods are linear in their parameters and are restricted to RKHSs, they are amenable to somewhat straightforward mathematical analysis, and are very well analyzed because of this.



(a) Data from classes A and B.

(b) Linear boundary separating data.

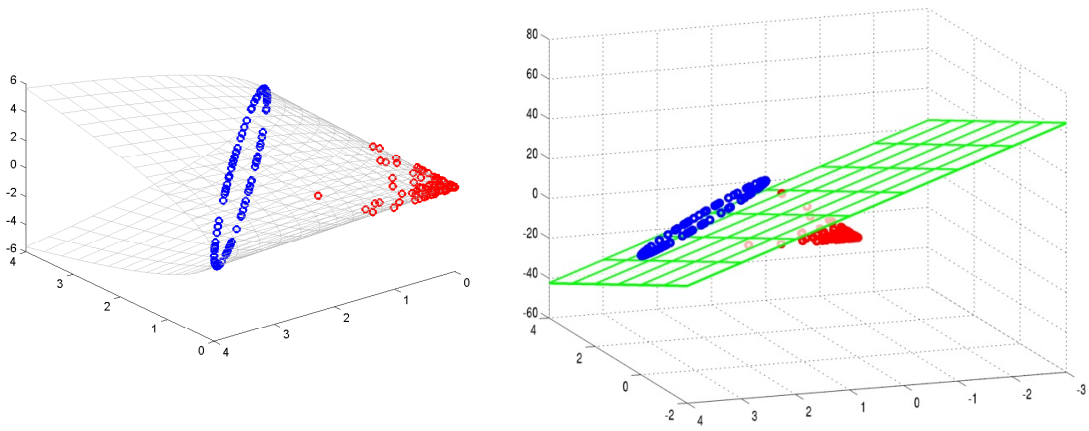
Figure S1: Example of linearly separable data. Any simple linear learning algorithm e.g. perceptron, finds a solution.



(a) Data from classes A and B.

(b) Linear boundary fails to separate data.

Figure S2: Example of nonlinearly separable data. Perceptron fails to find a solution, and diverges.



(a) Nonlinear mapping of data. (b) Linear boundary in new space.

Figure S3: If we map the same data using a nonlinear map $\phi(x, y) := (x^2, y^2, 2xy)$, the perceptron now finds a solution in 3 dimensions.

Author Biography

Miao Liu is a research staff member in the AI Science Department at IBM T. J. Watson Research Center, Yorktown Heights NY. Prior to joining IBM in 2016, he was a Postdoctoral Associate in the Laboratory of Information and Decision System (LIDS) at Massachusetts Institute of Technology (MIT), where he worked on scalable Bayesian nonparametric methods for solving multiagent learning and planning problems. He received a Ph.D. degree in Electrical and Computer Engineering from Duke University in 2014. He received both his B.S. and M.S. degrees in Electronics and Information Engineering from Huazhong University of Science and Technology, in Wuhan, China in 2005 and 2007, respectively. Dr. Liu was a co-author of the best student paper at IROS2017 and received nomination of the best multi-robot paper in ICRA2017. His research interests include statistical machine learning, AI, and robotics.

Bruno Castro da Silva is an associate professor at the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS), in Brazil. Prior to that he was a postdoctoral associate at the Aerospace Controls Laboratory, at MIT. He received his Ph.D. in Computer Science from the University of Massachusetts, working under the supervision of Prof. Andrew Barto, in 2014. Before that he received a B.S. degree *cum laude* in Computer Science from the Federal University of Rio Grande do Sul in 2004, and an MSc. degree from the same university in 2007. Bruno has worked in different occasions as a visiting researcher at the Laboratory of Computational Embodied Neuroscience, in Rome, Italy, developing novel control algorithms for the iCub robot. His research interests lie in the intersection of machine learning, reinforcement learning, optimal control theory, and robotics, and include the construction of reusable motor skills, active learning, efficient exploration of large state-spaces and Bayesian optimization applied to control.

Girish Chowdhary is an assistant professor at the University of Illinois at Urbana-Champaign and affiliated with Electrical and Computer Engineering, Agricultural and Biological Engineering, and the UIUC Coordinated Science Laboratory (CSL). He is the director of the Distributed Autonomous Systems laboratory at UIUC. He holds a PhD (2010) from Georgia Institute of Technology in Aerospace Engineering. He was a postdoc at the Laboratory for Information and Decision Systems (LIDS) of the Massachusetts Institute of Technology for about two years (2011-2013). He was an assistant professor at Oklahoma State Universitys Mechanical and Aerospace Engineering department (2013-2016). Prior to joining Georgia Tech, he also worked with the German Aerospace Center's (DLR's) Institute of Flight Systems for around three years (2003-2006). His undergraduate institution was the Royal Melbourne Institute of Technology in Australia. Girish's ongoing research interest is in theoretical insights and practical algorithms for adaptive autonomy.

Shih-Yuan Liu is a Senior Research Scientist at nuTonomoy Inc. Previously, he was a postdoctoral associate at Laboratory for Information and Decision Systems (LIDS) and Aerospace Controls Laboratory (ACL) at the Massachusetts Institute of Technology (MIT). He received the Ph.D. degree in Mechanical Engineering in Controls from University of California, Berkeley in 2014. His research interests include control, path-planning, coordination, and teleoperation of autonomous ground and aerial vehicles in dynamic environments.

Jonathan P. How is the Richard C. Maclaurin Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology. He received a B.A.Sc. from the University of Toronto in 1987 and his S.M. and Ph.D. in Aeronautics and Astronautics from MIT in 1990 and 1993, respectively. He then studied for two years at MIT as a postdoctoral associate for the Middeck Active Control Experiment (MACE) that flew onboard the Space Shuttle Endeavour in March 1995. Prior to joining MIT in 2000, he was an Assistant Professor in the Department of Aeronautics and Astronautics at Stanford University. He is the Editor-in-chief of the IEEE Control Systems Magazine and an Associate Editor for the AIAA Journal of Aerospace Information Systems. Professor How was the recipient of the 2002 Institute of Navigation Burka Award, a Boeing Special Invention award in 2008, the IFAC Automatica award for best applications paper in 2011, the AeroLion Technologies Outstanding Paper Award for the Journal Unmanned Systems in 2015, won the IEEE Control Systems Society Video Clip Contest in 2015, and received the AIAA Best Paper in Conference Awards in 2011, 2012, and 2013. He is a Fellow of AIAA and a senior member of IEEE.