Pattern Recognition 41 (2008) 3271 – 3286

Contents lists available at ScienceDirect

Pattern Recognition

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

Kernels, regularization and differential equations

Florian Steinke*, Bernhard Schölkopf

Max-Planck-Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen, Germany

ARTICLE INFO

Article history:
Received 18 March 2008
Received in revised form 3 June 2008
Accepted 5 June 2008

Keywords:
Positive definite kernel
Differential equation
Gaussian process
Reproducing kernel Hilbert space

ABSTRACT

Many common machine learning methods such as support vector machines or Gaussian process inference make use of positive definite kernels, reproducing kernel Hilbert spaces, Gaussian processes, and regularization operators. In this work these objects are presented in a general, unifying framework and interrelations are highlighted.

With this in mind we then show how linear stochastic differential equation models can be incorporated naturally into the kernel framework. And vice versa, many kernel machines can be interpreted in terms of differential equations. We focus especially on ordinary differential equations, also known as dynamical systems, and it is shown that standard kernel inference algorithms are equivalent to Kalman filter methods based on such models.

In order not to cloud qualitative insights with heavy mathematical machinery, we restrict ourselves to finite domains, implying that differential equations are treated via their corresponding finite difference equations.

1. Introduction

Kernel methods are commonly used in the field of pattern recognition. For example, the authors of Ref. [1] have developed a support vector machine (SVM)-based face detector that works in real time on video data, and Ref. [2] uses SVMs for the tracking of humans with extensive pose articulation. Moreover, unsupervised detection of brain activation patterns is explored by [3] using one-class SVMs. The authors of Ref. [4] determine structured error patterns in microarray data using probabilistic kernel methods, and Ref. [5] uses a similar approach for processing motion capture data. Many such pattern recognition methods use SVMs for binary classification [6–8]. However, kernel methods are also employed for multi-class classification [9], regression [10], novelty detection [11], semi-supervised learning [12] and dimensionality reduction [13]. Gaussian processes (GPs) are the Bayesian versions of kernel methods. They have also been applied to classification [14,15], regression [16,17] or dimensionality reduction [18]. All these kernel methods are built around some common notions and objects, which are explained in this paper in a simple unifying way.

As depicted in Fig. 1, support vector machines can be thought of as follows. They first map the training and test input data into a potentially infinite dimensional feature space, a *reproducing kernel Hilbert space* (RKHS), and then classify the data with the help of a separating hyperplane. Since there are often many hyperplanes that separate the training data points, SVMs select the hyperplane with the largest *margin*, that is, the largest distance between the hyperplane and the data points. However, what is the intuitive meaning of distance in this feature space? One way to understand such distances is to explicitly choose a specific feature function Φ of which all components have some problem-dependent meaning. However, often the RKHS and its corresponding norm are only defined implicitly via the choice of a *kernel function* $k(x,y)=\Phi(x)^{\mathrm{T}}\Phi(y)$. In this case, the interpretation is not as straightforward. It was noted by Ref. [19] that any kernel function is related to a specific *regularization operator*. The present paper explains this connection in a simple but very general form, and we show how it can help to better understand SVMs and other related kernel machines.

Furthermore, it turns out that for the commonly used Gaussian (RBF) kernel, the feature space is a subset of the space of all functions from the input domain to the real numbers, and the corresponding regularization operator is an infinite sum of derivative operators [20]. We generalize this result and show that all translation-invariant kernel functions are related to differential operators. The corresponding homogeneous differential equations (DEs) are a useful tool for understanding the meaning of specific kernel functions. However, we could also exploit this relation in the inverse direction and construct kernels that are specifically adapted to problems involving DE models. To make this point clearer, let us consider a simple regression example from physics, which can be visualized easily and which we will thus use throughout the paper. Assume that we have acquired

* Corresponding author. Tel.: +49 7071 601571; fax: +49 7071 601552.
E-mail addresses: steinke@tuebingen.mpg.de (F. Steinke), bs@tuebingen.mpg.de (B. Schölkopf).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.06.011

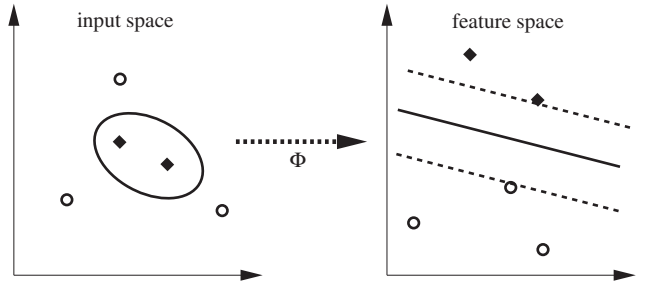

Fig. 1. Support vector machines map input data points via Φ into a potentially infinite dimensional feature space. The classification then proceeds by finding the separating hyperplane with the largest margin between the classes. However, what is the meaning of distance in this feature space? Especially if the feature space is only defined implicitly via a kernel function $k(x,y)=\Phi(x)^{\mathrm{T}}\Phi(y)$?.

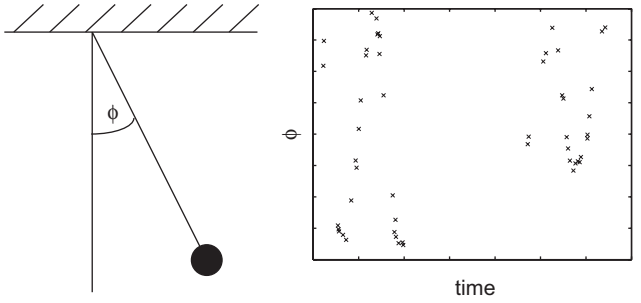

Fig. 2. (left) Schematic view of a pendulum, and (right) 50 noisy measurements of the pendulum's angle $\phi(t_i)$ at times t_i, $i=1,\ldots,50$.

measurements of a pendulum's position at given time instances, as depicted in Fig. 2. We are then interested in two problems:

Firstly, we will discuss how to optimally reconstruct the full time course of the pendulum's position. The pendulum's dynamics can be described approximately by a simple linear DE, and estimating the full state trajectory from few measurements is equivalent to classical state estimation in linear dynamical systems. For this task one typically employs a variant of the Kalman filter. On the other hand, the problem of reconstructing a function from a finite number of measurements is also the goal of non-parametric regression techniques, such as the kernel-based methods support vector machines/support vector regression (SVR) or GP inference. In this paper, we will show how the knowledge of a model DE can be included into kernel methods, and that these are closely related to Kalman filter-based approaches.

Secondly, we will explore how to learn about properties of the pendulum from the given measurements. In particular this will aim at determining parameters of the DE that characteristically describes the pendulum, a task that is commonly known as linear system identification. We will show how model selection methods for kernel methods such as cross-validation or marginal likelihood optimization can be used for system identification purposes. As for state estimation, these machine learning-inspired approaches turn out to be equivalent to other well-known system identification methods, such as prediction error methods.

Having these two objectives in mind, we will first describe kernel methods in a relatively broad way that is not specifically tailored towards DEs. However, this framework will allow us to straightforwardly understand the close links between linear DEs and kernel methods as a special case. We mostly focus on ordinary linear DEs, also known as dynamical systems, but will also give examples of linear partial differential equations (PDEs). Other linear operator equations could also be dealt with similarly. By DEs we will in this paper always mean stochastic DEs, since these can be nicely incorporated into kernel methods. Stochastic DEs are a superset of normal DEs, since any DE can be converted into a stochastic DE by setting the noise level to zero.

1.1. Finite domains

The current paper is formulated in terms of finite domains. Functions to be estimated are assumed to map finite domains to $\mathbb{R}$ or $\mathbb{R}^n$. In the pendulum example imagine time to be discretized into many small time steps. The use of finite domains thus means that whenever we speak of DEs in this paper we actually mean discretized versions thereof, that is, the corresponding finite difference equations.

In the authors' opinion, finite domains are just the right level of simplification needed for an easy, yet very far-reaching exposition of the matter. The restriction to finite domains simplifies the required mathematics dramatically. Functions on finite domains are finite dimensional vectors, requiring only simple linear algebra for analysis instead of more involved functional analysis. Existence and convergence of sums/integrals is trivial for finite domains, and point evaluations are described by inner products with unit vectors instead of functionals involving Dirac-delta distributions. Finite domains also allow one to define Gaussian densities for function-valued random variables. This is not possible for infinite dimensional functions, at least not with respect to the standard Lebesgue measure, which does not exist for infinite dimensional function spaces [21].

Despite these important simplifications, little qualitative expression power is lost. Most well-known results on kernels can be easily derived and motivated for finite domains. Reasonably smoothly varying functions can be approximated well by their finite dimensional piecewise-linear counterparts, which, in most cases, allow DEs to be converted straightforwardly into qualitatively equivalent finite difference equations. Finally, there are also some common settings for machine learning that naturally deal with finite domains, for example graph-based or transductive learning.

There are, of course, also certain shortcomings of a finite domain approach. Generally speaking, we cannot answer questions regarding the limiting behaviour for ever smaller discretization steps. Note that while such limiting processes on continuous domains typically exist, see e.g. Ref. [22] for one-dimensional domains, they often have some additional surprising properties, some of which are at first sight in conflict with our understanding of the corresponding model for finite domains. For example, the sample paths of Brownian motion are continuous, yet nowhere differentiable [22]. This implies that the corresponding RKHS norm, defined below, is infinite for each sample path almost surely. While the RKHS is thus a null space under the measure of the continuous time process, the mean of non-parametric regression with a finite number of data points is nevertheless guaranteed to be an element of the RKHS, a very surprising fact. Also, if we define our models via discrete regularization operators or inverse covariances as defined below and then take the limit of step size to zero, then the marginal distributions of these continuous processes are often not identical to the finite distributions. For example, for the linear difference equation $\boldsymbol{x}_i=(\mathbf{1}+\boldsymbol{A}\Delta t)\boldsymbol{x}_{i-1}$ the exact discretization of the continuous analog would be $\boldsymbol{x}_i=\exp(\boldsymbol{A}\Delta t)\boldsymbol{x}_{i-1}$. While these expressions are similar for small step sizes Δt they are not identical. This fact is sometimes important for computational reasons, since by construction the discrete models often have some sparsity structures in the inverse covariance and these are not, in general, preserved for the marginals.

The aim of this paper is to offer a simple intuitive introduction to the kernel framework and to show its connections to DEs. We thus concentrate solely on finite domains. Note that this means that when speaking of processes in this paper, we just mean distributions

over functions on a given *fixed* finite domain. We do not make statements about what happens if one or more points are added to the domain of the model, and the defined processes are not assumed to be marginals of their continuous analogs.

1.2. Overview

The remainder of the paper is structured as follows: after introducing some notation in Section 2, we define in Section 3 a framework of basic objects used in kernel methods, and we explain how these objects are interrelated. Thereafter, we describe the use of these objects for SVR in Section 3.2, for GP regression in Section 3.3, and for vector-valued regression in Section 3.4. In Section 4, we discuss a typical kernel-machine regression model and show its relation to linear stochastic DEs. We demonstrate how to develop kernel functions from linear state-space models or higher-order DEs. We show that the resulting inference methods are equivalent to Kalman filter-based methods. The pendulum and other examples are presented in detail in Section 5. In Section 6 we discuss the practical implications of the link between kernel machines and linear stochastic DEs. We summarize our conclusions in Section 7.

For better readability, we have restricted the main part of the paper to real-valued kernels, and postpone the more natural, slightly more technical treatment involving complex numbers to Appendix A. It will appear throughout the text that, with regularization theory in mind, conditionally positive definite (cpd) kernels arise quite naturally. We have transferred all parts dealing with cpd kernels to Appendix B, where we present an extension of the kernel framework to cpd kernels.

1.3. Related work

Most of the mathematical results of this paper are not the authors' original work, but have been mentioned in different contexts before. Our contribution is to reformulate them in a unified, easily understandable framework, the simple language of finite domains. Furthermore, we reinterpret them to highlight parallels between kernel methods and linear DEs.

There is a large body of literature on kernels and DEs in many different communities, and we only cite some relevant books containing overviews of their respective fields as well as further references. Many machine learning-related facts about kernels and regularization methods are taken from Ref. [8], as well as Ref. [23] for the Bayesian interpretation. Sources in the statistics literature include [24,25], and in approximation theory [26]. For an overview of linear stochastic dynamical systems and their estimation we refer to Ref. [27].

The connection between stochastic processes and splines was first explored in Ref. [28]. It is also well known that thin-plate/cubic splines minimize the second derivative [29,26]. Connections between regularization operators and kernel functions are explained in Refs. [20,30], and general linear operator equations are solved with GPs in Ref. [31]. A unifying survey of the theory of kernels, RKHSs, and GPs has been undertaken by Ref. [32]. However, they do not use finite domains, which complicates their study and they do not mention the link with differential or operator equations. Approaches that directly employ kernel methods towards the estimation of stochastic DE models are proposed in Refs. [33,34].

2. Notation

We consider functions $f : \mathscr{X} \to \mathbb{R}$, where the domain $\mathscr{X}$ is a finite set, $|\mathscr{X}| = N$. When considering dynamical systems we will typically set $\mathscr{X}$ to be an evenly discretized interval and assume N to be large. Other examples of finite domains are discretized regions of higher dimensional spaces, but also finite sets of graphs, texts, or any other type of objects.

We denote by $\mathscr{H}$ the space of all functions $f : \mathscr{X} \to \mathbb{R}$. f is fully described by the $\mathbb{R}^N$-vector $\boldsymbol{f} = (f(x_1), \ldots, f(x_N))^{\mathrm{T}}$. Vectors and matrices are denoted in bold font, but if an element of $\mathscr{H}$ is thought of as a function from $\mathscr{X}$ to $\mathbb{R}$, we use the corresponding normal font character. For points $x_i \in \mathscr{X}$ we define *location vectors/functions* by $\boldsymbol{\delta}_{x_i} = (\delta_{ij})_{j=1,\ldots,N}$, where δ_{ij} is the Kronecker symbol. The inner product of these with a function $\boldsymbol{f} \in \mathscr{H}$ yields $\boldsymbol{\delta}_{x_i}^{\mathrm{T}} \boldsymbol{f} = f(x_i)$. Thus, location vectors correspond to Dirac-delta functions centred at the point x_i for continuous, infinite domains.

Linear operators $\boldsymbol{G} : \mathscr{H} \to \mathscr{H}$ are isomorphic to matrices in $\mathbb{R}^{N \times N}$. Therefore, any function $g : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ uniquely determines a linear operator $\boldsymbol{G} : \mathscr{H} \to \mathscr{H}$ through $\boldsymbol{G}_{ij} = \boldsymbol{\delta}_{x_i}^{\mathrm{T}} \boldsymbol{G} \boldsymbol{\delta}_{x_j} = g(x_i, x_j)$ and vice versa. The columns of $\boldsymbol{G}$ will be noted by $\boldsymbol{G}_{x_i} = \boldsymbol{G}\boldsymbol{\delta}_{x_i}$; they are real-valued functions on $\mathscr{X}$. For a set $X = \{x_i | i = 1, \ldots, m\} \subseteq \mathscr{X}$ of points, $\boldsymbol{G}_X$ will denote the $m \times m$ submatrix of $\boldsymbol{G}$ corresponding to X.

3. The kernel framework

In non-parametric regression, we are given observations $(x_i, y_i) \in \mathscr{X} \times \mathbb{R}$, $i = 1, \ldots, m$, $m \leqslant N$, and the goal is to predict the value y_* for arbitrary test points $x_* \in \mathscr{X}$. SVR estimates a prediction function $f : \mathscr{X} \to \mathbb{R}$, $y_* = f(x_*)$, as the minimizer of a functional like

$$\min_{\boldsymbol{f} \in \mathscr{H}} \| \boldsymbol{R}\boldsymbol{f} \|^2 + C \, \mathrm{Loss}(\{(x_i, y_i, f(x_i)) | i = 1, \ldots, m\}). \tag{1}$$

On the one hand, f should be close to the observed data as measured through a *loss function* $\mathrm{Loss} : (\mathscr{X} \times \mathbb{R} \times \mathbb{R})^m \to \mathbb{R}$. On the other hand, f should be *regular* as measured by the *regularization operator* $\boldsymbol{R} : \mathscr{H} \to \mathscr{G}$, where $\mathscr{G}$ is any finite dimensional Hilbert space. These two objectives are relatively weighted through the *regularization parameter* C. Note that SVMs also use the same setting for binary classification. The classes are represented as $y = \pm 1$. First a real-valued function $f : \mathscr{X} \to \mathbb{R}$ is estimated and then thresholded to obtain the binary class predictions. Unlike radial basis function networks [20,35], SVMs use the hinge loss $|yf(x) - 1|_+$ where $|x|_+ = x$ if $x > 0$ and $|x|_+ = 0$ otherwise.

Many questions arise around objective (1). How are $\| \boldsymbol{R}\boldsymbol{f} \|^2$ and the commonly used function space norm $\| \boldsymbol{f} \|_K^2$ related? This will lead to the notion of RKHSs. The N-dimensional problem (1) can be solved using a smaller m-dimensional equivalent involving *kernel functions*. But how does $\boldsymbol{R}$ relate to the chosen kernel function? Can one interpret (1) in a Bayesian way? For example, with the help of GPs? The current section will answer the above questions in a simple, yet precise way for finite domains. We will furthermore show the interrelations between the terms mentioned above.

Throughout the main part of this paper we assume that $\boldsymbol{R}$ is a one-to-one operator. This will lead to a framework with positive definite kernels. If $\boldsymbol{R}$ is not one-to-one, cpd kernels arise. All definitions and theorems derived for the positive definite case in the current section are extended to the cpd case in Appendix B.

3.1. Regularization operators, kernels, RKHS, and GPs

Fig. 3 depicts the most common objects in the kernel framework. We will explain them below, starting with the covariance operator. The covariance operator is not commonly used in the kernel literature, but we introduce it as a useful abstraction in the centre of the framework. While it does not in itself have a special meaning, it helps us to unify the links between the other "leaf" objects. With the covariance operator in mind, the reader may then easily derive additional direct links.

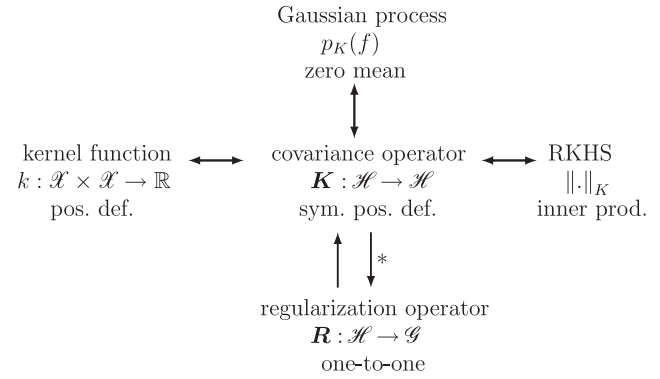

Fig. 3. Common objects in the kernel framework and their interrelations. Arrows denote that one can uniquely be determined from the other (the * denotes that this connection is not unique).

Definition 1 (*Covariance operator*). A *covariance operator* $\boldsymbol{K}$ is a positive definite matrix of size $N \times N$, i.e. for all $\boldsymbol{f} \in \mathscr{H}, \boldsymbol{f} \neq \boldsymbol{0}$, it is $\boldsymbol{f}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{f} > 0$.

A first interpretation of the covariance operator which gives $\boldsymbol{K}$ its name is given through its use in GPs.

Definition 2 (*Gaussian process*). A *GP* is a distribution over all functions $f : \mathscr{X} \to \mathbb{R}$ such that for any linear functional $w : \mathscr{H} \to \mathbb{R}$ the value $w(\boldsymbol{f}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{f}$ is a real-valued, normally distributed random variable.

This definition taken from Ref. [21] is tailored to the case where f is infinite dimensional, and no Lebesgue density exists in $\mathscr{H}$. For finite $\mathscr{X}$, it simply implies that the distribution has a density $p_K(f)$ over the functions in $\mathscr{H}$, and that this density is a multivariate Gaussian. Note that this means that in the finite dimensional setting, distributions over functions can be described via standard multivariate Gaussian distributions. Given a covariance operator $\boldsymbol{K}$ we can define a special zero mean GP by

$$p_K(f) = N(0, \boldsymbol{K}) \propto \exp(-\tfrac{1}{2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{f}^{2}). \tag{2}$$

Conversely, given a GP, its covariance matrix is a valid positive definite covariance operator.

The covariance operator also allows one to define another well-known object.

Definition 3 (*Kernel function*). A symmetric function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is called a *positive definite kernel function*, if for all subsets $X \subseteq \mathscr{X}$, $X = \{x_1, \ldots, x_m\}$, $m \leqslant N$, and all $0 \neq \boldsymbol{\alpha} \in \mathbb{R}^m$, it holds that

$$\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j k(x_i, x_j) = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{K}_X\boldsymbol{\alpha} = \left(\sum_{i=1}^{m}\alpha_i\boldsymbol{\delta}_{x_i}\right)^{\mathrm{T}}\boldsymbol{K}\left(\sum_{j=1}^{m}\alpha_j\boldsymbol{\delta}_{x_j}\right) > 0.$$

By definition, kernel functions give rise to a positive definite covariance operator $\boldsymbol{K}_{\mathscr{X}}$. Conversely, a covariance operator $\boldsymbol{K}$ defines a kernel function through $k(x_i, x_j) = \boldsymbol{K}_{ij} = \boldsymbol{\delta}_{x_i}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\delta}_{x_j}$, since positive definiteness of $\boldsymbol{K}$ implies that $\boldsymbol{K}_X$, too, is positive definite for all $X \subseteq \mathscr{X}$.

Kernel functions naturally lead to the definition of specially adapted function spaces.

Definition 4 (*Reproducing Kernel Hilbert Space*). A Hilbert space $(\mathscr{S}, (.,.)_S)$, $\mathscr{S} \subseteq \mathscr{H}$, of functions $f : \mathscr{X} \to \mathbb{R}$ is called a *RKHS*, if the evaluation functionals $\delta_{x_i} : \mathscr{H} \to \mathbb{R}$ defined by $\delta_{x_i}(f) = \boldsymbol{\delta}_{x_i}^{\mathrm{T}}\boldsymbol{f} = f(x_i)$ are continuous for all $x_i \in \mathscr{X}$, i.e. $|\delta_{x_i}(f)| \leqslant C\|\boldsymbol{f}\|_S$ for all $\boldsymbol{f} \in \mathscr{S}$.

As for the definition of GPs, this formulation of the definition of RKHSs is tailored towards the continuous domain case. The definition ensures that point evaluations of functions in $\mathscr{S}$ are well-defined, which is not obvious for functions on continuous domains, for example, L_2 functions. Well-defined point evaluations are, of course, necessary for machine learning methods that deal with point-wise data measurements. In the finite domain setting, the definition of RKHSs is quite trivial. It implies that $\mathscr{H}$ with any inner product $(.,.)_S$ is an RKHS, also with the usual L_2 inner product. The proof is found in Appendix C, together with the proof of the following lemma which summarizes some useful results about RKHSs.

Lemma 5. *The following statements hold for RKHS* $(\mathscr{H}, (.,.)_S)$:

(1) *There exists a unique element* $\boldsymbol{S}_{x_i} \in \mathscr{H}$ *for each* $x_i \in \mathscr{X}$, *the* representer, *such that*

$$\delta_{x_i}(f) = f(x_i) = (\boldsymbol{S}_{x_i}, \boldsymbol{f})_S$$

for all $\boldsymbol{f} \in \mathscr{H}$. *This property is called the* reproducing property.

(2) *The function* $s : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ *defined by* $s(x_i, x_j) = (\boldsymbol{S}_{x_i}, \boldsymbol{S}_{x_j})_S$ *is a positive definite kernel function in the sense of Definition* 3.

Let the operator $\boldsymbol{S} : \mathscr{H} \to \mathscr{H}$ *be defined by* $\boldsymbol{S}_{ij} = s(x_i, x_j)$.

(3) *Any inner product* $(\boldsymbol{f}, \boldsymbol{g})_S$ *can be uniquely expressed in the form* $\boldsymbol{f}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{g}$ *where* $\boldsymbol{T}$ *is a positive definite operator.*

(4) $s(x_i, x_j) = \boldsymbol{T}_{ij}^{-1}$ *or equivalently* $\boldsymbol{S} = \boldsymbol{T}^{-1}$.

(5) *The kernel* s *defines the inner product* $(.,.)_S$ *uniquely.*

The above lemma implies that for a given covariance operator $\boldsymbol{K}$ one can define an RKHS $(\mathscr{H}, (.,.)_K)$ by setting

$$(\boldsymbol{f}, \boldsymbol{g})_K \equiv \boldsymbol{f}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{g}.$$

Then the representer of this RKHS is identical with the kernel function $\boldsymbol{K}\boldsymbol{\delta}_{x_i}$ derived from $\boldsymbol{K}$ via $k(x_i, x_j) = \boldsymbol{K}_{ij}$. Since the relation between kernel and inner product is unique, one could also construct a unique valid covariance operator from a given RKHS.

The definitions so far have been purely technical, but we can give them a practical meaning when considering them in conjunction with a regularization operator as used in the SVR objective (1).

Definition 6 (*Regularization operator*). A *regularization operator* $\boldsymbol{R} : \mathscr{H} \to \mathscr{G}$ is a one-to-one linear operator. Here, $\mathscr{G}$ is any finite dimensional Hilbert space.

If we use $\boldsymbol{K} = (\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R})^{-1}$, then by Lemma 5 it is

$$\|\boldsymbol{f}\|_K^2 = \boldsymbol{f}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{f} = \boldsymbol{f}^{\mathrm{T}}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{f} = \|\boldsymbol{R}\boldsymbol{f}\|^2.$$

That means that if $\|\boldsymbol{R}\boldsymbol{f}\|$ measures the *regularity* of $f : \mathscr{X} \to \mathbb{R}$, then the RKHS norm exactly equals the regularity measure. In the SVR objective (1) regular functions are thus preferred over less regular ones. Furthermore, the related GP is

$$p_K(f) = N(0, \boldsymbol{K}) \propto \exp(-\tfrac{1}{2}\|\boldsymbol{R}\boldsymbol{f}\|^2)$$

implying that under this distribution regular functions are more likely than less regular ones. The most likely functions are those which exactly fulfill the *regularity/model equation*

$$\boldsymbol{R}\boldsymbol{f} = 0.$$

Note that since $\boldsymbol{R}$ is assumed to be one-to-one, only the zero function can fulfill the model equation exactly. Non-vanishing functions

Table 1
Summary of the objects of the positive definite kernel framework and their interrelations

Entity	Symbol	Relations
Kernel function	$k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$	$k(x_i, x_j) = \boldsymbol{K}_{i,j} = \boldsymbol{\delta}_{x_i}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\delta}_{x_j}$
	$\boldsymbol{K}_{x_i} : \mathscr{X} \to \mathbb{R}$	$k(x_i, x_j) = (\boldsymbol{K}_{x_i}, \boldsymbol{K}_{x_j})_K$
		$k(x_i, x_j) = \boldsymbol{\delta}_{x_i}^{\mathrm{T}} (\boldsymbol{R}^{\mathrm{T}} \boldsymbol{R})^{-1} \boldsymbol{\delta}_{x_j}$
		$k(x_i, x_j) = \mathrm{Cov}_{\boldsymbol{f} \sim p_K}(f(x_i), f(x_j))$
Covariance op.	$\boldsymbol{K} : \mathscr{H} \to \mathscr{H}$	$\boldsymbol{K}_{i,j} = k(x_i, x_j)$,
		$\boldsymbol{K} = (\boldsymbol{R}^{\mathrm{T}} \boldsymbol{R})^{-1} = \mathrm{Cov}_{\boldsymbol{f} \sim p_K}(\boldsymbol{f}, \boldsymbol{f})$
RKHS	$(.,.)_K : \mathscr{H} \times \mathscr{H} \to \mathbb{R}$	$(\boldsymbol{f}, \boldsymbol{g})_K = \boldsymbol{f}^{\mathrm{T}} \boldsymbol{K}^{-1} \boldsymbol{g} = \boldsymbol{f}^{\mathrm{T}} \boldsymbol{R}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{g}$
	$\|.\|_K . : \mathscr{H} \to \mathbb{R}$	$\|\boldsymbol{f}\|_K = (\boldsymbol{f}, \boldsymbol{f})_K^{1/2} = \|\boldsymbol{R}\boldsymbol{f}\|$
Gaussian process	$p_K : \mathscr{H} \to \mathbb{R}$	$p_K(f) = N(0, \boldsymbol{K})$
		$p_K(f) \propto \exp(-\frac{1}{2} \|\boldsymbol{f}\|_K^2)$
		$p_K(f) \propto \exp(-\frac{1}{2} \|\boldsymbol{R}\boldsymbol{f}\|^2)$
Regularization op.	$\boldsymbol{R} : \mathscr{H} \to \mathscr{G}$	($\boldsymbol{R} = \sqrt{\boldsymbol{K}^{-1}}$, not unique)

$\mathrm{Cov}_{\boldsymbol{x} \sim p(\boldsymbol{x})}(x_i, x_j)$ denotes the covariance between x_i and x_j under a distribution of $\boldsymbol{x}$ with density $p(\boldsymbol{x})$. If the arguments are vectors, the corresponding covariance matrix is meant.

violate this equation by an amount that is determined by the structure of $\boldsymbol{R}$. If non-trivial functions are to be considered fully regular, that is, $\|\boldsymbol{R}\boldsymbol{f}\| = 0$, then $\boldsymbol{R}$ cannot be one-to-one. This case is discussed in Appendix B.

Given a covariance operator $\boldsymbol{K}$, we can compute an associated regularization operator $\boldsymbol{R}$ as $\boldsymbol{R} = \sqrt{\boldsymbol{K}^{-1}}$. However, note that if we transform $\boldsymbol{R} \to \boldsymbol{K} \to \boldsymbol{R}$ in this way we will not necessarily recover the same regularization operator we started from. The original $\boldsymbol{R}$ does not have to be quadratic and even if it is, taking the root would set all originally negative eigenvalues of $\boldsymbol{R}$ to positive.

The objects of the kernel framework and their interrelations are summarized in Table 1.

3.2. Support vector machines

With the above definitions the SVR objective (1) can be rewritten as

$$\min_{\boldsymbol{f} \in \mathscr{H}} \|\boldsymbol{f}\|_K^2 + C \, \mathrm{Loss}(x_i, y_i, f(x_i))_{i=1,\ldots,m}. \tag{3}$$

This optimization problem over the whole function space $\mathscr{H}$, i.e. over N variables where N is potentially large, can be reduced to a typically much smaller m-dimensional optimization problem using kernel functions. To see this, we will derive the famous representer theorem in two steps. The proofs are found in Appendix C.

The first step, which is interesting in itself, shows a general property of RKHSs: Any function in an RKHS can be decomposed into a set of kernel functions and its $\mathscr{H}$-orthogonal complement. If the complement is understood as a function from $\mathscr{X}$ to $\mathbb{R}$, then it has function value zero at all kernel centres.

Lemma 7. *Given distinct points* $X = \{x_i | i = 1, \ldots, m\}$, $m \leqslant N$, *any* $\boldsymbol{f} \in \mathscr{H}$ *can be uniquely written as* $\boldsymbol{f} = \sum_{i=1}^m \alpha_i \boldsymbol{K}_{x_i} + \boldsymbol{\rho}$, $\boldsymbol{\alpha} \in \mathbb{R}^m$, *where* $\boldsymbol{\rho} \in \mathscr{H}$ *satisfies the conditions* $\rho(x_i) = (\boldsymbol{K}_{x_i}, \boldsymbol{\rho})_K = 0$, $i = 1, \ldots, m$.

The second step then is as follows.

Theorem 8 (*Representer theorem*). *Given* $m \leqslant N$ *distinct points* $X = \{x_i | i = 1, \ldots, m\}$ *and labels* $\{y_i | i = 1, \ldots, m\} \subseteq \mathbb{R}$ *the minimizer* $\boldsymbol{f}$ *of Eq.* (3) *has the form* $\boldsymbol{f}_{\boldsymbol{\alpha}} = \sum_{i=1}^m \alpha_i \boldsymbol{K}_{x_i}$. $\boldsymbol{\alpha} \in \mathbb{R}^m$ *minimizes the expression*

$$\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K}_X \boldsymbol{\alpha} + C \mathrm{Loss}(x_i, y_i, f_{\boldsymbol{\alpha}}(x_i))_{i=1,\ldots,m}. \tag{4}$$

If the loss is convex, $\boldsymbol{\alpha}$ *is determined uniquely.*

Remark. f can also be expanded in another function system, say $\boldsymbol{f} = \sum_{j=1}^L c_j \boldsymbol{\phi}_j$. Then $\min_{\boldsymbol{c} \in \mathbb{R}^L} \boldsymbol{c}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{c} + \mathrm{Loss}(x_i, y_i, \boldsymbol{f}_{\boldsymbol{c}}(x_i))_{i=1,\ldots,m}$ with $\boldsymbol{M}_{ij} = \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{R}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\phi}_j$ is the optimization problem corresponding to Eq. (1), see e.g. Refs. [25,36]. This is also a convex problem and can sometimes be solved very efficiently if, for example, compactly supported basis functions are used [36]. However, one only finds the optimal solution within the span of the selected basis functions. A globally optimal solution in $\mathscr{H}$ would, in general, require $L = N$ basis functions. Furthermore, $\boldsymbol{M}_{ij} = \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{R}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\phi}_j$ has to be computed for all i,j which could be challenging.

3.3. GP inference

The SVR objective (1) can also be interpreted from a Bayesian perspective. Assume a two step-model where firstly a latent function $f : \mathscr{X} \to \mathbb{R}$ is drawn from the *GP prior* $p_K(f)$ with covariance operator $\boldsymbol{K}$, and where subsequently the measurements are determined from this function as described by a *local likelihood* $p(\boldsymbol{y}|f) = p(\boldsymbol{y}|\boldsymbol{f}_X)$, where $\boldsymbol{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$ and $X = \{x_1, \ldots, x_m\}$. A common example of a local likelihood is the i.i.d. likelihood, that is, $p(\boldsymbol{y}|f) = \prod_i p(y_i | f(x_i))$. The posterior for local likelihoods is

$$p(f | \boldsymbol{y}, X) \propto p(\boldsymbol{y}|f) p_K(f) \propto p(\boldsymbol{y}|\boldsymbol{f}_X) \exp(-\tfrac{1}{2} \|\boldsymbol{R}\boldsymbol{f}\|^2),$$

and the maximum a posteriori (MAP) estimate is

$$\arg\max_{\boldsymbol{f} \in \mathscr{H}} p(f | \boldsymbol{y}, X) = \arg\min_{\boldsymbol{f} \in \mathscr{H}} \tfrac{1}{2} \|\boldsymbol{R}\boldsymbol{f}\|^2 - \log p(\boldsymbol{y}|\boldsymbol{f}_X).$$

So if one can identify $-\log p(\boldsymbol{y}|\boldsymbol{f}_X)$ with $\mathrm{Loss}(\{(x_i, y_i, f(x_i)) | i = 1, \ldots, m\})$, which is possible, for example, for the common squared loss, then SVR is just a MAP estimate of a GP model. Note, however, that in some well-known cases such as, for example, the hinge loss, this identification is in a strict sense not possible. The resulting likelihood would not be normalizable with respect to $\boldsymbol{y}$. Nevertheless, if one is willing to work with unnormalized models, the equivalence holds in general. The qualitative meaning of the prior is the same in any case.

Bayesian statistics is typically not only interested in the MAP estimate of $f(x_*)$ but in the full predictive distribution,

$$p(f(x_*) | \boldsymbol{y}, X) \propto \int p(\boldsymbol{y}|\boldsymbol{f}_X) p_K(f) \, \mathrm{d}\boldsymbol{f}_{\mathscr{X} \setminus x_*}.$$

Here, we have used the notation that for every set $I = \{x_{i_1}, \ldots, x_{i_k}\} \subseteq \mathscr{X}$, $\mathrm{d}\boldsymbol{f}_I$ means $\mathrm{d}f(x_{i_1}), \ldots, \mathrm{d}f(x_{i_k})$. Because of the local likelihood we can then split the $N - 1$ dimensional integral as follows:

$$p(f(x_*) | \boldsymbol{y}, X) \propto \int p(\boldsymbol{y}|\boldsymbol{f}_X) \underbrace{\left(\int p_K(f) \, \mathrm{d}\boldsymbol{f}_{\mathscr{X} \setminus X \cup x_*} \right)}_{= p_K(\boldsymbol{f}_{X \cup x_*})} \mathrm{d}\boldsymbol{f}_X.$$

So if an analytic expression of the marginal $p_K(\boldsymbol{f}_{X \cup x_*})$, which is independent of the data, could be computed, then only an m-dimensional integral would have to be solved for inference. Such an expression is given in the following theorem, which just expresses a standard property of Gaussian distributions. Since it reduces the work from N dimensions to m dimensions similar to the representer theorem 8, one could call it the *Bayesian representer theorem*.

Theorem 9. *Given* $m \leqslant N$ *distinct points* $X = \{x_1, \ldots, x_m\} \subseteq \mathscr{X}$ *the GP* $p_K(f)$ *has the marginals*

$$p_K(\boldsymbol{f}_X) = \frac{1}{\sqrt{(2\pi)^m |\boldsymbol{K}_X|}} \exp(-\tfrac{1}{2} \boldsymbol{f}_X^{\mathrm{T}} \boldsymbol{K}_X^{-1} \boldsymbol{f}_X) = N(0, \boldsymbol{K}_X).$$

This property is often used to construct GPs: Given a kernel function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ one stores the values corresponding to X into

a square matrix $\boldsymbol{K}_X$ and sets $p(\boldsymbol{f}_X) = N(0, \boldsymbol{K}_X)$. Using standard formulas for conditioning Gaussian distributions and block-partitioned matrix inversion one can show that this construction is *consistent*, i.e. for all $X' \subseteq \mathscr{X}, X \cap X' = \emptyset$ it holds that $p(\boldsymbol{f}_X) = \int p(\boldsymbol{f}_{X \cup X'}) \mathrm{d}\boldsymbol{f}_{X'}$. By Kolmogorov's extension theorem, or by simply using $X = \mathscr{X}$ in our finite dimensional case, this yields a GP on all of $\mathscr{X}$.

3.4. Vector-valued regression

Consider now regression from $\mathscr{X}$ to $\mathbb{R}^n$, $n > 1$. We will show that the kernel framework explained above can be easily extended to this case. The function space of all functions $\boldsymbol{f} : \mathscr{X} \to \mathbb{R}^n$ will be denoted by $\mathscr{H}^n$. We can represent such a function as a vector $\underline{\boldsymbol{f}}$ in $\mathbb{R}^{nN}$. Denoting the component functions by $f^i : \mathscr{X} \to \mathbb{R}$ it is $\underline{\boldsymbol{f}} \equiv (\boldsymbol{f}^{1\mathrm{T}} \cdots \boldsymbol{f}^{n\mathrm{T}})^{\mathrm{T}}$. The standard inner product in $\mathscr{H}^n$ is $\underline{\boldsymbol{f}}^{\mathrm{T}} \underline{\boldsymbol{g}} = \sum_{j=1}^{n} \boldsymbol{f}^{j\mathrm{T}} \boldsymbol{g}^j$. The unit vector $\underline{\boldsymbol{\delta}}^j_{x_i}$, i.e. the location vector for location x_i and the j-th component, then has the j-th component equal to $\boldsymbol{\delta}_{x_i}$ and all others equal to zero. It is $\boldsymbol{\delta}^{j\mathrm{T}}_{x_i} \underline{\boldsymbol{f}} = f^j(x_i)$. Linear operators $\underline{\boldsymbol{A}} : \mathscr{H}^n \to \mathscr{H}^n$ are isomorphic to $\mathbb{R}^{(Nn)\times(Nn)}$ matrices.

Theorem 10. *The function space $\mathscr{H}^n$ is isomorphic to the space $\tilde{\mathscr{H}}$ of all functions from $\tilde{\mathscr{X}} = \mathscr{X} \times \{1, \ldots, n\}$ to $\mathbb{R}$.*

This obvious theorem includes all we need in order to work with vector-valued functions: As $\mathscr{X}$ is a finite set, so is $\tilde{\mathscr{X}}$. All the above theory on kernels, regularization operators, and GPs applies. For example, using the regularization operator $\underline{\boldsymbol{R}} : \mathscr{H}^n \to \mathscr{G}$, the corresponding kernel function is

$$k(x_i, x_j)^{lm} = k((x_i, l), (x_j, m)) = \boldsymbol{\delta}^{l\mathrm{T}}_{x_i} (\underline{\boldsymbol{R}}^{\mathrm{T}} \underline{\boldsymbol{R}})^{-1} \boldsymbol{\delta}^m_{x_j}. \tag{5}$$

To construct a sensible regularizer $\underline{\boldsymbol{R}}$, a similarity measure between points in $\tilde{\mathscr{X}}$ is needed. Since in many applications it is not clear how to compare different components of $\underline{\boldsymbol{f}}$, it is common to use a block-diagonal regularizer $\underline{\boldsymbol{R}} = \mathrm{diag}(\boldsymbol{R}^1, \ldots, \boldsymbol{R}^n)$, i.e. regularizing each component separately. The corresponding kernel function then has the vector form

$$\underline{\boldsymbol{K}}^j_{x_i} = (0, \ldots, 0, \boldsymbol{K}^{j\mathrm{T}}_{x_i}, 0, \ldots, 0)^{\mathrm{T}},$$

with the individual kernel functions $\boldsymbol{K}^j_{x_i} = (\boldsymbol{R}^{j,T} \boldsymbol{R}^j)^{-1} \boldsymbol{\delta}_{x_i}$ in the corresponding components. The joint covariance matrix $\underline{\boldsymbol{K}}$ is block-diagonal in this case. If the loss/likelihood term does not imply a dependency between different components, such as, for example, the quadratic loss, then each dimension can be treated separately. However, there are also numerous situations where a joint regularization makes sense. Examples are shown in the next section.

The theory as described here was mentioned in Ref. [32]. Ref. [37] introduces a slightly different formalism employing operator-valued kernel functions in this context. However, the derived representer theorem is equivalent to the simple approach presented here.

Note that one could also reorder the entries in $\underline{\boldsymbol{f}}$; for example, we could define $\underline{\boldsymbol{f}} = (\boldsymbol{f}(x_1)^{\mathrm{T}}, \ldots, \boldsymbol{f}(x_N)^{\mathrm{T}})^{\mathrm{T}}$. While in this section we have used a special notation for vector-valued functions in order to highlight the differences, we will from now on use normal vector notation also for vector-valued functions to keep the notation simple.

3.5. Inhomogeneous regularization

As shown in the next section, there are numerous cases where one would like to have $\|\boldsymbol{Rf} - \boldsymbol{u}\|$, $\boldsymbol{u} \neq 0$, as the regularizer in the SVR objective (1) or equivalently use non-zero means for GPs.

Since for $\boldsymbol{f} = 0$, $\|\boldsymbol{Rf} - \boldsymbol{u}\| = \|\boldsymbol{u}\| \neq 0$, $\|\boldsymbol{Rf} - \boldsymbol{u}\|$ cannot be used as a norm in an RKHS. To circumvent this problem, note that since $\boldsymbol{R}$ is assumed to be one-to-one $\boldsymbol{R}^{-1}\boldsymbol{u}$ exists uniquely and can be computed without regard to the measurement data. We can then base any inference on $\tilde{\boldsymbol{f}} = \boldsymbol{f} - \boldsymbol{R}^{-1}\boldsymbol{u}$, adapting the loss term appropriately. The regularization term then reads $\|\boldsymbol{R}\tilde{\boldsymbol{f}}\| = \|\boldsymbol{Rf} - \boldsymbol{u}\|$, which represents a true norm for $\tilde{\boldsymbol{f}}$. The kernel framework can now be applied as described above.

4. Kernels and DEs

SVR and GP inference both use an a priori model that can be expressed in the form

$$\boldsymbol{Rf} \approx 0, \tag{6}$$

Functions $f : \mathscr{X} \to \mathbb{R}$ which fulfill Eq. (6) to a high degree as measured by $\|\boldsymbol{Rf}\|$, the two-norm of the residual, are preferred to functions that significantly violate the equation.

In this section we discuss a common choice for $\boldsymbol{R}$, namely linear stochastic DEs. If the input domain is one-dimensional, one speaks of ordinary differential equations (ODEs) or dynamical systems, and for multivariate input these are PDEs. Since this paper is restricted to finite domains, the term DE should be understood as meaning finite difference equations throughout. In most cases, the differences are negligible for discretization steps that are sufficiently small.

Linking DEs and kernel machines is useful both from a machine learning perspective as well as from a perspective focused primarily on work with DEs.

From a machine learning point of view, stochastic DEs can be seen as an ideal prior model. They describe *local* properties of the function f, that is, how the function value at one point relates to function values in the neighbourhood. On a *global* level, stochastic DEs do not constrain the function very much, because small local noise contributions can add up over longer distances. Thus, this prior is well-suited to situations where we a priori do not know much about the global structure of the target function, but we assume that locally it should not vary too much or only in a certain predefined manner.

From a DE point of view, it is useful to have all the machinery of kernel methods at hand. With these, one can estimate the *state/trajectory* of the DE model, that is, the function described by the DE. One can also estimate the DE or its parameters, a task commonly known as *system identification*. Both problems are ubiquitous throughout natural science, statistics and engineering.

4.1. Linear state-space models

Linear state-space models are the most common models in the class of ODEs, or dynamical systems [27]. They are classically given as

$$\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{x}_{i-1} + \boldsymbol{B}\boldsymbol{u}_i + \boldsymbol{\epsilon}^{(P)}_i, \quad i = 1, \ldots, N-1 \tag{7}$$

$$\boldsymbol{y}_i = \boldsymbol{C}\boldsymbol{x}_i + \boldsymbol{D}\boldsymbol{u}_i + \boldsymbol{\epsilon}^{(M)}_i, \quad i = 1, \ldots, N-1. \tag{8}$$

The *model* equation (7) states that the hidden *states* $\boldsymbol{x}_i \in \mathbb{R}^n$ follow a stochastic difference equation with external user-defined control $\boldsymbol{u}_i \in \mathbb{R}^k$ and i.i.d. *process noise* $\boldsymbol{\epsilon}^{(P)}_i$, which is Gaussian-distributed with mean zero and covariance $\boldsymbol{\Sigma}_P$. The *likelihood* of the *measurements* $\boldsymbol{y}_i \in \mathbb{R}^m$ is defined via Eq. (8). The measurements are linear combinations of the state and the control with additive i.i.d. Gaussian *measurement noise* $\boldsymbol{\epsilon}^{(M)}_i$ with mean zero and covariance $\boldsymbol{\Sigma}_M$. The *initial state* $\boldsymbol{x}_0$ is independently Gaussian-distributed with mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$. Note that the assumption that the process noise is Gaussian-distributed is in fact a very natural one if the finite difference equations ought to be discretizations of a continuous stochastic model. In this case, the distribution of a finite difference model

should not depend on the discretization step size. Suppose we split one interval into M smaller steps; then the joint process noise in this interval is $\sum_{i=1}^{M} \varepsilon_i^{(P)}$, where the $\varepsilon_i^{(P)}$ are i.i.d. random variables. If the variance of the $\varepsilon_i^{(P)}$ is finite, then the sum will have a Gaussian distribution for large M, regardless of the distribution of the $\varepsilon_i^{(P)}$. Thus, if the process noise has finite variance, the only valid distribution that can be refined on an ever smaller grid is the Gaussian distribution.

We now interpret the state-space model in terms of the kernel framework.

Theorem 11. *The linear state-space model* (7) *defines a GP over trajectories* $\boldsymbol{x} : \mathscr{X} \to \mathbb{R}^n$, $\mathscr{X} = \{0, .., N-1\}$. *Mean and covariance for* $i,j \in \mathscr{X}$ *are given as*

$$\boldsymbol{\mu}_i = \mathbb{E}(\boldsymbol{x}_i) = \boldsymbol{A}^i \boldsymbol{\mu}_0 + \sum_{l=1}^{i} \boldsymbol{A}^{i-l} \boldsymbol{B} \boldsymbol{u}_l, \tag{9}$$

$$\begin{aligned} \boldsymbol{K}_{i,j} &= \mathbb{E}((\boldsymbol{x}_i - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_j)^{\mathrm{T}}) \\ &= \boldsymbol{A}^i \boldsymbol{\Sigma}_0 \boldsymbol{A}^{j,T} + \sum_{l=1}^{\min(i,j)} \boldsymbol{A}^{i-l} \boldsymbol{\Sigma}_P \boldsymbol{A}^{j-l,\mathrm{T}}. \end{aligned} \tag{10}$$

Proof (*Dynamical systems view*). Since all (conditional) distributions of the $\boldsymbol{x}_i$ are Gaussian, so is the joint distribution of $\boldsymbol{x} : \mathscr{X} \to \mathbb{R}^n$, i.e. it is a GP. Furthermore, it is

$$\boldsymbol{x}_i = \boldsymbol{A}^i \boldsymbol{x}_0 + \sum_{l=1}^{i} \boldsymbol{A}^{i-l} (\boldsymbol{B} \boldsymbol{u}_l + \boldsymbol{\epsilon}_l^{(P)}).$$

Using the independence assumptions, Eqs. (9) and (10) follow. □

Proof (*Kernel view*). Eq. (7) can be written equivalently as

$$\underbrace{\begin{pmatrix} \boldsymbol{\Sigma}_0^{-1/2} & & & \\ & \boldsymbol{\Sigma}_P^{-1/2} & & \\ & & \cdots & \\ & & & \boldsymbol{\Sigma}_P^{-1/2} \end{pmatrix} \begin{pmatrix} \boldsymbol{1} & & & \\ -\boldsymbol{A} & \boldsymbol{1} & & \\ & \cdots & \cdots & \\ & & -\boldsymbol{A} & \boldsymbol{1} \end{pmatrix}}_{=\boldsymbol{R}} \times \underbrace{\begin{pmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \cdots \\ \boldsymbol{x}_{N-1} \end{pmatrix}}_{=\boldsymbol{x}} - \underbrace{\begin{pmatrix} \boldsymbol{\Sigma}_0^{-1/2} \boldsymbol{\mu}_0 \\ \boldsymbol{\Sigma}_P^{-1/2} \boldsymbol{B} \boldsymbol{u}_1 \\ \cdots \\ \boldsymbol{\Sigma}_P^{-1/2} \boldsymbol{B} \boldsymbol{u}_{N-1} \end{pmatrix}}_{=\boldsymbol{u}} = \boldsymbol{\epsilon},$$

where the deviations $\boldsymbol{\epsilon} \in \mathbb{R}^{Nn}$ are i.i.d. Gaussian-distributed with mean zero and covariance one. Since, for any initial state $\boldsymbol{x}_0$ there exists exactly one solution of the system, i.e. one trajectory $\boldsymbol{x}$ that follows Eq. (7), the $\boldsymbol{R}$ thus constructed is one-to-one and defines a valid regularization operator. Using the theory from Section 3, the model is then equivalent to a GP with mean $\boldsymbol{\mu} = \boldsymbol{R}^{-1}\boldsymbol{u}$ and covariance $\boldsymbol{K} = (\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R})^{-1}$. Formulas (9) and (10) can be verified by checking that $\boldsymbol{R}\boldsymbol{\mu} = \boldsymbol{u}$ and $\boldsymbol{K}(\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}) = (\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R})\boldsymbol{K} = \boldsymbol{1}$. □

The GP equivalent to Eq. (7) has the density

$$p(x) \propto \exp(-\tfrac{1}{2}\|\boldsymbol{R}\boldsymbol{x} - \boldsymbol{u}\|^2). \tag{11}$$

This expression has a nice, simple interpretation: trajectories x that follow the model DE (7) are a priori the most likely functions $\boldsymbol{x} : \mathscr{X} \to \mathbb{R}^n$, and deviations from the equation are penalized quadratically.

So far, we have shown that linear state-space models define GP distributions on trajectories $\boldsymbol{x} : \mathscr{X} \to \mathbb{R}^n$. Whether any GP can be written as a linear state-space model depends on whether the reader considers models with state dimension N—or infinite state dimension in the continuous case—as valid state-space models. An introduction to infinite dimensional systems can be found in Ref. [38]. Imagine an arbitrary GP $p(z) = N(\boldsymbol{\mu}, \boldsymbol{K})$ for $z : \mathscr{X} \to \mathbb{R}$. One could simply set $\boldsymbol{x}_0 = \boldsymbol{z}$, i.e. $\boldsymbol{\mu}_0 = \boldsymbol{\mu}$, $\boldsymbol{\Sigma}_0 = \boldsymbol{K}$, and then propagate with $\boldsymbol{A} = \boldsymbol{1}$, $\boldsymbol{u}_i = 0$, and $\boldsymbol{\Sigma}_P = \boldsymbol{0}$. Alternatively, one could use the decomposition $p(z) = p(z_0)p(z_1|z_2), \ldots, p(z_{N-1}|z_0, \ldots, z_{N-2})$ to formulate a state-space model. Since for arbitrary covariances $\boldsymbol{K}$, we cannot assume special Markov properties, we would need again an N-dimensional state-space to represent the GP. For special $\boldsymbol{K}$, however, this construction may allow one to exploit Markov properties of the GP, and thus a representation with a much lower state dimension.

4.2. Linear DEs and the Fourier transform

Kernel methods are often motivated via regularization in the Fourier domain [8]. At the same time, derivative operators reduce to simple multiplications in the Fourier domain. This leads us to examine more closely the connection between DEs and Fourier space penalization in this section.

Assume $\mathscr{X}$ to be the discretized real line, i.e. $\mathscr{X} = \{i/h | i = 1, \ldots, N\}$, $h > 0$, and let $L(\lambda) = \sum_{i=0}^{n} a_i \lambda^i$ be an n-th order polynomial. Consider the linear ODE

$$L(\boldsymbol{D})\boldsymbol{f} = \sum_{i=0}^{n} a_i \boldsymbol{D}^i \boldsymbol{f} = 0, \tag{12}$$

where $\boldsymbol{D}$ is the first derivative operator and $f : \mathscr{X} \to \mathbb{R}$. For the remainder of the chapter we will assume periodic boundary conditions, allowing the use of the discrete Fourier transform to express the derivative operator. Periodic systems are in general not causal, since random events in the future could propagate forward to influence the past. However, for stable linear systems these effects can be neglected for large enough domains, because the contribution of any state onto future state values decays to zero eventually. The natural formulation of the Fourier transform in terms of complex exponentials requires the use of complex-valued linear algebra. For ease of presentation we have omitted this so far, however, all definitions and theorems can also be formulated with complex numbers, as sketched in Appendix A. We will also assume that $L(\boldsymbol{D})$ is one-to-one. Unfortunately, there are common examples where this is not the case, e.g. for the second derivative used for thin-plate splines. Regularization with non-one-to-one operators requires the use of the cpd kernels as described in Appendix B. For discrete $\mathscr{X}$, a straightforward approximation of the continuous derivative is the approximate derivative operator $\boldsymbol{D}$ given as follows in the case of periodic boundary conditions,

$$D = \frac{1}{h} \begin{pmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \\ 1 & & & -1 \end{pmatrix}. \tag{13}$$

$\boldsymbol{D}$ can be diagonalized in the Fourier basis, $\boldsymbol{D} = \sum_{k=1}^{N} \boldsymbol{u}_k w_k \boldsymbol{u}_k^{\mathrm{T}}$, where $w_k = 1/h(\exp(\mathrm{i}(2\pi/N)k) - 1)$ and $\boldsymbol{\delta}_{x_j}^{\mathrm{T}} \boldsymbol{u}_k = \exp(\mathrm{i}(2\pi/N)jk)$. It is well-known that functions of $\boldsymbol{D}$ can be computed by applying equivalent operations to the eigenvalues of w_k. In particular, the corresponding kernel function then is

$$k(x_l, x_m) = (\overline{L(\boldsymbol{D})}^{\mathrm{T}} L(\boldsymbol{D}))^{-1}_{lm} = \boldsymbol{\delta}_{x_l}^{\mathrm{T}} (\overline{L(\boldsymbol{D})}^{\mathrm{T}} L(\boldsymbol{D}))^{-1} \boldsymbol{\delta}_{x_m} \tag{14}$$

$$= \sum_{k=1}^{N} \boldsymbol{\delta}_{x_l}^{\mathrm{T}} \boldsymbol{u}_k \frac{1}{\overline{L(w_k)} L(w_k)} \boldsymbol{u}_k^{\mathrm{T}} \boldsymbol{\delta}_{x_m} \tag{15}$$

$$= \sum_{k=1}^{N} \frac{1}{|L(w_k)|^2} \exp\left(\mathrm{i}\frac{2\pi}{N} k(l-m)\right). \tag{16}$$

Thus, the kernel $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is the (discrete) Fourier transform of $g(w_k) = 1/|L(w_k)|^2$. Since g is real-valued, the Fourier transform of it is also real and additionally symmetric. The corresponding kernel function then is real-valued and only depends on the distance between x_l and x_m, $d = |l - m|$, that is, it is translation-invariant.

Let us motivate Eq. (12) from a regularization point of view. High derivatives are described by polynomials $L(\lambda)$ of high order, in which case $\|L(\boldsymbol{D})\boldsymbol{f}\|^2 = \sum_k \boldsymbol{f}^{\mathrm{T}}\boldsymbol{u}_k |L(w_k)|^2 \boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{f}$ strongly penalizes high frequencies. The corresponding kernel then contains few high frequency components and is thus relatively smooth.

One can also discuss the reverse derivation from a translation-invariant kernel function on $\mathscr{X}$ to a differential regularization operator. Translation-invariance implies that the covariance operator $\boldsymbol{K}$ is diagonal in the Fourier basis. In order to derive a DE, invert the eigenvalues of $\boldsymbol{K}$, take the square root, and interpolate the result by a polynomial L of at most degree N. Eq. (12) then yields the model that is implicitly used when performing regression with this kernel. A famous example is the Gaussian kernel, $k(x_i, x_j) \propto \exp(-|i-j|^2/2\sigma^2)$. The discrete Fourier transform is difficult to compute analytically in this case, so we approximate it with its continuous counterpart for large N and small step sizes. The continuous Fourier transform of a Gaussian is again a Gaussian with variance σ^{-2}. Inverting and taking the square root, we derive a function $\exp((\sigma^2/4)w^2)$, whose Taylor expansion is $L(w) = \sum_{n=0}^{\infty} (\sigma^{2n}/2^{2n}n!)w^{2n}$. Replacing w by the derivative ∂_x, we re-derive the result of Ref. [20]. They state that the Gaussian kernel is equivalent to regularization with derivatives of all (even) orders,

$$\boldsymbol{R} = \sum_{n=0}^{\infty} \frac{\sigma^{2n}}{2^{2n}n!}\boldsymbol{D}^{2n}. \tag{17}$$

A larger σ leads to a stronger penalization of high derivatives, i.e. to smoother functions.

The introduction of the Fourier transform above also leads to a discrete version of Bochner's theorem [39]. While the original theorem in continuous domains deals with positive semi-definite functions, we can make a stronger statement involving positive definiteness for finite domains: A translation-invariant function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$, $k(x_i, x_j) = \phi(i-j)$, is a positive definite kernel function if and only if the (discrete) Fourier transform of ϕ is positive. Since the Fourier transform of ϕ is identical with the eigenvalues of $\boldsymbol{K}$, and we do not have to be concerned with the existence and regularity of Fourier transforms in finite domains, the result, in our case, is trivial.

4.3. Linear stochastic PDEs

A general form of discrete stochastic linear PDEs for $f : \mathscr{X} \to \mathbb{R}$ is

$$f(x_i) = \sum_{x_j \in \mathscr{N}_i} a_{ij} f(x_j) + \varepsilon_i^{(P)}, \quad x_i \in \mathscr{X}, \tag{18}$$

where $\mathscr{N}_i \subset \mathscr{X}$ is the set of neighbours of x_i, $a_{ij} \in \mathbb{R}$, and $\boldsymbol{\epsilon}^{(P)}$ is i.i.d. zero mean Gaussian noise with covariance $\boldsymbol{K}_i$. Since Eq. (18) is a linear equation system in $\boldsymbol{f}$, it is a valid kernel model equation (6). If the x_i are placed on a regular grid and periodic boundary conditions are assumed, the Fourier transform methods from the previous section can also be applied for this multivariate setting.

Note that apart from being a discretized stochastic PDE, Eq. (18) is also one form of writing Gaussian Markov random fields. Additionally, graph-based learning involving the graph Laplacian can be written in this form. This noteworthy fact implies that multiple methods in physics, control theory, image processing, PDE theory, machine learning and statistics all use the same underlying model.

4.4. State estimation and system identification using kernels

Both GP and SVR regression can be interpreted as optimal state estimators if the kernel is chosen with respect to a DE as described above. Both methods try to minimize the deviation of the estimated trajectory from the DE $\boldsymbol{Rf} = 0$ and at the same time try to minimize the distance to the measured data points, where the distance is measured either through a loss function in the SVR case or through a likelihood in the probabilistic setting. An optimal trade-off between these potentially contradicting targets is obtained. Furthermore, SVR and GP regression can both be used for system identification. In SVR one typically chooses the kernel to minimize the cross-validation error on the training set. In GP regression one tries to find the kernel function that maximizes the marginal likelihood, that is, the complete likelihood of the training data and latent function $f : \mathscr{X} \to \mathbb{R}$ marginalized over the latents. Since each DE can be related to a specific kernel function, optimizing for the best kernel in a class of kernels derived from DEs is equivalent to choosing the most appropriate DE model for the given data set. More formally, assume, for example, that we are interested in a DE model of the form

$$L_{\boldsymbol{\theta}}(\boldsymbol{D})\boldsymbol{f} = \sum_{i=0}^{\theta_0} \theta_{i+1}\boldsymbol{D}^i\boldsymbol{f} = 0. \tag{19}$$

Optimizing for the best parameters $\boldsymbol{\theta}$ of the corresponding kernel function $\boldsymbol{K}_{\boldsymbol{\theta}} = (L_{\boldsymbol{\theta}}(\boldsymbol{D})^{\mathrm{T}} L_{\boldsymbol{\theta}}(\boldsymbol{D}))^{-1}$ is equivalent to determining the best differential model of the above form.

The possibility of using kernel machines to estimate the state and the parameters of DEs has been noticed by Ref. [33] in a spline context, and by Ref. [34] who use SVR and cross-validation.

Before discussing the practical implications of this matter, we present some examples highlighting the kernel framework and its connections to DEs.

5. Examples

5.1. The pendulum—state estimation

Consider again the pendulum in Fig. 2. According to Newton's third law, the free motion dynamics of the angle of the pendulum is *approximately* described by the second-order linear DE

$$ml^2\ddot{\phi}(t) + \lambda\dot{\phi}(t) + mgl\phi(t) = 0, \tag{20}$$

where m is the mass of the pendulum, l the length, g the gravitational constant and $\lambda > 0$ a damping factor. Eq. (20) is only approximately correct for two qualitatively different reasons. Firstly, it is only the linearization around the rest position of a truly nonlinear DE. The true gravitational effect is $mgl\sin(\phi(t))$ which for small $\phi(t)$ is similar to $mgl\phi(t)$. Secondly, there may be many, potentially random influences on the pendulum which are not known or cannot in principle be observed. For example, the viscosity of the surrounding air could change slightly due to local temperature changes, or more drastically a by-passer could simply hit the pendulum. Both model mismatch and stochastic influences can be modeled as process noise in a stochastic DE system, rendering this a versatile model.

Fourier space method: The pendulum equation (20) can be written in the operator form

$$L(\partial_x)f(x) = (\partial_x^2 + c_1\partial_x + c_2 I)f(x) = 0, \tag{21}$$

where $I : \mathscr{H} \to \mathscr{H}$ it the identity operator. We discretize an input interval into $N = 4096$ steps and apply the Fourier framework from Section 4.2 to derive a translation-invariant kernel $k(x_i, x_j) = (L(\boldsymbol{D})^{\mathrm{T}} L(\boldsymbol{D}))^{-1}_{ij}$. The resulting kernel and a GP regression with this kernel for the pendulum data in Fig. 2 (right) is shown in Fig. 4.

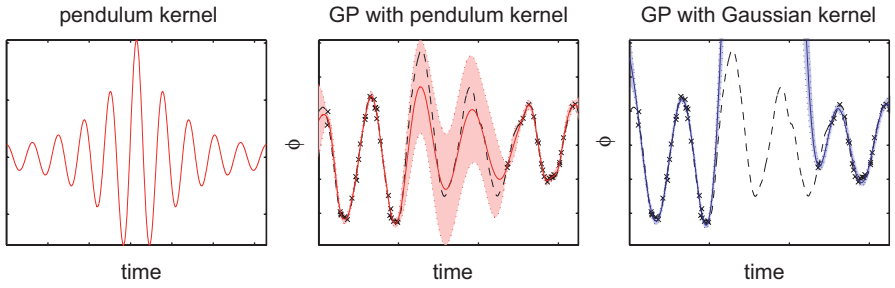

Fig. 4. (left) Kernel function $k(x_i,.)$ derived from the differential equation (20) describing a pendulum. Fourier space transforms with periodic boundary conditions were used. The resulting kernel is translation invariant, x_i is chosen in the middle of the interval. (middle) The 50 data points from Fig. 2, denoted by black crosses, are regressed using a Gaussian process with the pendulum kernel, left, and a Gaussian i.i.d. likelihood. The solid red line denotes the mean of the posterior GP, the shaded area plus–minus one marginal standard deviation of the function values. The dashed black line shows the true sample path from which the data points were generated. (right) GP regression as in the middle figure; however, with a Gaussian kernel.

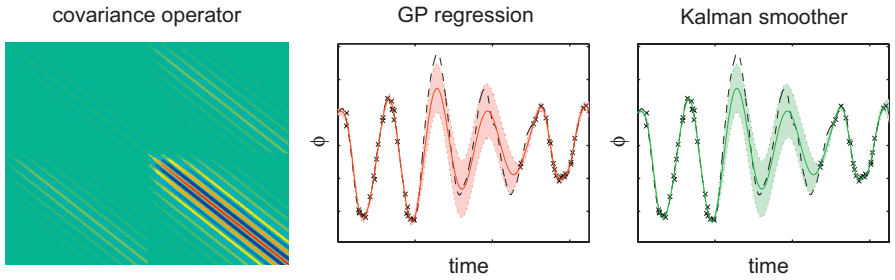

Fig. 5. (left) The covariance matrix derived from the differential equation describing a pendulum (20) using a state-space formulation with initial condition. Since the state-space is two-dimensional the kernel function has for each position pair i, j four entries. Two entries describe the covariance within each component, the two others the cross-covariances. (middle) Gaussian process regression using the kernel from the left figure and the 50 data points from Fig. 2. The solid red line denotes the mean of the posterior GP, the shaded area plus–minus one marginal standard deviation for the function values. The dashed black line is the original sample path. (right) Equivalent results produced by a Kalman smoother.

Observe that the GP regression with the kernel adapted to the pendulum is able to nicely follow the true sample path (middle). While a GP regression with a standard Gaussian kernel yields comparable results in regions where many data points are observed, it performs much worse in the middle where no observations are recorded. This can be explained as follows. Since the a priori model of $\boldsymbol{f}$ in terms of a stochastic DE, $\boldsymbol{Rf} = \boldsymbol{\epsilon} \sim N(0, \sigma^2 \boldsymbol{1})$, allows violations of the exact DE $\boldsymbol{Rf} = 0$, multiple observations can override the model. However, in regions with no observations the prior is more important. Since the Gaussian kernel encodes for the wrong prior model (17) its predictions are especially bad in these regions.

State-space view: The pendulum equation (20) can equally be written as a state-space model with a two-dimensional state, $n = 2$. Then it is

$$\boldsymbol{A} = h\begin{pmatrix} 0 & 1 \\ -\lambda/ml^2 & -g/l \end{pmatrix} + \boldsymbol{1}, \quad \boldsymbol{C} = (1\ 0), \quad \boldsymbol{B} = \boldsymbol{D} = 0,$$

$$\boldsymbol{K} = \begin{pmatrix} 0 & \\ & \sigma^{(P),2} \end{pmatrix}, \quad \boldsymbol{H} = \sigma^{(M),2},$$

where we used $N = 4096$, $h = 0.003$, $\mu_0 = (0.2, 0.1)^{\mathrm{T}}$, $\Sigma_0 = 10^{-5}\boldsymbol{1}$, $\lambda/ml^2 = 25$, $g/l = 1$, $\sigma^{(P)} = 0.085$, and $\sigma^{(M)} = 0.02$. The data samples for the pendulum—see Fig. 2 (right)—were drawn from this model.

The covariance operator for this state-space model computed by Eq. (10) is colour-coded in Fig. 5 (left). Observe the oscillations when fixing a row or column which corresponds to fixing a kernel centre x_i and observing the kernel function $\boldsymbol{K}_{x_i}$. Fig. 5 (middle) shows the marginal posterior mean and variances when performing GP regression using the kernel from the left figure and the data from Fig. 2 (right). Note that the results are up to numerical errors identical to the solution of a Kalman smoother [40], as shown in Fig. 4 (right). This fact is discussed in more detail in Section 6.

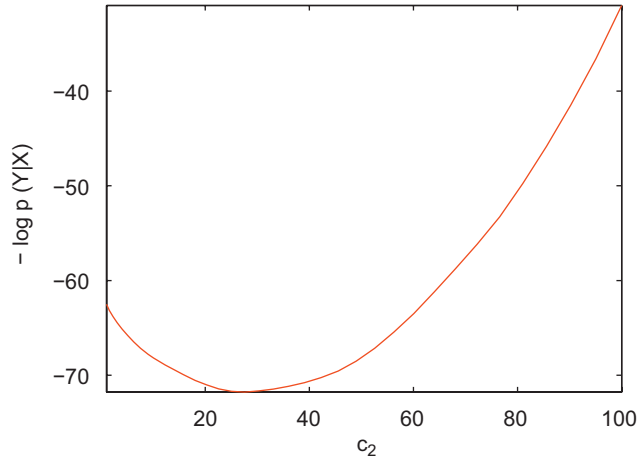

Fig. 6. The negative log marginal likelihood of a Gaussian process regression for the pendulum data set in Fig. 2. Different parameters c_2 are used for the pendulum-adapted kernel in Fig. 4. The minimum of the negative log marginal likelihood is obtained for $c_{2,min} = 27.5$, the true value is $c_{2,true} = 25$.

5.2. The pendulum—parameter estimation

In Fig. 6 we show results from a simple system identification task, i.e. determining the parameter c_2 of the pendulum model (21). We use the pendulum kernel in Fig. 4 and maximize the marginal likelihood of a GP regression model for the optimal value of c_2, where c_1 is assumed to be known. The maximum is attained for a value c_2 close to the true model. We also computed the marginal

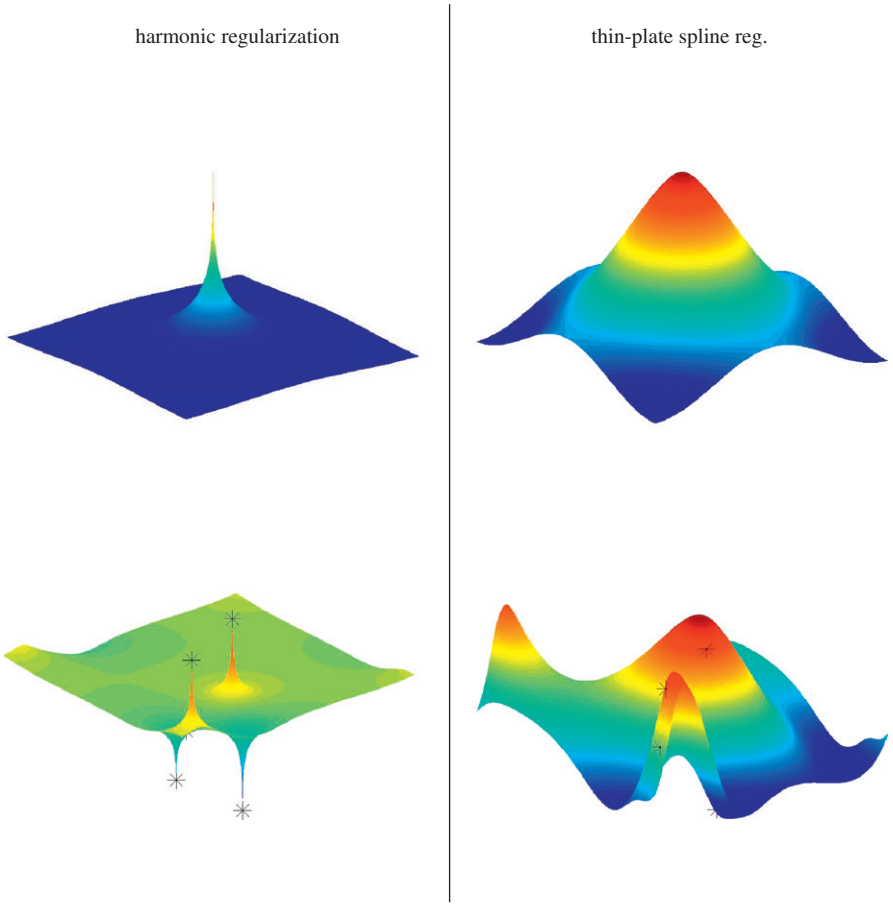

Fig. 7. For a two-dimensional domain $\mathscr{X}$ with periodic boundary conditions, the kernel functions $\boldsymbol{R}_{x_i}$ for harmonic and thin-plate spline regularization are shown in the top row. x_i is chosen in the middle of $\mathscr{X}$. Below we show the mean of a GP regression with these kernels and five data points, denoted as black stars.

likelihood for GP regression with a Gaussian kernel. The maximal marginal likelihood for a Gaussian kernel with automatically chosen parameters is 20 orders of magnitude smaller than for the pendulum kernel. In a Bayesian interpretation the data thus strongly prefers a pendulum-adapted model over the standard Gaussian kernel model.

5.3. Two-dimensional PDEs

In this section we discuss kernels for two-dimensional domains. We show how the harmonic and the thin-plate spline regularizer that both build on derivatives and can be interpreted as stochastic PDEs can be incorporated into the kernel framework.

Next, we show examples of harmonic and thin-plate spline regularization in the kernel framework.

As mentioned in Section 4.3, the Fourier transform can also be applied for functions on higher-dimensional domains, and derivative operators can also be translated into multiplications in this setting. Consider a rectangular grid with $N^2 = 256^2$ points and periodic boundary conditions. The discrete derivative $\boldsymbol{D}^1$ in the first direction and the derivative $\boldsymbol{D}^2$ in the second direction are both diagonal in the tensor Fourier basis $\boldsymbol{u}_{k^1} \otimes \boldsymbol{u}_{k^2}$, where $(\boldsymbol{\delta}_{x_l} \otimes \boldsymbol{\delta}_{x_m})^{\mathrm{T}} \boldsymbol{u}_{k^1} \otimes \boldsymbol{u}_{k^2} = \exp i((2\pi/N)(lk^1 + mk^2))$ and the eigenvalues are $w_{k^1 \otimes k^2} = w_{k^1} w_{k^2}$, $k^1, k^2 = 1, \ldots, N$.

Harmonic regularization results from penalizing the Jacobian of $f\colon \mathscr{X} \to \mathbb{R}$, that is, all first derivatives,

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{D}^1 \\ \boldsymbol{D}^2 \end{pmatrix}.$$

This results in $\|\boldsymbol{R}\boldsymbol{f}\|^2 = \boldsymbol{f}^{\mathrm{T}} \boldsymbol{\Delta} \boldsymbol{f}$, where $\boldsymbol{\Delta} = \boldsymbol{D}^{1\mathrm{T}} \boldsymbol{D}^1 + \boldsymbol{D}^{2\mathrm{T}} \boldsymbol{D}^2$ is the (discrete) Laplace operator. Functions minimizing this expression, the so-called harmonic energy, effectively minimize the graph's area and are thus very common in many fields of research, especially computer graphics [41]. Since constant functions are not penalized by $\boldsymbol{R}$, the cpd framework for non-one-to-one $\boldsymbol{R}$ has to be used in this case, see Appendix B. Postponing a more detailed discussion, the most important change here is to use the pseudoinverse instead of the inverse for deriving the kernel, $\boldsymbol{K} = (\boldsymbol{R}^{\mathrm{T}} \boldsymbol{R})^{+}$. This operation is easily performed using the two-dimensional fast Fourier transform.

The thin-plate splines energy penalize the Hessian of $f : \mathscr{X} \to \mathbb{R}$, that is, all second derivatives,

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{D}^1 \boldsymbol{D}^1 \\ \boldsymbol{D}^1 \boldsymbol{D}^2 \\ \boldsymbol{D}^2 \boldsymbol{D}^1 \\ \boldsymbol{D}^2 \boldsymbol{D}^2 \end{pmatrix}.$$

The energy leaves linear functions unpenalized, thus we again have to use the cpd framework and correspondingly the pseudoinverse.

In Fig. 7, we show the resulting kernels for harmonic and thin-plate spline regularization. Furthermore, we show results of approximating five randomly chosen data points with a GP regression with the respective kernels. Note that the harmonic kernel is sharply peaked, but the regression output stays in the convex hull of the training output values, the famous mean value property of harmonic maps. The thin-plate spline solution is much smoother, but occasionally overshoots the training values.

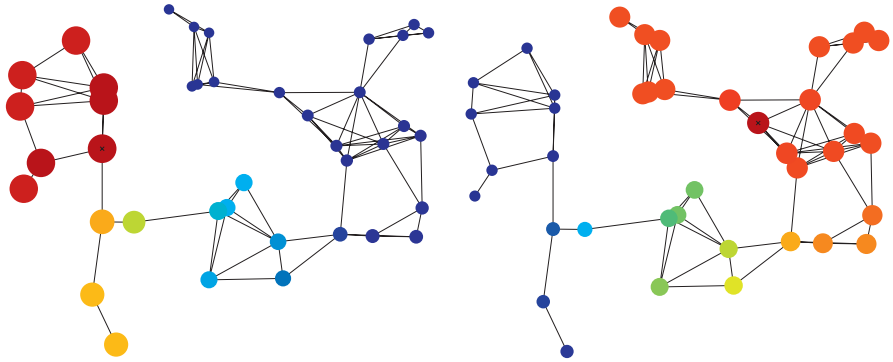

Fig. 8. Kernel corresponding to a graph Laplacian as regularizer $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}$. The kernel functions $\boldsymbol{R}_{x_i}$ are encoded in the colour and the size of the nodes. Vertex x_i is marked with a black cross, the edges of the graph are shown in black.

5.4. Graph Laplacian

Since graph domains are naturally finite, graph-based learning is a good example of where the finite domain kernel framework directly applies without the need for discretization.

The graph Laplacian is an approximation of the true Laplacian Δ on graphs [42]. Kernels on graphs based on the graph Laplacian are described by Ref. [43], they are used for semi-supervised learning by Ref. [44] and Ref. [45] use them in GPs on finite image domains for image super-resolution. The graph Laplacian $\boldsymbol{\Delta}_G$ for a graph $G=(E,\mathscr{X})$ with edges E and vertices $\mathscr{X}$ is given by $\boldsymbol{\Delta}_G = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{W}_{ij}$ is the weight of edge $(i,j) \in E$, 0 if $(i,j) \notin E$, and the degree matrix $\boldsymbol{D}$ is diagonal with entries $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$. We use an ε-neighbourhood graph constructed from 40 random points in $[0,1]^2$, $\varepsilon = 0.2$, i.e. $(i,j) \in E$ if and only if $\|x_i - x_j\| < \varepsilon$. Edge weights $\boldsymbol{W}_{ij}$ are set as $\boldsymbol{W}_{ij} = \exp(-\|x_i - x_j\|^2/\varepsilon^2)$.

As in the above section, setting $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R} = \boldsymbol{\Delta}_G$ leads to the problem that $\boldsymbol{\Delta}_G$ is not one-to-one. Functions $\boldsymbol{f}$ constant on a connected component have $\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\Delta}_G\boldsymbol{f} = 0$, a fact commonly used in spectral clustering [46]. Thus, in order to derive a kernel we again use the pseudoinverse. For more details see Appendix B.

Fig. 8 shows the resulting kernel function $\boldsymbol{K}_{x_i}$. The closer a point is to x_i the larger its corresponding kernel values. Equivalently, under the corresponding GP prior the correlation of the function value at a certain point with the function value at x_i is the stronger the closer the point is to x_i. Note that the distance is measured in terms of the geodesic distance intrinsic to the graph, not the Euclidean distance of the embedding space.

6. Discussion

We have shown that common linear DE models can be flawlessly integrated into the kernel framework and that trajectory/state estimation and system identification can both be performed with kernel machines such as SVR or GP regression. However, there are already many well-established algorithms for state estimation and system identification. In this section, we discuss how kernel methods relate to these standard methods, and when one should prefer which type of algorithm.

State estimation in the linear state-space model described in Section 4.1 is classically dominated by the Kalman filter/smoother [40] and its variants [27]. For such models the Kalman filter algorithm is also equivalent to graphical model message-passing algorithms [47]. Since all these models perform optimal state estimation in the state-space model as do kernel methods such as GP regression or SVR, the results of the two types of methods are identical. The Kalman filter can be interpreted as just an efficient way of computing GP regression exploiting the special features of (low-dimensional) linear state-space models. SVR is slightly different in that it typically uses an ε-insensitive linear loss function [8] which corresponds to a different likelihood model. For a quadratic loss, however, the output of an SVR will be identical to the mean estimate of a Kalman smoother. It is interesting to note that even without considering equivalence of the underlying model assumptions, kernel methods can be related to Kalman filter-like algorithms. For dynamical systems, the matrix $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}$, whose inverse yields the covariance operator, is block-tridiagonal. Ref. [48] propose an algorithm to invert such matrices in linear time using a forward–backward scheme that is closely reminiscent of the Kalman smoother algorithm.

Considering system identification for linear ODEs, there exist many different algorithms in the control community such as subspace identification, Fourier space methods or prediction error methods [27]. Statisticians classically use expectation maximization (EM), which maximizes the marginal likelihood of the model, that is, the likelihood of the observed outputs given the parameters with the hidden states integrated out. The marginal likelihood can be efficiently computed using a Kalman smoother. As for the case of state estimation, all these methods are at least qualitatively equivalent to kernel machine model selection algorithms. The marginal likelihood is also used in GP regression for kernel selection. The cross-validation error can be seen as an approximation of the negative marginal likelihood or the prediction error, which also links SVR regression to this picture.

Since we have argued above that kernel methods are largely equivalent to standard algorithms for treating DEs, we might ask in which context may one benefit from using kernel methods. Kernel methods are to be understood here as algorithms that explicitly compute the kernel function and that perform batch inference by minimizing/integrating an expression of the dimension m, where m is the number of measured data points. Conversely, all classical algorithms work sequentially, performing inference without explicitly computing the kernel function.

For one-dimensional problems, that is, ODEs or dynamical systems, Kalman filter or graphical model-based methods concentrate on the chain-like structure of the model. They give rise to many $O(N)$ algorithms for computing marginal means, marginal variances or the marginal likelihood, where N is the number of discretization steps. If only m measurements, $m \ll N$, are given, this effort can be reduced to $O(m)$ with a little precomputation, summarizing many small steps without observations into one large step. In contrast, kernel-based methods working with the full covariance matrix typically scale around $O(m^3)$ for regression or computing the marginal likelihood. Furthermore, such methods have to compute the kernel function for the given dynamical system. Using the Fourier framework described in Section 4.2, the fast Fourier transform takes $O(N \log N)$ time, and

using the state-space model, the kernel is given explicitly by Eq. (10). One advantage of the kernel view for dynamical systems is that it yields direct access to all pairwise marginal distributions, even for non-neighbouring points, which is not obvious with sequential algorithms.

For multidimensional problems, that is, in PDEs, the kernel method's view on the joint problem is more useful in practical terms, since message-passing is difficult due to many loops and is not guaranteed to yield the optimal solution [47]. However, in this case, too, the kernel cannot be computed analytically but has to be derived either through a fast Fourier transform or, in the worst case, through matrix inversion, which scales like $O(N^3)$. If one aims at estimating the whole latent function $f : \mathcal{X} \to \mathbb{R}$, then direct optimization of problem (1) may be advantageous in comparison with computing the kernels first and then optimizing the kernelized problem. For example, in graph-based learning one typically solves the estimation problem directly in the so-called primal. However, if the graph were given in advance and the labels of the nodes were only uncovered at a later time, it would be advantageous to precompute the kernel functions, since regression to yield all of $f : \mathcal{X} \to \mathbb{R}$ could then be performed in $O(m^3)$ instead of $O(N^3)$.

In sum, one could say that the connection between kernels and DEs will typically not yield faster or better algorithms, except in a few special cases. However, it may help to gain deeper theoretical understanding of both kernel methods and DEs. For example, the connection presented shows that given a state-space model and measurements, the posterior covariances between states at different time points are not dependent on the observations; they are simply given through the covariance matrix $\boldsymbol{K}$. This insight is not obvious from looking at the Kalman update equations. Conversely, the existence of an $O(N)$ inversion algorithm for tridiagonal matrices is not surprising when formulating the inversion in terms of a Kalman filter state estimation problem.

6.1. Nonlinear extensions

This paper has so far solely focused on linear DEs or equivalently on linear regularization operators. However, there is great interest in nonlinear models in many fields, and it is natural to ask whether any of the insights presented above carry over to such a situation.

The disappointing answer is that most of the results are critically dependent on the linearity assumption. If $\boldsymbol{R}$ is not a linear operator, then $\|\boldsymbol{Rf}\|$ does not define a norm. Also, interpreting the kernel as the Green's function of $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}$, that is, the solution of $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{K}_{x_i} = \boldsymbol{\delta}_{x_i}$, does not make sense, since the solution of nonlinear differential problems $\boldsymbol{Rf} = \boldsymbol{u}$ cannot in general be represented in terms of a linear sum of such Green's functions as in the linear case. Also, corresponding probability distributions over functions $f : \mathcal{X} \to \mathbb{R}$ are then, in general, not Gaussian any more, and often cannot be described through an analytic expression at all.

Kernel methods are sometimes used for nonlinear systems, typically in the form that $\boldsymbol{x}_{i+1} = \boldsymbol{f}(\boldsymbol{x}_i)$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ is described by a kernel regression. However, such kernel methods should not be mixed up with the type of kernels we discussed here, since in this paper the kernels were functions of time, not of the preceding state. Furthermore, such one-step-ahead prediction with kernels is not associated with a Gaussian process over trajectories in $\mathcal{H}$, nor does it yield an SVR problem of type (1) over trajectories.

While these are strong negative statements, the dual view of DEs—either in terms of local conditional distributions or more kernel-like as joint distributions over whole functions—may still help to shape intuitions for the nonlinear case and may help to develop new approximate inference algorithms. For example, [49] investigate the joint N-dimensional state distribution of a nonlinear DE, and approximate it using an N-variate GP distribution corresponding to a low order linear DE. Their key calculation is motivated in finite dimensions and is then extended to continuous domains. Conversely, one could also ask whether sequential inference schemes for nonlinear DEs such as the extended Kalman filter, the unscented Kalman filter [50], or sequential Monte Carlo methods [51] can be transferred to other, potentially multivariate, nonlinear kernel-like problems.

7. Conclusion

We have presented a joint framework for kernels, RKHSs, Gaussian processes and regularization operators. All these objects are closely related to each other. Given the theoretical framework, it is natural to see stochastic linear DEs as important examples of regularization operators. We have discussed ordinary as well as partial linear differential equations.

While the exposition is kept simple through the use of the finite domain assumption, note that most results also hold for infinite/continuous domains and we hope the readers will be able to realize this when making comparisons with existing work. An exact treatment for infinite, continuous domains often requires advanced mathematical machinery [21,22,26], and we have thus concentrated on the finite dimensional case, which mostly yields qualitatively similar results.

A good understanding of all the mentioned interrelations between different methods and communities will help the readers to select suitable algorithms for specific problems and may guide their intuition in developing new methods, for example, for dealing with nonlinear DEs. One potential future application may be to explore the meaning of kernel PCA [19] for kernels derived from dynamical systems, which to our knowledge has not yet been studied.

Appendix A. Complex-valued functions and kernels

For finite domains $\mathcal{X}$, complex-valued functions $f : \mathcal{X} \to \mathbb{C}$ are isomorphic to elements in $\mathbb{C}^N = \mathcal{H}$. Some basics of linear algebra in $\mathbb{C}^N$ are as follows: Set $\boldsymbol{f}^* = \overline{\boldsymbol{f}}^{\mathrm{T}}$. The inner product in $\mathbb{C}^N$ is $\boldsymbol{f}^*\boldsymbol{g} = \sum_i \overline{f(x_i)} g(x_i)$ and thus satisfies $\boldsymbol{f}^*\boldsymbol{g} = \overline{\boldsymbol{g}^*\boldsymbol{f}}$. A matrix $\boldsymbol{A}$ is called symmetric or hermitian, if $\boldsymbol{A}^* = \overline{\boldsymbol{A}^{\mathrm{T}}} = \boldsymbol{A}$. Hermitian matrices have real eigenvalues λ_i and an orthogonal basis of eigenfunctions $\{\boldsymbol{u}_i\}_{i=1,\ldots,N}$, thus, $\boldsymbol{f}^*\boldsymbol{A}\boldsymbol{f}$ is real for any $\boldsymbol{f} \in \mathcal{H}$.

Complex-valued algebra does not interfere with the kernel framework. All definitions, theorems and proofs of Section 3 hold if the functions are understood as complex-valued and the appropriate inner product is used. For example the positive definite kernel condition then states that $\sum_{i,j} \overline{\alpha_i} \alpha_j k(x_i, x_j) > 0$, where the sum is real-valued, since $\boldsymbol{K}$ is a hermitian matrix by assumption. We will not be more explicit here, but just state the following theorem, that shows that the complex-valued theory consistently reduces to the real-valued one described in Section 3, if all involved entities are in fact real.

Theorem 12. *With the notation of the SVR objective* (3) *and the Representer theorem* 8 *the following holds*: *if the observation values* $\{y_i | i = 1, \ldots, m\}$ *and the kernel* $\boldsymbol{K}$ *are real-valued and the loss term is a non-decreasing function of* $|f_{\boldsymbol{\alpha}}(x_i) - y_i|$, *then the function* $f_{\boldsymbol{\alpha}} : \mathcal{X} \to \mathbb{C}$ *minimizing* (3) *is real-valued and additionally all coefficients* $\boldsymbol{\alpha}$ *in Theorem* 8 *are real.*

Proof. Assume $f = \boldsymbol{f}^{\Re} + i\boldsymbol{f}^{\Im} \in \mathcal{H}, \boldsymbol{f}^{\Re}, \boldsymbol{f}^{\Im} \in \mathbb{R}^N$. Then

$$\|\boldsymbol{f}\|_K^2 = \|\boldsymbol{f}^{\Re}\|_K^2 + \|\boldsymbol{f}^{\Im}\|_K^2 + 2\underbrace{\Im(\boldsymbol{f}^{\Im\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{f}^{\Re})}_{=0, \text{ as } \boldsymbol{K} \text{ is real}} \tag{A.1}$$

is minimized for $\boldsymbol{f}^{\Im}=0$. Similarly, the loss term is minimized for $\boldsymbol{f}^{\Im}=0$, since the loss of $|f(x_i)-y_i|^2=(\boldsymbol{\delta}_{x_i}^{\mathrm{T}}\boldsymbol{f}^{\Re}-y_i)^2+(\boldsymbol{\delta}_{x_i}^{\mathrm{T}}\boldsymbol{f}^{\Im})^2$ is by assumption larger that the loss of $|f^{\Re}(x_i)-y_i|^2$. Thus the combined minimum is attained for $\boldsymbol{f}^{\Im}=0$. It is $\boldsymbol{f}_X=\boldsymbol{K}_X\boldsymbol{\alpha}$ and $\boldsymbol{K}_X$ is real and positive definite, thus one-to-one. It follows that $\boldsymbol{f}_X\in\mathbb{R}^m$ requires $\boldsymbol{\alpha}\in\mathbb{R}^m$. □

Appendix B. The cpd world

Regularization operators $\boldsymbol{R}^c$ which are not one-to-one motivate the use of the cpd framework. For example, regularizing with the first derivative yields zero penalty for all constant functions, thus $\boldsymbol{R}^c$ cannot be one-to-one in this case.

Most kernel results in Section 3 can be extended to cpd kernels. However, special care has to be taken of the null space of the regularization operator. The description in this section will use the complex-valued setting as introduced in Appendix A above.

B.1. The pseudoinverse

Consider a hermitian matrix $\boldsymbol{A}$ with orthonormal eigendecomposition $\boldsymbol{A}=\sum_i\boldsymbol{u}_i\lambda_i\boldsymbol{u}_i^*$. If $\boldsymbol{A}$ is not one-to-one, i.e. $\exists i:\lambda_i=0$, then we can define the (Moore–Penrose) pseudoinverse of $\boldsymbol{A}$ by

$$\boldsymbol{A}^+=\sum_{i=1,\lambda_i\neq 0}^{N}\boldsymbol{u}_i\frac{1}{\lambda_i}\boldsymbol{u}_i^*.$$

Lemma 13. *For $\boldsymbol{A}$ as above and $\boldsymbol{P}=\sum_{\{i|\lambda_i=0\}}\boldsymbol{u}_i\boldsymbol{u}_i^*$ the orthogonal projection from $\mathscr{H}$ to the null space $\mathscr{N}$ of $\boldsymbol{A}$, we have*

(1) $(\boldsymbol{A}^+)^*=\boldsymbol{A}^+$;
(2) $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A}=\boldsymbol{A}$, $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+=\boldsymbol{A}^+$, *and* $\boldsymbol{A}^+\boldsymbol{A}=\mathbf{1}_{\mathscr{N}^\perp}$;
(3) $[\boldsymbol{P},\boldsymbol{A}]=0$ *where* $[\boldsymbol{A},\boldsymbol{P}]=\boldsymbol{A}\boldsymbol{P}-\boldsymbol{P}\boldsymbol{A}$;
(4) *If* $(\mathbf{1}-\boldsymbol{P})\boldsymbol{A}(\mathbf{1}-\boldsymbol{P})$ *is positive definite on* $\mathscr{N}^\perp$, *then* $(\mathbf{1}-\boldsymbol{P})\boldsymbol{A}^+(\mathbf{1}-\boldsymbol{P})$ *is also positive definite on that subspace.*

B.2. The cpd kernel framework

Fig. B1 depicts the most common objects for the cpd setting in parallel to Fig. 3. The structures and interrelations are very similar to the positive definite case, see Section 3.1, but a non-empty null space of $\boldsymbol{R}^c$ requires a few changes.

Throughout this section we will assume that the *regularization operator* $\boldsymbol{R}^c:\mathscr{H}\to\mathscr{G}$ is an arbitrary operator from $\mathscr{H}$ to some linear space $\mathscr{G}$. We do *not* assume that it is one-to-one. We denote its null space of dimension $0\leqslant M\leqslant N$ as $\mathscr{P}$ and let $\boldsymbol{P}$ be the orthogonal projection from $\mathscr{H}$ to $\mathscr{P}$. If $\boldsymbol{R}^c$ is not one-to-one, neither is $\boldsymbol{R}^{c*}\boldsymbol{R}^c$, and we cannot define the covariance operator as the inverse of this matrix. Instead, we redefine the *covariance operator* $\boldsymbol{K}^c$ to be a symmetric positive semi-definite matrix, i.e.

$$\boldsymbol{f}^*\boldsymbol{K}^c\boldsymbol{f}\geqslant 0,\quad \forall\boldsymbol{f}\in\mathscr{H}. \tag{B.1}$$

The covariance operator is then related to the regularization operator $\boldsymbol{R}^c$ as

$$\boldsymbol{K}^c=(\boldsymbol{R}^{c*}\boldsymbol{R}^c)^+. \tag{B.2}$$

Note that the null space of $\boldsymbol{K}^c$ is also $\mathscr{P}$. The corresponding *Gaussian process* $p_{K^c}(f)$ has the form

$$p_{K^c}(f)=N^U(0,\boldsymbol{K}^c)\propto\exp(-\tfrac{1}{2}\|\boldsymbol{R}^c\boldsymbol{f}\|^2), \tag{B.3}$$

where $N^U(.,.)$ is an unnormalized Gaussian density. If the dimension M of the null space $\mathscr{P}$ is greater than zero, then $p_{K^c}(f)$ cannot be normalized since the density is constant in the directions of $\mathscr{P}$, $\|\boldsymbol{R}^c\boldsymbol{p}\|=0$ for $\boldsymbol{p}\in\mathscr{P}$. However, an unnormalizable prior may nevertheless be useful and lead to a valid posterior, if the likelihood constrains possible functions f enough.

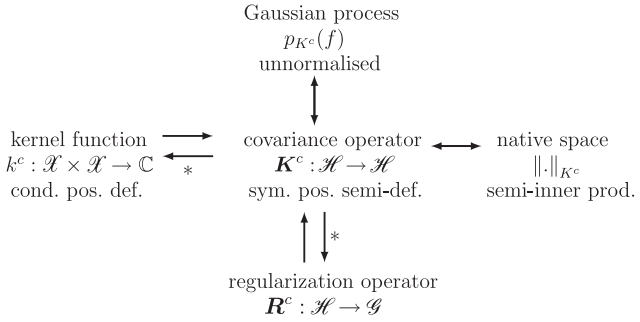

Fig. B1. Common objects in the cpd kernel framework and their interrelations. Arrows denote that one can uniquely be determined from the other (the $*$ denotes that this connection is not unique). A semi-inner product is an inner product which is only positive semi-definite.

We define a semi-inner product $(.,.)_{K^c}$ by

$$(\boldsymbol{f},\boldsymbol{g})_{K^c}=\boldsymbol{f}^{\mathrm{T}}\boldsymbol{R}^{c*}\boldsymbol{R}^c\boldsymbol{g}=\boldsymbol{f}^{\mathrm{T}}\boldsymbol{K}^{c+}\boldsymbol{g}. \tag{B.4}$$

A semi-inner product is an inner product which is only positive semi-definite, the corresponding semi-norm $\|.\|_{K^c}$ is also only positive semi-definite. The tuple $(\mathscr{H},(.,.)_{K^c})$ then is not a Hilbert space, we follow Ref. [26] and call it a *native space*.

$(\mathscr{H},(.,.)_{K^c})$ can be converted into an RKHS in two ways: firstly, by restricting the function space to $(\mathscr{P}^\perp,(.,.)_{K^c})$. The second alternative is to extend the inner product to $(\boldsymbol{f},\boldsymbol{g})_S=(\boldsymbol{f},\boldsymbol{g})_{K^c}+\boldsymbol{f}^*\boldsymbol{P}\boldsymbol{g}$, such that $(\mathscr{H},(.,.)_S)$ is an RKHS.

When discussing *cpd kernel functions* there are some additional subtleties not encountered in the positive definite case.

Definition 14. A symmetric function $k^c:\mathscr{X}\times\mathscr{X}\to\mathbb{C}$ is called cpd with respect to the linear space $\mathscr{P}\subseteq\mathscr{H}$, if for all distinct points $x_1,\ldots,x_m\in\mathscr{X}$, $m\leqslant N$, and all $0\neq\boldsymbol{\alpha}\in\mathbb{C}^m$ with

$$\sum_{j=1}^{m}\alpha_j\overline{p(x_j)}=\sum_{j=1}^{m}\alpha_j\boldsymbol{p}^*\boldsymbol{\delta}_{x_j}=0,\quad\forall\boldsymbol{p}\in\mathscr{P} \tag{B.5}$$

we have that

$$\sum_{i=1}^{m}\sum_{j=1}^{m}\overline{\alpha_i}\alpha_j k^c(x_i,x_j)$$
$$=\boldsymbol{\alpha}^*\tilde{\boldsymbol{K}^c}_X\boldsymbol{\alpha}=\left(\sum_{i=1}^{m}\alpha_i\boldsymbol{\delta}_{x_i}\right)^*\tilde{\boldsymbol{K}^c}\left(\sum_{j=1}^{m}\alpha_j\boldsymbol{\delta}_{x_j}\right)>0, \tag{B.6}$$

where $\tilde{\boldsymbol{K}^c}$ is the operator given as $\tilde{\boldsymbol{K}^c}_{ij}=k^c(x_i,x_j)$.

In other words, if $\boldsymbol{f}=\sum_{i=1}^{m}\alpha_i\boldsymbol{\delta}_{x_i}$, $\boldsymbol{\alpha}\neq 0$, and $\boldsymbol{f}^*\boldsymbol{p}=0\ \forall\boldsymbol{p}\in\mathscr{P}$, then $\boldsymbol{f}^*\tilde{\boldsymbol{K}^c}\boldsymbol{f}>0$. Or equivalent but shorter, $\tilde{\boldsymbol{K}^c}$ is positive definite on $\mathscr{P}^\perp$.

It is important to note, that the operator $\tilde{\boldsymbol{K}^c}$ which is composed from the cpd kernel function values is not necessarily equal to the covariance operator $\boldsymbol{K}_c$, and there exists famous counter examples, e.g. thin-plate spline kernel functions. The definition of a cpd kernel function with respect to $\mathscr{P}$ just implies that $\tilde{\boldsymbol{K}^c}$ be positive definite on $\mathscr{P}^\perp$, it does *not* make any claim about the behaviour on $\mathscr{P}$. For example, thin-plate spline kernels [26] yield matrices $\tilde{\boldsymbol{K}^c}$ which

have $\boldsymbol{f}^*\tilde{\boldsymbol{K}}^c\boldsymbol{f}<0$ for some $\boldsymbol{f}\in\mathscr{P}$. This contradicts the positive semi-definiteness assumption of the covariance operator $\boldsymbol{K}^c$, which was enforced since surely $\|\boldsymbol{f}\|_{K^c}^2=\|\boldsymbol{R}_c\boldsymbol{f}\|^2\geqslant 0$ for all $\boldsymbol{f}\in\mathscr{H}$.

This problem can be circumvented by setting

$$\boldsymbol{K}^c=(\boldsymbol{1}-\boldsymbol{P})\tilde{\boldsymbol{K}}^c(\boldsymbol{1}-\boldsymbol{P}). \tag{B.7}$$

Due to the projection step the assignment of a cpd kernel function to a covariance operator is not unique. If $\{\boldsymbol{p}_i\}_{i=1,\dots,M}$ is an orthonormal basis of $\mathscr{P}$, then Eq. (B.7) implies that

$$\begin{aligned}\boldsymbol{K}_{ij}^c&=\boldsymbol{\delta}_{x_i}^*(\boldsymbol{1}-\boldsymbol{P})\tilde{\boldsymbol{K}}^c(\boldsymbol{1}-\boldsymbol{P})\boldsymbol{\delta}_{x_j}\\&=k^c(x_i,x_j)-\sum_l p_l(x_i)(\boldsymbol{p}_l^*\tilde{\boldsymbol{K}}^c{}_{x_j})\\&\quad-\sum_m(\tilde{\boldsymbol{K}}^c{}^*_{x_i}\boldsymbol{p}_m)p_m(x_j)+\sum_{l,m}p_l(x_i)(\boldsymbol{p}_l^*\boldsymbol{p}_m)p_m(x_i).\end{aligned} \tag{B.8}$$

Note that above we have made an important assumption that does not in general hold for infinite domains and thus requires a slightly different formalism when extended to this setting. We have assumed that an L_2-type inner product exists in $\mathscr{H}$. While we could restrict the space of functions $\mathscr{H}$ to $L_2(\mathscr{X})$ for infinite domains, this is not natural for our purposes. Since we aim at regularizing with $\|\boldsymbol{R}^c\boldsymbol{f}\|$ we only need this expression to be well defined. We do not need that $\boldsymbol{f}$ itself has a finite L_2 norm, it could be an element of a larger space than $L_2(\mathscr{X})$. For example, using $\mathscr{X}=\mathbb{R}$ and regularizing with the first derivative we could include constant functions into $\mathscr{H}$ even though an L_2-type inner product between two linear functions on $\mathbb{R}$ does not exist. While for finite domains it is trivially $\mathscr{H}\subseteq L_2(\mathscr{X})$, Ref. [26] gives an account for more general function spaces $\mathscr{H}$ and infinite domains. Specifically, he uses a slightly different projection for relating the covariance operator with the kernel function in Eqs. (B.7) and (B.8).

The results of this section are summarized in Table B1.

B.3. Support vector machines

Employing regularization operators which are not necessarily one-to-one leads to SVR which is slightly different from the positive definite case. As in Section 3.2, Lemma 7; we first present a useful decomposition of an arbitrary function in $\mathscr{H}$ and then the representer theorem follows.

Definition 15. A set $X=\{x_i|i=1,\dots,m\}\subseteq\mathscr{X}$, $m\leqslant N$, of points is called *unisolvent* with respect to the linear space $\mathscr{P}\subseteq\mathscr{H}$, $\dim(\mathscr{P})\leqslant m$, if the only solution for $p(x_i)=0$ with $\boldsymbol{p}\in\mathscr{P}$, $i=1,\dots,m$ is $\boldsymbol{p}=0$.

Lemma 16. *Given distinct points $X=\{x_i|i=1,\dots,m\}$, $m\leqslant N$, which are unisolvent with respect to $\mathscr{P}$, any $\boldsymbol{f}\in\mathscr{H}$ can be written like*

$$\boldsymbol{f}=\sum_{i=1}^m\alpha_i\boldsymbol{K}_{x_i}^c+\sum_{j=1}^M\beta_j\boldsymbol{p}_j+\boldsymbol{\rho}, \tag{B.9}$$

where $\{\boldsymbol{p}_j\}_{j=1,\dots,M}$ is a basis of $\mathscr{P}$ and $\boldsymbol{\alpha}\in\mathbb{C}^m$, $\boldsymbol{\beta}\in\mathbb{C}^M$, and $\boldsymbol{\rho}\in\mathscr{H}$ are uniquely determined and satisfy the following conditions:

$$\sum_{i=1}^m\alpha_i\overline{p_j(x_i)}=\boldsymbol{p}_j^*\left(\sum_{i=1}^m\alpha_i\boldsymbol{\delta}_{x_i}\right)=0,\quad j=1,\dots,M, \tag{B.10}$$

$$\boldsymbol{\rho}(x_i)=0,\quad i=1,\dots,m. \tag{B.11}$$

Furthermore, $\|\boldsymbol{f}\|_{K^c}^2$ can then be written as $\|\boldsymbol{f}\|_{K^c}^2=\boldsymbol{\alpha}^*\boldsymbol{K}_X^c\boldsymbol{\alpha}+\|\boldsymbol{\rho}\|_{K^c}^2$.

Table B1
Summary of the objects of the conditionally positive definite kernel framework and their interrelations

Entity	Symbol	Relations
cpd Kernel func.	$k^c:\mathscr{X}\times\mathscr{X}\to\mathbb{C}$	$k^c(x_i,x_j)=\tilde{\boldsymbol{K}}^c{}_{ij}$
Covariance op.	$\boldsymbol{K}^c:\mathscr{H}\to\mathscr{H}$	$\boldsymbol{K}^c=(\boldsymbol{1}-\boldsymbol{P})\tilde{\boldsymbol{K}}^c(\boldsymbol{1}-\boldsymbol{P})$ $\boldsymbol{K}^c=(\boldsymbol{R}^{c*}\boldsymbol{R}^c)^+$
Native space	$(.,.)_{K^c}:\mathscr{H}\times\mathscr{H}\to\mathbb{C}$	$(\boldsymbol{f},\boldsymbol{g})_{K^c}=\boldsymbol{f}^*\boldsymbol{K}^{c+}\boldsymbol{g}=\boldsymbol{f}^*\boldsymbol{R}^{c*}\boldsymbol{R}^c\boldsymbol{g}$
	$\|.\|_{K^c}.:\mathscr{H}\to\mathbb{R}$	$\|\boldsymbol{f}\|_{K^c}=(\boldsymbol{f},\boldsymbol{f})_{K^c}^{1/2}=\|\boldsymbol{R}^c\boldsymbol{f}\|$
Gaussian process	$p_{K^c}:\mathscr{H}\to\mathbb{R}$	$p_{K^c}(f)=N^U(0,\boldsymbol{K}^c)$ $p_{K^c}(f)\propto\exp(-\frac{1}{2}\|\boldsymbol{f}\|_{K^c}^2)$ $p_{K^c}(f)\propto\exp(-\frac{1}{2}\|\boldsymbol{R}^c\boldsymbol{f}\|^2)$
Regularization op.	$\boldsymbol{R}^c:\mathscr{H}\to\mathscr{G}$	$(\boldsymbol{R}^c=\sqrt{\boldsymbol{K}^{c+}}$, not unique)

Note that condition (B.10) ensures that $\sum_{i=1}^m\alpha_i\boldsymbol{\delta}_{x_i}\in\mathscr{P}^\perp$. Furthermore, it is $\sum_{i=1}^m\alpha_i\boldsymbol{K}_{x_i}^c=\boldsymbol{K}^c(\sum_{i=1}^m\alpha_i\boldsymbol{\delta}_{x_i})$, and $\boldsymbol{K}_{x_i}^c$ and $\tilde{\boldsymbol{K}}^c{}_{x_i}$ just differ by an element of $\mathscr{P}$. Thus, one could replace $\boldsymbol{K}_{x_i}^c$ in Eq. (B.9) by $\tilde{\boldsymbol{K}}^c{}_{x_i}$ without changing the expression. Practically that means that we can work directly with the cpd kernel function when performing SVR regression and do not have to use the more complicated expression (B.8) which includes projections.

Proof. The theorem states that $f(x_i)=\sum_{j=1}^m\alpha_j\boldsymbol{K}_{ji}^c+\sum_{j=1}^M\beta_jp(x_i)$, $i=1,\dots,m$, where $\sum_{i=1}^m\alpha_i\overline{p_j(x_i)}=0$, $j=1,\dots,M$. In matrix notation this is

$$\boldsymbol{K}_{\text{ext}}^c\begin{pmatrix}\boldsymbol{\alpha}\\\boldsymbol{\beta}\end{pmatrix}\equiv\begin{pmatrix}\boldsymbol{K}_X^c&\boldsymbol{T}\\\boldsymbol{T}^*&0\end{pmatrix}\begin{pmatrix}\boldsymbol{\alpha}\\\boldsymbol{\beta}\end{pmatrix}=\begin{pmatrix}\boldsymbol{f}_X\\0\end{pmatrix} \tag{B.12}$$

with $\boldsymbol{T}\in\mathbb{C}^{m\times M}$ defined by $\boldsymbol{T}_{ij}=p_j(x_i)$. This system is uniquely solvable for $(\boldsymbol{\alpha},\boldsymbol{\beta})$ because of the following argument due to Ref. [26, p. 117]: Suppose that $(\boldsymbol{\alpha},\boldsymbol{\beta})$ lies in the null space of $\boldsymbol{K}_{\text{ext}}^c$. Then we have

$$\begin{aligned}&\boldsymbol{K}_X^c\boldsymbol{\alpha}+\boldsymbol{T}\boldsymbol{\beta}=0,\\&\boldsymbol{T}^*\boldsymbol{\alpha}=0.\end{aligned}$$

$\boldsymbol{K}_X^c$ is positive definite for all $\boldsymbol{\alpha}$ that satisfy the second equation. Multiplying the first equation by $\boldsymbol{\alpha}^*$ yields $0=\boldsymbol{\alpha}^*\boldsymbol{K}_X^c\boldsymbol{\alpha}+(\boldsymbol{T}^*\boldsymbol{\alpha})^*\boldsymbol{\beta}=\boldsymbol{\alpha}^*\boldsymbol{K}_X^c\boldsymbol{\alpha}$. Due to positive definiteness, we can conclude that $\boldsymbol{\alpha}=0$ and thus $\boldsymbol{T}\boldsymbol{\beta}=0$. Since X is a unisolvent set of points, this implies $\boldsymbol{\beta}=0$.

Returning to the inhomogeneous system (B.12) it can be shown [24] using block matrix inversion theorems that

$$\boldsymbol{\alpha}=(\boldsymbol{K}_X^c-\boldsymbol{K}_X^c\boldsymbol{T}(\boldsymbol{T}^*\boldsymbol{K}_X^c\boldsymbol{T})^+\boldsymbol{T}^*\boldsymbol{K}_X^c)\boldsymbol{f}_X, \tag{B.13}$$

$$\boldsymbol{\beta}=(\boldsymbol{T}^*\boldsymbol{K}_X^c\boldsymbol{T})^+\boldsymbol{T}^*\boldsymbol{K}_X^c\boldsymbol{f}_X. \tag{B.14}$$

Finally, set $\boldsymbol{\rho}=\boldsymbol{f}-\sum_{i=1}^m\alpha_i\boldsymbol{K}_{x_i}^c+\sum_{j=1}^M\beta_j\boldsymbol{p}_j$. □

Using this decomposition, the representer theorem for cpd kernels is straight-forward as in the positive definite case.

Theorem 17 (*Representer theorem*). *Given distinct, unisolvent points $X=\{x_i|i=1,\dots,m\}\subseteq\mathscr{X}$, $m\leqslant N$, and labels $\{y_i|i=1,\dots,m\}\subseteq\mathbb{C}$, $C\in\mathbb{R}$, the minimizer of*

$$\|\boldsymbol{f}\|_{K^c}^2+C\ \text{Loss}(\{(x_i,y_i,f(x_i))|i=1,\dots,m\}) \tag{B.15}$$

has the form $\boldsymbol{f}_{\boldsymbol{\alpha},\boldsymbol{\beta}}=\sum_{i=1}^m\alpha_i\boldsymbol{K}_{x_i}^c+\sum_{j=1}^M\beta_j\boldsymbol{p}_j$. $\boldsymbol{\alpha}\in\mathbb{C}^m$, $\boldsymbol{\beta}\in\mathbb{C}^M$ minimize the expression

$$\boldsymbol{\alpha}^*\boldsymbol{K}_X^c\boldsymbol{\alpha}+C\ \text{Loss}(\{(x_i,y_i,f(x_i))|i=1,\dots,m\}) \tag{B.16}$$

subject to the conditions

$$\sum_{i=1}^m\alpha_i\overline{p_j(x_i)}=0,\quad j=1,\dots,M. \tag{B.17}$$

B.4. GP inference

The decomposition in Lemma 16 is also the key to compute the marginals of an unnormalized Gaussian process. As in Section 3.3 we will call this the *GP representer theorem* for the cpd case.

Theorem 18. *For $X \subseteq \mathscr{X}$ unisolvent with respect to $\mathscr{P}$, the marginal distribution $p_{K^c}(\boldsymbol{f}_X) \propto N^U(0, \boldsymbol{M}^+)$ under the joint GP $p_{K^c}(f) \propto N^U(0, \boldsymbol{K}^c)$ is given by*

$$\boldsymbol{M} = \boldsymbol{K}_X^c - \boldsymbol{K}_X^c \boldsymbol{T}(\boldsymbol{T}^* \boldsymbol{K}_X^c \boldsymbol{T})^+ \boldsymbol{T}^* \boldsymbol{K}_X^c, \tag{B.18}$$

where $\{\boldsymbol{p}_j\}_{j=1,\dots,M}$ is a basis of $\mathscr{P}$ and $\boldsymbol{T}_{ij} = p_j(x_i)$.

Proof. By Lemma 16 any $\boldsymbol{f} \in \mathscr{H}$ can be written as $\boldsymbol{f} = \sum_{i=1}^m \alpha_i \boldsymbol{K}_{x_i}^c + \sum_{j=1}^M \beta_j \boldsymbol{p}_j + \boldsymbol{\rho}$ where $\rho(x_i) = 0, i = 1, \dots, m$. Therefore $\boldsymbol{\rho}$ is independent of $\boldsymbol{f}_X$. Furthermore with Eq. (B.13) it is

$$\begin{aligned} \|f\|_{K^c}^2 &= \boldsymbol{\alpha}^* \boldsymbol{K}_X^c \boldsymbol{\alpha} + \|\boldsymbol{\rho}\|_{K^c}^2 \\ &= \boldsymbol{f}_X^* (\boldsymbol{K}_X^c - \boldsymbol{K}_X^c \boldsymbol{T}(\boldsymbol{T}^* \boldsymbol{K}_X^c \boldsymbol{T})^+ \boldsymbol{T}^* \boldsymbol{K}_X^c) \boldsymbol{f}_X + \|\boldsymbol{\rho}\|_{K^c}^2 \\ &= \boldsymbol{f}_X^* \boldsymbol{M} \boldsymbol{f}_X + \|\boldsymbol{\rho}\|_{K^c}^2. \end{aligned}$$

From that it follows that

$$\begin{aligned} p(\boldsymbol{f}_X) &\propto \int \exp\left(-\frac{1}{2}\|\boldsymbol{R}^c \boldsymbol{f}\|^2\right) \mathrm{d}\boldsymbol{f}_{\mathscr{X}\setminus X} \\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{f}_X^* \boldsymbol{M}_X^+ \boldsymbol{f}_X\right) \underbrace{\int \exp\left(-\frac{1}{2}\|\boldsymbol{\rho}\|_{K^c}^2\right) \mathrm{d}\boldsymbol{f}_{\mathscr{X}\setminus X}}_{=const} \\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{f}_X^* \boldsymbol{M}_X^+ \boldsymbol{f}_X\right). \qquad \square \end{aligned}$$

B.5. Transitions between the cpd and the positive definite world

Imagine a family of regularization operators $\boldsymbol{R}_\theta : \mathscr{H} \to \mathscr{G}$ continuously parameterized by $\theta \in U$ where $U \subseteq \mathbb{R}$ is an open neighbourhood of 0. Assume that $\boldsymbol{R}_\theta$ is one-to-one for all θ except for $\theta = 0$. Thus, for $\theta = 0$ we have to use the cpd framework, for $\theta \neq 0$ we should use the positive definite scheme. However, the limit of $\boldsymbol{K}_\theta$ for $0 \neq \theta \to 0$ is not equal to $\boldsymbol{K}_{\theta=0}^c$. The limit does not even exist since in the positive definite case the kernel is the inverse of $\boldsymbol{R}^*\boldsymbol{R}$ which diverges for $\theta \to 0$. On the other hand, the SVR objective function

$$V(\theta, \boldsymbol{f}) \equiv \|\boldsymbol{R}_\theta \boldsymbol{f}\|^2 + C \operatorname{Loss}(\{(x_i, y_i, f(x_i)) | i = 1, \dots, m\}) \tag{B.19}$$

depends continuously on θ. Thus one might hope that the minimizer also depends continuously on θ.

The following theorem which is novel to our knowledge shows that this apparent problem of continuity can be resolved. It shows especially that, while the kernel is diverging for $\theta \to 0$, the SVR solution for $\theta \neq 0$ converges for $\theta \to 0$, and that the limiting element is equal to the cpd SVR solution for $\theta = 0$.

Theorem 19. *Let $\boldsymbol{R}_\theta : \mathscr{H} \to \mathscr{G}$ depend continuously differentiable on $\theta \in U$, $U \in \mathbb{R}^{\mathrm{d}}$ an open neighbourhood of 0 and let $\boldsymbol{R}_\theta$ be one-to-one if and only if $\theta \neq 0$. Let $\mathscr{P}$ be the null space of $\boldsymbol{R}_{\theta=0}$. Furthermore, let $X = \{x_i | i = 1, \dots, m\} \subseteq \mathscr{X}$, $m \leqslant N$, be a set of distinct points unisolvent with respect to $\mathscr{P}$ with corresponding observations $\{y_i | i = 1, \dots, m\} \subseteq \mathbb{C}$. The minimizer $\boldsymbol{f}_\theta = \arg\min_{\boldsymbol{f} \in \mathscr{H}} V(\theta, \boldsymbol{f})$ depends continuously on θ, if $\operatorname{Loss}(\{(x_i, y_i, f(x_i)) | i = 1, \dots, m\})$ is strictly convex and twice continuously differentiable with respect to the $f(x_i)$.*

Proof. As a first step note that $V(\theta, \boldsymbol{f})$ is strictly convex in $\boldsymbol{f}$ for all $\theta \in U$. Both $\|\boldsymbol{R}_\theta \boldsymbol{f}\|^2$ and $\operatorname{Loss}(\{(x_i, y_i, f(x_i)) | i = 1, \dots, m\})$ are convex with respect to $\boldsymbol{f}$ for all θ. If $\theta \neq 0$ then $\|\boldsymbol{R}_\theta \boldsymbol{f}\|^2$ is strictly convex and so is the sum ("strictly convex + convex = strictly convex"). If $\theta = 0$ then $\|\boldsymbol{R}_\theta \boldsymbol{f}\|^2$ is constant in the direction of vectors $\boldsymbol{p} \in \mathscr{P}$. However, for these $\boldsymbol{p}$ at least one of the $p(x_i)$, $i = 1, \dots, m$, is not equal to zero since X is unisolvent. Thus, the loss term is strictly convex with respect to ε where $\boldsymbol{f}_\varepsilon = \boldsymbol{f} + \varepsilon \boldsymbol{p}$, and so is the whole objective function.

Since $V(\theta, \boldsymbol{f})$ is strictly convex in $\boldsymbol{f}$ and continuously differentiable, the unique minimum for given θ is determined by

$$F(\theta, \boldsymbol{f}) \equiv \frac{\partial}{\partial \boldsymbol{f}} V(\theta, \boldsymbol{f}) = 0.$$

By assumption $F : U \times \mathbb{C}^N \to \mathbb{C}^N$ is continuously differentiable and $(\partial/\partial \boldsymbol{f}) F(\theta, \boldsymbol{f}) = (\partial^2/\partial \boldsymbol{f}^2) V(\theta, \boldsymbol{f})$ is invertible since the objective is strictly convex. Using the implicit function theorem [52, p. 292] there exists a continuous function $f_\theta : U \to \mathscr{H}$ with $F(\theta, \boldsymbol{f}_\theta) = 0$. $\square$

Given this theorem one could argue that the cpd framework is unnecessary: if the goal is to regularize with a non-one-to-one operator $\boldsymbol{R}$ one could just use a slightly perturbed version of $\boldsymbol{R}$ which actually is one-to-one and for which one could use the positive definite framework. The solution of a SVR would then not differ very much from the unperturbed result. However, if $\boldsymbol{R}^*\boldsymbol{R}$ is nearly singular the corresponding covariance operator $\boldsymbol{K} = (\boldsymbol{R}^*\boldsymbol{R})^{-1}$ will have some large values. Computations with such a kernel will then be numerically unstable, and it is better to use the cpd framework instead.

Appendix C. Additional proofs

In the finite domains, $\mathscr{H}$ with any inner product $(.,.)_S$ is an RKHS, also with the usual L_2 inner product. To see this note that in $\mathbb{R}^N$ all norms are equivalent and $|\delta_{x_i}(f)| = |f(x_i)| \leqslant \|\boldsymbol{f}\|_1 \leqslant C\|\boldsymbol{f}\|_S$.

Proof (*Lemma 5*). (1) Riesz's theorem.

(2) Since the functionals δ_{x_i} are linearly independent, so are their representers $\boldsymbol{S}_{x_i}$. Then for $\boldsymbol{\alpha} \neq 0$ it is $\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j s(x_i, x_j) = \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j (\boldsymbol{S}_{x_i}, \boldsymbol{S}_{x_j})_S = \|\sum_{i=1}^m \alpha_i \boldsymbol{S}_{x_i}\|_S^2 > 0$.

(3) Set $\boldsymbol{T}_{ij} = (\boldsymbol{\delta}_{x_i}, \boldsymbol{\delta}_{x_j})_S$. Then for any $\boldsymbol{f} = \sum_i f(x_i)\boldsymbol{\delta}_{x_i}$, $\boldsymbol{g} = \sum_i g(x_i)\boldsymbol{\delta}_{x_i}$, it is $(\boldsymbol{f}, \boldsymbol{g})_S = \sum_{i,j} f(x_i) g(x_j) (\boldsymbol{\delta}_{x_i}, \boldsymbol{\delta}_{x_j})_S = \sum_{i,j} f(x_i) g(x_j) \boldsymbol{T}_{ij} = \boldsymbol{f}^{\mathrm{T}} \boldsymbol{T} \boldsymbol{g}$.

(4) Using the reproducing property on $\boldsymbol{\delta}_{x_i}$, $\delta_{ij} = (\boldsymbol{S}_{x_i}, \boldsymbol{\delta}_{x_j})_S = \boldsymbol{\delta}_{x_i}^{\mathrm{T}} \boldsymbol{S} \boldsymbol{T} \boldsymbol{S} \boldsymbol{\delta}_{x_j}$ and $\delta_{ij} = (\boldsymbol{\delta}_{x_i}, \boldsymbol{S}_{x_j})_S = \boldsymbol{\delta}_{x_i}^{\mathrm{T}} \boldsymbol{T} \boldsymbol{S} \boldsymbol{\delta}_{x_j}$ for all $x_i, x_j \in \mathscr{X}$ implies the claim.

(5) Since necessarily $\boldsymbol{S} = \boldsymbol{T}^{-1}$ and $\boldsymbol{T}$ uniquely defines the inner product, the last claim follows. $\square$

Proof (*Lemma 7*). $\boldsymbol{f}$ is the sum of a part $\boldsymbol{f}_{\boldsymbol{\alpha}}$ in the span of the $\boldsymbol{K}_{x_i}$, $x_i \in X$, and the $\boldsymbol{K}$-orthogonal complement $\boldsymbol{\rho}$. The orthogonality condition $(\boldsymbol{K}_{x_i}, \boldsymbol{\rho})_K = 0$ implies $\rho(x_i) = 0$. Since $\boldsymbol{K}$ is positive definite, so is the submatrix $\boldsymbol{K}_X$. Therefore the system $\boldsymbol{f}_X = \boldsymbol{K}_X \boldsymbol{\alpha}$ is uniquely solvable for $\boldsymbol{\alpha} \in \mathbb{R}^m$. $\square$

Proof (*Theorem 8*). Following Lemma 7, and $\boldsymbol{f} \in \mathscr{H}$ can be written as $\boldsymbol{f} = \boldsymbol{f}_{\boldsymbol{\alpha}} + \boldsymbol{\rho}$ with $(\boldsymbol{f}_{\boldsymbol{\alpha}}, \boldsymbol{\rho})_K = 0$. The objective can then be written as

$$\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K}_X \boldsymbol{\alpha} + \|\boldsymbol{\rho}\|_K^2 + C \operatorname{Loss}(x_i, y_i, f_{\boldsymbol{\alpha}}(x_i))_{i=1,\dots,m}.$$

The loss term is independent of $\boldsymbol{\rho}$ because $\rho(x_i) = 0, i = 1, \dots, m$, and thus the objective is minimized for $\boldsymbol{\rho} = 0$. Convexity of the loss and the uniqueness of the map between $\boldsymbol{f}_\alpha$ and $\boldsymbol{\alpha}$, Lemma 7, imply that the whole objective here is convex in $\boldsymbol{\alpha}$. Thus, the minimum is unique in this case. $\square$

References

[1] W. Kienzle, G. Bakir, M. Franz, B. Schölkopf, Face detection—efficient and rank deficient, Advances in Neural Information Processing Systems, vol. 17, MIT Press, Cambridge, MA, 2005, pp. 673–680.

[2] L. Zhang, B. Wu, R. Nevatia, Detection and tracking of multiple humans with extensive pose articulation, in: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[3] X. Song, G. Iordanescu, A. Wyrwicz, One-class machine learning for brain activation detection, in: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[4] G. Sanguinetti, M. Milo, M. Rattray, N. Lawrence, Accounting for probe-level noise in principal component analysis of microarray data, Bioinformatics 21 (19) (2005) 3748–3754.

[5] N.D. Lawrence, J. Quiñonero-Candela, Local distance preservation in the GP-LVM through back constraints, in: Proceedings of the International Conference in Machine Learning, 2006, pp. 513–520.

[6] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[7] C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discovery 2 (2) (1998) 121–167.

[8] B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002 URL ⟨http://www.learning-with-kernels.org⟩.

[9] J. Weston, C. Watkins, Support vector machines for multiclass pattern recognition, in: Proceedings of the Seventh European Symposium on Artificial Neural Networks, 1999.

[10] A. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.

[11] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.

[12] O. Chapelle, B. Schölkopf, A. Zien, Semi-supervised Learning, MIT Press, Cambridge, MA, 2006.

[13] B. Schölkopf, A. Smola, K. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.

[14] C. Williams, D. Barber, Bayesian classification with Gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 20 (12) (1998) 1342–1351.

[15] M. Opper, O. Winther, Gaussian processes for classification: mean-field algorithms, Neural Comput. 12 (11) (2000) 2655–2684.

[16] C. Williams, C. Rasmussen, Gaussian processes for regression, Advances in Neural Information Processing Systems, vol. 8, 1996, pp. 514–520.

[17] N. Lawrence, M. Seeger, R. Herbrich, Fast sparse Gaussian process methods: the informative vector machine, Advances in Neural Information Processing Systems, vol. 15, 2003, pp. 609–616.

[18] N. Lawrence, Gaussian process latent variable models for visualisation of high dimensional data, Advances in Neural Information Processing Systems, vol. 16, 2004.

[19] A. Smola, B. Schölkopf, K. Müller, The connection between regularization operators and support vector kernels, Neural Networks 11 (4) (1998) 637–649.

[20] F. Girosi, M. Jones, T. Poggio, Priors stabilizers and basis functions: from regularization to radial, tensor and additive splines, A.I. Memo No. 1430, MIT Press, Cambridge, MA, 1993.

[21] V. Bogachev, Gaussian Measures, AMS, New York, 1998.

[22] B. Oksendal, Stochastic differential equations: an introduction with applications, sixth ed., Springer, Berlin, 2002.

[23] C.E. Rasmussen, C.K. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, MA, 2006.

[24] G. Wahba, Spline models for observational data, SIAM, Philadelphia, PA, 1990.

[25] J.O. Ramsay, B.W. Silverman, Functional Data Analysis, second ed., Springer, Berlin, 2005.

[26] H. Wendland, Scattered Data Approximation, Cambridge University Press, Cambridge, UK, 2005.

[27] L. Ljung, System Identification—Theory for the User, second ed., Prentice-Hall, Upper Saddle River, NJ, 1999.

[28] G. Kimeldorf, G. Wahba, A correspondence between bayesian estimation on stochastic processes and smoothing by splines, Ann. Math. Stat. 41 (2) (1970) 495–502.

[29] W. Madych, S. Nelson, Multivariate interpolation and conditionally positive definite functions. II, Math. Comput. 54 (189) (1990) 211–230.

[30] A. Smola, B. Schölkopf, K.-R. Müller, The connection between regularization operators and support vector kernels, Neural Networks 11 (1998) 637–649.

[31] T. Graepel, Solving noisy linear operator equations by Gaussian processes: application to ordinary and partial differential equations, in: Proceedings of the 20th International Conference on Machine Learning, vol. 20, 2003, pp. 234–241.

[32] M. Hein, O. Bousquet, Kernels, associated structures and generalizations, Technical Report 127, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2004.

[33] N.E. Heckman, J.O. Ramsay, Penalized regression with model-based penalties, Can. J. Stat. 28 (2000) 241–258.

[34] F. Steinke, B. Schölkopf, Machine learning methods for estimating operator equations, in: Proceedings of the 14th IFAC Symposium on System Identification, SYSID06, Elsevier, Amsterdam, 2006, pp. 1–6.

[35] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural network architectures, Neural Comput. 7 (1995) 219–267.

[36] C. Walder, B. Schölkopf, O. Chapelle, Implicit surface modelling with a globally regularised basis of compact support, Comput. Graphics Forum 25 (3) (2006) 635–644.

[37] C. Micchelli, M. Pontil, On learning vector-valued functions, Neural Comput. 17 (1) (2005) 177–204.

[38] R. Curtain, H. Zwart, An Introduction to Infinite Dimensional Linear Systems Theory, Springer, Berlin, 1995.

[39] S. Bochner, Monotone Funktionen, Stieltjessche integrate und harmonische analyse, Math. Ann. 108 (1933) 378–410.

[40] R. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1) (1960) 35–45.

[41] M. Floater, K. Hormann, Surface parameterization: a tutorial and survey, Advances in Multiresolution for Geometric Modelling, vol. 1, Springer, Berlin, 2005.

[42] M. Hein, J.-Y. Audibert, U. von Luxburg, Graph Laplacians and their convergence on random neighborhood graphs, J. Mach. Learn. Res. 8 (2007) 1325–1370.

[43] A. Smola, R. Kondor, Kernels and regularization on graphs, in: Proceedings of the Conference on Learning Theory, Springer, Berlin, 2003.

[44] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, in: Proceedings of the 20th International Conference on Machine Learning, vol. 20, 2003.

[45] M. Tipping, C. Bishop, Bayesian image super-resolution, in: Advances in Neural Information Processing Systems, vol. 15, 2003, pp. 1279–1286.

[46] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.

[47] M. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, An introduction to variational methods for graphical models, Mach. Learning 37 (2) (1999) 183–233.

[48] Y. Huang, W. McColl, Analytical inversion of general tridiagonal matrices, J. Phys. A: Math. General 30 (1997) 7919–7933.

[49] C. Archambeau, D. Cornford, M. Opper, J. Shawe-Taylor, Gaussian process approximations of stochastic differential equations, J. Mach. Learn. Res., in: Workshop and Conference Proceedings, vol. 1, 2007, pp. 1–16.

[50] S. Julier, J. Uhlmann, A new extension of the Kalman filter to nonlinear systems, in: I. Kadar (Ed.), Proceedings of the Conference on Signal Processing, Sensor Fusion, and Target Recognition VI, vol. 3068, 1997, pp. 182–193.

[51] A. Doucet, N. de Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer, Berlin, 2001.

[52] H. Heuser, Lehrbuch der Analysis, Teil 2, B. G. Teubner, Stuttgart, Germany, 1991.

About the Author—FLORIAN STEINKE earned a Diplom in Physics from the Eberhard-Karls-Universität, Tübingen, in 2005. He is currently pursuing a Ph.D. at the Max Planck Institute for Biological Cybernetics in Tübingen under the supervision of Bernhard Schölkopf. His diploma thesis on Modeling Human Heads with Implicit Surfaces won the DAGM-SMI prize for the best diploma thesis on Pattern Recognition in Germany in the academic year 2005–2006.

About the Author—BERNHARD SCHÖLKOPF was born in Stuttgart on 20 February, 1968. He received an M.Sc. in Mathematics and the Lionel Cooper Memorial Prize from the University of London in 1992, followed in 1994 by the Diplom in Physics from the Eberhard-Karls-Universität, Tübingen. Three years later, he obtained a Doctorate in Computer Science from the Technical University Berlin. His thesis on Support Vector Learning won the annual dissertation prize of the German Association for Computer Science (GI). In 1998, he won the prize for the best scientific project at the German National Research Center for Computer Science (GMD). He has researched at AT&T Bell Labs, at GMD FIRST, Berlin, at the Australian National University, Canberra, and at Microsoft Research Cambridge (UK). He has taught at Humboldt University, Technical University Berlin, and Eberhard-Karls-University Tübingen. In July 2001, he was appointed scientific member of the Max Planck Society and director at the MPI for Biological Cybernetics; in October 2002, he was appointed Honorarprofessor for Machine Learning at the Technical University Berlin. In 2006, he received the J.K. Aggarwal Prize of the International Association for Pattern Recognition. He has been the program chair of COLT and NIPS and serves on the editorial boards of JMLR, IEEE PAMI and IJCV.