Black Box Quantiles for Kernel Learning

Anthony Tompkins[*1] **Ransalu Senanayake**[*1] **Philippe Morere**[*1] **Fabio Ramos**[1,2]
[1]School of Computer Science, The University of Sydney, Australia. [2]NVIDIA, USA.

Abstract

Kernel methods have been successfully used in various domains to model nonlinear patterns. However, the structure of the kernels is typically handcrafted for each dataset based on the experience of the data analyst. In this paper, we present a novel technique to learn kernels that best fit the data. We exploit the measure-theoretic view of a shift-invariant kernel given by the Bochner's theorem, and automatically learn the measure in terms of a parameterized quantile function. This flexible black box quantile function, evaluated on Quasi-Monte Carlo samples, builds up quasi-random Fourier feature maps that can approximate arbitrary kernels. The proposed method is not only general enough to be used in any kernel machine, but can also be combined with other kernel design techniques. We learn expressive kernels on a variety of datasets, verifying the methods ability to automatically discover complex patterns without being guided by human expert knowledge.

1 Introduction

Even though we live in an era where data is abundant, it still requires expertise knowledge to infer useful information from data. In order to make machine learning algorithms accessible to data analysts without prolonged experience, it is vital to automate all phases of machine learning algorithms. To this end, there have been several initiatives to automate both inference and parameter optimization in machine learning algorithms [1, 2, 3, 4, 5]. Although these black box [6, 7] and gray box [8] algorithms are becoming popular in various sub-disciplines of machine learning, these concepts are scarcely discussed in kernel methods [9] irrespective of their popularity and widespread applications.

As with connectionists paradigms such as deep neural networks, kernel methods gained popularity due to their capacity to capture nonlinear patterns [9, 10]. Although kernel methods are valuable when data are scarce, they are equally flexible in big data settings [11, 12, 13]. In order to use kernel methods such as support vector machines, Gaussian process regression, etc., it typically requires an experienced user to pick a kernel or a composition of kernels from a known pool of kernels such as periodic, squared-exponential radial basis function (RBF), Matérn, etc. [14]. Therefore, it is necessary to automate the art of choosing kernels so as to perform data analysis in an end-to-end fashion. On another facet, there are a variety of applications in the real-world where pre-defining the kernel is inexpedient because data becomes available only after deploying the algorithm. For instance, in online robotics applications, the robot is required to adapt the kernel to account for the changes in the terrain [15] or to adjust the properties of the control policies according to the feedback [16, 17].

The majority of kernel learning techniques focuses on choosing the kernel from a mixture of popular known kernels and learning their weights [18, 19]. In contrast, in this paper, we attempt to learn kernels in a more general and black box fashion without being restricted to known kernels. We represent a shift-invariant kernel with a parameterized quantile function. This enables us to seamlessly change the structure of the kernel (i.e. the kernel function) by merely adjusting the parameters of the quantile function. This process generates novel shift-invariant kernels that are guaranteed to be positive-definite. For this reason, these kernels can simply be used in any kernelized inference algorithm such as kernelized regression or classification by optimizing the parameters of the black box quantile (BBQ) alongside the parameters of the inference algorithm. Treating the underlying quantile as a black box does not hinder the interpretability of the inference algorithm since the reconstructed kernel itself can still be understood as a similarity function.

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. [*]Equal contribution.

2 Related work and background

2.1 Kernel learning

For a non-empty set $\mathcal{X} \subset \mathbb{R}^D$, let us denote the kernel function as $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$. By adding the inner product structure $\langle \cdot, \cdot \rangle$, a kernel can be represented as $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}$ for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, where $\phi : \mathcal{X} \mapsto \mathcal{H}$ is a mapping from the low dimensional input space $\mathcal{X}$ into a possibly infinite-dimensional Hilbert space $\mathcal{H}$. Intuitively, these inner product kernels quantify the similarity between two input points. The simplest kernel is the linear kernel given by $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^* \mathbf{x}'$. However, arguably, the most popular kernel is the squared-exponential kernel $k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-2\gamma^2 \|\mathbf{x} - \mathbf{x}'\|_2^2)$ with parameters σ^2 and γ^2. If less smooth fittings are intended, then the user would choose kernels such as the neural network or Matérn $\frac{3}{2}$ [14]. Similarly, if there is a seasonal pattern it is desirable to use a periodic kernel $k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-2\gamma^2 \sin^2(\frac{\rho\pi}{2}|\mathbf{x} - \mathbf{x}'|))$ with the periodicity parameter ρ.

Because of the inner product, the kernel function is, i) positive-definite, ii) conjugate symmetric $k(\mathbf{x}, \mathbf{x}') = k^*(\mathbf{x}', \mathbf{x})$, and iii) satisfies linearity in the first argument of the kernel. Due to these properties, functional composition, and convolution of kernels result in another valid kernel. Even though kernels constructed with these operations are capable of computing complex patterns, the choice of kernels and their combinations often depend on the domain knowledge and physical observations of the data analyst. An intuitive guide on how to combine popular kernels is given in [20]. Nonetheless, because the manual construction of a sophisticated kernel is not straightforward, similar to a typical model selection problem in statistics, researchers have attempted to learn kernels as multi-task learning and through expensive optimization techniques [18, 21, 22]. These approaches to finding sophisticated kernels is known as kernel learning. In this paper, we revisit kernel learning with novel techniques developed in recent years to approximate kernels by a dot product of basis functions. In this regard, the spectral domain representation of the kernel defined in Theorem 1 and Corollary 1 has gained popularity [19, 23].

Theorem 1 *(Bochner's Theorem) [24] The sufficient and necessary conditions for the existence of a continuous positive-definite function $\hat{\mu} : \mathbb{R}^D \to \mathbb{C}$ for all $\mathbf{x} \in \mathbb{R}^D$ is,*

$$\hat{\mu}(\mathbf{x}) = \int_{\mathbb{R}^D} e^{-i\mathbf{x}^\top \boldsymbol{\omega}} \mathrm{d}\mu(\boldsymbol{\omega}), \tag{1}$$

where μ is a finite non-negative Borel measure on $\mathbb{R}^D$.

Corollary 1 *If the measure μ in Theorem 1 is a probability measure with $\hat{\mu}(0) = 1$ and has a probability density function (pdf) f_Ω on the random variable Ω with its realization $\boldsymbol{\omega} \in \mathbb{R}^D$, then $\hat{\mu}(\mathbf{x} - \mathbf{x}') =: k(\mathbf{x}, \mathbf{x}')$ is a continuous, stationary, and positive-definite covariance function that satisfies,*

$$k(\mathbf{x}, \mathbf{x}') = \int_{\mathbb{R}^D} e^{-i(\mathbf{x}-\mathbf{x}')^\top \boldsymbol{\omega}} f_\Omega(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}. \tag{2}$$

In a series of pioneering work by Wilson et al. [19, 25], the covariance function of a Gaussian process prior is modeled by making use of a result similar to Corollary 1 as a spectral representation [14]. Taking advantage of mixture representations and sampling from a pdf has been explored in [26, 27]. We leave this discussion for Section 5. In the next section, we discuss the generic sampling based approximation for kernels with a fixed structure (i.e. kernels are not learned).

2.2 Random Fourier features

Although dot product kernels can capture non-linear patterns as their features are represented in an infinite dimensional space, naively computing these kernels for large datasets is computationally prohibitive as it requires to evaluate the kernel for each pair of points. To alleviate this issue, Rahimi et al. [23] proposed to pick a known probability density function $f_\Omega(\omega)$ in such a way that the required kernel can be reconstructed using Corollary 1. For instance, samples drawn from a standard normal distribution can reconstruct a squared exponential kernel. More generally, representing the known pdf using a finite number of Monte-Carlo (MC) samples $\{\omega_m \overset{\text{iid}}{\sim} f_\Omega(\omega)\}_{m=1}^M$ creates a finite dimensional approximation of the feature map $\hat{\phi}(\mathbf{x}) \in \mathbb{C}^M$ [28, 29, 30]. That is,

$$k(\mathbf{x}, \mathbf{x}') \approx \frac{1}{M} \sum_{m=1}^{M} e^{-i(\mathbf{x}-\mathbf{x}')^\top \omega_m} = \langle \hat{\phi}(\mathbf{x}), \hat{\phi}(\mathbf{x}') \rangle_{\mathbb{C}}. \tag{3}$$

The approximate feature map can be decomposed into, $\hat{\phi}(\mathbf{x}) = \frac{1}{\sqrt{M}}[e^{-i\mathbf{x}^\top \omega_1}, \ldots, e^{-i\mathbf{x}^\top \omega_M}] \in \mathbb{C}^M$. For real valued kernels, (3) can be further reduced into cosine and sine terms [23]. This work has gained attention in recent years because of the simplicity, solid theoretical basis [31, 32], and superiority in various applications [17, 33]. Further advantages and outstanding performance of randomization based algorithms have been discussed in [34, 35, 36, 37].

3 Black box quantile kernels

In this section, we propose black box quantiles (BBQ) as an alternative parameterization of the probability

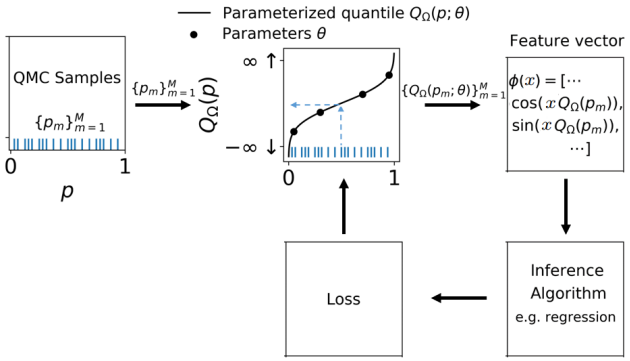


Figure 1: A summary of the algorithm.

Data: $x_{train}, y_{train}, x_{test}, y_{test}$
Result: Optimal kernel parameters θ_*
Optimal inference algorithm parameters w_*
$p = \{p_m\}_{m=1}^M$ low discrepancy sequence
Initialize θ
while *inference loss not converged* **do**
  Construct quantile $Q(\cdot;\theta)$ from θ
  $\hat{\phi}(Q) \leftarrow$ Compute features as in (6)
  Train inference model using $\hat{\phi}(Q)$
  Compute inference loss as in Sec. 3.3
  $\theta \leftarrow$ Next parameter from optimiser
end
return Best θ, corresponding w_*

Algorithm 1: BBQ algorithm

measure to capture complex patterns in data. The method is general and can be used with other kernel composition techniques. Without loss of generality, BBQ features are introduced for univariate data x in Sections 3.1 to 3.3 and a summary of the algorithm is given in Figure 1 and Algorithm 1. The treatment for multidimensional data is discussed in Sections 4 and 5.

3.1 Parameterized quantiles

As discussed in Section 2.2, known pdfs are used in [23] to construct popular kernels using Corollary 1. Unfortunately, to approximate complicated kernels, the method requires the specification of generic pdfs which can be notoriously difficult as pdfs need to satisfy $\int_{-\infty}^{\infty} f_\Omega(\omega)\mathrm{d}\omega = 1$. Observing the definition of Bochner's theorem (Theorem 1) given in a measure-theoretic view, we can alternatively prescribe the probability distribution in terms of a parameterized quantile function.

If the measure in Theorem 1 is defined by the cumulative distribution function (cdf), $p = P(\Omega \leq \omega) =: F_\Omega(\omega) : \mathbb{R} \to [0,1]$, then its quantile function is $\inf\{\omega \in \mathbb{R}; p \leq F_\Omega(\omega)\} =: Q_\Omega : [0,1] \to \mathbb{R}$. Intuitively, with strict monotonicity and continuity assumptions, the quantile can be regarded as the inverse of the cdf. Quantile functions of known distributions are widely used in many application domains in statistics [38, 39]. In Corollary 2, we obtain the kernel by a parameterized quantile function.

Corollary 2 *Following Theorem 1, if $Q_\Omega(p;\theta)$ is a quantile function associated with a random variable Ω and parameterized by θ, the associated continuous, positive-definite, and shift invariant kernel, $k(x,x';\theta)$, is given by,*

$$k(x,x';\theta) = \int_{[0,1]} e^{-i(x-x')Q_\Omega(p;\theta)}\mathrm{d}p. \tag{4}$$

Our objective is to implicitly learn a sophisticated kernel by explicitly learning the parameters θ that define a quantile function. To this end, the properties that need to be satisfied are given in Theorem 2.

Theorem 2 *The function Q_Ω is a quantile function iff,*

1. $\lim_{p\to 0} Q_\Omega(p) = -\infty$ *and* $\lim_{p\to 1} Q_\Omega(p) = \infty$,
2. $Q_\Omega(p)$ *is a nondecreasing function of* p,
3. $\lim_{p\uparrow p_0} Q_\Omega(p) = Q_\Omega(p_0), \forall p_0 \in [0,1]$, *i.e.* $Q_\Omega(p)$ *is left-continuous.*

Proof sketch: Writing Q_Ω using probability functions verifies the necessary conditions. For sufficiency, the existence of a sample space $\mathcal{S}$, a probability function P on $\mathcal{S}$, and a random variable Ω defined on $\mathcal{S}$ such that Q_Ω is a quantile function should be verified. A more detailed axiomatic description of probability and quantiles can be found in [40, 41, 42].

Since building kernels by direct integration in Equation 2 is intractable, we adopt a Monte Carlo approximation similar to random Fourier features discussed in Section 2.2, such that $k(\mathbf{x},\mathbf{x}';\theta) \approx \langle\hat{\phi}(\mathbf{x};\theta), \hat{\phi}(\mathbf{x}';\theta)\rangle_{\mathbb{C}}$. However, in this setting, the parameterized quantile is evaluated on a low-discrepancy sequence of quasi-Monte Carlo (QMC) samples. QMC approximations are known to be superior over MC approximations irrespective of the dimensionality [43, 44, 45] and with known kernels [45]. With $\{p_m\}_{m=1}^M$ QMC samples, the approximated feature map is given by,

$$\hat{\phi}(x;\theta) = \frac{1}{\sqrt{M}}[e^{-ixQ_\Omega(p_1;\theta)}, \ldots, e^{-ixQ_\Omega(p_M;\theta)}] \in \mathbb{C}^M. \tag{5}$$

For real valued kernels, this can be further simplified

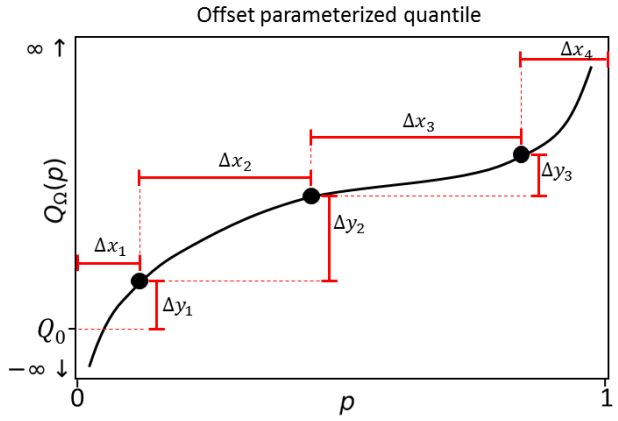


Figure 2: Diagram of the offset parameterized quantile

as,

$$\hat{\phi}(x;\theta) = \sqrt{\frac{2}{M}}\Big[\cos\big(xQ_\Omega(p_1;\theta)\big),\ldots,\cos\big(xQ_\Omega(p_M;\theta)\big), \quad (6)$$
$$\sin\big(xQ_\Omega(p_1;\theta)\big),\ldots,\sin\big(xQ_\Omega(p_M;\theta)\big)\Big],$$

where $\hat{\phi}(\mathbf{x};\theta) \in \mathbb{R}^{2M}$. The objective is to learn $Q_\Omega(p;\theta)$ by adjusting θ.

3.2 BBQ parameterization

In this section we demonstrate how to parameterize an arbitrary quantile such that it can be evaluated for $p \in [0,1]$. We restrict ourselves to fully continuous functions and therefore condition 3 in Theorem 2 is not of our interest. Intuitively, we are interested in seeking a parameterization technique to represent a non-decreasing continuous function with vertical asymptotes at 0 and 1.

In order to satisfy properties listed in Theorem 2, there are numerous ways to parametrize a valid quantile function such as Bernstein polynomials [46]. While such complex quantile formulations can surely be advantageous, the aim of this paper is to lay the foundation to quantile induced kernels and demonstrate their importance through various applications. Therefore, in the following sections we restrict ourselves to coordinates in the space of the quantile function $Q(p) \in \mathbb{R}$. This allows using isotonic regression or constrained interpolation on monotonically increasing points to interpolate the quantile function. Specifically, we use two interpolation methods that enjoy flexible coordinate based parameterization and also ensure the necessary condition of functional monotonocity: *Linear* and *Monotonic Cubic Hermite* interpolation. While the simplest method of linear interpolation guarantees monotonicity on monotonic interpolating coordinates, naive cubic interpolation does not. Strict conditions must be set on the interpolant's tangents [47] in order to guarantee the induced kernel is valid. Specifically, we use the Piecewise Cubic Hermite Interpolation (PCHIP) method [47] however various other methods exist [48, 49].

As shown in Figure 2, we represent our N interpolating points in terms of horizontal and vertical *offsets* from some vertical origin Q_0. Specifically, for N interpolating points,

$$\begin{aligned} \text{learn } & \Delta x_1,\ldots,\Delta x_{N+1} \in [0,1], \\ & \Delta y_1,\ldots,\Delta y_N \in \mathbb{R}, \\ \text{s.t. } & \sum_{i=1}^{N+1} \Delta x_i = 1, \\ & Q_0 \in \mathbb{R}, \end{aligned}$$

where Q_0 is a vertical origin from which all offsets are defined, Δx_i and Δy_i are offsets with respect to Q_0. To enforce the necessary constraint that all Δx_i sum up to 1, Δx_i are simply chosen in $[0,1]$ and normalized.

Even with this simple and explicit parameterization of the quantile function, we demonstrate that we are able to learn complex kernels as well as avoid overfitting as the number of interpolating points increases, by using the negative log-marginal likelihood as a loss function.

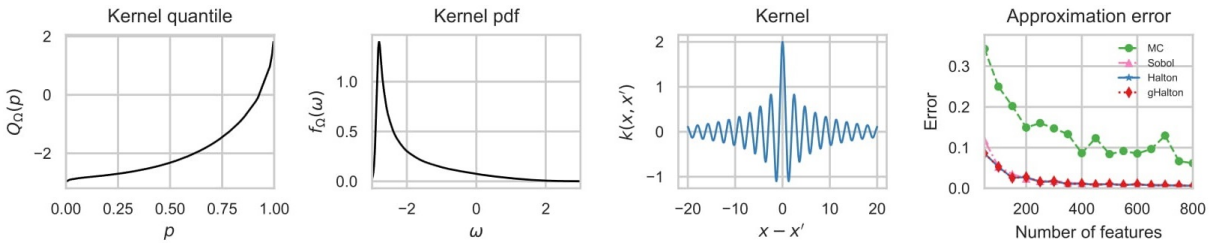
Handling asymptotes. Valid quantile functions feature asymptotes at $p = 0$ and $p = 1$ which interpolation techniques do not guarantee. Asymptotes are constructed by using modified inverse functions $\frac{a_0}{p} + b_0$ and $\frac{a_1}{1-p} + b_1$ around $p = 0$ and $p = 1$ respectively. Parameters a_0, b_0, a_1, and b_1 are chosen to ensure quantile function and derivative continuity at both interpolation end points.

3.3 Learning BBQ parameters

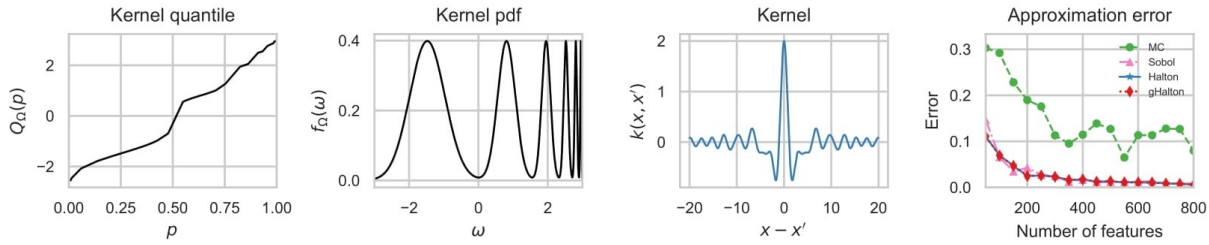
Quantile parameters θ are learned by minimizing the inference algorithm loss $\mathcal{L}_{inference}$ (eg. regression loss). Examples of losses include Mean Square Error (MSE), computed on an validation data, and Negative Log Marginal Likelihood (NLML), computed on training data. Depending on the quantile parameterization used, constraints on θ may apply, and hence, one generally needs to solve,

$$\text{minimize}_\theta \quad \mathcal{L}_{inference}(\theta) \quad (7a)$$
$$\text{subject to} \quad g(\theta) \leq b, \quad (7b)$$

where g and b reflect constraints from the chosen quantile parameterization. Extended literature on optimization provides a wide range of local and global derivative-free optimizers to efficiently solve this problem, such as ADAM, Bayesian optimization [50], DIRECT [51] or COBYLA [52].



(a) Skewed



(b) Combed

Figure 3: Experiment 4.1: Reconstructing corresponding kernels from two different quantile functions. Though not used in the algorithm, as visualized in the second column, kernels generated by BBQ can have corresponding pdfs that are arbitrarily complex and almost impossible to explicitly learn. BBQ parameterization allows implicitly modeling such complex distributions. The last column shows normalized Frobenius errors for different sampling methods.

4 Experiments

The BBQ algorithm with parameterizations discussed in Section 3 and supplementary materials was implemented in python. We conducted a series of experiments on a computer with 16 GB RAM to validate the proposed method. All datasets were normalized between -1 and 1. Bayesian Linear Regression (BLR) [53] was used for inference, similar to [26]. Further details are provided in the supplementary materials. The code is available at `github.com/MushroomHunting/black-box-quantile-kernels`.

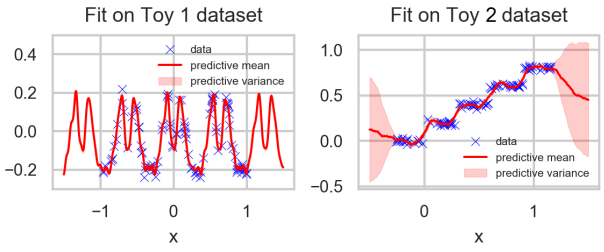


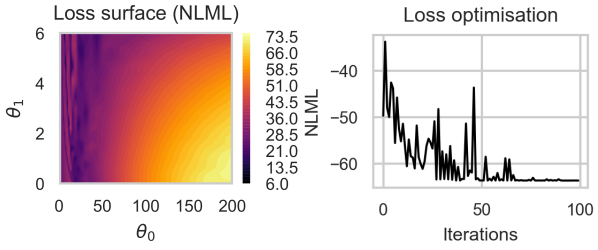
Figure 4: Experiment 4.2: Fits on toy datasets (periodic and steps) using BBQ features.

4.1 Quality of kernel approximation

In this section, we verify that the parameterized quantile functions can easily construct expressive black box kernels with popular kernels. Furthermore, by utilizing QMC sequences, we also demonstrate that the true kernel can be approximated significantly faster than standard MC samples which are used in pdf-based methods such as [23, 27]. To show this, as shown in Figure 3 and supplementary materials, complex probability distributions whose kernels are known in closed from were chosen. In order to show the importance of QMC sampling on BBQ kernel learning, kernels were approximated using three low-discrepancy sampling methods—Sobol, Halton, and generalized Halton sequences—in addition to Monte-Carlo sampling.

In order to assess the quality of approximation, we calculated the normalized Frobenius norm error $\|K_t - \hat{K}\|_F / \|K_t\|_F$ where K_t and $\hat{K}$ are the true and approximated kernel Gram matrices, respectively. For the true and approximate kernel Gram matrices 4000 points were sampled on the interval $[-10, 10]$ and vary the number of features over the range $[50, 1000]$.

While low-discrepancy sequences have been considered for standard kernels [45], their efficacy has not been demonstrated for the more general family of kernels induced by arbitrary quantile functions. For the various complicated quantiles in Figure 3, and their corresponding pdfs and kernels, irrespective of the number of features, the error is always smaller with QMC.



(a) (b)

Figure 5: Experiment 4.2: An example of loss (NLML) surface for 2-parameter quantile and loss per optimization iteration.

4.2 Various aspects of learning BBQ kernels

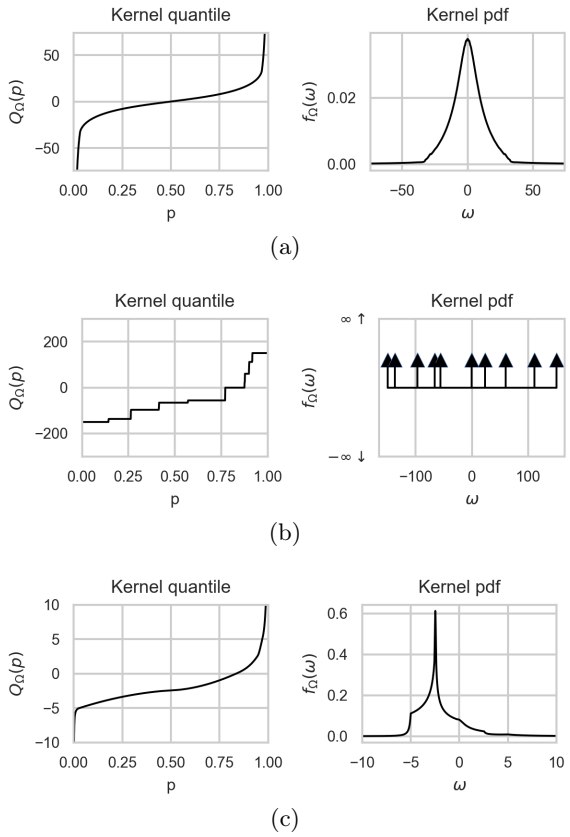
Using two toy datasets (Figure 4) we illustrate that Bayesian linear regression with BBQ features can model nonlinear patterns. Interestingly, rather than handcrafting and composing periodic and RBF kernels, the BBQ parameterization allowed automatically learning the appropriate kernel, making both interpolation and extrapolation possible. Figure 5a shows the error surface. The objective is to determine quantile parameters θ that minimize the loss. An instance of gradient descent to find an optima is shown in Figure 5b.

In order to demonstrate the BBQ algorithm's capacity to learn highly complex kernels, two toy datasets (Figure 4) were used to learn the quantile functions. Even though pdfs and quantiles are merely two different representions of the probability, the proposed quantile parameterization leads to generating flexible and arbitrarily complex kernels. Highlighting this property, as shown in Figure 6, the proposed method was able to learn complex quantiles that would otherwise have been challenging, if not impossible, to pragmatically learn even with a large finite mixture of pdfs or a composition of known kernels.

4.3 Learning complex patterns and extrapolation

We experiment on various real-world datasets from the UCI machine learning repository[1]. *CO2* and *passenger* are periodic datasets evaluating extrapolation capabilities. Datasets *concrete* and *noise* feature higher dimensions of 5 and 8 respectively. We further tests on three in-filling texture datasets from [54] *pores*, *rubber tread*. For multidimensional datasets, one black-box quantile per dimension was learned. For extrapolation datasets, *passenger* and *CO2* we compose BBQ features with a linear kernel $k_{lin} + k_{BBQ}$ and $k_{lin} + k_{lin} \times k_{BBQ}$, respectively, as discussed in Section 5.

[1] https://archive.ics.uci.edu/ml/index.php

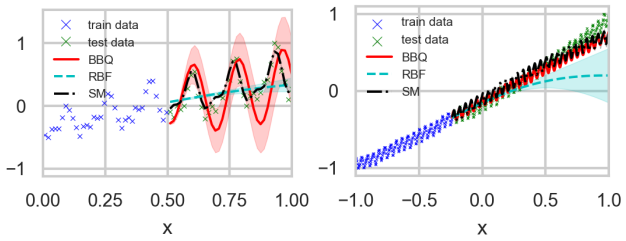


(a) (b) (c)

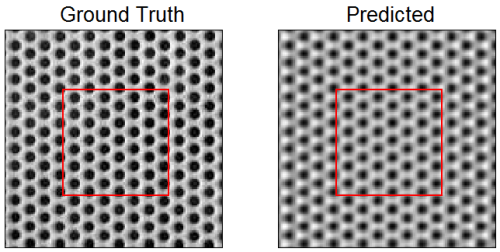
Figure 6: Experiment 4.2: Learned kernel quantile functions and their corresponding kernel pdfs. Slight variation in quantile function result varied PDFs: unimodal, diracs, multimodal or skewed.

Conventionally, different Gaussian process methods such as [55, 56] with fixed kernels are used to capture nonlinear patterns. In addition to these, we compare Bayesian linear regression augmented with BBQ features against the standard Random Fourier Features (RFFs) for the RBF kernel [23] as well as Spectral Mixture (SM) kernels [19]. Since SM kernels are somewhat sensitive to initialization, we run SM kernels 10 times and report the best result. In order to compare textures that are in a regular grid, we used 3000 inducing points with bicubic interpolation, in a sparse approximation method akin to, though not exactly the same, KISS-GP [57]. We also compare with Doubly Stochastic Deep GPs [58] which can learn complex patterns in data because of the deep structure, though a complex kernel is not explicitly learned.

Methods are evaluated in terms of RMSE and Mean Negative Log Loss (MNLL). The smaller these metrics are the more accurate the model is. Unlike RMSE, MNLL takes into account both the mean and variance of predictions [53]. Occasionally full GPs with SM kernels perform better which could be explained by



(a) Airline dataset (b) CO2 dataset



(c) Pores dataset

Figure 7: Experiment 4.3: (a)-(b) Extrapolation on two real datasets. On both datasets, BLR with BBQ, SM both discover periodicity while RBF only finds the general trend. This shows BBQ features can easily be composed with non-stationary kernels (here linear) to learn global trends. (c) Intra-filling task on *pores* texture dataset (train set outside red square, test set inside) and prediction with 300 BBQ features respectively. See supplementary materials for all plots.

better estimates of uncertainty. Results displayed in Table 1 show the superior performance of using BBQ features. In comparison, indicating the importance of learning the kernel, the standard RFFs (RFF-RBF) consistently scores higher errors.

Fits on individual data sets for various methods are displayed in Figure 7, showing both SM and BBQ identify the data periodicity, while RFF-RBF only manages to follow the global trend. All three textures with corresponding qunatiles (one quantile per dimension) are shown in the supplementary materials. Finally, BLR with BBQ features has complexity $\mathcal{O}(M^2N)$ resulting in much faster runtime than SM of complexity $\mathcal{O}(N^3)$. This difference is especially noticeable on moderately sized datasets such as textures, where BLR-BBQ runs in minutes on a desktop computer compared to hours for SM.

4.4 Effect of the number of quantile parameters

We designed an experiment to show the empirical influence of increasing number of qunatile parameters on BBQ regression error. See Figure 8 for results

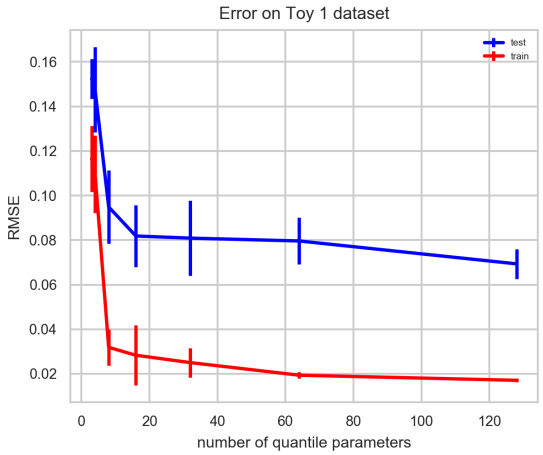


Figure 8: Effect of increasing number of quantile parameters.

on CO2 dataset. The training and testing errors decrease with the increased number of parameters and then levels off. In this case, training by optimizing NLML (using Bayesian linear regression as the model) does not lead to overfitting, even when the model is overparameterized. Nevertheless, note that this is not a straightforward comparison because the errors depend not only on the quantile parameterization, but also on the inference model, loss function, and the optimizer.

5 Discussion

A non-negative definite kernel fully defines a probability distribution and this representation is commonly referred to as a characteristic function in statistics. It is possible to specify the uncertainty of a quantity using the characteristic function, cdf, pdf, or quantile, and with some assumptions the following equalities hold: $\phi(x) = \mathbb{E}[e^{-ix\Omega}] = \int_{\mathbb{R}} e^{-ix\omega} dF_\Omega(\omega) = \int_{\mathbb{R}} e^{-ix\omega} f_\Omega(\omega) \mathrm{d}\omega = \int_0^1 e^{-ixQ_\Omega(p)} \mathrm{d}p$. In this paper, we demonstrated that the quantile representation can be conveniently used to learn complex kernels in a data-driven way.

BBQ features have inherent similarities to SM kernels—the former uses the quantile representation while the later uses the pdf representation. Although SM kernels are attractive, as with any periodic kernel based methods, they are sensitive to initialization of hyperparameters. In a more generic Bayesian nonparametric setting, Oliva et al. [27] further scale this as a mixture of pdfs by exploiting sampling based techniques which require expensive Markov chain Monte Carlo (MCMC) techniques. In BBQs, we attempt to circumvent these issues by a different representation which leads to flexible parametrization and optimization. Parameter optimization in this

Table 1: Experiment 4.3: Loss metrics on all real datasets. We used RMSE and MNLL for spectral mixture, RBF with random Fourier features, the proposed technique (BBQ), Sparse Gaussian Process Regression (SGPR) [56], Sparse Variational Gaussian Process (SVGP) [55], and Doubly Stochastic Deep GP (DSDeepGP) [58].

Loss	Method	CO2	Passenger	Concrete	Noise	Rubber	Pores	Tread
RMSE	BBQ	**0.068**	**0.096**	**0.124**	0.138	**0.248**	**0.256**	**0.114**
	SM	0.083	0.102	0.465	**0.132**	0.395	0.795	0.513
	RFF-RBF	0.245	0.270	0.164	0.184	0.687	1.739	0.326
	SGPR	0.190	0.262	0.138	0.164	0.315	0.586	0.276
	SVGP	0.191	0.262	0.176	0.201	0.3176	0.5853	0.1436
	DSDeepGP	0.446	0.396	0.178	0.174	0.3256	0.5853	0.1508
MNLL	BBQ	**-1.242**	**-0.610**	**-0.577**	-0.173	1.336	**0.337**	**-0.754**
	SM	-0.604	-0.441	0.743	**-0.570**	**0.523**	1.386	1.022
	RFF-RBF	-0.368	14.310	3.173	7.569	18.351	122.689	1.057
	SGPR	-0.695	0.516	-0.545	-0.392	0.268	0.885	0.328
	SVGP	-0.686	0.503	-0.308	-0.181	0.274	0.885	-0.501
	DSDeepGP	1.454	1.361	0.111	0.032	0.306	0.884	-0.339

framework uses NLML and is less prone to overfitting.

It is straightforward to combine BBQ features with other potentially non-stationary kernels to enable additional expressiveness. We support this notion with Claim 1.

Claim 1 *We have the equivalence of kernel composition operations in the kernel space and feature space [59].*

$$\begin{aligned}(k_1 + k_2)(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \\ &= [\phi_1(\mathbf{x})\phi_2(\mathbf{x})][\phi_1(\mathbf{x}')\phi_2(\mathbf{x}')]^\top,\end{aligned} \tag{8}$$

defines the sum of two feature maps, and

$$\begin{aligned}(k_1 \times k_2)(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') \times k_2(\mathbf{x}, \mathbf{x}') \\ &= \sum_i^{n,m} \phi_{1,2}^{(i)}(\mathbf{x})\phi_{1,2}^{(i)}(\mathbf{x}'),\end{aligned} \tag{9}$$

defines the product of two feature maps, where $\phi_{1,2}(\mathbf{x}) = \phi_1 \times \phi_2$ *is the Cartesian product.*

In experiments, to handle multiple dimensions, different quantiles were used on a per-dimension basis analogous to Automatic Relevance Determination (ARD) in kernel based methods [60, 61]. As an alternative method to deal with multi-dimentional data, it is possible to concatenate feature vectors for multiple dimensions. Consider a D-dimensional dataset having N data points $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D) \in \mathbb{R}^{N \times D}$. If f is an inference model such as linear regression, a composite model similar to Generalized Additive Model (GAM) [62] $f(\mathbf{X}) = \sum_{d=1}^{D} f_d(\mathbf{x}_d)$ can be composed. For instance, consider the linear model $f(\mathbf{x}) = \mathbf{w}\hat{\phi}^\top(\mathbf{x}; \theta)$ with BBQ features $\hat{\phi} \in \mathbb{R}^{N \times 2M}$ and corresponding coefficients $\mathbf{w} \in \mathbb{R}^{1 \times 2M}$. Then, the model for multidimensional data is $f(\mathbf{X}) = \sum_{d=1}^{D} \mathbf{w}_d \hat{\phi}^\top(\mathbf{x}_d; \theta_d) = \mathbf{W}^\top \hat{\Phi}(\mathbf{X}; \Theta)$ where $\hat{\Phi}(\mathbf{X}; \Theta) = \|_{d=1}^{D} \hat{\phi}(\mathbf{x}_d; \theta_d) \in \mathbb{R}^{N \times 2MD}$ and $\mathbf{W} = \|_{d=1}^{D} \mathbf{w}_d \in \mathbb{R}^{1 \times 2MD}$ with $\|$ indicating vector concatenation. Algorithmically, it is possible to learn an individual quantile for each dimension and then concatenate corresponding features to create a $2MD$ dimensional feature vector for the inference algorithm.

Although the aforementioned treatments to handle multi-dimentional data are straightforward, note that they cannot capture correlation between covariates. Though the out of scope of this paper, the parameterized quantile representation naturally opens the door to explicitly representing multidimensional variations using copulas that are widely studied in Statistics [63].

6 Conclusion

We proposed a technique to automatically learn highly expressive kernels that fit the data best. To do this, we parameterized a quantile function and learned the parameters using stochastic gradient descent. With the use of Bayesian linear regression, we have shown that inducing a kernel by a quantile function allows one to precisely take advantage of Quasi-Monte Carlo sampling to reduce approximation error.

Inspired by recent ideas in automated machine learning, fundamental connections with harmonic analysis and measure theory, we believe more general and flexible representations of kernels will open doors to compelling new directions in Bayesian inference techniques and kernel learning. As future work, it would be interesting to investigate connections with Copula methods [63], non-stationary kernels [64], and alternative quantile function parameterizations.

References

[1] Z. Ghahramani, "The automatic statistician," 2014.

[2] D. Tran, M. D. Hoffman, R. A. Saurous, E. Brevdo, K. Murphy, and D. M. Blei, "Deep probabilistic programming," in *International Conference on Learning Representations*, 2017.

[3] D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei, "Edward: A library for probabilistic modeling, inference, and criticism," *arXiv preprint arXiv:1610.09787*, 2016.

[4] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems*, 2015.

[5] I. Google, "Cloud AutoML," 2018.

[6] I. Valera and Z. Ghahramani, "Automatic discovery of the statistical types of variables in a dataset," in *International Conference on Machine Learning*, 2017.

[7] T. V. Nguyen and E. V. Bonilla, "Automated variational inference for gaussian process models," in *Advances in Neural Information Processing Systems*, 2014.

[8] P. Galliani, A. Dezfouli, E. Bonilla, and N. Quadrianto, "Gray-box inference for structured gaussian process models," in *Artificial Intelligence and Statistics*, 2017.

[9] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, 1998.

[10] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, 1995.

[12] B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M.-F. F. Balcan, and L. Song, "Scalable kernel methods via doubly stochastic gradients," in *Advances in Neural Information Processing Systems*, 2014.

[13] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," in *Uncertainty in Artificial Intelligence*, Citeseer, 2013.

[14] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*, Springer, 2004.

[15] R. Senanayake and F. Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping," in *Conference on Robot Learning*, 2017.

[16] Z. Marinho, A. Dragan, A. Byravan, B. Boots, S. Srinivasa, and G. Gordon, "Functional gradient motion planning in reproducing kernel hilbert spaces," *arXiv preprint arXiv:1601.03648*, 2016.

[17] A. Rajeswaran, K. Lowrey, E. V. Todorov, and S. M. Kakade, "Towards generalization and simplicity in continuous control," in *Advances in Neural Information Processing Systems*, 2017.

[18] D. Duvenaud, *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014.

[19] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *International Conference on Machine Learning*, 2013.

[20] D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, "Structure discovery in nonparametric regression through compositional kernel search," *arXiv preprint arXiv:1302.4922*, 2013.

[21] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine learning research*, 2004.

[22] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the twenty-first international conference on Machine learning*, ACM, 2004.

[23] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Neural Information Processing Systems (NIPS)*, 2007.

[24] S. Bochner, *Vorlesungen über Fouriersche Integrale: von S. Bochner*. Akad. Verl.-Ges., 1932.

[25] A. G. Wilson, E. Gilboa, A. Nehorai, and J. P. Cunningham, "Gpatt: Fast multidimensional pattern extrapolation with gaussian processes," *arXiv preprint arXiv:1310.5288*, 2013.

[26] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *The Journal of Machine Learning Research*, 2010.

[27] J. B. Oliva, A. Dubey, A. G. Wilson, B. Póczos, J. Schneider, and E. P. Xing, "Bayesian nonparametric kernel-learning," in *Artificial Intelligence and Statistics*, 2016.

[28] B. Sriperumbudur and Z. Szabó, "Optimal rates for random fourier features," in *Advances in Neural Information Processing Systems*, pp. 1144–1152, 2015.

[29] A. Rudi and L. Rosasco, "Generalization properties of learning with random features," in *Advances in*

Neural Information Processing Systems, pp. 3215–3225, 2017.

[30] F. Bach, "On the equivalence between kernel quadrature rules and random feature expansions," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 714–751, 2017.

[31] D. J. Sutherland and J. Schneider, "On the error of random fourier features," 2013.

[32] K. Choromanski, M. Rowland, T. Sarlos, V. Sindhwani, R. Turner, and A. Weller, "The geometry of random features," in *International Conference on Artificial Intelligence and Statistics*, 2018.

[33] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE, 2009.

[34] L. A. Pastur, "Spectra of random self adjoint operators," *Russian mathematical surveys*, 1973.

[35] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomisation in learning," in *Neural Information Processing Systems (NIPS)*, 2008.

[36] D. Lopez-Paz, K. Muandet, and B. Recht, "The randomized causation coefficient," *The Journal of Machine Learning Research*, 2015.

[37] H. Mania, A. Guy, and B. Recht, "Simple random search provides a competitive approach to reinforcement learning," *arXiv preprint arXiv:1803.07055*, 2018.

[38] G. Steinbrecher and W. T. Shaw, "Quantile mechanics," *European journal of applied mathematics*, 2008.

[39] P. Sankaran, N. U. Nair, and N. Midhu, "A new quantile function with applications to reliability analysis," *Communications in Statistics-Simulation and Computation*, 2016.

[40] G. Casella and R. L. Berger, *Statistical inference*. Duxbury Pacific Grove, CA, 2002.

[41] E. Parzen, "Quantile functions, convergence in quantile, and extreme value distribution theory.," tech. rep., Texas A and M univ college station inst of statistics, 1980.

[42] C. P. Chambers, "An axiomatization of quantiles on the domain of distribution functions," *Mathematical Finance*, 2009.

[43] J. Dick and F. Pillichshammer, *Digital nets and sequences: discrepancy theory and quasi–Monte Carlo integration*. Cambridge University Press, 2010.

[44] J. Dick, F. Y. Kuo, and I. H. Sloan, "High-dimensional integration: the quasi-monte carlo way," *Acta Numerica*, 2013.

[45] J. Yang, V. Sindhwani, H. Avron, and M. Mahoney, "Quasi-monte carlo feature maps for shift-invariant kernels," in *International Conference on Machine Learning*, 2014.

[46] S. Han, X. Liao, D. Dunson, and L. Carin, "Variational gaussian copula inference," in *Artificial Intelligence and Statistics*, 2016.

[47] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, 1980.

[48] M. Steffen, "A simple method for monotonic interpolation in one dimension," *Astronomy and Astrophysics*, 1990.

[49] R. W. Stineman, "A consistently well-behaved method of interpolation," *Creative Computing*, 1980.

[50] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.

[51] M. J. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in optimization and numerical analysis*, Springer, 1994.

[52] J. M. Gablonsky and C. T. Kelley, "A locally-biased form of the direct algorithm," *Journal of Global Optimization*, 2001.

[53] C. Bishop, "Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn," *Springer, New York*, 2007.

[54] A. G. Wilson, E. Gilboa, A. Nehorai, and J. P. Cunningham, "Fast kernel learning for multidimensional pattern extrapolation," in *Advances in Neural Information Processing Systems 27*, 2014.

[55] J. Hensman, A. G. d. G. Matthews, and Z. Ghahramani, "Scalable variational gaussian process classification," in *Proceedings of AISTATS*, 2015.

[56] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Artificial Intelligence and Statistics*, 2009.

[57] A. Wilson and H. Nickisch, "Kernel interpolation for scalable structured gaussian processes (kiss-gp)," in *International Conference on Machine Learning*, 2015.

[58] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep gaussian processes," in *Advances in Neural Information Processing Systems*, 2017.

[59] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[60] D. J. MacKay, "Introduction to Gaussian processes," *NATO ASI Series F Computer and Systems Sciences*, 1998.

[61] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012.

[62] T. J. Hastie, "Generalized additive models," in *Statistical models in S*, Routledge, 2017.

[63] R. B. Nelsen, *An introduction to copulas*. Springer Science & Business Media, 2007.

[64] S. Remes, M. Heinonen, and S. Kaski, "Nonstationary spectral kernels," in *Advances in Neural Information Processing Systems*, 2017.