Journal of Machine Learning Research 23 (2022) 1-47 Submitted 1/21; Revised 11/21; Published 2/22

Gauss-Legendre Features for Gaussian Process Regression

Paz Fink Shustin PAZFINK@MAIL.TAU.AC.IL
Department of Applied Mathematics
Tel Aviv University
Tel Aviv, 69978, Israel

Haim Avron HAIMAV@TAUEX.TAU.AC.IL
Department of Applied Mathematics
Tel Aviv University
Tel Aviv, 69978, Israel

Editor: John Shawe-Taylor

Abstract

Gaussian processes provide a powerful probabilistic kernel learning framework, which allows learning high quality nonparametric regression models via methods such as Gaussian process regression. Nevertheless, the learning phase of Gaussian process regression requires massive computations which are not realistic for large datasets. In this paper, we present a Gauss-Legendre quadrature based approach for scaling up Gaussian process regression via a low rank approximation of the kernel matrix. We utilize the structure of the low rank approximation to achieve effective hyper parameter learning, training and prediction. Our method is very much inspired by the well-known random Fourier features approach, which also builds low-rank approximations via numerical integration. However, our method is capable of generating high quality approximation to the kernel using an amount of features which is poly-logarithmic in the number of training points, while similar guarantees will require an amount that is at the very least linear in the number of training points when using random Fourier features. Furthermore, the structure of the low-rank approximation that our method builds is subtly different from the one generated by random Fourier features, and this enables much more efficient hyperparameter learning. The utility of our method for learning with low-dimensional datasets is demonstrated using numerical experiments.

1. Introduction

Gaussian processes (GPs) (Williams and Rasmussen, 2006) provide a powerful probabilistic kernel learning framework, which allows learning high quality nonparametric regression models via methods such as Gaussian process regression (GPR). Indeed, GP based methods are widely used in machine learning and statistics. They have been applied to a wide variety of problems, such as data visualization, Bayesian optimization (Snoek et al., 2012), modeling dynamics and spatial data analysis (Stein, 1999). One of the key advantages of the GP formulation of kernel regression is that the marginal likelihood is a function of the kernel hyperparameters, and that it can be computed via a closed-form formula. By maximizing the marginal likelihood, one can learn the hyperparameters from the data, thereby tuning the method in a principled manner.

However, learning GPs comes with a hefty computational price-tag. Given a training set of n points of dimension d, exact GPR requires solving a (usually dense) linear equation, and thus requires $O(n^3)$ FLOPs. Prediction costs $O(nd)$ FLOPs per test point. Such costs are problematic for datasets with more than a few thousand points. The situation is even more severe if we consider the hyperparameter learning phase: here the cost is $O(n^3)$ FLOPs per hyperparameter in a learning iteration (assuming we use a first-order optimization method). Hyperparameter learning of exact kernel models on large-scale data is even more unrealistic than training such models.

Given the ubiquity of GPs, it is unsurprising that there is a rich literature on scaling GP-based method, e.g. (Quiñonero-Candela and Rasmussen, 2005) and (Wilson and Nickisch, 2015). One attractive approach is to approximate the kernel matrix (also known as covariance matrix) $\mathbf{K}_{\boldsymbol{\theta}}$ as a sum of a diagonal matrix (often a multiple of the identity) and a low rank matrix (Stein, 2014):

$$\mathbf{K}_{\boldsymbol{\theta}} \approx \mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^* + \mathbf{D}(\boldsymbol{\theta}) \tag{1}$$

In the above, $\mathbf{K}_{\boldsymbol{\theta}} \in \mathbb{R}^{n\times n}$ denotes the kernel matrix, where the subscript $\boldsymbol{\theta}$ denotes the dependence of the kernel matrix on the hyperparameters $\boldsymbol{\theta}$ (discussion of our notation appears in Section 2.1), $\mathbf{Z}$ has $s \ll n$ columns, and $\mathbf{W}(\boldsymbol{\theta}), \mathbf{D}(\boldsymbol{\theta})$ are diagonal matrices. The various steps of GPR can be much more efficiently conducted on a kernel whose kernel matrix has the structure of the righthand side of Eq. (1), e.g. training takes $O(ns^2)$ (see Section 4.2 for details on efficient GPR with low-rank approximations with an even more restricted structure in which $\mathbf{D}(\boldsymbol{\theta})$ is a multiple of the identity).

In the kernel learning literature, methods for forming a low rank approximation of kernels can be roughly split into two approaches: methods that use data-dependent basis functions, and methods that use independent basis functions. An example for the first kind is the Nyström method (Williams and Seeger, 2001). Such methods utilize the given training data, and thus may outperform methods that use independent basis functions, especially when there is a large gap in the eigenspectrum. However, data dependence can incur additional costs. For example, the Nyström method requires keeping some of the data as part of the model.

Another class of methods for building low rank approximations of kernel matrices are methods that use independent basis functions, and thus approximate the kernel function directly. One such important and highly influential method is the *random Fourier features* approach suggested by Rahimi and Recht in 2007 (Rahimi and Recht, 2008). Following the publication of (Rahimi and Recht, 2008), there has been extensive research on random features, including works that attempt to improve the approximation quality of the method (e.g., (Sutherland and Schneider, 2015; Choromanski et al., 2018)), works that focused on using in random features to learn huge datasets (e.g. (Huang et al., 2014; Avron and Sindhwani, 2016)), and works that focused on theoretical analysis of random features (e.g. (Yang et al., 2012; Sriperumbudur and Szabó, 2015; Avron et al., 2017; Li et al., 2019)). The previous list is far from exhaustive. In the context of our work, worth mentioning is Avron et al. (Avron et al., 2017) which showed that if the kernel matrix of the approximate kernel spectrally approximates the kernel matrix of the true kernel then the excess risk when using kernel ridge regression with the approximate kernel is not much larger than the excess risk when using the true kernel.

Random Fourier features, and random features methods in general, are based on writing the kernel function as an integral and then using numerical integration schemes in order to construct a low rank approximation of that function[1]. In random Fourier features, a shift-invariant kernel is rewritten as an integral via an application of Bochner's theorem, and Monte-Carlo integration is used to build the low rank approximation. The use of Quasi Monte-Carlo in lieu of Monte-Carlo integration was explored Avron et al. (Avron et al., 2016). Bach explored the connection between random Fourier features and kernel quadrature rules in (Bach, 2017), however without providing any practically useful explicit mappings for kernels. Monte-Carlo and Quasi-Monte Carlo integration admit only a slow convergence rate. As a consequence, the number of features required for a spectral approximation when using Monte-Carlo or Quasi Monte-Carlo integration must be polynomial in a quality parameter of the spectral approximation. In this paper we argue that in the context of Gaussian process regression a stronger notion of spectral equivalence is required. The slow convergence rate of Monte-Carlo or Quasi Monte-Carlo integration implies that at best the number of features required for spectral equivalence is linear in the number of training points, which is obviously undesirable.

1. A rank k bivariate function $f(\mathbf{x}, \mathbf{y})$ is a function that can be written as $f(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{k} \sigma_j \phi_j(\mathbf{x})\psi_j(\mathbf{y})$ for some $\sigma_1, \ldots, \sigma_k, \phi_1, \ldots, \phi_k, \psi_1, \ldots, \psi_k$ (Townsend and Trefethen, 2013).

Random features approaches based on Monte-Carlo and Quasi Monte-Carlo suffer from another serious defect when it comes to GPR: the low rank approximation they build has the form

$$\mathbf{K}_{\boldsymbol{\theta}} \approx \mathbf{Z}(\boldsymbol{\theta})\mathbf{Z}(\boldsymbol{\theta})^* + \mathbf{D}(\boldsymbol{\theta}) \tag{2}$$

While for training and prediction, this structure works equally as well as the structure in Eq. (1), when it comes to hyperparameter learning this is no longer the case; see Section 4.3.

One can construct faster converging low-rank approximations using numerical quadrature rules such as Gaussian quadrature. Dao et al. considered the use of Gaussian quadrature in the context of kernel learning (Dao et al., 2017). Gaussian quadrature rule is a method for numerically approximating *weighted* integrals (i.e., integrals of the form $\int_{-\infty}^{\infty} f(x)w(x)dx$ where $w(x) \geq 0$ is a weight function) that is optimal in some formal sense. In the context of approximating kernel functions, the weight function $w(\cdot)$ is determined by the kernel function and the value of the hyperparameters. Once the weight function has been determined, in order to use a Gaussian quadrature the nodes and weights corresponding to that particular weight function must be computed. Efficient algorithms exist, but these algorithms require the computation of integrals as well. For a single kernel, that is when using a fixed value of the hyperparameters, and when using a fixed number of quadrature features, computing the nodes and weights is a one-time offline task. However, if the hyperparameters are not fixed, e.g. when they are set using hyperparameter learning, Gaussian quadrature becomes unrealistic. Furthermore, the fact that the nodes and weights change with the hyperparameters implies that we must use an approximation of the form of Eq. (2) and not of Eq. (1), which is less desirable. The connection between random Fourier features and quadrature rules was also explored by Munkhoeva et al. (Munkhoeva et al., 2018). One noticeable limitation of quadrature rules is the curse of dimensionality, since the scale of their computational requirements increases with dimension. Thus, this approach is more applicable for low-dimensional datasets, such as spatial data.

Low rank approximations for kernels matrices have also been widely used in the statistics literature, and in particular the spatial statistics literature (Cressie and Johannesson, 2008; Eidsvik et al., 2012; Banerjee et al., 2008; Finley et al., 2009; Katzfuss and Cressie, 2012). Possible limitations of the low rank approximation approach in the context of spatial statistics have been noted in (Quiñonero-Candela and Rasmussen, 2005; Banerjee et al., 2008; Stein et al., 2007; Sang et al., 2011), and analyzed mathematically by Stein (Stein, 2014). The use of random features in the context of spatial statistics was explored by Ton et al. (Ton et al., 2018).

In this paper, we propose a quadrature based low-rank approximation approach for efficient GPR involving a wide class of kernels which includes shift-invariant kernels (i.e., stationary covariance functions). Unlike previous literature which uses quadrature features in the context of GPR, our method forms an approximation of the form of Eq. (1), and so is able to efficiently perform hyperparameter learning in addition to training and prediction. Our method achieves this by using a fixed set of quadrature nodes and weights, and designing the approximation so that varying the hyperparameters corresponds to only changing the integrand.

Specifically, our method uses a Gauss-Legendre quadrature. Gauss-Legendre quadrature is a Gaussian quadrature for the uniform weight function on a finite interval. Thus, the weight function does not change with the hyperparameters, and with it the quadrature nodes and weight stay fixed, whereas only the integrand varies. Changing only the integrand translates to a simplified parametric form for the approximate kernel matrix (Eq. (1)) which is more amenable to efficient computations. Our proposed method, which we call *Gauss-Legendre Features,* is described in Section 4.

The Gauss-Legendre quadrature is designed to approximate integrals with an integration area which is a finite interval. However, for most widely-used kernels the integrand has infinite support. We address this issue by utilizing the fact that for such kernels the integrand decays quickly, so we can approximate the integral by truncating the integration area. The truncation cutoff is determined by a parameter of our method. Another parameter is the number of features (i.e., quadrature nodes) used in the approximation. In order to set these two parameters correctly, we need a method for assessing the quality of one kernel function approximation by another. To that end, we introduce

the notion of *spectral equivalence*, and argue that if one parameterized family of kernels is spectrally equivalent to another one, then that first family is a good surrogate for the second family in the context of GPR. These results are summarized in Section 3.

We rigorously analyze how to set the truncation cutoff and the number of features to achieve spectral equivalence (these results are reported in Section 5). Here another advantage of using Gauss-Legendre quadrature becomes evident: the Gauss-Legendre quadrature converges much faster than the Monte-Carlo or Quasi Monte-Carlo integration, so typically the number of features is sublinear in the training size. Indeed, for widely used kernels like the Gaussian kernel and the Matèrn kernel, the number of features required when using Gauss-Legendre features is poly-logarithmic in the training size (see Section 6). A sublinear number of features is also likely achievable using Gaussian quadrature (kernel learning using Gaussian quadrature is suggested by Dao et al. (Dao et al., 2017), however without proving spectral equivalence). Yet, as explained this is rather problematic for hyperparameter learning, and in general requires a large overhead for computing the quadrature nodes and weights.

Finally, empirical results (Section 7) clearly demonstrate the superiority of our proposed method over classical random Fourier features when conducting Gaussian process regression on low-dimensional datasets.

2. Preliminaries

2.1 Notations and Basic Definitions

We consider all vectors as column vectors, unless otherwise stated. For a vector $\mathbf{x}$ or a matrix $\mathbf{A}$, the notation $\mathbf{x}^*$ or $\mathbf{A}^*$ denotes the Hermitian transpose. The $n \times n$ identity matrix is denoted by $\mathbf{I}_n$. A Hermitian matrix $\mathbf{A}$ is positive semidefinite (PSD) if $\mathbf{x}^*\mathbf{A}\mathbf{x} \geq 0$ for every vector $\mathbf{x}$. Also, for any Hermitian matrices $\mathbf{A}$, $\mathbf{B}$ of the same size, the notation $\mathbf{A} \preceq \mathbf{B}$ means that $\mathbf{B} - \mathbf{A}$ is PSD.

We consider n pairs of training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}$, where $\mathbf{x}$ denotes the input vector of dimension d and y denotes a scalar response. A *kernel function* (aka *covariance function*) is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which is *positive definite*, i.e. for every $m \in \mathbb{N}$ and $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^d$, the matrix $\mathbf{K} \in \mathbb{R}^{m \times m}$ defined by $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is PSD. The matrix $\mathbf{K}$ is known by various names: *kernel matrix, Gram matrix, covariance matrix.* Given a dataset $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, we will conveniently use $\mathbf{X}$ to denote the n-by-d matrix whose rows are $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and use $\mathbf{K}(\mathbf{X}, \mathbf{X})$ to denote the kernel matrix corresponding to the kernel k with data $\mathbf{X}$. For another kernel $\tilde{k}$ we will use $\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X})$ to denote the kernel matrix.

In many cases we will deal with parameterized families of kernels $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$, where $\boldsymbol{\theta}$ represents the hyperparameters vector, and Θ is a set of possible parameters values. The kernel matrix corresponding to $k_{\boldsymbol{\theta}}$ is denoted by $\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})$. We also group the responses $y_1, \ldots, y_n$ into a single vector $\mathbf{y} \in \mathbb{R}^n$.

The *Kullback–Leibler divergence* (abbreviated K*L-divergence* henceforth) is a well established metric for how much one distribution is different from a reference distribution. We denote the KL-divergence between two probability distributions $\mathcal{P}$ on $\mathcal{Q}$ by $D_{\mathbf{KL}}(\mathcal{P}, \mathcal{Q})$, and recall the following is a well established result[2]: if $\mathcal{N}_1 = \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$ and $\mathcal{N}_2 = \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$ are two multivariate normal distributions, we have

$$D_{\mathbf{KL}}(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{2}\mathbf{Tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\right) + \frac{1}{2}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\right)^{\mathrm{T}} \boldsymbol{\Sigma}_1^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\right) + \frac{1}{2}\left(\log\det\boldsymbol{\Sigma}_1 - \log\det\boldsymbol{\Sigma}_0\right) - \frac{n}{2} \quad (3)$$

2. The exact definition of the KL-divergence is not important, since we always use Eq. (3) when working with it.

2.2 Gaussian Process Regression

Gaussian Process Regression (GPR) is a Bayesian nonparametric approach for regression. First, the following regression model is assumed:

$$y = f(\mathbf{x}) + \varepsilon, \quad \varepsilon \overset{i.i.d}{\sim} \mathcal{N}\left(\mathbf{0}, \sigma_n^2\right)$$

(σ_n^2 is a (hyper)parameter). Additionally, it is assumed that f is a *Gaussian Process*, $f(\mathbf{x}) \sim \mathcal{GP}\left(\boldsymbol{\mu}\left(\mathbf{x}\right), k(\mathbf{x}, \mathbf{x}')\right)$, where k is the kernel function. This means that for any set of data points $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^d$ the vector $\mathbf{f} \in \mathbb{R}^m$ defined by $\mathbf{f}_j = f(\mathbf{x}_j)$ $(j = 1, \ldots, m)$ is a Gaussian random vector with mean defined by $\mu(\mathbf{X}) = [\mu(\mathbf{x}_1) \cdots \mu(\mathbf{x}_m)]^{\mathrm{T}}$ and the covariance matrix $\mathbf{K}(\mathbf{Z}, \mathbf{Z})$. Throughout the paper we assume, for the sake of simplicity, that the mean function $\mu(\mathbf{x})$ is zero. This simplifies the formulas while not really restricting generality (a nonzero mean can be easily handled by shifting). Under these assumptions, $y \sim \mathcal{N}(0, \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I})$. Under these priors, the expected predictive value for $f(\mathbf{x})$ at a test $\mathbf{x}$ is (Williams and Rasmussen, 2006)

$$f(\mathbf{x}) \approx \mathbf{K}(\mathbf{x}, \mathbf{X})\left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}_n\right)^{-1}\mathbf{y}\,.$$

Consequently, training is conducted by computing the vector

$$\boldsymbol{\alpha} \coloneqq \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}_n\right)^{-1}\mathbf{y}\,.$$

From these formulas, we see that assuming that evaluating the kernel function takes $O(d)$ operations and that $\boldsymbol{\alpha}$ is computed using direct factorization, training takes $O(n^3)$ operations and prediction takes $O(nd)$ operations.

The previous description is for a fixed kernel k. Typically, the kernel $k_{\boldsymbol{\theta}}$ has *hyperparameters* which we represent throughout the paper by the vector $\boldsymbol{\theta}$. The hyperparameters are usually constrained to some possible set of hyperparameters values Θ, and thus defines a parameterized family of kernels $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$. *Hyperparameter learning* refers to the process of determining the value of the hyperparameters directly from the training data, and is considered one of the important advantages of the GP framework. This is typically conducted by maximizing the log marginal likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) \coloneqq -\frac{1}{2}\mathbf{y}^{\mathrm{T}}\left(\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}_n\right)^{-1}\mathbf{y} - \frac{1}{2}\log\det(\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}_n) - \frac{n}{2}\log 2\pi \tag{4}$$

In order to maximize $\mathcal{L}(\boldsymbol{\theta})$ using a first-order optimization method, it is required to compute its gradients. Using direct methods, computing the gradient takes $O(n^3|\boldsymbol{\theta}|)$ where $|\boldsymbol{\theta}|$ represents the number of hyperparameters in $\boldsymbol{\theta}$.

2.3 Random Fourier Features

Random Fourier Features (RFF) (Rahimi and Recht, 2008), is one of the most popular methods for constructing a low rank approximation of kernels and scaling up kernel methods. The method targets shift-invariant kernels, i.e. kernels of the form $k(\mathbf{x}, \mathbf{x}') = k_0(\mathbf{x} - \mathbf{x}')$ for a positive definite function $k_0(\cdot)$.

RFF is motivated by a simple consequence of Bochner's Theorem: for every shift-invariant kernel for which $k_0(0) = \sigma_f^2$ there is a probability measure μ and possibly a corresponding probability density function $p(\cdot)$, both on $\mathbb{R}^d$, such that

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2\int_{\mathbb{R}^d} e^{-2\pi i\boldsymbol{\eta}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}')}d\mu(\boldsymbol{\eta}) = \sigma_f^2\int_{\mathbb{R}^d} e^{-2\pi i\boldsymbol{\eta}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}')}p(\boldsymbol{\eta})d\boldsymbol{\eta}$$

Let us assume that the density $p(\cdot)$ exists. If one chooses $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ randomly according to $p(\cdot)$, and defines $\varphi(\mathbf{x}) = \frac{1}{\sqrt{s}}\left(e^{-2\pi i\boldsymbol{\eta}_1^{\mathrm{T}}\mathbf{x}}, \ldots, e^{-2\pi i\boldsymbol{\eta}_s^{\mathrm{T}}\mathbf{x}}\right)^*$, then

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2\mathbb{E}_{\boldsymbol{\eta}_1,\ldots\boldsymbol{\eta}_s}\left[\varphi\left(\mathbf{x}\right)^*\varphi\left(\mathbf{x}'\right)\right]\,.$$

Hence, an approximated kernel can be defined:

$$\tilde{k}^{(\mathrm{RFF})}(\mathbf{x},\mathbf{x}') \coloneqq \varphi(\mathbf{x})^*\varphi(\mathbf{x}') = \frac{\sigma_f^2}{s}\sum_{j=1}^{s} e^{-2\pi i \boldsymbol{\eta}_j^{\mathrm{T}}(\mathbf{x}-\mathbf{x}')}\,.$$

The kernel matrix corresponding to the approximate kernel is

$$\tilde{\mathbf{K}}^{(\mathrm{RFF})}(\mathbf{X},\mathbf{X}) = \mathbf{Z}\mathbf{Z}^*$$

where $\mathbf{Z} \in \mathbb{C}^{n\times s}$ to be the matrix whose m^{th} row is $\varphi(\mathbf{x}_m)^*$. The low rank structure of $\tilde{\mathbf{K}}^{(\mathrm{RFF})}(\mathbf{X},\mathbf{X})$ allows more efficient training ($O(ns^2)$) and predictions ($O(sd)$), which are attractive if $s \ll n$.

The previous description is for a fixed kernel (and fixed hyperparameters). When using GPR with hyperparameter learning we are dealing parameterized family of kernels $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$. This case has not been considered in Rahimi and Recht original work (Rahimi and Recht, 2008). We discuss it in Section 4.3. Note that, there exist variants of RFF. One example is the modified RFF method (Avron et al., 2017) which suggests to sample $\mathbf{Z}$ using a different sampling distribution from the one used in classical RFF. In this method, another probability density function $q(\cdot)$ whose support includes that of $p(\cdot)$ is considered, such that $\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_s$ are chosen randomly according to $p(\cdot)/q(\cdot)$. Then, the approximate kernel is

$$\tilde{k}^{(\mathrm{MRF})}(\mathbf{x},\mathbf{x}') \coloneqq \frac{\sigma_f^2}{s}\sum_{j=1}^{s} \frac{p(\boldsymbol{\eta}_j)}{q(\boldsymbol{\eta}_j)} e^{-2\pi i \boldsymbol{\eta}_j^{\mathrm{T}}(\mathbf{x}-\mathbf{x}')}\,.$$

We consider this variant in the experiments reported in Section 7.

2.4 Gauss-Legendre Quadrature

Quadrature rules are a deterministic way to approximate integrals. A celebrated method for(mostly, one dimensional) numerical integration is Gaussian quadrature. The idea of Gaussian quadrature is to approximate integrals $\int_\Omega f(\boldsymbol{\eta})d\mu(\boldsymbol{\eta}) \approx \sum_{j\in\{1,\ldots,s\}^d} w_j f(\boldsymbol{\eta}_j)$ where the number of nodes s yields exactness of the approximation for all polynomials of degree up to $2s-1$. The nodes and weights depend on the measure $d\mu$ and the parameter s, and can be computed efficiently using a corresponding family of orthogonal polynomials (Hale and Townsend, 2013). Usually, the number of nodes s is chosen such that the quadrature error satisfies

$$\left|\int_\Omega f(\boldsymbol{\eta})d\mu(\boldsymbol{\eta}) - \sum_{j\in\{1,\ldots,s\}^d} w_j f(\boldsymbol{\eta}_j)\right| \le \Delta$$

for a certain Δ, i.e., is maximized to obtain an error at most Δ. Gaussian quadrature is very accurate, and thus beneficial in particular when dealing with "well behaved" functions (e.g., analytic functions or functions whose derivatives are smooth up to a certain degree (Trefethen, 2013)). Once the weight function is fixed, and the number of nodes (s) is chosen, the nodes and weights are fixed, and are computed only once.

Gauss-Legendre quadrature is a type of Gaussian quadrature for approximating integrals of the form $\int_{[-1,1]^d} f(\boldsymbol{\eta})d\boldsymbol{\eta}$, i.e. the uniform weight function is considered for the domain $[-1,1]^d$. The one dimensional weights of Gauss-Legendre quadrature are

$$w_j = \frac{2}{(1-\eta_j^2)(L_s'(\eta_j))^2}, \quad j = 1,\ldots,s$$

where L_s is the s-th Legendre polynomial, and the nodes $\eta_1,\ldots,\eta_s$ are simply the roots of L_s (Abramowitz and Stegun, 1972, p. 887). Higher dimension Gauss-Legendre quadarture are obtained by tensoring the one dimensional rule.

In this paper, we consider kernels that can be written as an integral form (e.g., as in the RFF method described in the previous section). The integral is typically a product of a feature map that depends on the family of the kernel (e.g., shift-invariant kernel) and a weight function which depends on the kernel itself. Thus, it seems natural to approximate this integral using a Gaussian quadrature. Indeed, such an approximation was considered in (Dao et al., 2017; Munkhoeva et al., 2018). However, all the aforementioned works considered fixed kernel parameters, and did not consider varying the kernel parameters as is done in hyperparameter learning for GPR. When using a Gaussian quadarture based on the weight function related to the kernel at hand, the weight function itself depends on the hyperparameters. This means that the nodes and weights have to be recomputed in every iteration. Furthermore, it is unclear how in that case one will be able to compute the derivative of the marginal likelihood if gradient descent is used to maximize the marginal likelihood.

Instead, in this work we decided to use Gauss-Legendre quandrature, truncating the integral and pushing the (truncated) weight function to be part of the integrand. We remark that Clenshaw-Curtis (considered with uniform weight function) would probably have worked similarly well, but Gauss-Legendre is optimal for the uniform weight function, so we chose it.

3. Spectrally Equivalent Kernel Approximations

Our strategy for scaling up GPR is based on approximating the kernel $k_{\boldsymbol{\theta}}$ by an approximate kernel $\tilde{k}_{\boldsymbol{\theta}}$ that is low-rank in some sense which will become apparent in the next section. This raises the question: how can we determine whether $\tilde{k}_{\boldsymbol{\theta}}$ indeed approximates $k_{\boldsymbol{\theta}}$ well? Avron et al. (Avron et al., 2017) suggested that in the context of kernel ridge regression, *spectral approximations* of the kernel matrices allow us to reason about how well one kernel is approximated by another. The argument in (Avron et al., 2017) is based on risk bounds for given fixed hyperparameters, and so is less appropriate for GPR where hyperparameter learning is common practice. In this section, we introduce the notion of *spectral equivalence*, a stronger form of spectral approximation, and connect it to hyperparameter learning in GPR.

Assume a bounding set $\mathcal{X} \subseteq \mathbb{R}^d$ for the data, then given a dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathbb{R}$, the general assumption when using a kernel k is that

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}(\mathbf{X}, \mathbf{X}))\,.$$

If, however, we would have used the kernel $\tilde{k}$, then the assumption would have been

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}))\,.$$

Thus, a measure on how well $\tilde{k}$ approximates k might be devised by measuring how much $\mathcal{N}(\boldsymbol{\mu}, \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}))$ is different from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}(\mathbf{X}, \mathbf{X}))$. The KL-divergence is a well-established measure on how different one probability distribution is from a reference distribution, so arguably, $\tilde{k}$ approximates k well if the KL-divergence $D_{\mathbf{KL}}\left(\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}(\mathbf{X}, \mathbf{X})), \mathcal{N}(\boldsymbol{\mu}, \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}))\right)$ is small. Indeed, the use of KL-divergence as such a measure was suggested in the literature on spatial data analysis (Banerjee et al., 2008; Sang and Huang, 2012; Stein, 2014).

The notion of spectral equivalence, which we develop below, is a measure of how two matrices are close to one another. To connect it to the KL-divergence, which we use to measure how well $\tilde{k}$ approximates k, we have the following lemma, which implies that if the covariance matrices of two multivariate distributions are close, then the KL-divergence is small.

Lemma 1 *Suppose that $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma}_0, , \boldsymbol{\Sigma}_1 \in \mathbb{R}^{n \times n}$ are two symmetric positive definite matrices. Suppose that*

$$(1 - n^{-1})\boldsymbol{\Sigma}_0 \preceq \boldsymbol{\Sigma}_1 \preceq (1 + n^{-1})\boldsymbol{\Sigma}_0\,. \tag{5}$$

Then,

$$D_{\mathbf{KL}}\left(\mathcal{N}(\boldsymbol{\mu}, \Sigma_0), \mathcal{N}(\boldsymbol{\mu}, \Sigma_1)\right) \leq 1 + O(n^{-1})\,.$$

We first need the following Lemma.

Lemma 2 *Suppose that* $\mathbf{A}$ *and* $\mathbf{B}$ *are two symmetric positive definite matrices of order* $n \times n$*, such that*

$$(1 - n^{-1})\mathbf{B} \preceq \mathbf{A} \preceq (1 + n^{-1})\mathbf{B}\,. \tag{6}$$

Then, there exists $\gamma_1, \ldots, \gamma_n \in [-n^{-1}, n^{-1}]$ *such that*

$$\log\det\mathbf{A} - \log\det\mathbf{B} = \sum_{i=1}^{n} \log(1+\gamma_i)\,.$$

Proof Let $\lambda_1, \ldots, \lambda_n$ denote the sorted eigenvalues of $\mathbf{B}$, and $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n$ denote the sorted eigenvalues of $\mathbf{A}$, so

$$\log\det\mathbf{B} = \sum_{i=1}^{n} \log\lambda_i, \quad \log\det\mathbf{A} = \sum_{i=1}^{n} \log\tilde{\lambda}_i\,.$$

Eq. (6) implies that there exist $\gamma_1, \ldots, \gamma_n \in [-n^{-1}, n^{-1}]$ such that $\tilde{\lambda}_i = (1+\gamma_i)\lambda_i$. Hence,

$$\begin{aligned}
\log\det\mathbf{A} &= \sum_{i=1}^{n} \log\tilde{\lambda}_i \\
&= \sum_{i=1}^{n} \log(1+\gamma_i)\lambda_i \\
&= \log\det\mathbf{B} + \sum_{i=1}^{n} \log(1+\gamma_i)
\end{aligned}$$

and that completes the proof. ■

Proof [Proof of Lemma 1]Since for two symmetric positive definite matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \preceq \mathbf{B}$ implies $\mathbf{B}^{-1} \preceq \mathbf{A}^{-1}$, Eq. (5) implies that

$$\left(1 - \frac{1}{n+1}\right)\boldsymbol{\Sigma}_0^{-1} = \frac{1}{1+n^{-1}}\boldsymbol{\Sigma}_0^{-1} \preceq \boldsymbol{\Sigma}_1^{-1} \preceq \frac{1}{1-n^{-1}}\boldsymbol{\Sigma}_0^{-1} = \left(1 + \frac{1}{n-1}\right)\boldsymbol{\Sigma}_0^{-1}\,.$$

Multiplying by $\boldsymbol{\Sigma}_0^{1/2}$ on the right and left sides gives

$$\left(1 - \frac{1}{n+1}\right)\mathbf{I}_n \preceq \boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0^{1/2} \preceq \left(1 + \frac{1}{n-1}\right)\mathbf{I}_n$$

i.e., the eigenvalues of $\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0^{1/2}$ are bounded in the interval $[1-(n+1)^{-1}, 1+(n-1)^{-1}]$. Thus,

$$\mathbf{Tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\right) = \mathbf{Tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_0^{1/2}\right) = \mathbf{Tr}\left(\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0^{1/2}\right) \leq n\left(1 + \frac{1}{n-1}\right)\,.$$

Also, from Lemma 2, there exist $\gamma_1, \ldots, \gamma_n \in [-n^{-1}, n^{-1}]$ such that

$$\log\det\boldsymbol{\Sigma}_1 - \log\det\boldsymbol{\Sigma}_0 = \sum_{i=1}^{n} \log(1+\gamma_i)\,.$$

Using Eq. (3), we obtain

$$\begin{aligned} D_{\mathbf{KL}}\left(\mathcal{N}(\boldsymbol{\mu},\Sigma_0),\mathcal{N}(\boldsymbol{\mu},\Sigma_1)\right) &= \frac{1}{2}\mathbf{Tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\right) + \frac{1}{2}(\log\det\boldsymbol{\Sigma}_1 - \log\det\boldsymbol{\Sigma}_0) - \frac{n}{2} \\ &\leq \frac{n}{2}\left(1+\frac{1}{n-1}\right) + \frac{1}{2}\sum_{i=1}^{n}\log\left(1+\gamma_i\right) - \frac{n}{2} \\ &\leq \frac{n}{2}\left(\frac{1}{n-1} + \log\left(1+\frac{1}{n}\right)\right) \\ &= \frac{1}{2} + \frac{1}{2(n-1)} + \frac{n}{2}\left(\frac{1}{n} + O\left(\frac{1}{n^2}\right)\right) \\ &= 1 + O\left(\frac{1}{n}\right). \end{aligned}$$

■

Lemma 1 motivates the following definitions:

Definition 3 *We say that a n-by-n symmetric matrix $\mathbf{A}$ is* spectrally equivalent *to another n-by-n symmetric matrix $\mathbf{B}$ if*

$$(1-n^{-1})\mathbf{B} \preceq \mathbf{A} \preceq (1+n^{-1})\mathbf{B}. \tag{7}$$

Remark 4 *Spectral equivalence is a strong type of* spectral approximation. *Indeed, for a $\Delta \geq 0$, a matrix $\mathbf{A}$ is a Δ-spectral approximation of another matrix $\mathbf{B}$, if $(1-\Delta)\mathbf{B} \preceq \mathbf{A} \preceq (1+\Delta)\mathbf{B}$ (Avron et al., 2017), so spectral equivalence is spectral approximation with $\Delta = n^{-1}$. As the previous lemmas show, this is the level of approximation required to make the KL-divergence between the distributions defined by two covariance matrices to be small.*

Definition 5 *Let $n \geq 1$ be an integer, and $\mathcal{X}$ be a data domain. Two positive definite kernels k and $\tilde{k}$ are n-spectrally equivalent* on domain $\mathcal{X}$ *if for every $\mathbf{X}$ with n rows in $\mathcal{X}$, the kernel matrix $\tilde{\mathbf{K}}(\mathbf{X},\mathbf{X})$ is spectrally equivalent to the kernel matrix $\mathbf{K}(\mathbf{X},\mathbf{X})$.*

The last definition uses two specific kernels, k and $\tilde{k}$. In GPR, we usually use a parameterized family of kernels $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$, where $\boldsymbol{\theta}$ represents the hyperparameters, and Θ is a set of possible parameter values. We generally assume that Θ is bounded. Boundedness of Θ is necessary, since without it, it is possible to drive the kernel matrix to identity, thereby making it impossible to approximate it using a low rank matrix. We then approximate each kernel $k_{\boldsymbol{\theta}}$ by $\tilde{k}_{\boldsymbol{\theta}}$, that is we use the parameterized family $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$. We say that the parameterized family $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ approximates the parameterized family $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ well if for every $\boldsymbol{\theta}\in\Theta$ the kernel $\tilde{k}_{\boldsymbol{\theta}}$ approximates $k_{\boldsymbol{\theta}}$ well, as is captured by the following definition.

Definition 6 *Two parameterized families of positive definite kernels $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ and $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ are said to be n-spectrally equivalent on domain $\mathcal{X}$ if for every $\boldsymbol{\theta}\in\boldsymbol{\Theta}$, $k_{\boldsymbol{\theta}}$ and $\tilde{k}_{\boldsymbol{\theta}}$ are n-spectrally equivalent over $\mathcal{X}$.*

In light of Lemma 1, if two parameterized families $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ and $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ are n-spectrally equivalent, then for any parameters $\boldsymbol{\theta}$ and any dataset consisting of n data points, the distributions on the response assumed by the two GP models induced by these families are close in the sense that the KL-divergence is close to 1.

4. Gauss-Legendre Features

In this section, we present our proposed method (Gauss-Legendre Features) and show how it can be used to perform efficient Gaussian process regression. Our method includes two important parameter vectors: $\mathbf{U}$ and $\mathbf{s}$. In the next section, we show how these parameters can be set in order to obtain an approximation that is spectrally equivalent to the true kernel.

4.1 Feature Map

We begin by describing the Gauss-Legendre feature map. The proposed method builds feature maps for kernel families that can be written in the following form:

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \sigma_f^2 \int_{\mathbb{R}^d} \varphi(\mathbf{x},\boldsymbol{\eta})\varphi(\mathbf{x}',\boldsymbol{\eta})^* p(\boldsymbol{\eta};\boldsymbol{\theta}_0) d\boldsymbol{\eta} + \sigma_n^2 \gamma(\mathbf{x}-\mathbf{x}') \,. \tag{8}$$

In the above,

$$\gamma(\mathbf{z}) \coloneqq \begin{cases} 1 & \mathbf{z} = 0 \\ 0 & \mathbf{z} \neq 0 \end{cases},$$

the function $\varphi : \mathcal{X} \times \mathbb{R}^d \to \mathbb{C}$ is such that for every $\mathbf{x} \in \mathcal{X}$ the function $\varphi(\mathbf{x},\cdot)$ is even-symmetric (i.e., for every $\boldsymbol{\eta} \in \mathbb{R}^d$, $\varphi(\mathbf{x},\boldsymbol{\eta}) = \varphi(\mathbf{x},-\boldsymbol{\eta})^*$), $\boldsymbol{\theta} = [\boldsymbol{\theta}_0, \sigma_f^2, \sigma_n^2]$, and for every $\boldsymbol{\theta}_0$ the function $p(\cdot;\boldsymbol{\theta}_0)$ is an even probability density on $\mathbb{R}^d$. Note that $\boldsymbol{\theta}_0$ can be a vector.

Note that in Eq. (8) we included a ridge term $\sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$. Typically, the ridge term is omitted from the kernel but appears in various equations involving the kernel matrix due to the Gaussian noise assumption in the GPR model. While we could state our theory in the more traditional way of having the noise term outside of the kernel, the definitions and theorems statements will be somewhat more cumbersome. We found it more convenient to include σ_n^2 as part of the vector of the parameter set $\boldsymbol{\theta}$, and include the ridge term in the kernel definition. The resulting equations are the same.

There are quite a few kernel families that adhere to this structure. For example, due to Bochner's theorem, a shift-invariant kernel with an additional noise level term can be written in the form

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \sigma_f^2 \int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}\boldsymbol{\eta}} p(\boldsymbol{\eta};\boldsymbol{\theta}_0) d\boldsymbol{\eta} + \sigma_n^2 \gamma(\mathbf{x}-\mathbf{x}') \,. \tag{9}$$

So, we can use $\varphi(\mathbf{x},\boldsymbol{\eta}) = e^{-i\mathbf{x}^{\mathrm{T}}\boldsymbol{\eta}}$ to cast shift invariant kernels in the form of Eq. (8).

The underlying idea of Gauss-Legendre features is to first truncate the integral Eq. (9) to the domain $\mathcal{Q}_{\mathbf{U}} = \prod_{k=1}^d [-U_k, U_k]$, for some $\mathbf{U} = (U_1,\ldots,U_d)^{\mathrm{T}}$, and then approximate the truncated integral using a tensorized Gauss-Legendre quadrature. Since the weight function is uniform for the Gauss-Legendre quadrature rule, it enables the method to fit a variety of kernel classes. The domain $\mathcal{Q}_{\mathbf{U}}$ might depend on $\mathcal{X}$ and Θ, but not on the concrete dataset $\mathbf{x}_1,\ldots,\mathbf{x}_n$. Let $(\chi_1^{(m)}, w_1^{(m)}),\ldots,(\chi_m^{(m)}, w_m^{(m)})$ denote the nodes and weights of the m-point Gauss-Legendre. Assume we are given a list of quadrature size for each dimension: $\mathbf{s} = (s_1,\ldots,s_d)$. Let $s = \prod_{k=1}^d s_k$. The approximation then reads:

$$\begin{aligned}
k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') &\approx \sigma_f^2 \int_{\mathcal{Q}_U} \varphi(\mathbf{x},\boldsymbol{\eta})\varphi(\mathbf{x}',\boldsymbol{\eta})^* p(\boldsymbol{\eta};\boldsymbol{\theta}_0) d\boldsymbol{\eta} + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}') \\
&\approx \sigma_f^2 \sum_{j_1=1}^{s_1} \cdots \sum_{j_d=1}^{s_d} w_{j_1\cdots j_d} p(\hat{\boldsymbol{\eta}}_{j_1\cdots j_d};\boldsymbol{\theta}_0)\varphi(\mathbf{x},\hat{\boldsymbol{\eta}}_{j_1\ldots j_d})\varphi(\mathbf{x}',\hat{\boldsymbol{\eta}}_{j_1\cdots j_d})^* + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}') \\
&= \sigma_f^2 \sum_{j=1}^{s} h_j(\boldsymbol{\theta}_0)\varphi(\mathbf{x},\boldsymbol{\eta}_j)\varphi(\mathbf{x}',\boldsymbol{\eta}_j)^* + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')
\end{aligned} \tag{10}$$

where

$$\hat{\boldsymbol{\eta}}_{j_1\cdots j_d} \coloneqq \begin{bmatrix} \eta_{j_1}^{(s_1)} \\ \vdots \\ \eta_{j_d}^{(s_d)} \end{bmatrix} = \begin{bmatrix} U_1 \cdot \chi_{j_1}^{(s_1)} \\ \vdots \\ U_d \cdot \chi_{j_d}^{(s_d)} \end{bmatrix}$$

and

$$w_{j_1 \cdots j_d} = \prod_{k=1}^{d} U_k w_{j_k}^{(s_k)} .$$

In Eq. (10), we assumed we have a bijective mapping $a_1, \ldots, a_s$ between $\{1, \ldots, s\}$ and $\{1, \ldots, s_1\} \times \cdots \times \{1, \ldots, s_d\}$ and then defined:

$$\boldsymbol{\eta}_j \coloneqq \hat{\boldsymbol{\eta}}_{a_j} \qquad h_j(\boldsymbol{\theta}_0) \coloneqq w_{a_j} p(\boldsymbol{\eta}_j; \boldsymbol{\theta}_0) .$$

Finally, the parameterized family of approximate kernels is

$$\tilde{k}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') \coloneqq \sigma_f^2 \sum_{j=1}^{s} h_j(\boldsymbol{\theta}_0) \varphi(\mathbf{x}, \boldsymbol{\eta}_j) \varphi(\mathbf{x}', \boldsymbol{\eta}_j)^* + \sigma_n^2 \gamma(\mathbf{x} - \mathbf{x}') . \tag{11}$$

Note that the conditions that $p(\cdot; \boldsymbol{\theta}_0)$ is even and $\varphi(\mathbf{x}, \cdot)$ is even-symmetric, coupled with the fact that the Gauss-Legendre quadrature is symmetric, ensures that $\tilde{k}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$ is always real. We stress that since the weights in Gauss-Legendre quadrature are fixed once $\mathbf{U}$ and $\mathbf{s}$ are chosen, we need to compute them only once. It is also worth noting that Clenshaw-Curtis quadrature probably would work similarly to Gauss-Legendre quadrature, when using the uniform density function (Trefethen, 2008, 2013). Nevertheless, since Gauss-Legendre quadrature is optimal for the uniform weight function, we decided to use it and not Clenshaw-Curtis.

Consider the first term on the right hand side of Eq. (11). It is a bivariate function that can be written as a sum of s separable bivariate functions. Thus, we can informally view s as the rank of the decomposition, and if s is small, then this is a low-rank approximation. The parameterized family of approximate kernels $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$ is composed of kernels that can be written as a low-rank bivariate function plus a ridge term. In the next subsection, we show how to utilize this low-rank structure in order to efficiently perform Gaussian process regression.

Of course, the crucial question is how do we choose $\mathbf{U} = (U_1, \ldots, U_d)$ and $\mathbf{s} = (s_1, \ldots, s_d)$. We want to choose these parameters such that $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$ and $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$ are n-spectrally equivalent over the domain $\mathcal{X}$, where n is the target dataset size (since GPR is nonparametric, the effective rank of the kernel matrix goes to infinity when n goes to infinity, so it is impossible to approximate the kernel matrix well with a matrix of fixed rank, i.e., with s fixed, as n goes to infinity). We discuss this question in the next section. In the reminder of this section, we discuss how to efficiently perform GPR using the approximate kernel family $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$.

4.2 Efficient Gaussian Process Regression

As a first and crucial step, we show how to write the kernel matrix of $\tilde{k}_{\boldsymbol{\theta}}$ as a low-rank matrix plus a ridge term. Given a dataset $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, let $\mathbf{X}$, as usual, denote the matrix whose row j is $\mathbf{x}_j^{\mathrm{T}}$. Define the matrix

$$\mathbf{Z} \in \mathbb{C}^{n \times s}, \quad \mathbf{Z}_{lj} \coloneqq \varphi(\mathbf{x}_l, \boldsymbol{\eta}_j) .$$

Note that $\mathbf{Z}$ depends on $\mathbf{U}$ and $\mathbf{s}$, but not on the hyperparameters $\boldsymbol{\theta}$. Next, define

$$\mathbf{W} : \Theta \to \mathbb{R}_+^{s \times s}, \quad \mathbf{W}(\boldsymbol{\theta}) \coloneqq \begin{bmatrix} h_1(\boldsymbol{\theta}_0) & & \\ & \ddots & \\ & & h_s(\boldsymbol{\theta}_0) \end{bmatrix} .$$

We now have

$$\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X}) = \sigma_f^2 \mathbf{Z} \mathbf{W}(\boldsymbol{\theta}) \mathbf{Z}^* + \sigma_n^2 \mathbf{I}_n .$$

Notice that dependence on $\boldsymbol{\theta}$ is confined to the diagonal matrix $\mathbf{W}(\boldsymbol{\theta})$. This will be very helpful in deriving efficient formulas for GPR.

We now discuss each of the various stages of GPR separately. For simplicity, we assume that the GP prior has zero mean ($\mu = 0$).

Training. Given $y_1, \ldots, y_n$, training usually amounts to computing the vector

$$\boldsymbol{\alpha} \coloneqq \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}$$

where $\mathbf{y} = [y_1, \ldots, y_n]^{\mathrm{T}}$. However, in our case, in order to utilize the structure of $\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})$, we instead compute:

$$\mathbf{w} \coloneqq \mathbf{Z}^*\boldsymbol{\alpha} = \mathbf{W}(\boldsymbol{\theta})^{-1}(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1})^{-1}\mathbf{Z}^*\mathbf{y}\,.$$

In the above, the second equality is a simple consequence of the Woodbury matrix identity. Since $\mathbf{W}(\boldsymbol{\theta})$ has positive diagonal, $\mathbf{w}$ can be computed using $O(ns^2)$ operations, discounting the cost of computing $\mathbf{Z}$.

Prediction. Given a test set $\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_t^{(t)}$ which are distinct from the training set, the predicted (mean) vector $\mathbf{y}^{(t)} = [y_1^{(t)}, \ldots, y_t^{(t)}]^{\mathrm{T}}$ is defined by $\mathbf{y}^{(t)} \coloneqq \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)}, \mathbf{X})\boldsymbol{\alpha}$. Let

$$\mathbf{Z}^{(t)} \in \mathbb{C}^{t\times s}, \quad \mathbf{Z}^{(t)}_{lj} \coloneqq \varphi(\mathbf{x}_l^{(t)}, \boldsymbol{\eta}_j)\,.$$

Then, we have

$$\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)}, \mathbf{X}) = \sigma_f^2\mathbf{Z}^{(t)}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^*$$

and

$$\begin{aligned}\mathbf{y}^{(t)} &= \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)}, \mathbf{X})\boldsymbol{\alpha}\\ &= \sigma_f^2\mathbf{Z}^{(t)}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^*\boldsymbol{\alpha}\\ &= \sigma_f^2\mathbf{Z}^{(t)}\mathbf{W}(\boldsymbol{\theta})\mathbf{w}\,.\end{aligned}$$

The predictive variance is then

$$\begin{aligned}\mathbf{Y}^{(t)} &= \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)}, \mathbf{X}^{(t)}) - \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)}, \mathbf{X})\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})^{-1}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)}, \mathbf{X})^*\\ &= \sigma_f^2\mathbf{Z}^{(t)}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^{(t)*} - \sigma_f^2\mathbf{Z}^{(t)}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^*\left(\sigma_f^2\mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^* + \sigma_n^2\mathbf{I}_n\right)^{-1}\mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^{(t)*}\\ &= \sigma_f^2\mathbf{Z}^{(t)}\mathbf{W}(\boldsymbol{\theta})\left(\mathbf{I}_s - \sigma_n^{-2}\mathbf{W}(\boldsymbol{\theta})^{-1}\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1}\right)^{-1}\mathbf{Z}^*\mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\right)\mathbf{Z}^{(t)*}\\ &= \sigma_f^2\mathbf{Z}^{(t)}\left(\mathbf{I}_s - \sigma_n^{-2}\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1}\right)^{-1}\mathbf{Z}^*\mathbf{Z}\right)\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^{(t)*}\\ &= \sigma_f^2\mathbf{Z}^{(t)}\left(\mathbf{I}_s - \sigma_n^{-2}\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1}\right)^{-1}\mathbf{Z}^*\mathbf{Z}\right)\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^{(t)*}\end{aligned}$$

Hence, once we have $\mathbf{w}$ (computed during training), we can compute $\mathbf{y}^{(t)}$ using $O(ts)$ operations, discounting the cost of compute $\mathbf{Z}^{(t)}$, and $\mathbf{Y}^{(t)}$ using $O(s^3 + ts^2)$ operations.

Hyperparameter Learning. Hyperparameter learning amounts to finding the hyperparameters $\boldsymbol{\theta}$ which maximize the log marginal likelihood. To do so, we need to be to able to efficiently compute the log marginal likelihood and its gradient. It is well known that the likelihood is given by

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^{\mathrm{T}}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y} - \frac{1}{2}\log\det\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X}) - \frac{n}{2}\log 2\pi \tag{12}$$

and the derivatives are given by

$$\frac{\partial\mathcal{L}}{\partial\theta_i} = -\frac{1}{2}\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})}{\partial\theta_i}\right) + \frac{1}{2}\mathbf{y}^{\mathrm{T}}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})}{\partial\theta_i}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y} \tag{13}$$

where θ_i represents a hyperparameter in $\boldsymbol{\theta}$.

Proposition 7 *After an $O(ns^2)$ preprocessing step of computing $\mathbf{Z}^*\mathbf{Z}$, and discounting the cost of computing the partial derivatives of p with respect to the hyperparamters, the log marginal likelihood $\mathcal{L}(\boldsymbol{\theta})$ and the gradient $\nabla\mathcal{L}(\boldsymbol{\theta})$ can be computed in $O(ns + s^3 + s|\boldsymbol{\theta}|)$ arithmetic operations, where $|\boldsymbol{\theta}|$ represents the number of hyperparameters in $\boldsymbol{\theta}$. Furthermore, the amount of memory storage required is $O(s^2)$.*

Proof First, let us consider the computation of the likelihood. For the first term in Eq. (12), note that

$$\boldsymbol{\alpha}(\boldsymbol{\theta}) = \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\mathbf{y} = \sigma_n^{-2}(\mathbf{y} - \sigma_f^2\mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{w}(\boldsymbol{\theta}))$$

where $\mathbf{w}(\boldsymbol{\theta}) = \mathbf{W}(\boldsymbol{\theta})^{-1}(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1})^{-1}\mathbf{Z}^*\mathbf{y}$. In the previous equations, we made the dependence of $\boldsymbol{\alpha}$ and $\mathbf{w}$ on $\boldsymbol{\theta}$ explicit. Obviously, once $\mathbf{Z}^*\mathbf{Z}$ has been computed (an $O(ns^2)$ preprocessing step), we can compute both $\mathbf{w}(\boldsymbol{\theta})$ and $\boldsymbol{\alpha}(\boldsymbol{\theta})$ in $O(ns + s^3)$. The first term in Eq. (12) is now equal to $-\mathbf{y}^{\mathrm{T}}\boldsymbol{\alpha}(\boldsymbol{\theta})/2$.

For the second term in Eq. (12), using the matrix determinant lemma, we have

$$\log\det\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X}) = n\log\sigma_n^2 + s\log\sigma_f^2 + \log\det\mathbf{W}(\boldsymbol{\theta}) + \log\det(\sigma_f^{-2}\mathbf{W}(\boldsymbol{\theta})^{-1} + \sigma_n^{-2}\mathbf{Z}^*\mathbf{Z})\,.$$

Since $\mathbf{W}(\boldsymbol{\theta})$ is diagonal, once $\mathbf{Z}^*\mathbf{Z}$ has been computed, we can compute this term in $O(s^3)$ operations.

Next, let us consider the computation of each derivative of the likelihood according to Eq. (13). The crucial observations are:

$$\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_f^2} = \mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^*, \quad \frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_n^2} = \mathbf{I}_n, \quad \frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\theta_i} = \sigma_f^2\mathbf{Z}\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*$$

(using θ_i to denote an hyperparameter in $\boldsymbol{\theta}_0$) where using last equality amounts to computing partial derivative $\partial p(\boldsymbol{\eta}_j;\boldsymbol{\theta}_0)/\partial\theta_i$ for $j = 1,\ldots,s$. For the second term in Eq. (13) we have,

$$\begin{aligned}
\boldsymbol{\alpha}(\boldsymbol{\theta})^{\mathrm{T}}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_f^2}\boldsymbol{\alpha}(\boldsymbol{\theta}) &= \mathbf{w}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{W}(\boldsymbol{\theta})\mathbf{w}(\boldsymbol{\theta}) \\
\boldsymbol{\alpha}(\boldsymbol{\theta})^{\mathrm{T}}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_n^2}\boldsymbol{\alpha}(\boldsymbol{\theta}) &= \|\boldsymbol{\alpha}(\boldsymbol{\theta})\|_2^2 \\
\boldsymbol{\alpha}(\boldsymbol{\theta})^{\mathrm{T}}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\theta_i}\boldsymbol{\alpha}(\boldsymbol{\theta}) &= \sigma_f^2\mathbf{w}(\boldsymbol{\theta})^{\mathrm{T}}\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{w}(\boldsymbol{\theta})
\end{aligned}$$

so this term can be computed in $O(s)$ operations once we compute $\mathbf{w}(\boldsymbol{\theta})$ and $\boldsymbol{\alpha}(\boldsymbol{\theta})$ (which are computed during the computation of the likelihood). For the first term in Eq. (13), let

$$\mathbf{F}(\boldsymbol{\theta}) \coloneqq \sigma_f^2(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1})^{-1}\mathbf{Z}^*\mathbf{Z}\,.$$

Again, once $\mathbf{Z}^*\mathbf{Z}$ has been computed, $\mathbf{F}(\boldsymbol{\theta})$ can be computed in $O(s^3)$ operations. Now, using the Woodbury formula and cyclicality of the trace, we have

$$\begin{aligned}
\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_f^2}\right) &= \sigma_f^{-2}\mathbf{Tr}\left(\mathbf{F}(\boldsymbol{\theta})\right) \\
\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_n^2}\right) &= \sigma_n^{-2}(n - \mathbf{Tr}\left(\mathbf{F}(\boldsymbol{\theta})\right)) \\
\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\theta_i}\right) &= \sigma_n^{-2}\sigma_f^2\mathbf{Tr}\left(\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\mathbf{Z}\right) - \sigma_n^{-2}\sigma_f^2\mathbf{Tr}\left(\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})\right)
\end{aligned} \tag{14}$$

(the calculations leading to these formulas appear in Appendix A). Thus, once $\mathbf{Z}^*\mathbf{Z}$, $\mathbf{F}(\boldsymbol{\theta})$ and the diagonal of $\mathbf{Z}^*\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})$ have been computed, all these computations can be done in $O(s|\boldsymbol{\theta}|)$. Note that the diagonal of $\mathbf{Z}^*\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})$ can be computed using $O(s^2)$ operations once we have $\mathbf{Z}^*\mathbf{Z}$ and $\mathbf{F}(\boldsymbol{\theta})$.

In terms of memory storage, notice that once $\mathbf{Z}^*\mathbf{Z}$ and $\mathbf{Z}^*\mathbf{y}$ have been computed there is no longer any need for $\mathbf{Z}$ (which requires $O(ns)$ in storage). However, in order to compute $\mathbf{Z}^*\mathbf{Z}$ and $\mathbf{Z}^*\mathbf{y}$ we do not need to form all of $\mathbf{Z}$ in memory, but rather can stream over the training set transforming every training point using φ and accumulating its contribution to $\mathbf{Z}^*\mathbf{Z}$ and $\mathbf{Z}^*\mathbf{y}$. Thus, the dominant storage cost is for holding $\mathbf{Z}^*\mathbf{Z}$ which is $O(s^2)$. ■

The computations can be performed more stably by utilizing various matrix identities. We delegate the details to Appendix A.

To have a computational advantage in the training and prediction steps we need $s = o(n)$. However, since for most kernels computing the gradient of the likelihood requires $O(n^3|\boldsymbol{\theta}|)$, our method has a computational advantage in the hyperparameter learning phase even if $s = \Theta(n)$.

4.3 Comparison to Other Methods

Our proposed method is very much inspired by the Random Fourier Features (RFF) method (Rahimi and Recht, 2008). Although originally defined only for shift-invariant kernels, the method can be easily generalized for kernels of the form of Eq. (8). We refer to the generalization as *Monte-Carlo Features* (MCF). RFF is a special case of MCF. In particular, given some fixed parameters $\boldsymbol{\theta}$, an MCF approximate kernel is

$$\tilde{k}_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \frac{\sigma_f^2}{s}\sum_{j=1}^{s}\varphi(\mathbf{x},\boldsymbol{\eta}_j)\varphi(\mathbf{x}',\boldsymbol{\eta}_j)^* + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$$

where $\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_s$ are sampled from the density function $p(\boldsymbol{\eta};\boldsymbol{\theta}_0)$. As before, we include the ridge term $\sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$ in the kernel definition. Thus, the kernel matrix approximation is

$$\tilde{\mathbf{K}}_{\boldsymbol{\theta}}^{(\mathrm{MCF})}(\mathbf{X},\mathbf{X}) = \sigma_f^2\mathbf{Z}(\boldsymbol{\theta})\mathbf{Z}(\boldsymbol{\theta})^* + \sigma_n^2\mathbf{I}_n$$

where

$$\mathbf{Z}(\boldsymbol{\theta}) \in \mathbb{C}^{n\times s}, \quad \mathbf{Z}(\boldsymbol{\theta})_{lj} \coloneqq \varphi(\mathbf{x}_l,\boldsymbol{\eta}_j)\,.$$

Note that, according to (Avron et al., 2017), to obtain $\mathbf{K}$ and $\tilde{\mathbf{K}}_{\boldsymbol{\theta}}^{(\mathrm{MCF})}$ that are n-spectrally equivalent, the number of features of RFF should be $s = O(n^3)$, in which case the approximation is more expensive than exact computation. Obviously, training and prediction can be efficiently executed by utilizing the identity plus low-rank structure of $\tilde{\mathbf{K}}_{\boldsymbol{\theta}}^{(\mathrm{MCF})}$ much in the same way as we have done for Gauss-Legendre features, and indeed this is the reason the method was developed (Rahimi and Recht, 2008).

The above developments were for a *fixed* $\boldsymbol{\theta}$. However, it is less clear how to define a *family* of approximations for various $\boldsymbol{\theta}$, and perform hyperparameter learning. A key issue is that the kernel approximation should vary smoothly with $\boldsymbol{\theta}$, so obviously fresh $\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_s$ cannot be sampled differently for every $\boldsymbol{\theta}$. It is outside the scope of this paper to consider how to use MCF to define parameterized families of kernel approximation suitable for hyperparameter learning. Nevertheless, since we wish to use RFF as a baseline for complexity comparisons and numerical experiments, we show how it is possible to use the specific case of RFF to form parameterized families of kernel approximation and perform hyperparameter learning for a restricted family of kernels that includes the Gaussian and Matèrn kernels.

Specifically, we will consider a restricted class of shift-invariant whose kernel has the following specific form:

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \sigma_f^2 k_0(\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')) + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}') \tag{15}$$

where $k_0(\cdot)$ is a positive definite function, and $\mathbf{L}$ is diagonal with positive entries $L_1, \ldots, L_d$ which are part of parameter vector $\boldsymbol{\theta}$ (i.e., $\boldsymbol{\theta} = [L_1, \ldots L_d, \sigma_f^2, \sigma_n^2]$). Note that the number of variables in $\boldsymbol{\theta}_0$ is equal to the dimension d. Gaussian and Matèrn kernels are examples of such kernels. In this case we can write $k_{\boldsymbol{\theta}}$ in the form of Eq. (8), where $\varphi(\mathbf{x}, \boldsymbol{\eta}) = e^{-i\mathbf{x}^{\mathrm{T}}\boldsymbol{\eta}}$. We further assume that $p(\boldsymbol{\eta}; \mathbf{L})$ is such that sampling a random vector $\boldsymbol{\eta}$ is the same as sampling from the distribution defined by $p(\cdot; \mathbf{I}_d)$ and scaling the vector by $\mathbf{L}^{-1}$. Thus, for such kernels we can sample $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ once from $p(\cdot; \mathbf{I}_d)$ and view any change in $\boldsymbol{\theta}_0 = \mathbf{L}$ as corresponding change in $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$. Concretely, letting

$$\mathbf{W} = \begin{bmatrix} \boldsymbol{\eta}_1 & \cdots & \boldsymbol{\eta}_s \end{bmatrix},$$

the feature matrix is

$$\mathbf{Z}(\mathbf{L}) = \frac{1}{\sqrt{s}} \exp\left(-i\mathbf{X}\mathbf{L}^{-1}\mathbf{W}\right)$$

and the kernel matrix is

$$\tilde{\mathbf{K}}_{\boldsymbol{\theta}}^{(\mathrm{RFF})}(\mathbf{X}, \mathbf{X}) = \sigma_f^2 \mathbf{Z}(\mathbf{L})\mathbf{Z}(\mathbf{L})^* + \sigma_n^2 \mathbf{I}_n \,.$$

However, the crucial point is that now $\mathbf{Z}(\mathbf{L})$ depends smoothly on $\mathbf{L}$, so we can compute gradients. Formulas quite similar to the ones derived in the previous section can be derived (we omit most details), with the main difference being in taking the derivative of the kernel matrix with respect to the parameters in $\mathbf{L}$. Here we have

$$\begin{aligned}
\frac{\partial \tilde{\mathbf{K}}_{\boldsymbol{\theta}}^{(\mathrm{RFF})}(\mathbf{X}, \mathbf{X})}{\partial L_k} &= \sigma_f^2 \frac{\partial(\mathbf{Z}\mathbf{Z}^*)(\mathbf{L})}{\partial L_k} = \sigma_f^2 \left(\frac{\partial \mathbf{Z}(\mathbf{L})}{\partial L_k} \mathbf{Z}(\mathbf{L})^* + \mathbf{Z} \frac{\partial \mathbf{Z}(\mathbf{L})^*}{\partial L_k} \right) \\
\frac{\partial \mathbf{Z}(\mathbf{L})}{\partial L_k} &= -\frac{i}{\sqrt{s}} \left(\mathbf{X} \frac{\partial \mathbf{L}^{-1}}{\partial L_k} \mathbf{W} \odot \exp\left(-i \cdot \mathbf{X}\mathbf{L}^{-1}\mathbf{W}\right) \right) = -i \left(\mathbf{X} \frac{\partial \mathbf{L}^{-1}}{\partial L_k} \mathbf{W} \odot \mathbf{Z}(\mathbf{L}) \right) \\
\frac{\partial \mathbf{Z}(\mathbf{L})^*}{\partial L_k} &= \frac{i}{\sqrt{s}} \left(\mathbf{W}^{\mathrm{T}} \frac{\partial \mathbf{L}^{-1}}{\partial L_k} \mathbf{X}^{\mathrm{T}} \odot \exp\left(i \cdot \mathbf{X}\mathbf{L}^{-1}\mathbf{W}\right) \right) = i \left(\mathbf{W}^{\mathrm{T}} \frac{\partial \mathbf{L}^{-1}}{\partial L_k} \mathbf{X}^{\mathrm{T}} \odot \mathbf{Z}(\mathbf{L})^* \right) = i \left(\mathbf{X} \frac{\partial \mathbf{L}^{-1}}{\partial L_k} \mathbf{W} \odot \mathbf{Z}(\mathbf{L}) \right)^* .
\end{aligned}$$

In terms of complexity, the main difference between Gauss-Legendre Features and RFF is that for the former the matrix $\mathbf{Z}^*\mathbf{Z}$ stays constant when $\boldsymbol{\theta}$ varies, and so the product can be computed once, while for the latter $\mathbf{Z}(\mathbf{L})^*\mathbf{Z}(\mathbf{L})$ varies, and so changes every iteration. This adds an additional cost of $O(ns^2)$ operations for every gradient computation. Furthermore, we need to compute $\partial \mathbf{Z}(\mathbf{L})/\partial L_k$ for $k = 1, \ldots, d$ in each iteration, each costing $O(nsd)$ operations, for a total of $O(nsd^2)$ operations. Furthermore, since $\mathbf{Z}(\mathbf{L})$ changes in each iteration, and $\mathbf{Z}(\mathbf{L})$ features in many of the equations, we cannot compute the matrix $\mathbf{Z}(\mathbf{L})^*\mathbf{Z}(\mathbf{L})$ once and reduce storage costs to $O(s^2)$, and using $\mathbf{Z}(\mathbf{L})$ implicitly many times will incur a large overhead. Thus, for RFF the storage cost is $O(ns)$.

Table 1: Computational complexities (arithmetic operations) comparison between Gauss-Legendre Features and Random Fourier Features for kernels of the form of Eq. (15) (e.g., non-isotropic Gaussian an Matèrn kernels). In the table, n is the size of the training set, t is the size of the test set, s is the approximation rank, and I is the number of gradient computations for hyperparameter learning (e.g., number of gradient descent iterations).

	Gauss-Legendre Features	Random Fourier Features
Inference	$O(ns^2)$	$O(ns^2)$
Prediction	$O(nt)$	$O(nt)$
Hyperparameter learning:	$O(ns^2 + I(ns + s^3 + sd))$	$O(I(ns^2 + nsd^2))$

A similar issue will likely arise when using features based on Gaussian quadrature, like Dao et al. suggested (Dao et al., 2017) (that paper does not discuss GPR hyperparameter learning). When $\boldsymbol{\theta}$ changes, the distribution that defines Gaussian quadrature changes. Unlike Guass-Legendre features which use fixed nodes, for Gaussian quadrature features the quadrature nodes change with $\boldsymbol{\theta}$, which prevents the use of a fixed feature matrix $\mathbf{Z}(\boldsymbol{\theta})$. Furthermore, when performing hyperparameter learning with Gaussian quadrature features, we need to not only compute the quadrature weights but also compute their derivatives. We note that the formulas for the modified RFF are similar, where $\partial\mathbf{Z}(\mathbf{L})/\partial L_k$ and $\partial\mathbf{Z}(\mathbf{L})^*/\partial L_k$ will have an additional term that originates in pointwise multiplying the RFF matrix $\mathbf{Z}$ by $[p(\boldsymbol{\eta}_1)/q(\boldsymbol{\eta}_1),\ldots,p(\boldsymbol{\eta}_s)/q(\boldsymbol{\eta}_s)]$, as described in Section 2.3. However, this does not changes the costs and issues mentioned above.

5. Parameter Computation

In order to complete the description of our method, we need to specify how to choose $\mathbf{U}$ and $\mathbf{s}$. First, we show how to set $\mathbf{U}$ and $\mathbf{s}$ for a fixed $\boldsymbol{\theta}\in\Theta$ and n such that $k_{\boldsymbol{\theta}}$ and $\tilde{k}_{\boldsymbol{\theta}}$ are n-spectrally equivalent. We then consider how to set a fixed $\mathbf{U}$ and $\mathbf{s}$ such that the two families $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ and $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ are n-spectrally equivalent.

5.1 From Matrix Approximation to Integral Approximation

For now (and until subsection 5.4) let us assume that $\boldsymbol{\theta}$ is fixed. Our goal is to set $\mathbf{U}$ and $\mathbf{s}$ such that $k_{\boldsymbol{\theta}}$ and $\tilde{k}_{\boldsymbol{\theta}}$ are n-spectrally equivalent, i.e. for every dataset $\mathbf{x}_1,\ldots,\mathbf{x}_n\in\mathcal{X}$

$$(1-n^{-1})\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})\preceq\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})\preceq(1+n^{-1})\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})\,.$$

In other words, we want to set $\mathbf{U}$ and $\mathbf{s}$ such that for every $\mathbf{v}\in\mathbb{R}^n$,

$$(1-n^{-1})\mathbf{v}^{\mathrm{T}}\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})\mathbf{v}\le\mathbf{v}^{\mathrm{T}}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})\mathbf{v}\le(1+n^{-1})\mathbf{v}^{\mathrm{T}}\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})\mathbf{v}\,. \tag{16}$$

Henceforth, for conciseness, we drop $\mathbf{X}$ from the expressions, although the various expressions implicitly depend on $\mathbf{X}$.

Let

$$\mathbf{z}(\boldsymbol{\eta})\coloneqq\begin{bmatrix}\varphi(\mathbf{x}_1,\boldsymbol{\eta})\\ \vdots\\ \varphi(\mathbf{x}_n,\boldsymbol{\eta})\end{bmatrix}.$$

Then,

$$\mathbf{v}^{\mathrm{T}}\mathbf{K}_{\boldsymbol{\theta}}\mathbf{v}=\sigma_f^2\int_{\mathbb{R}^d}|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta}+\sigma_n^2\|\mathbf{v}\|_2^2$$

and

$$\mathbf{v}^{\mathrm{T}}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}\mathbf{v}=\sigma_f^2\sum_{j=1}^{s}h_j(\boldsymbol{\theta}_0)|\mathbf{z}(\boldsymbol{\eta}_j)^*\mathbf{v}|^2+\sigma_n^2\|\mathbf{v}\|_2^2\,.$$

Since rescaling $\mathbf{v}$ rescales all the terms in the previous inequality, we can assume without loss of generality that $\mathbf{v}^{\mathrm{T}}\mathbf{K}_{\boldsymbol{\theta}}\mathbf{v}=1$. In that case, Eq. (16) is equivalent to

$$\left|\int_{\mathbb{R}^d}|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta}-\sum_{j=1}^{s}h_j(\boldsymbol{\theta}_0)|\mathbf{z}(\boldsymbol{\eta}_j)^*\mathbf{v}|^2\right|\le\frac{1}{\sigma_f^2n}\,. \tag{17}$$

Thus, the nodes $\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_s$ and weights $h_1(\boldsymbol{\theta}_0)\ldots,h_s(\boldsymbol{\theta}_0)$ function as a quadrature approximation.

5.2 Truncating the Integral

As alluded earlier, we approach the quadrature approximation Eq. (17) by first truncating the integral and then using a Gauss-Legendre quadrature for the truncated integral. In other words, we write

$$\left|\int_{\mathbb{R}^d} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta} - \sum_{j=1}^{s} h_j(\boldsymbol{\theta}_0)|\mathbf{z}(\boldsymbol{\eta}_j)^*\mathbf{v}|^2\right| \le \left|\int_{\mathbb{R}^d} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta} - \int_{\mathcal{Q}_\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta}\right| + \left|\int_{\mathcal{Q}_\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta} - \sum_{j=1}^{s} h_j(\boldsymbol{\theta}_0)|\mathbf{z}(\boldsymbol{\eta}_j)^*\mathbf{v}|^2\right| \tag{18}$$

where $\mathcal{Q}_\mathbf{U} = \prod_{k=1}^{d}[-U_k, U_k]$. We set $\mathbf{U}$ such that the first term is smaller than $\sigma_f^{-2}n^{-1}/2$, and set each of the components in $\mathbf{s}$ to be large enough so that the second term is also smaller than $\sigma_f^{-2}n^{-1}/2$.

Obviously, we want to set the components in $\mathbf{U}$ to be as small as possible, to limit the integration area. Having a smaller integration area allows us to use smaller values in $\mathbf{s}$. The minimal values in $\mathbf{U}$ such that the first term is bounded by $\sigma_f^{-2}n^{-1}/2$ depends on how quickly $p(\boldsymbol{\eta};\boldsymbol{\theta}_0)$ decays as $\|\boldsymbol{\eta}\|_\infty \to \infty$: the faster the density decays, the smaller is the region where the function value has a significant contribution. Therefore, in our analysis, we distinguish between four classes of decay of $p(\cdot;\boldsymbol{\theta}_0)$, and analyze each on its own.

For a diagonal $\mathbf{L}$, with positive entries $L_1, \ldots, L_d$ on the diagonal (i.e., $\mathbf{L} = \mathbf{diag}\,(L_1, \ldots, L_d)$), let us define:

$$\mathcal{P}_{C,\mathbf{L}} \coloneqq \left\{p : \mathbb{R}^d \to \mathbb{R}_+ \text{ measurable} \,|\, \int_{\mathbb{R}^d} p(\boldsymbol{\eta})d\boldsymbol{\eta} = 1,\, p(\boldsymbol{\eta}) \le C \cdot \prod_{k=1}^{d} \frac{1}{1+L_k^2\eta_k^2}\right\}$$

$$\mathcal{P}^{(r)}_{C,\mathbf{L}} \coloneqq \left\{p : \mathbb{R}^d \to \mathbb{R}_+ \text{ measurable} \,|\, \int_{\mathbb{R}^d} p(\boldsymbol{\eta})d\boldsymbol{\eta} = 1,\, p(\boldsymbol{\eta}) \le C \cdot \left(1+\|\mathbf{L}\boldsymbol{\eta}\|_2^2\right)^{-r}\right\}$$

$$\mathcal{E}^{(1)}_{C,\mathbf{L}} \coloneqq \left\{p : \mathbb{R}^d \to \mathbb{R}_+ \text{ measurable} \,|\, \int_{\mathbb{R}^d} p(\boldsymbol{\eta})d\boldsymbol{\eta} = 1,\, p(\boldsymbol{\eta}) \le C \cdot e^{-\|\mathbf{L}\boldsymbol{\eta}\|_1}\right\}$$

$$\mathcal{E}^{(2)}_{C,\mathbf{L}} \coloneqq \left\{p : \mathbb{R}^d \to \mathbb{R}_+ \text{ measurable} \,|\, \int_{\mathbb{R}^d} p(\boldsymbol{\eta})d\boldsymbol{\eta} = 1,\, p(\boldsymbol{\eta}) \le C \cdot e^{-\|\mathbf{L}\boldsymbol{\eta}\|_2^2}\right\}.$$

The families $\mathcal{P}_{C,\mathbf{L}}$ and $\mathcal{P}^{(r)}_{C,\mathbf{L}}$ include densities that decay at a polynomial rate or faster, while $\mathcal{E}^{(1)}_{C,\mathbf{L}}$ includes densities that decay at exponential rate or faster, and $\mathcal{E}^{(2)}_{C,\mathbf{L}}$ includes densities that decay at a square exponential rate or faster. Table 2 shows four well known kernels, their corresponding densities, and their decay class.

The following proposition specifics how to set $\mathbf{U}$ based on the decay class and the maximum value of φ.

Proposition 8 *Suppose that, M_R is such that for every $\boldsymbol{\eta} \in \mathbb{R}^d$ and every $\mathbf{x} \in \mathcal{X}$ we have $|\varphi(\mathbf{x},\boldsymbol{\eta})| \le M_R$. Then, the following establishes a $\mathbf{U}^{(\min)} = (U_1^{(\min)}, \ldots, U_d^{(\min)})$ such that if $\mathbf{U} \ge \mathbf{U}^{(\min)}$ (where we interpret the inequality as entrywise) then we have*

$$\left|\int_{\mathbb{R}^d} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta} - \int_{\mathcal{Q}_\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta}\right| \le \frac{1}{2\sigma_f^2 n} \tag{19}$$

for every $\mathbf{v}$ such that $\mathbf{v}^{\mathrm{T}}\mathbf{K}_{\boldsymbol{\theta}}\mathbf{v} = 1$.

Table 2: Example of decay classes for a few well known kernel function. In the table below, $\mathbf{L}$ is a diagonal matrix with non-negative diagonal entries.

	$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}')$	$p(\boldsymbol{\eta};\boldsymbol{\theta}_0)$	Decay Class
Non-isotropic Gaussian	$\exp(-\|\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\|_2^2//2)$	$(2\pi)^{-d/2}\prod_{k=1}^{d}\ell_k\exp(-\|\mathbf{L}\boldsymbol{\eta}\|_2^2)$ $L_k=\ell_k/\sqrt{2}$	$\mathcal{E}^{(2)}_{C,\mathbf{L}}$, $C=(2\pi)^{-d/2}\prod_{k=1}^{d}\ell_k$
Non-isotropic Cauchy	$2^d\prod_{k=1}^{d}\frac{\ell_k}{\ell_k^2+(\mathbf{x}-\mathbf{x}')_k^2}$	$\exp(-\|\mathbf{L}\boldsymbol{\eta}\|_1)$ $L_k=\ell_k$	$\mathcal{E}^{(1)}_{C,\mathbf{L}}$, $C=1$
Non-isotropic Laplacian	$\exp(-\|\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\|_1)$	$\pi^{-d}\prod_{k=1}^{d}\frac{\ell_k}{1+\ell_k^2\eta_k^2}$ $L_k=\ell_k$	$\mathcal{P}_{C,\mathbf{L}}$, $C=\pi^{-d}\prod_{k=1}^{d}\ell_k$
Matèrn	$\frac{2^{1-\nu}}{\Gamma(\nu)}(\sqrt{2\nu}\|\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\|_2)^{\nu}\cdot K_\nu(\sqrt{2\nu}\|\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\|_2)$	$\frac{\Gamma(\nu+d/2)}{\pi^{d/2}\Gamma(\nu)(2\nu)^{d/2}}\prod_{k=1}^{d}\ell_k\left(1+\|\mathbf{L}\boldsymbol{\eta}\|_2^2\right)^{-(\nu+d/2)}$ $L_k=\ell_k/\sqrt{2\nu}$	$\mathcal{P}^{(r)}_{C,\mathbf{L}}$ $r=\nu+d/2$ $C=\frac{\Gamma(\nu+d/2)}{\Gamma(\nu)(2\pi\nu)^{d/2}}\prod_{k=1}^{d}\ell_k$

1. *If* $p(\cdot;\boldsymbol{\theta}_0)\in\mathcal{P}_{C,\mathbf{L}}$, $U_k^{(\min)}\coloneqq\frac{1}{L_k}\cot\left(L_k\left(\frac{4CM_R^2\sigma_f^2n^2}{\sigma_n^2}\right)^{-1/d}\right)$.

2. *If* $p(\cdot;\boldsymbol{\theta}_0)\in\mathcal{P}^{(r)}_{C,\mathbf{L}}$ *and* $r>d/2$*:*

   (a) *for* $d=2$, *set* $U_k^{(\min)}\coloneqq\frac{1}{L_k}\sqrt{\sqrt[r-1]{\frac{\pi CM_R^2\sigma_f^2n^2}{(r-1)\sigma_n^2L_1L_2}}-1}$.

   (b) *for any* $d\neq2$, *let* $x>0$ *be the solution to the equation*

$$\frac{\pi^{d/2}CnM_R^2}{2^{d-2}\Gamma\left(\frac{d}{2}\right)(2r-d)\sigma_n^2\prod_{k=1}^{d}L_k}x^{d-2r}\left|{}_2F_1\left(r-d/2,r;r-d/2+1;-x^{-2}\right)\right|=\frac{1}{2\sigma_f^2n}\,. \quad (20)$$

   (${}_2F_1$ *is the hypergeometric function). Then set* $U_k^{(\min)}\coloneqq x/L_k$. *We have*

$$U_k^{(\min)}\leq\frac{1}{L_k}\left(\frac{\pi^{d/2}CM_R^2\sigma_f^2n^2}{2^{d-2}\Gamma\left(\frac{d}{2}\right)(2r-d)\sigma_n^2\prod_{k=1}^{d}L_k}\right)^{1/(2r-d)}.$$

3. *If* $p(\cdot;\boldsymbol{\theta}_0)\in\mathcal{E}^{(1)}_{C,\mathbf{L}}$, $U_k^{(\min)}\coloneqq\frac{1}{L_k}\ln\left(\frac{1}{L_k}\left(\frac{4CM_R^2\sigma_f^2n^2}{\sigma_n^2}\right)^{1/d}\right)$.

4. *If* $p(\cdot;\boldsymbol{\theta}_0)\in\mathcal{E}^{(2)}_{C,\mathbf{L}}$, $U_k^{(\min)}\coloneqq\frac{1}{L_k}\sqrt{\ln\left(\frac{\sqrt{\pi}}{L_k}\left(\frac{2^{2-d}CM_R^2\sigma_f^2n^2}{\sigma_n^2}\right)^{1/d}\right)}$.

Remark 9 *We recommend to set* $\mathbf{U}$ *to* $\mathbf{U}^{(\min)}$. *For* $\mathcal{P}_{C,\mathbf{L}}$, $\mathcal{E}^{(1)}_{C,\mathbf{L}}$ *and* $\mathcal{E}^{(2)}_{C,\mathbf{L}}$ *we give explicit formulas for* $\mathbf{U}^{(\min)}$. *For* $\mathcal{P}^{(r)}_{C,\mathbf{L}}$, *it is defined implicitly as the solution to a nonlinear equation. We recommend finding the solution numerically using root-finding methods. We also give an explicit upper bound for the value of* $\mathbf{U}^{(\min)}$, *which can be used if one wishes to avoid solving a non-linear equation, however, those upper bounds tend to be loose. Nevertheless, the upper bound is used later to derive asymptotic bounds on* s.

Proof First, since $\mathbf{v}^{\mathrm{T}}\mathbf{K}_{\boldsymbol{\theta}}\mathbf{v} = 1$, we have

$$\begin{aligned}
|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 &= \left|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{K}_{\boldsymbol{\theta}}^{-1/2}\mathbf{K}_{\boldsymbol{\theta}}^{1/2}\mathbf{v}\right|^2 \\
&\leq (\mathbf{z}(\boldsymbol{\eta})^*\mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{z}(\boldsymbol{\eta}))\cdot(\mathbf{v}^{\mathrm{T}}\mathbf{K}_{\boldsymbol{\theta}}\mathbf{v}) \\
&= \mathbf{z}(\boldsymbol{\eta})^*\mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{z}(\boldsymbol{\eta}) \\
&\leq \sigma_n^{-2}\|\mathbf{z}(\boldsymbol{\eta})\|_2^2 \\
&\leq n\sigma_n^{-2}M_R^2
\end{aligned} \tag{21}$$

where the first inequality is due to the Cauchy-Schwartz inequality, the second inequality follows from observing that the smallest eigenvalue of $\mathbf{K}_{\boldsymbol{\theta}}$ is bigger than or equal to σ_n^2, and the last inequality is due to the fact that every entry in $\mathbf{z}(\boldsymbol{\eta})$ has absolute value that is smaller or equal to M_R.

The case of $p \in \mathcal{P}_{C,\mathbf{L}}$: In this case,

$$\begin{aligned}
\left|\int_{\mathbb{R}^d}|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta} - \int_{\mathcal{Q}_U}|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta}\right| &= \left|\int_{|\boldsymbol{\eta}|\geq\mathbf{U}}|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta}\right| \\
&\leq \frac{2CnM_R^2}{\sigma_n^2}\left|\int_{U_1}^{\infty}\cdots\int_{U_d}^{\infty}\prod_{k=1}^{d}\frac{1}{1+L_k^2\eta_k^2}d\eta_1\cdots d\eta_d\right| \\
&= \frac{2CnM_R^2}{\sigma_n^2}\prod_{k=1}^{d}\left|\int_{U_k}^{\infty}\frac{1}{1+L_k^2\eta_k^2}d\eta_k\right| \\
&= \frac{2CnM_R^2}{\sigma_n^2}\prod_{k=1}^{d}\frac{1}{L_k}\left(\frac{\pi}{2}-\arctan(L_kU_k)\right).
\end{aligned}$$

So, in order for Eq. (19) to hold, we set

$$U_k^{(\min)} = \frac{1}{L_k}\tan\left(\frac{\pi}{2} - L_k\left(\frac{4CM_R^2\sigma_f^2n^2}{\sigma_n^2}\right)^{-1/d}\right) = \frac{1}{L_k}\cot\left(L_k\left(\frac{4CM_R^2\sigma_f^2n^2}{\sigma_n^2}\right)^{-1/d}\right)$$

The case of $p \in \mathcal{P}_{C,\mathbf{L}}^{(r)}$: In this case,

$$\begin{aligned}
\left|\int_{\mathbb{R}^d}\left|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}\right|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta} - \int_{\mathcal{Q}_{\mathbf{U}}}\left|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}\right|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta}\right| &= \left|\int_{|\boldsymbol{\eta}|\geq\mathbf{U}}|\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta}\right| \\
&\leq \frac{2CnM_R^2}{\sigma_n^2}\left|\int_{U_1}^{\infty}\cdots\int_{U_d}^{\infty}\left(1+L_1^2\eta_1^2+\ldots+L_d^2\eta_d^2\right)^{-r}d\eta_1\cdots d\eta_d\right| \\
&= \frac{2CnM_R^2}{\sigma_n^2\prod_{k=1}^{d}L_k}\left|\int_{L_1U_1}^{\infty}\cdots\int_{L_dU_d}^{\infty}\left(1+\eta_1^2+\ldots+\eta_d^2\right)^{-r}d\eta_1\cdots d\eta_d\right| \\
&\leq \frac{2CnM_R^2}{\sigma_n^2\prod_{k=1}^{d}L_k}\left|\int_0^{\pi/2}\int_m^{\infty}\int_0^{\pi/2}\cdots\int_0^{\pi/2}\left(\frac{t^{d-1}}{(1+t^2)^r}\prod_{j=1}^{d-2}\sin^j\phi_{d-1-j}\right)d\phi_1\cdots d\phi_{d-2}dtd\theta\right| \\
&= \frac{2CnM_R^2}{\sigma_n^2\prod_{k=1}^{d}L_k}\cdot\frac{\pi}{2}\left|\int_m^{\infty}\frac{t^{d-1}}{(1+t^2)^r}\prod_{j=1}^{d-2}\left(\int_0^{\pi/2}\sin^j\phi_{d-1-j}d\phi_{d-1-j}\right)dt\right| \\
&= \frac{\pi^{d/2}CnM_R^2}{2^{d-2}\Gamma\left(\frac{d}{2}\right)\sigma_n^2\prod_{k=1}^{d}L_k}\cdot\left|\int_m^{\infty}\frac{t^{d-1}}{(1+t^2)^r}dt\right| \\
&= \frac{\pi^{d/2}CnM_R^2}{2^{d-1}\Gamma\left(\frac{d}{2}\right)\sigma_n^2\prod_{k=1}^{d}L_k}\cdot\left|\int_{m^2}^{\infty}\frac{t^{d/2-1}}{(1+t)^r}dt\right|
\end{aligned} \tag{22}$$

where $m = \min_{1\le k\le d}\{L_k U_k\}$, and in the second inequality we use d-dimensional ($d \ge 2$) spherical coordinates (Blumenson, 1960):

$$\begin{aligned}
\eta_1 &= t\cos\phi_1 \\
2 \le k \le d-2:\ \eta_k &= t\cos\phi_k \prod_{j=1}^{k-1}\sin\phi_j \\
\eta_{d-1} &= t\sin\theta \prod_{j=1}^{d-2}\sin\phi_j \\
\eta_d &= t\cos\theta \prod_{j=1}^{d-2}\sin\phi_j
\end{aligned}$$

where $\|\boldsymbol{\eta}\| = t$, $0 \le \phi_j \le \pi/2$, $0 \le \theta < \pi/2$, $m \le t < \infty$, and the Jacobian is

$$J = t^{d-1}\prod_{j=1}^{d-2}\sin^j \phi_{d-1-j}\,.$$

Also, in the fourth equality we use the following property of the beta function:

$$\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} = \mathrm{B}\left(x,y\right) = 2\int_0^{\pi/2}\left(\sin\phi\right)^{2x-1}\left(\cos\phi\right)^{2y-1}d\phi$$

with $y = 1/2$ and $x = (j+1)/2$, for any $j = 1,\ldots,d-2$, that is

$$\int_0^{\pi/2}\sin^j\phi_{d-1-j}d\phi_{d-1-j} = \frac{\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{j+1}{2}\right)}{2\Gamma\left(\frac{j+2}{2}\right)}$$

which implies that

$$\prod_{j=1}^{d-2}\left(\int_0^{\pi/2}\sin^j\phi_{d-1-j}d\phi_{d-1-j}\right) = \frac{\Gamma\left(\frac{1}{2}\right)^{d-2}}{2^{d-2}\Gamma\left(\frac{d}{2}\right)} = \frac{\pi^{d/2-1}}{2^{d-2}\Gamma\left(\frac{d}{2}\right)}\,.$$

Now, we write the last integral of (22) in terms of incomplete beta function, as follows:

$$\begin{aligned}
&\int_{m^2}^{\infty}\frac{t^{d/2-1}}{(1+t)^r}dt = \\
&\quad\left[t = \frac{\tilde{t}}{1-\tilde{t}}\right] = \int_{m^2/(1+m^2)}^{1}\tilde{t}^{d/2-1}\cdot(1-\tilde{t})^{r-d/2-1}d\tilde{t} \\
&\quad\left[\tilde{t} = \frac{1}{1-u}\right] = (-1)^{1+d/2-r}\int_{-m^{-2}}^{0}u^{r-d/2-1}(1-u)^{-r}du \\
&\qquad\qquad = (-1)^{d/2-r}\int_{0}^{-m^{-2}}u^{r-d/2-1}(1-u)^{-r}du \\
&\qquad\qquad = \frac{2m^{d-2r}}{2r-d}\,{}_2F_1\left(r-d/2,r;r-d/2+1;-m^{-2}\right)
\end{aligned}$$

The expression in the last equality is the analytic continuation of the beta function $\mathrm{B}_{-m^{-2}}\left(r-d/2,1-r\right)$ (Olver et al., 2010, Sections 8.17, 15.4), i.e.

$$\mathrm{B}_{-m^{-2}}\left(r-d/2,1-r\right) = \frac{2m^{d-2r}}{2r-d}\,{}_2F_1\left(r-d/2,r;r-d/2+1;-m^{-2}\right)\,.$$

Therefore, we obtain

$$\left|\int_{\mathbb{R}^d} |\mathbf{z}(\boldsymbol{\eta})\boldsymbol{\alpha}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta} - \int_{\mathcal{Q}_\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^*\boldsymbol{\alpha}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta}\right| \leq \frac{\pi^{d/2} CnM_R^2}{2^{d-2}\Gamma\left(\frac{d}{2}\right)(2r-d)\sigma_n^2 \prod_{k=1}^d L_k} m^{d-2r} \cdot \left|{}_2F_1\left(r-d/2, r; r-d/2+1; -m^{-2}\right)\right| .$$

In order for Eq. (19) to hold, let $x > 0$ be the solution of the equation

$$\frac{\pi^{d/2} CnM_R^2}{2^{d-2}\Gamma\left(\frac{d}{2}\right)(2r-d)\sigma_n^2 \prod_{k=1}^d L_k} x^{d-2r} \left|{}_2F_1\left(r-d/2, r; r-d/2+1; -x^{-2}\right)\right| = \frac{1}{2\sigma_f^2 n}$$

and then set $U_k^{(\min)} = x/L_k$.

Note that if we replace the expression after the second equality in (22) with the upper bound

$$\frac{2CnM_R^2}{\sigma_n^2 \prod_{k=1}^d L_k} \left|\int_{U_1}^\infty \cdots \int_{U_d}^\infty \left(\eta_1^2 + \ldots + \eta_d^2\right)^{-r} d\eta_1 \cdots d\eta_d\right|$$

then, in order for Eq. (19)to hold, we can set

$$U_k^{(\min)} = \frac{1}{L_k}\left(\frac{\pi^{d/2} CM_R^2 \sigma_f^2 n^2}{2^{d-2}(2r-d)\Gamma\left(\frac{d}{2}\right)\sigma_n^2 \prod_{k=1}^d L_k}\right)^{1/(2r-d)} .$$

Note also that for $d = 2$, we have the simplest case of spherical coordinates, which yields

$$U_k^{(\min)} = \frac{1}{L_k}\sqrt{\sqrt[r-1]{\frac{\pi C M_R^2 \sigma_f^2 n^2}{(r-1)\sigma_n^2 L_1 L_2}} - 1} .$$

Also, for $d = 1$ we do not need spherical coordinates. In that case, we have

$$\begin{aligned}\left|\int_{\mathbb{R}^d} |\mathbf{z}(\boldsymbol{\eta})\boldsymbol{\alpha}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta} - \int_{\mathcal{Q}_\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^*\boldsymbol{\alpha}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta}\right| &= \frac{2nCM_R^2}{\sigma_n^2}\left|\int_U^\infty \frac{1}{(1+L_1^2\eta^2)^r} d\eta\right| \\ &= \frac{nCM_R^2}{\sigma_n^2 L_1}\left|\int_{L_1^2U^2}^\infty \frac{s^{-1/2}}{(1+s)^r} ds\right| \\ &= \frac{2nCM_R^2 L_1^{1-2r} U^{1-2r}}{(2r-1)\sigma_n^2 L_1}\left|{}_2F_1\left(r-1/2, r; r+1/2; -L_1^{-2}U^{-2}\right)\right| .\end{aligned}$$

Now, $U^{(\min)}$ that equates the last expression with $1/2\sigma_f^2 n$ is obtained by solving the same equation obtained for $d \geq 3$, only with $d = 1$.

Note that if we bound the first integral similarly to the bound in the case $d \geq 3$, we obtain the same formula for $U^{(\min)}$ only with $d = 1$.

The case of $p \in \mathcal{E}_{C,\mathbf{L}}^{(1)}$: In this case,

$$\begin{aligned}\left|\int_{\mathbb{R}^d} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta} - \int_{\mathcal{Q}_U} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta}\right| &= \left|\int_{|\boldsymbol{\eta}|\geq\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta}\right| \\ &\leq \frac{2CnM_R^2}{\sigma_n^2}\left|\int_{U_1}^\infty e^{-L_1\eta_1} d\eta_1 \cdots \int_{U_d}^\infty e^{-L_d\eta_d} d\eta_d\right| \\ &= \frac{2CnM_R^2}{\sigma_n^2}\prod_{k=1}^d \left|\int_{U_k}^\infty e^{-L_k\eta_k} d\eta_k\right| \\ &= \frac{2CnM_R^2}{\sigma_n^2}\prod_{k=1}^d \frac{e^{-L_k U_k}}{L_k}\end{aligned}$$

So, in order for Eq. (19) to hold we set

$$U_k^{(\min)} = \frac{1}{L_k} \ln \left(\frac{1}{L_k} \left(\frac{4CM_R^2 \sigma_f^2 n^2}{\sigma_n^2} \right)^{1/d} \right) .$$

The case of $p \in \mathcal{E}^{(2)}_{C,\mathbf{L}}$: In this case,

$$\begin{aligned}
\left| \int_{\mathbb{R}^d} |\mathbf{z}(\boldsymbol{\eta})^* \mathbf{v}|^2 p(\boldsymbol{\eta}) d\boldsymbol{\eta} - \int_{\mathcal{Q}_U} |\mathbf{z}(\boldsymbol{\eta})^* \mathbf{v}|^2 p(\boldsymbol{\eta}) d\boldsymbol{\eta} \right| &= \left| \int_{|\boldsymbol{\eta}| \geq \mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^* \mathbf{v}|^2 p(\boldsymbol{\eta}) d\boldsymbol{\eta} \right| \\
&\leq \frac{2CnM_R^2}{\sigma_n^2} \left| \int_{U_1}^{\infty} e^{-\frac{L_1^2 \eta_1^2}{2}} d\eta_1 \cdots \int_{U_d}^{\infty} e^{-\frac{L_d^2 \eta_d^2}{2}} d\eta_d \right| \\
&= \frac{\pi^{d/2} CnM_R^2}{2^{d/2-1} \sigma_n^2} \prod_{k=1}^{d} \frac{\mathbf{erfc}\left(\frac{L_k U_k}{\sqrt{2}} \right)}{L_k} \\
&\leq \frac{\pi^{d/2} CnM_R^2}{2^{d/2-1} \sigma_n^2} \prod_{k=1}^{d} \frac{e^{-\frac{L_k^2 U_k^2}{2}}}{L_k}
\end{aligned}$$

where we used the bound $\mathbf{erfc}\,(x) \leq e^{-x^2}$ for non-negative x, which follows from (Craig, 1991). In this paper it was shown that for any non-negative x

$$\mathbf{erfc}\,(x) = \frac{2}{\pi} \int_0^{\frac{\pi}{2}} e^{-\frac{x^2}{\sin^2 \theta}} d\theta$$

and therefore $\mathbf{erfc}\,(x) = \frac{2}{\pi} \int_0^{\frac{\pi}{2}} e^{-\frac{x^2}{\sin^2 \theta}} d\theta \leq \frac{2}{\pi} \int_0^{\frac{\pi}{2}} e^{-x^2} d\theta = e^{-x^2}$. So, in order for Eq. (19) to hold we set

$$U_k^{(\min)} = \frac{1}{L_k} \sqrt{2 \ln \left(\frac{\sqrt{\pi}}{L_k} \left(\frac{2^{2-d/2} CM_R^2 \sigma_f^2 n^2}{\sigma_n^2} \right)^{1/d} \right)} .$$

■

5.3 Approximating the Truncated Integral

The nodes $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ and their weights $h_1(\boldsymbol{\theta}_0), \ldots, h_s(\boldsymbol{\theta}_0)$ are simply rescaled multivariate Gauss-Legendre quadrature nodes and weights[3], and so $\sum_{j=1}^{s} h_j(\boldsymbol{\theta}_0) |\mathbf{z}(\boldsymbol{\eta}_j)^* \mathbf{v}|^2$ is a quadrature approximation of $\int_{\mathcal{Q}_\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^* \mathbf{v}|^2 p(\boldsymbol{\eta}; \boldsymbol{\theta}_0) d\boldsymbol{\eta}$. In this section we derive a lower bound on $\mathbf{s}$ that guarantees that

$$\left| \int_{\mathcal{Q}_\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^* \mathbf{v}|^2 p(\boldsymbol{\eta}; \boldsymbol{\theta}_0) d\boldsymbol{\eta} - \sum_{j=1}^{s} h_j(\boldsymbol{\theta}_0) |\mathbf{z}(\boldsymbol{\eta}_j)^* \mathbf{v}|^2 \right| \leq \frac{1}{2\sigma_f^2 n} .$$

The bound on $\mathbf{s}$ depends on $\mathbf{U}$. Together with Proposition 8, we completely specify how to build the quadrature approximation so that Eq. (17) holds.

3. Gauss-Legendre quadrature is defined for one dimensional integrals. In multivariate Gauss-Legendre quadrature we refer to the quadrature obtained by tensorizing the one-dimensional quadrature.

5.3.1 Decay of Chebyshev Coefficients for Multivariate Functions

Our analysis relies on generalizations of existing decay bounds for Chebyshev expansions of analytic functions in one dimension to multivariate functions. In this subsection, we introduce these results.

Classical decay bounds for Chebyshev expansions of analytic functions in one dimension are based on bounding the function values on the Bernstein ellipse (see (Mason and Handscomb, 2002, Section 1.4) for further details). For multivariate functions, a polyellipse is used instead.

Definition 10 *A* Bernstein ellipse *is an open region in the complex plane which bounded by an ellipse with foci* ± 1*. A* Bernstein polyellipse in d-dimensions *is a cartesian product of* d *Bernstein ellipses.*

Let $\rho(U,\beta) \coloneqq \beta/(2U) + \sqrt{\beta^2/(4U^2)+1}$*. Given* $\mathbf{U} = (U_1,\ldots,U_d) > 0$ *and a singularity point* $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_d) > 0$*, denote.*

$$E_{\mathbf{U},\boldsymbol{\beta}} \coloneqq \left\{ \mathbf{z} \in \mathbb{C}^d : \left| z_k + \sqrt{z_k^2 - U_k^2} \right| < U_k \rho(U_k,\beta_k) \qquad \forall k = 1,\ldots,d \right\}.$$

Note that $E_{1,\boldsymbol{\beta}\oslash\mathbf{U}}$*, where* $\boldsymbol{\beta} \oslash \mathbf{U}$ *denotes entrywise division between* $\boldsymbol{\beta}$ *and* $\mathbf{U}$*, is a Bernstein polyellipse, and that* $E_{\mathbf{U},\boldsymbol{\beta}} = \mathbf{U} \odot E_{1,\boldsymbol{\beta}\oslash\mathbf{U}}$*. So,* $E_{\mathbf{U},\boldsymbol{\beta}}$ *is a polyellipse with foci at* $\pm U_k$*.*

For a multivariate analytic function f *on* $[-1,1]^d$*, the multivariate tensorised Chebyshev expansion is given by*

$$f(\mathbf{x}) = \sum_{j_1,\ldots,j_d=0}^{\infty} a_{j_1\ldots j_d} T_{j_1}(x_1)\cdots T_{j_d}(x_d)$$

where the coefficients are given by

$$a_{j_1\ldots j_d} = \frac{2^{d-m}}{\pi^d} \int_{\mathcal{Q}_1} \frac{f(x_1,\ldots,x_d)\, T_{j_1}(x_1)\cdots T_{j_d}(x_d)}{\sqrt{1-x_1^2}\cdots\sqrt{1-x_d^2}} dx_1\cdots dx_d$$

where $m \coloneqq \#\{j_k : j_k = 0\}$*.*

The following is a generalization of classical results for one dimension (Trefethen, 2013, Theorem 8.1, Theorem 8.2):

Theorem 11 *Let* f *be an analytic function on* $[-1,1]^d$ *and analytically continuable to* $E_{1,\boldsymbol{\beta}\oslash\mathbf{U}}$ *where it satisfies* $|f(x_1,\ldots,x_d)| \le M$ *for some* $M > 0$*. Then, for all* $j_1,\ldots,j_d$*,*

$$|a_{j_1\ldots j_d}| \le \frac{2^{d-m} M}{\rho_1^{j_1}\cdots\rho_d^{j_d}}.$$

Although Theorem 11 has essentially been proven in (Wang and Zhang, 2020), for completeness we include in Appendix B our proof of Theorem 11 which is based on a different technique.

5.3.2 Bounding the Integration Error

We have the following result:

Theorem 12 *Given* $\mathbf{U} = (U_1,\ldots,U_d)$ *such that* $U_1,\ldots,U_d > 0$*, and* $\boldsymbol{\beta}$ *such that* $\beta_1,\ldots,\beta_d > 0$*, let* $E_{\mathbf{U},\boldsymbol{\beta}}$ *be the polyellipse such that in dimension* j *the foci is* $\pm U_j$ *and passes through* $i\beta_j/2$*, and let* $\boldsymbol{\rho} = (\rho_1,\ldots,\rho_d)$ *with* $\rho_j \coloneqq \rho(U_j,\beta_j)$ *(for* $j = 1,\ldots,d$*) denote the sum of the semi-axes in each dimension. Assume that either* $p(\cdot;\boldsymbol{\theta}_0) \in \mathcal{P}_{C,\mathbf{L}}$ *or* $p(\cdot;\boldsymbol{\theta}_0) \in \mathcal{P}^{(r)}_{C,\mathbf{L}}$ *where* $r > d/2$*, or* $p \in \mathcal{E}^{(1)}_{C,\mathbf{L}}$ *or* $p \in \mathcal{E}^{(2)}_{C,\mathbf{L}}$*. Furthermore, assume that if* $\mathbf{z}(\boldsymbol{\eta}) = \mathbf{a}(\boldsymbol{\eta}) + i\mathbf{b}(\boldsymbol{\eta})$*, then for each* $1 \le j \le n$ *the functions*

$\mathbf{a}_j, \mathbf{b}_j$ *are analytic on* $\mathbb{R}^d$. *Finally, assume that* $p(\cdot;\boldsymbol{\theta}_0)$ *has an analytic continuation* $\hat{p}(\cdot;\boldsymbol{\theta}_0)$ *to* $E_{\mathbf{U},\boldsymbol{\beta}}$. *Let* $\hat{\mathbf{z}}(\cdot)$ *denote the analytic continuation of* $\mathbf{z}(\cdot)$. *Denote*

$$\begin{aligned} M_R &\coloneqq \sup_{\boldsymbol{\eta}\in\mathbb{R}^d} \|\mathbf{z}(\boldsymbol{\eta})\|_\infty \\ M_{\mathbf{U},\boldsymbol{\beta}} &\coloneqq \sup_{\boldsymbol{\eta}\in E_{\mathbf{U},\boldsymbol{\beta}}} \|\hat{\mathbf{z}}(\boldsymbol{\eta})\|_\infty \\ C_{\mathbf{U},\boldsymbol{\beta}} &\coloneqq \sup_{\boldsymbol{\eta}\in E_{\mathbf{U},\boldsymbol{\beta}}} |\hat{p}(\boldsymbol{\eta};\boldsymbol{\theta}_0)|\,. \end{aligned}$$

Then, for

$$s_k \geq \frac{\frac{1}{d}\ln\left(2^{2d+2}M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}\sigma_n^{-2}\sigma_f^2 n^2\right) + \ln U_k - \ln(\rho_k - 1)}{2\ln\rho_k} + 1,\ k = 1,\ldots,d,$$

we have

$$\left|\int_{\mathcal{Q}_\mathbf{U}} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta} - \sum_{j=1}^{s} h_j(\boldsymbol{\theta}_0)|\mathbf{z}(\boldsymbol{\eta}_j)^*\mathbf{v}|^2\right| \leq \frac{1}{2\sigma_f^2 n}\,.$$

Note that in that case

$$s = \prod_{k=1}^{d} s_k = O\left((2d)^{-d}\prod_{k=1}^{d}\frac{\ln\left(2^{2d+2}M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}U_k^d\sigma_n^{-2}\sigma_f^2 n^2\right) - d\ln(\rho_k - 1)}{\ln\rho_k}\right)$$

Proof For conciseness, we drop $\boldsymbol{\theta}_0$ from p throughout the proof. For convenience, we use the following form of the quadrature rule

$$\sum_{j=1}^{s} h_j(\boldsymbol{\theta}_0)|\mathbf{z}(\boldsymbol{\eta}_j)^*\mathbf{v}|^2 = \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d} w_{j_1\cdots j_d} p(\boldsymbol{\eta}_{j_1\cdots j_d})|\mathbf{z}(\boldsymbol{\eta}_{j_1\cdots j_d})^*\mathbf{v}|^2$$

as presented in Section 4.

Denote $f_\mathbf{v}(\boldsymbol{\eta}) = |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2$. Also denote $\tilde{p}(\boldsymbol{\chi}) \coloneqq p(U_1\chi_1,\ldots,U_d\chi_d)$ and $\tilde{f}_\mathbf{v}(\boldsymbol{\chi}) \coloneqq f_\mathbf{v}(U_1\chi_1,\ldots,U_d\chi_d)$. Note that

$$\left|\int_{\mathcal{Q}_\mathbf{U}} f_\mathbf{v}(\boldsymbol{\eta})p(\boldsymbol{\eta})d\boldsymbol{\eta} - \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d} w_{j_1\ldots j_d} p(\boldsymbol{\eta}_{j_1\cdots j_d}) f_\mathbf{v}(\boldsymbol{\eta}_{j_1\ldots j_d})\right| =$$

$$\left(\prod_{k=1}^{d} U_k\right)\left|\int_{[-1,1]^d} \tilde{f}_\mathbf{v}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi})d\boldsymbol{\chi} - \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d}\tilde{w}_{j_1\ldots j_d}\tilde{p}(\boldsymbol{\chi}_{j_1\ldots j_d})\tilde{f}_\mathbf{v}(\boldsymbol{\chi}_{j_1\ldots j_d})\right|$$

where $\tilde{w}_{j_1\ldots j_d} = \prod_{k=1}^{d} w_{j_k}^{(s_k)}$. The sum in the right-hand side is a quadrature approximation of $\tilde{f}_\mathbf{v}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi})$, which we analyze.

To that end, we first bound the analytic continuation of $\tilde{f}_\mathbf{v}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi})$ on $E_{1,\boldsymbol{\beta}\oslash\mathbf{U}}$ (where $\boldsymbol{\beta}\oslash\mathbf{U}$ denotes entrywise division). For every $\boldsymbol{\eta}\in E_{\mathbf{U},\boldsymbol{\beta}}$ we have (similar to the derivation of Eq. (21)):

$$|\hat{f}_\mathbf{v}(\boldsymbol{\eta})| = \left||\hat{\mathbf{z}}(\boldsymbol{\eta})^*\mathbf{v}|^2\right| \leq \sigma_n^{-2}\|\hat{\mathbf{z}}(\boldsymbol{\eta})\|_2^2 \leq n\sigma_n^{-2}M_{\mathbf{U},\boldsymbol{\beta}}^2\,.$$

Thus, $|\hat{f}_\mathbf{v}(\boldsymbol{\eta})\hat{p}(\boldsymbol{\eta})| \leq n\sigma_n^{-2}M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}$ and $|\hat{\tilde{f}}_\mathbf{v}(\boldsymbol{\chi})\hat{\tilde{p}}(\boldsymbol{\chi})| \leq n\sigma_n^{-2}M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}$. We can now apply quadrature approximation bounds on $\tilde{f}_\mathbf{v}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi})$ to bound the error

$$e_s \coloneqq \left|\int_{[-1,1]^d}\tilde{f}_\mathbf{v}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi})d\boldsymbol{\chi} - \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d}\tilde{w}_{j_1\ldots j_d}\tilde{f}_\mathbf{v}(\boldsymbol{\chi}_{j_1\ldots j_d})\tilde{p}(\boldsymbol{\chi}_{j_1\ldots j_d})\right|.$$

Let

$$\tilde{f}_{\mathbf{v}}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi}) = \sum_{k_1,\dots,k_d=0}^{\infty} a_{k_1\dots k_d} T_{k_1}(\chi_1)\cdots T_{k_d}(\chi_d)$$

be the multivariate Chebyshev expansion of $\tilde{f}_{\mathbf{v}}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi})$, and let P_{2s-1} be the truncated expansion:

$$P_{2s-1}(\boldsymbol{\chi}) \coloneqq \sum_{k_1=0}^{2s_1-1} \cdots \sum_{k_d=0}^{2s_d-1} a_{k_1\dots k_d} T_{k_1}(\chi_1)\cdots T_{k_d}(\chi_d)\,.$$

Similarly to the strategy employed in (Ubaru et al., 2017) and (Trefethen, 2013, Theorem 19.3), we have

$$\begin{aligned}
e_s = &\left| \int_{[-1,1]^d} \tilde{f}_{\mathbf{v}}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi})d\boldsymbol{\chi} - \int_{[-1,1]^d} P_{2s-1}(\boldsymbol{\chi})d\boldsymbol{\chi} + \right. \\
&\left. \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d} \left(P_{2s-1}(\boldsymbol{\chi}_{j_1\dots j_d})\tilde{w}_{j_1\dots j_d}\right) - \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d} \left(\tilde{f}_{\mathbf{v}}(\boldsymbol{\chi}_{j_1\dots j_d})\tilde{p}(\boldsymbol{\chi}_{j_1\dots j_d})\tilde{w}_{j_1\dots j_d}\right) \right| \\
= &\left| \int_{[-1,1]^d} \left(\tilde{f}_{\mathbf{v}}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi}) - P_{2s-1}(\boldsymbol{\chi})\right) d\boldsymbol{\chi} - \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d} \left(\tilde{f}_{\mathbf{v}}(\boldsymbol{\chi}_{j_1\dots j_d})\tilde{p}(\boldsymbol{\chi}_{j_1\dots j_d}) - P_{2s-1}(\boldsymbol{\chi}_{j_1\dots j_d})\right) \tilde{w}_{j_1\dots j_d} \right| \\
= &\left| \int_{-1}^{1}\cdots\int_{-1}^{1} \sum_{k_1=2s_1}^{\infty}\cdots\sum_{k_d=2s_d}^{\infty} a_{k_1\dots k_d} T_{k_1}(\chi_1)\cdots T_{k_d}(\chi_d)d\chi_1\cdots d\chi_d - \right. \\
&\left. \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d} \left(\sum_{k_1=2s_1}^{\infty}\cdots\sum_{k_d=2s_d}^{\infty} a_{k_1\dots k_d} T_{k_1}(\chi_{j_1}^{(s_1)})\cdots T_{k_d}(\chi_{j_d}^{(s_d)}) \right) \tilde{w}_{j_1\dots j_d} \right| \\
\le &\sum_{k_1=2s_1}^{\infty}\cdots\sum_{k_d=2s_d}^{\infty} |a_{k_1\dots k_d}| \left[\prod_{m=1}^{d} \int_{-1}^{1} |T_{k_m}(\chi_m)|\, d\chi_m + \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d} w_{j_1}^{(s_1)}\cdots w_{j_d}^{(s_d)} \left|T_{k_1}(\chi_{j_1}^{(s_1)})\right|\cdots\left|T_{k_d}(\chi_{j_d}^{(s_d)})\right| \right] \\
\le &\sum_{k_1=2s_1}^{\infty}\cdots\sum_{k_d=2s_d}^{\infty} |a_{k_1\dots k_d}| \left[2^d + \left(\sum_{j_1=1}^{s_1} w_{j_1}^{(s_1)}\right)\cdots\left(\sum_{j_d=1}^{s_d} w_{j_d}^{(s_d)}\right) \right] \\
\le &\sum_{k_1=2s_1}^{\infty}\cdots\sum_{k_d=2s_d}^{\infty} |a_{k_1\dots k_d}| \left[2^d + 2^d\right] \\
\le &\frac{2^{2d+1} n M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}}{\sigma_n^2} \sum_{k_1=2s_1}^{\infty}\cdots\sum_{k_d=2s_d}^{\infty} \frac{1}{\rho_1^{k_1}\cdots\rho_d^{k_d}} \\
= &\frac{2^{2d+1} n M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}}{\sigma_n^2} \cdot \prod_{k=1}^{d} \frac{1}{\rho_k^{2s_k-1}(\rho_k-1)}
\end{aligned}$$

where we use the bound

$$\tilde{f}_{\mathbf{v}}(\boldsymbol{\chi})\tilde{p}(\boldsymbol{\chi}) \le \frac{n M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}}{\sigma_n^2}\,.$$

In the first equality, we use the following equality

$$\int_{[-1,1]^d} P_{2s-1}(\boldsymbol{\chi})d\boldsymbol{\chi} = \sum_{j_1=1}^{s_1}\cdots\sum_{j_d=1}^{s_d} \tilde{w}_{j_1\dots j_d} P_{2s-1}(\boldsymbol{\chi}_{j_1\dots j_d})$$

which follows from the exactness of the Gauss-Legendre quadrature in one dimension:

$$\begin{aligned}
\int_{[-1,1]^d} P_{2s-1}(\boldsymbol{\chi})d\boldsymbol{\chi} &= \int_{[-1,1]^d} \sum_{k_1=0}^{2s_1-1} \cdots \sum_{k_d=0}^{2s_d-1} a_{k_1\ldots k_d} T_{k_1}(\chi_1)\cdots T_{k_d}(\chi_d)d\chi_1\cdots d\chi_d \\
&= \sum_{k_1=0}^{2s_1-1} \cdots \sum_{k_d=0}^{2s_d-1} a_{k_1\ldots k_d} \int_{[-1,1]^d} T_{k_1}(\chi_1)\cdots T_{k_d}(\chi_d)d\chi_1\cdots d\chi_d \\
&= \sum_{k_1=0}^{2s_1-1} \cdots \sum_{k_d=0}^{2s_d-1} a_{k_1\ldots k_d} \left(\int_{-1}^{1} T_{k_1}(\chi_1)d\chi_1\right)\cdots\left(\int_{-1}^{1} T_{k_d}(\chi_d)d\chi_d\right) \\
&= \sum_{k_1=0}^{2s_1-1} \cdots \sum_{k_d=0}^{2s_d-1} a_{k_1\ldots k_d} \left(\sum_{j_1=1}^{s_1} w_{j_1}^{(s_1)} T_{k_1}(\chi_{1,j_1})\right)\cdots\left(\sum_{j_d=1}^{s_d} w_{j_d}^{(s_d)} T_{k_d}(\chi_{d,j_d})\right) \\
&= \sum_{j_1=1}^{s_1} \cdots \sum_{j_d=1}^{s_d} w_{j_1}^{(s_1)}\cdots w_{j_d}^{(s_d)} \left(\sum_{k_1=0}^{2s_1-1} \cdots \sum_{k_d=0}^{2s_d-1} a_{k_1\ldots k_d} T_{k_1}(\chi_{1,j_1})\cdots T_{k_d}(\chi_{d,j_d})\right) \\
&= \sum_{j_1=1}^{s_1} \cdots \sum_{j_d=1}^{s_d} w_{j_1}^{(s_1)}\cdots w_{j_d}^{(s_d)} P_{2s-1}(\boldsymbol{\chi}_{j_1\ldots j_d})\,.
\end{aligned}$$

Hence,

$$\left|\int_{\mathcal{Q}_{\mathbf{U}}} |\mathbf{z}(\boldsymbol{\eta})^*\mathbf{v}|^2 p(\boldsymbol{\eta})d\boldsymbol{\eta} - \sum_{j_1=1}^{s_1} \cdots \sum_{j_d=1}^{s_d} w_{j_1\cdots j_d} |\mathbf{z}(\boldsymbol{\eta}_j)^*\mathbf{v}|^2\right| = e_s \prod_{k=1}^{d} U_k \le \frac{2^{2d+1} n M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}}{\sigma_n^2} \cdot \prod_{k=1}^{d} \frac{U_k}{\rho_k^{2s_k-1}(\rho_k-1)}\,.$$

Finally, bounding

$$\left(\frac{2^{2d+1} n M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}}}{\sigma_n^2}\right)^{1/d} \cdot \frac{U_k}{\rho_k^{2s_k-1}(\rho_k-1)} \le \frac{1}{\left(2\sigma_f^2 n\right)^{1/d}}$$

for each $k = 1,\ldots,d$, gives the bound from the theorem and the statement now follows immediately. ■

The last theorem allows us to compute the required $\mathbf{s}$ based on $\mathbf{U}$ and the singularities in $p(\cdot;\boldsymbol{\theta}_0)$. In Section 6, we show concrete examples for using Theorem 12 to bound $\mathbf{s}$ and $\mathbf{U}$. The main step is finding the polyellipse parameters. If the analytic extension of $p(\cdot;\boldsymbol{\theta}_0)$ has its first singularity at the pure imaginary value $\mathbf{x}_0 = \pm i\boldsymbol{\beta}$ for some $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_d)^{\mathrm{T}}$ such that $0 < \beta_k < U_k$, then $\tilde{p}(\cdot;\boldsymbol{\theta}_0)$ has its first singularity at $\tilde{\mathbf{x}}_0 = \pm i\boldsymbol{\beta} \oslash \mathbf{U}$. Thus, we can choose the ellipses parameters to be $\rho_k = \frac{\beta_k}{2U_k} + \sqrt{\frac{\beta_k^2}{4U_k^2}+1}$. Otherwise, we can choose $\beta_k = 2U_k$, i.e., $\rho_k = 1+\sqrt{2}$.

5.4 Handling a Parameter Domain Θ

Given a hyperparameter domain Θ, we want to set the parameters $\mathbf{U} = (U_1,\ldots,U_d)$ and $\mathbf{s} = (s_1,\ldots,s_d)$ such that Eq. (17) hold for every $\boldsymbol{\theta} \in \Theta$. This way, the parameterized family of positive definite kernel approximations $\{\tilde{k}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ given by Eq. (11) with these $\mathbf{U}$ and $\mathbf{s}$ is n-spectrally equivalent to $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ on the data domain $\mathcal{X}$. To do that, we need to find the worst-case (over $\boldsymbol{\theta} \in \Theta$) parameters $\mathbf{U}$ and $\mathbf{s}$. The following gives a general end-to-end statement.

Theorem 13 *Let*

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \sigma_f^2 \int_{\mathbb{R}^d} \varphi(\mathbf{x},\boldsymbol{\eta})\varphi(\mathbf{x}',\boldsymbol{\eta})^* p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta} + \sigma_n^2\gamma(\mathbf{x},\mathbf{x}')$$

be a parameterized family of kernels where $\boldsymbol{\theta} \in \Theta$. Suppose that:

1. *$|\varphi(\mathbf{x}, \boldsymbol{\eta})| \leq M_R$ for every $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\eta} \in \mathbb{R}^d$.*
2. *We set $\mathbf{U} \geq \sup_{\boldsymbol{\theta} \in \Theta} \mathbf{U}^{(\min)}(\boldsymbol{\theta})$ where $\mathbf{U}^{(\min)}(\boldsymbol{\theta})$ is the value set by Proposition 8 using parameters $\boldsymbol{\theta}$.*
3. *For every n points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ we set $\mathbf{z}(\boldsymbol{\eta}) = [\varphi(\mathbf{x}_1, \boldsymbol{\eta}), \ldots, \varphi(\mathbf{x}_n, \boldsymbol{\eta})]^{\mathrm{T}}$. If we write $\mathbf{z}(\boldsymbol{\eta}) = \mathbf{a}(\boldsymbol{\eta}) + i\mathbf{b}(\boldsymbol{\eta})$, then for each $1 \leq j \leq n$ the functions $\mathbf{a}_j, \mathbf{b}_j$ are analytic on $\mathbb{R}^d$.*
4. *$\boldsymbol{\beta}$ is such that $p(\cdot; \boldsymbol{\theta}_0)$ has an analytic continuation $\hat{p}(\cdot; \boldsymbol{\theta}_0)$ to $E_{\mathbf{U},\boldsymbol{\beta}}$ for all $\boldsymbol{\theta} \in \Theta$. Let $\hat{\mathbf{z}}(\cdot)$ denote the analytic continuation of $\mathbf{z}(\cdot)$.*

Let

$$M_{\mathbf{U},\boldsymbol{\beta}} \coloneqq \sup_{\boldsymbol{\eta} \in E_{\mathbf{U},\boldsymbol{\beta}}} \|\hat{\mathbf{z}}(\boldsymbol{\eta})\|_\infty$$
$$C_{\mathbf{U},\boldsymbol{\beta}} \coloneqq \sup_{\boldsymbol{\theta} \in \Theta,\, \boldsymbol{\eta} \in E_{\mathbf{U},\boldsymbol{\beta}}} |\hat{p}(\boldsymbol{\eta}; \boldsymbol{\theta}_0)| \ .$$

Then for

$$s_k \geq \frac{\frac{1}{d} \ln\left(2^{2d+2} M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}} \sigma_n^{-2} \sigma_f^2 n^2\right) + \ln U_k - \ln(\rho_k - 1)}{2 \ln \rho_k} + 1,\ k = 1, \ldots, d,$$

the parameterized family of kernel approximations given by Eq. (11) is n-spectrally equivalent to $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$ on the data domain $\mathcal{X}$. Furthermore, if we set $\mathbf{U} = \mathbf{U}^{(\min)}$ and $\mathbf{s}$ according the last lower bound we have

$$s = \prod_{k=1}^{d} s_k = O\left((2d)^{-d} \prod_{k=1}^{d} \frac{\ln\left(2^{2d+2} M_{\mathbf{U},\boldsymbol{\beta}}^2 C_{\mathbf{U},\boldsymbol{\beta}} U_k^d \sigma_n^{-2} \sigma_f^2 n^2\right) - d \ln(\rho_k - 1)}{\ln \rho_k}\right)$$

To use this theorem, one needs to bound the decay of the density functions $p(\cdot; \boldsymbol{\theta}_0)$ over Θ and calculate an upper bound on $C_{\mathbf{U},\boldsymbol{\beta}}$. In general, this might be hard, but luckily in most kernels display monotonicity in their hyperparameters that helps identify the worst case for $\boldsymbol{\theta}$ over Θ. For example, for the one dimensional Gaussian kernel $\exp(-\|\ell^{-1}(x - x')\|_2^2//2)$, the various parameters in the theorems monotonically increase as $\ell \to 0$. In the next section we given concrete examples for using Theorem 13 for the kernels listed in Table 2.

6. Examples of Feature Maps for Kernels

In this section, we show how to apply the theory presented in the previous section to design n-spectrally equivalent kernel approximations for a few widely used kernel functions. Throughout this section, we assume that n is fixed, the data domain is $\mathcal{X} \subseteq \mathbb{R}^d$, and the hyperparameter domain is Θ. Furthermore, we assume that we have a bounding box on the domain, i.e. $\mathcal{X} \subseteq \prod_{k=1}^{d} [-R_i/2, R_i/2]$ (obviously, such a bounding box can be easily computed from the input data). Let $\mathbf{R} = [R_1, \ldots, R_d]$.

6.1 Gaussian Kernel

Recall that the Gaussian kernel is

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\|\mathbf{x} - \mathbf{x}'\|_2^2 / 2\ell^2\right) + \sigma_n^2 \gamma(\mathbf{x} - \mathbf{x}')$$

($\boldsymbol{\theta} = [\ell, \sigma_f^2, \sigma_n^2]$) where we added a scaling factor σ_f^2 and included the ridge term $\sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$ in the kernel definition. Note that for conciseness, we consider the isotropic version; the formulas can be modified for the anisotropic case. As discussed in Section 4, by setting $\varphi(\mathbf{x}, \boldsymbol{\eta}) = e^{-i\mathbf{x}^{\mathrm{T}}\boldsymbol{\eta}}$, this kernel matches the form of Eq. (8). We assume that the hyperparameters are bounded as follows:

$$\Theta = \{[\ell, \sigma_n^2, \sigma_f^2] \,:\, \ell \geq \ell_0, \sigma_n^2 \geq \sigma_{n0}^2, \sigma_f^2 \leq \sigma_{f0}^2\}$$

where $\sigma_{n0}^2 > 0$ (i.e. we have a ridge term; our method is not able to approximate the Gaussian kernel in the absence of a ridge term).

The density is given by

$$p(\boldsymbol{\eta}; \ell) = \ell^d (2\pi)^{-d/2} \exp\left(-\|\boldsymbol{\eta}\|_2^2 \ell^2/2\right) .$$

Therefore, for $\boldsymbol{\theta} \in \Theta$, $p(\cdot;\ell) \in \mathcal{E}^{(2)}_{C,\mathbf{L}}$ with $C = \ell_0^d(2\pi)^{-d/2}$ and $\mathbf{L} = \frac{\ell_0}{\sqrt{2}}\mathbf{I}_d$. We also have $|\varphi(\mathbf{x}, \boldsymbol{\eta})| \leq 1$ for all $\mathbf{x}$ and $\boldsymbol{\eta}$, so we set $M_R = 1$. So, based on Proposition 8 we set

$$U_k = \frac{1}{\ell_0}\sqrt{2\ln\left(\left(\frac{2^{2-d}\sigma_{f0}^2 n^2}{\sigma_{n0}^2}\right)^{1/d}\right)} .$$

In addition, $p(\cdot;\ell)$ is analytic on $\mathbb{R}^d$, and in particular it is analytically continuable to the polyellipse $E_{\mathbf{U},\boldsymbol{\beta}}$ with $\beta_k = 2U_k$, $\rho_k = 1+\sqrt{2}$, as described in Theorem 12. Now we bound $p(\cdot;\ell_0)$ on the polyellipse as follows, for $x_k \in [-\sqrt{2}U_k, \sqrt{2}U_k]$, $y_k \in [-U_k, U_k]$:

$$\begin{aligned}
|p(\mathbf{x}+i\mathbf{y};\ell_0)| &= \left|\ell_0^d(2\pi)^{-d/2}\exp\left(-\ell_0^2/2\sum_{k=1}^d (x_k+iy_k)^2\right)\right| \\
&= \left|\ell_0^d(2\pi)^{-d/2}\exp\left(-\ell_0^2/2\sum_{k=1}^d x_k^2 - y_k^2 + 2ix_ky_k\right)\right| \\
&= \ell_0^d(2\pi)^{-d/2}\exp\left(\ell_0^2/2\sum_{k=1}^d y_k^2 - x_k^2\right) \\
&\leq \ell_0^d(2\pi)^{-d/2}\exp\left(\ell_0^2/2\sum_{k=1}^d y_k^2\right) \\
&\leq \ell_0^d(2\pi)^{-d/2}\exp\left(\ell_0^2/2\sum_{k=1}^d U_k^2\right) \\
&= \ell_0^d(2\pi)^{-d/2}\exp\left(\ell_0^2\|\mathbf{U}\|_2^2/2\right)
\end{aligned}$$

Hence, $C_{\mathbf{U},\boldsymbol{\beta}} = \ell_0^d(2\pi)^{-d/2}\exp\left(\ell_0^2\|\mathbf{U}\|_2^2/2\right)$.

Recall that $\mathbf{z} : \mathbb{R}^d \to \mathbb{C}^n$ is given by $\mathbf{z}(\boldsymbol{\eta})_j = e^{-i\boldsymbol{\eta}^{\mathrm{T}}\mathbf{x}_j}$, it is easy to verify that for every $\mathbf{v} \in \mathbb{R}^n$ the function $|\mathbf{z}(\cdot)^*\mathbf{v}|^2$ is an analytic function on $\mathbb{R}^d$. In particular, it is analytically continuable to the polyellipse $E_{\mathbf{U},\boldsymbol{\beta}}$ with $\beta_k = 2U_k$, $\rho_k = 1+\sqrt{2}$ as required by Theorem 12. Now we bound $|\mathbf{z}(\boldsymbol{\eta})_j|$ for each j on its corresponding ellipse. For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$: $|\mathbf{z}(\mathbf{x}+i\mathbf{y})_j| = |\mathbf{z}(i\mathbf{y})_j|$, so we need to bound $|\mathbf{z}(i\mathbf{y})_j|$ for $\mathbf{y} \in \prod_{k=1}^d [-U_k, U_k]$:

$$|\mathbf{z}(i\mathbf{y})_j| \quad = \quad \left|e^{-i\cdot i\mathbf{y}^{\mathrm{T}}\mathbf{x}_j}\right| = e^{\mathbf{y}^{\mathrm{T}}\mathbf{x}_j} \leq e^{\|\mathbf{y}\|_2\|\mathbf{x}_j\|_2} \leq e^{\|\mathbf{U}\|_2\|\mathbf{R}\|_2/2} .$$

Hence, $|\mathbf{z}(\boldsymbol{\eta})_j| \le e^{\|\mathbf{U}\|_2\|\mathbf{R}\|_2/2} =: M_{\mathbf{U},\boldsymbol{\beta}}$. Now, one can apply Theorem 13 with the these parameters and obtain that for

$$s_k \ge \frac{\frac{1}{d}\ln\left(2^{2d+2}\pi^{-d/2}\sigma_{n0}^{-2}\sigma_{f0}^2 n^2\right) + \frac{\ell_0^2}{2d}\|\mathbf{U}\|_2^2 + \frac{1}{d}\|\mathbf{U}\|_2\|\mathbf{R}\|_2 + \frac{1}{2}\ln\ln\left(\left(\frac{2^{2-d}\sigma_{f0}^2 n^2}{\sigma_{n0}^2}\right)^{1/d}\right) - \ln(\sqrt{2})}{2\ln(1+\sqrt{2})} + 1$$

we have the desired bound.

Since $\|\mathbf{U}\|_2 = O(\sqrt{\ln n})$ (in particular, $\|\mathbf{U}\|_2^2 = O(\ln n)$) and assuming the bounding box $\mathbf{R}$ is fixed then $s_k = O(\ln n)$ suffice and $s = \prod_{k=1}^d s_k = O((\ln n)^d)$ suffices.

6.2 Matèrn Kernel

Recall that the Matèrn kernel is

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}}{\ell}\|\mathbf{x}-\mathbf{x}'\|_2\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}}{\ell}\|\mathbf{x}-\mathbf{x}'\|_2\right) + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$$

($\boldsymbol{\theta} = [\ell, \sigma_f^2, \sigma_n^2]$) where we added a scaling factor σ_f^2 and an included the ridge term $\sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$ in the kernel definition. Note that for conciseness, we consider the isotropic version for a fixed ν; the formulas can be modified for the anisotropic case. As in the case of the Gaussian kernel, by setting $\varphi(\mathbf{x},\boldsymbol{\eta}) = e^{-i\mathbf{x}^{\mathrm{T}}\boldsymbol{\eta}}$, this kernel matches the form of Eq. (8). We assume that the hyperparameters are bounded as follows:

$$\Theta = \{[\ell, \sigma_n^2, \sigma_f^2] : \ell \ge \ell_0, \sigma_n^2 \ge \sigma_{n0}^2, \sigma_f^2 \le \sigma_{f0}^2\}$$

where $\sigma_{n0}^2 > 0$ (i.e. we have a ridge term; our method is not able to approximate the Matèrn kernel in the absence of a ridge term).

The density is given by

$$p(\boldsymbol{\eta};\ell) = \frac{\Gamma(\nu+d/2)\ell^d}{\Gamma(\nu)\,(2\nu\pi)^{d/2}}\left(1+\frac{\ell^2}{2\nu}\|\boldsymbol{\eta}\|_2^2\right)^{-(\nu+d/2)}.$$

Therefore $p \in \mathcal{P}^{(r)}_{C,\mathbf{L}}$ where we consider $C = \frac{\Gamma(\nu+d/2)\ell_0^d}{\Gamma(\nu)(2\nu\pi)^{d/2}}$, $\mathbf{L} = \frac{\ell_0}{\sqrt{2\nu}}\mathbf{I}_d$ and $r = \nu + d/2 > d/2$, i.e., $2r - d = 2\nu$. So we set U to be the numerical solution of Eq. (20) for $d = 1$, and we set U_k to be the numerical solution of Eq. (20) for $d \ge 2$. In addition, $p(\cdot;\ell)$ is analytic on $\mathbb{R}^d$ and it is analytically continuable to the polyellipse $E_{\mathbf{U},\boldsymbol{\beta}}$ with $\beta_k = \frac{\sqrt{2\nu}}{\ell_0\sqrt{d}}$, $\rho_k = \frac{\sqrt{2\nu}}{2\ell_0\sqrt{d}U_k} + \sqrt{\frac{2\nu}{4\ell_0^2 dU_k^2}+1}$, as required by Theorem 12. Now we bound $p(\cdot;\ell_0)$ on the polyellipse as follows, for $x_k \in \left[-\sqrt{\frac{\nu}{2\ell_0^2 d}+U_k^2}, \sqrt{\frac{\nu}{2\ell_0^2 d}+U_k^2}\right]$, $y_k \in \left[-\frac{\sqrt{2\nu}}{2\ell_0\sqrt{d}}, \frac{\sqrt{2\nu}}{2\ell_0\sqrt{d}}\right]$:

$$\begin{aligned}
|p(\mathbf{x}+i\mathbf{y};\ell_0)| &= \frac{\Gamma(\nu+d/2)\ell_0^d}{\Gamma(\nu)\,(2\nu\pi)^{d/2}}\left|1+\frac{\ell_0^2}{2\nu}\sum_{k=1}^d (x_k+iy_k)^2\right|^{-(\nu+d/2)} \\
&\le \frac{\Gamma(\nu+d/2)\ell_0^d}{\Gamma(\nu)\,(2\nu\pi)^{d/2}}\left(1-\frac{\ell_0^2}{2\nu}\sum_{k=1}^d y_k^2\right)^{-(\nu+d/2)} \\
&\le \frac{\Gamma(\nu+d/2)\ell_0^d}{\Gamma(\nu)\,(2\nu\pi)^{d/2}}\left(1-\frac{\ell_0^2}{2\nu}\sum_{k=1}^d \frac{\nu}{2\ell_0^2 d}\right)^{-(\nu+d/2)} \\
&= \frac{\Gamma(\nu+d/2)\ell_0^d}{\Gamma(\nu)\,(2\nu\pi)^{d/2}}\left(\frac{3}{4}\right)^{-(\nu+d/2)}
\end{aligned}$$

where the maximum value is obtained at the nearest points to the poles: $y_k = \pm\frac{\sqrt{2\nu}}{2\ell_0 d}$. Hence, $C_{\mathbf{U},\boldsymbol{\beta}} \coloneqq \frac{\Gamma(\nu+d/2)\ell_0^d}{\Gamma(\nu)(2\nu\pi)^{d/2}}\left(\frac{3}{4}\right)^{-(\nu+d/2)}$.

Recall that $\mathbf{z}:\mathbb{R}^d \to \mathbb{C}^n$ is given by $\mathbf{z}(\boldsymbol{\eta})_j = e^{-i\boldsymbol{\eta}^{\mathrm{T}}\mathbf{x}_j}$. Now we bound $|\mathbf{z}(\boldsymbol{\eta})_j|$ for each j on its corresponding ellipse. For any $\mathbf{x},\mathbf{y}\in\mathbb{R}^d$: $|\mathbf{z}(\mathbf{x}+i\mathbf{y})_j| = |\mathbf{z}(i\mathbf{y})_j|$, so we need to bound $|\mathbf{z}(i\mathbf{y})_j|$ for $\mathbf{y}\in\prod_{k=1}^d[-\beta_k/2,\beta_k/2]$:

$$|\mathbf{z}(i\mathbf{y})_j| \quad = \quad \left|e^{-i\cdot i\mathbf{y}^{\mathrm{T}}\mathbf{x}_j}\right| = e^{\mathbf{y}^{\mathrm{T}}\mathbf{x}_j} \le e^{\|\mathbf{y}\|_2\|\mathbf{x}_j\|_2} \le e^{\|\boldsymbol{\beta}\|_2\|\mathbf{R}\|_2/4}$$

Hence, $|\mathbf{z}(\boldsymbol{\eta})_j| \le e^{\|\boldsymbol{\beta}\|_2\|\mathbf{R}\|_2/4} =: M_{\mathbf{U},\boldsymbol{\beta}}$. Now, for the asymptotics, consider the upper bound for $\mathbf{U}$ in Theorem 13. If one denotes $\gamma = \ln\left(2^{2d+2}\pi^{-d/2}\frac{\Gamma(\nu+d/2)}{\Gamma(\nu)}\left(\frac{3}{4}\right)^{-(\nu+d/2)}\sigma_{n0}^{-2}\sigma_{f0}^2 n^2\right)$ and $\delta = \ln\left(\frac{\sigma_{n0}^{-2}\sigma_{f0}^2 n^2}{2^{d-1}\nu\mathrm{B}\left(\nu,\frac{d}{2}\right)}\right)$, then Theorem 13 can be applied with the these parameters and obtain that (for $d\ge 3$)

$$s_k \ge \frac{\frac{1}{2d}\|\boldsymbol{\beta}\|_2\|\mathbf{R}\|_2 + \frac{\gamma}{d} + \frac{\delta}{2\nu} - \ln\left(\beta_k/(2U_k) - 1 + \sqrt{\beta_k^2/(4U_k^2)+1}\right)}{2\ln\left(\beta_k/(2U_k) + \sqrt{\beta_k^2/(4U_k^2)+1}\right)} + 1$$

we have the desired bound.

Assuming the bounding box $\mathbf{R}$ is fixed, $s_k = O(\ln n)$ suffice and $s = \prod_{k=1}^d s_k = O((\ln n)^d)$ suffices.

6.3 Semigroup Kernels

The previous two examples were of shift-invariant kernels, and the feature mapping φ was based on Bochner's theorem. In this section, we demonstrate the application of our theory to a different type of kernels: semigroup kernels (Yang et al., 2014). These type of kernels require a slight modification of our setup, which we briefly describe below, but adjusting theory itself is technical and we omit it.

Semigroup kernels are well-suited for non-negative data, i.e. $\mathcal{X}\subseteq\mathbb{R}_+^d$, and require that the kernel value at $\mathbf{x}$ and $\mathbf{x}'$ depends only on the sum $\mathbf{x}+\mathbf{x}'$: $k(\mathbf{x},\mathbf{x}') = k_0(\mathbf{x}+\mathbf{x}')$. One example of such kernel is the reciprocal semigroup kernel:

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \sigma_f^2\prod_{k=1}^d\frac{\lambda}{x_k+x_k'+\lambda} + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$$

($\boldsymbol{\theta} = [\lambda,\sigma_f^2,\sigma_n^2]$) where we add a scaling factor σ_f^2 and an included the ridge term $\sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$ in the kernel definition. Berg et al. (Berg et al., 1984) showed that every semigroup kernel can be written in the following integral form, which is analogous to Eq. (9):

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \sigma_f^2\int_{\mathbb{R}_+^d} e^{-\boldsymbol{\eta}^{\mathrm{T}}(\mathbf{x}+\mathbf{x}')}p(\boldsymbol{\eta};\lambda)d\boldsymbol{\eta} + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')$$

where $p(\cdot;\lambda)$ is a probability density function which is supported only on $\mathbb{R}_+^d$. For the reciprocal semigroup kernel we have $p(\boldsymbol{\eta};\lambda) = \lambda e^{-\lambda\|\boldsymbol{\eta}\|_1} = \lambda e^{-\lambda\sum_{k=1}^d\eta_k}$, so $p(\cdot;\lambda)\in\mathcal{E}_{C,\mathbf{L}}^{(1)}$with $C=\lambda$, $\mathbf{L}=\lambda\mathbf{I}_d$, which is analytic on $\mathbb{R}_+^d$. Thus, we see that semigroup kernels can be represented as

$$k_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = \sigma_f^2\int_{\mathbb{R}_+^d}\varphi(\mathbf{x},\boldsymbol{\eta})\varphi(\mathbf{x}',\boldsymbol{\eta})^* p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta} + \sigma_n^2\gamma(\mathbf{x}-\mathbf{x}')\,,$$

which is almost the same as Eq. (8), except the integration area is $\mathbb{R}_+^d$ instead of $\mathbb{R}^d$. For semigroup kernels $\varphi(\mathbf{x},\boldsymbol{\eta}) = e^{-\boldsymbol{\eta}^{\mathrm{T}}\mathbf{x}}$.

The construction of Gauss-Legendre features is quite similar to the integration area is $\mathbb{R}^d$, except that we replace the assumption that $\mathcal{X} \subseteq \prod_{k=1}^d [-R_k/2, R_k/2]$ with $\mathcal{X} \subseteq \prod_{k=1}^d [0, R_k]$, the truncated integration area $\mathcal{Q}_{\mathbf{U}}$ with $\mathcal{H}_{\mathbf{U}} \coloneqq \prod_{k=1}^d [0, U_k]$, and the integration nodes and weights are obtained by linearly transforming $\mathcal{H}_{\mathbf{U}}$ (instead of $\mathcal{Q}_{\mathbf{U}}$) to $[-1,1]^d$ with the transformation $\eta_k = U_k \cdot \frac{\chi_k+1}{2}$ for $k = 1, \ldots, d$. We omit the details of the construction, since they mostly repeat the construction described in Section 5.

Now consider the reciprocal semigroup kernel. We assume that the hyperparameters are bounded as follows:

$$\Theta = \{[\lambda, \sigma_n^2, \sigma_f^2] \,:\, \lambda \geq \lambda_0, \sigma_n^2 \geq \sigma_{n0}^2, \sigma_f^2 \leq \sigma_{f0}^2\}$$

where $\sigma_{n0}^2 > 0$ (i.e., we have a ridge term). We set

$$U_k = \frac{1}{\lambda_0} \ln \left(\left(\frac{2\lambda_0^{1-d} \sigma_{f0}^2 n^2}{\sigma_{n0}^2} \right)^{1/d} \right) .$$

In addition, since $p(\cdot;\lambda)$ is analytic on $\mathbb{R}_+^d$, and in particular it is analytically continuable to the polyellipse $E_{\mathbf{U},\boldsymbol{\beta}}$ with $\beta_k = 2U_k$, $\rho_k = 1 + \sqrt{2}$. Now we bound the analytic continuation of $p(\cdot;\lambda_0)$ (which we also denote by $p(\cdot;\lambda_0)$) on the polyellipse as follows. For any $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{y} \in \mathbb{R}^d$, $|p(\mathbf{x}+i\mathbf{y};\lambda_0)| = \left|\lambda_0 e^{-\lambda_0 \sum_{k=1}^d x_k + iy_k}\right| = |p(\mathbf{x};\lambda_0)|$, so we need to bound $|p(\mathbf{x};\lambda_0)|$ for $x_k \in [U_k \frac{1-\sqrt{2}}{2}, U_k \frac{1+\sqrt{2}}{2}]$, $y_k \in [0, U_k]$:

$$|p(\mathbf{x};\lambda_0)| \quad = \quad \lambda_0 e^{-\lambda_0 \sum_{k=1}^d x_k} \leq \lambda_0 e^{\lambda_0 \sum_{k=1}^d U_k \frac{\sqrt{2}-1}{2}} = \lambda_0 e^{\lambda_0 \frac{\sqrt{2}-1}{2} \|\mathbf{U}\|_1} .$$

Hence, $C_{\mathbf{U},\boldsymbol{\beta}} \coloneqq \lambda_0 e^{\lambda_0 \frac{\sqrt{2}-1}{2} \|\mathbf{U}\|_1}$.

For semigroup kernels we use $\mathbf{z} : \mathbb{R}_+^d \to \mathbb{R}^n$ defined by $\mathbf{z}(\boldsymbol{\eta})_j = e^{-\boldsymbol{\eta}^{\mathrm{T}} \mathbf{x}_j}$ as the feature map. It can be seen that for every $\mathbf{v} \in \mathbb{R}^n$ the function $|\mathbf{z}(\cdot)^* \mathbf{v}|^2$ is an analytic function on $\mathbb{R}_+^d$, and we also have $|\mathbf{z}(\boldsymbol{\eta})_j| \leq 1 = M_R$ for $\boldsymbol{\eta} \geq 0$ and for all $1 \leq j \leq n$. In particular, it is analytically continuable to the pollyellipse $E_{\mathbf{U},\boldsymbol{\beta}}$ with $\beta_k = 2U_k$, $\rho_k = 1 + \sqrt{2}$. Now we bound $|\mathbf{z}(\boldsymbol{\eta})_j|^2$, where here $\mathbf{z}$ denotes the analytic continuation. Notice that for any $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{y} \in \mathbb{R}^d$, $|\mathbf{z}(\mathbf{x}+i\mathbf{y})_j| = |\mathbf{z}(\mathbf{x})_j|$, so we need to bound $|\mathbf{z}(\mathbf{x})_j|$ for each $x_k \in [U_k \frac{1-\sqrt{2}}{2}, U_k \frac{1+\sqrt{2}}{2}]$, $y_k \in [0, U_k]$:

$$|\mathbf{z}(\mathbf{x})_j| \quad = \quad e^{-\mathbf{x}^{\mathrm{T}} \mathbf{x}_j} = e^{-\sum_{k=1}^d x_k (\mathbf{x}_j)_k} \leq e^{\sum_{k=1}^d (U_k \frac{\sqrt{2}-1}{2} R_k)} = e^{\frac{\sqrt{2}-1}{2} \mathbf{U}^{\mathrm{T}} \mathbf{R}}$$

and each $\mathbf{x}_j$ satisfies $(\mathbf{x}_j)_k \leq R_k$. Hence, $|\mathbf{z}(\boldsymbol{\eta})_j| \leq e^{\frac{\sqrt{2}-1}{2} \mathbf{U}^{\mathrm{T}} \mathbf{R}} =: M_{\mathbf{U},\boldsymbol{\beta}}$. So we set

$$s_k \geq \left\lceil \frac{\frac{1}{d} \ln \left(2^{2d+2} \lambda_0 \sigma_n^{-2} \sigma_f^2 n^2\right) + \ln \left(\frac{\sqrt{2}-1}{2d} \|\mathbf{U}\|_1\right) + \frac{\sqrt{2}-1}{d} \mathbf{U}^{\mathrm{T}} \mathbf{R} + \ln \ln \left(\left(\frac{2\lambda_0^{1-d} \sigma_{f0}^2 n^2}{\sigma_{n0}^2} \right)^{1/d} \right) - \ln(\sqrt{2})}{2 \ln(1+\sqrt{2})} \right\rceil + 1 .$$

Since $\|\mathbf{U}\|_1 = O(\ln n)$ and assuming the bounding box $\mathbf{R}$ is fixed, then $s_k = O(\ln n)$ suffice and $s = \prod_{k=1}^d s_k = O((\ln n)^d)$ suffices.

It can be seen that in most examples, the dependence of the number of features on d is exponential.

7. Numerical Experiments

In this section, we report experiments evaluating the performance of our proposed quadrature based approach. Our goal is to show that indeed if $\mathbf{U}$ and $\mathbf{s}$ are set to be large enough, our method yields results that are essentially indistinguishable from using the exact kernel while offering faster hyperparameter learning, training and prediction. Clearly, from the theoretical results, our

method predominately applies to low-dimensional datasets (for example, such datasets are prevalent in spatial statistics), so we experiment with one dimensional and two dimensional datasets. We experiment both with the Gaussian kernel and the Matèrn kernel.

In the graphs, we label our method as GLF-GPR (standing for Gauss-Legendre Features Gaussian Process Regression). We use the following methods as a benchmark: exact GPR (labeled in the graphs as Exact-GPR), GPR based on random Fourier features (labeled RFF-GPR), and GPR based on modified random Fourier features (labeled MRF-GPR). As a performance metric, we use the MSE error on a test set (as a function of number of features) and the time to learn the hyperparameters. Training and prediction time of both GLF-GPR and RFF-GPR are essentially the same for the same number of features, and both are faster than Exact-GPR if the number of features is smaller than the training set size. Thus, when it comes to training and prediction time, it is sufficient to explore the test error as a function of the number of features. However, hyperparameter learning time can vary considerably between GLF-GPR and RFF-GPR, so we compare this quantity directly. MRF-GPR behaves similarly to RFF-GPR and is shown for comparison in the synthetic data experiments.

The various methods were implemented in MATLAB. Optimizing the hyperparameters was conducted using the MATLAB function `fmincon` after transforming the hyperparameters to a logarithmic scale. For each problem, we defined a hyperparameter domain, e.g.

$$\Theta = \{[\ell, \sigma_n^2, \sigma_f^2] : \ell_0 \le \ell \le \ell_1, \sigma_{n0}^2 \le \sigma_n^2 \le \sigma_{n1}^2, \sigma_{f1}^2 \le \sigma_f^2 \le \sigma_{f0}^2\}$$

and we take the initial hyperparameters for the optimization to be $[\ell_0, \sigma_{f0}^2, \sigma_{n0}^2]$. Running times were measured on a machine with two 3.2GHz Intel(R) Xeon(R) Gold 6134 CPUs, each having 8 cores, and 256GB RAM.

7.1 Synthetic Data

In this subsection, we report experiments on synthetically generated data. The data is generated by noisily sampling a predetermined function, i.e. samples are generated from the formula

$$y_i = f^\star(x_i) + \tau_i$$

where $f^\star$ is the true function and $\{\tau_i\}$ are i.i.d noise terms, distributed as normal variables with variance $\sigma_\tau^2 = 0.5^2$ (for 1D) or $\sigma_\tau^2 = 0.3^2\mathbf{I}_2$ (for 2D). In these experiments we use the isotropic Gaussian kernel.

First, we consider a one dimensional function:

$$f_1^\star(x) = \sin(2x) + \sin(6e^x) \tag{23}$$

The function was sampled equidistantly on $[-1, 1]$ with $n = 800$ samples. The results are reported in Figure 1, where we show how GLF-GPR with the number of quadrature points s compared to RFF-GPR.

In the top-right graph, we see that the log-likelihood of GLF-GPR merges with the log-likelihood of Exact-GPR for each of the hyperparameters, where the optimal hyperparameters are dashed in blue. RFF-GPR and MRF-GPR deviate considerably. This graph exemplifies that GLF-GPR can yield a good approximation to the exact log-likelihoods, while RFF-GPR and MRF-GPR yield poor approximations. We also see that GLF-GPR optimizes hyperparameters that are much closer to the exact values than RFF-GPR and MRF-FPR. The bottom-left plot shows the MSE error on the same test points. We see that the GLF-GPR error stabilizes on the Exact-GPR error even before the theoretical value of s. The bottom-right graph shows the runtime of the hyperparameter learning phase for different values of quadrature points s. GLF-GPR is clearly more efficient than Exact-GPR and mostly more efficient than RFF-GPR and MRF-GPR. As expected, as s becomes larger, GLF-GPR learns the hyperparameters much faster than Exact-GPR and RFF-GPR. Furthermore, GLF-GPR achieves a low error rate with fewer features than RFF-GPR and MRF-GPR, and thus

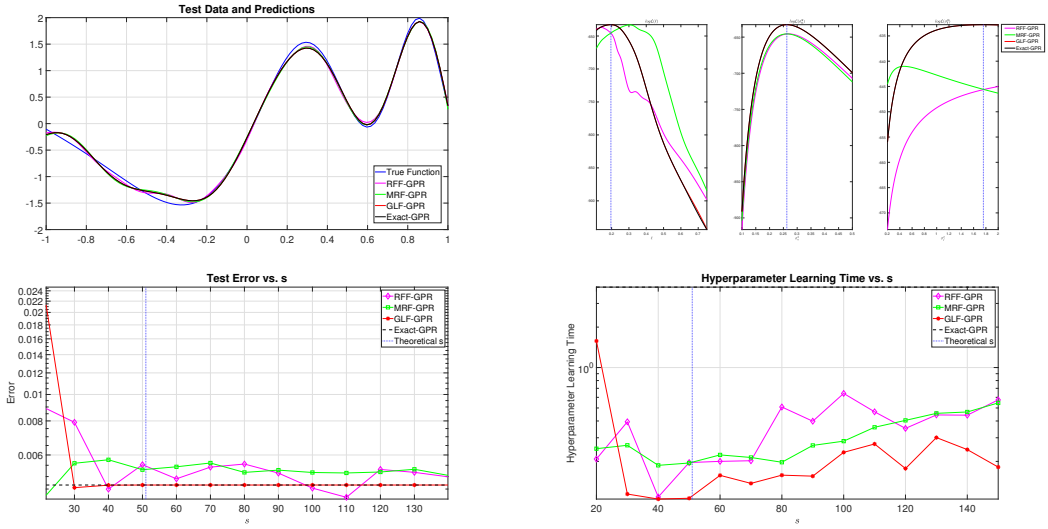

Figure 1: Results for data generated using the function $f_1^\star$. Top-left: true function, input data, and prediction. Top-right: log-likelihood for various sections of the hyperparameter range. Bottom-Left: Test error as a function of the number of features. Bottom-Right: hyperparameter learning time.

is able to do training, prediction, and hyperparameter learning much faster than RFF-GPR and MRF-GPR.

Next, we consider a two dimensional function:

$$f_2^\star(x_1, x_2) = (\sin(x_1) + \sin(10e^{x_1}))(\sin(x_2) + \sin(10e^{x_2}))\,. \tag{24}$$

The function was sampled on an uniform grid on $[-1, 1] \times [-1, 1]$ with $n = 4096$ samples. We consider $\ell_1 = \ell_2$ so $U_1 = U_2$ and $s_1 = s_2$, i.e., $s = s_1^2$. The results are reported in Figure 2.

Similar to the synthetic 1D experiment, in the top-right plot we see that GLF-GPR yields a good approximation to the exact log-likelihood. In the bottom-left plot we see that shows the GLF-GPR error stabilizes on error of Exact-GPR at a much smaller number of quadrature points than the theoretical s.

7.2 Natural Sound Modeling

Next we consider the natural sound benchmark used in (Wilson and Nickisch, 2015) (without hyperparameter learning) and (Dong et al., 2017) (with hyperparameter learning). The data is shown in the top-left graph of Figure 3. The goal is to recover contiguous missing regions in a waveform with $n = 59309$ training points. The test consists of 691 samples. The Gaussian kernel is used for learning.

Results are reported in Figure 3. In the bottom-left graph, we plot the test error as a function of the number of features. Initially, GLF-GPR produces poor results, but when s is large enough, the results are similar to the Exact-GPR (see also the top-right plot). We see that even when the number of quadrature points is smaller than the theoretical value required for spectral equivalence, GLF-GPR's error stabilizes on the Exact-GPR error. In contrast, RFF-GPR's error oscillates above Exact-GPR's error. In the bottom-right graph we see that the runtime of the hyperparameters learning phase is significantly smaller for GLF-GPR.

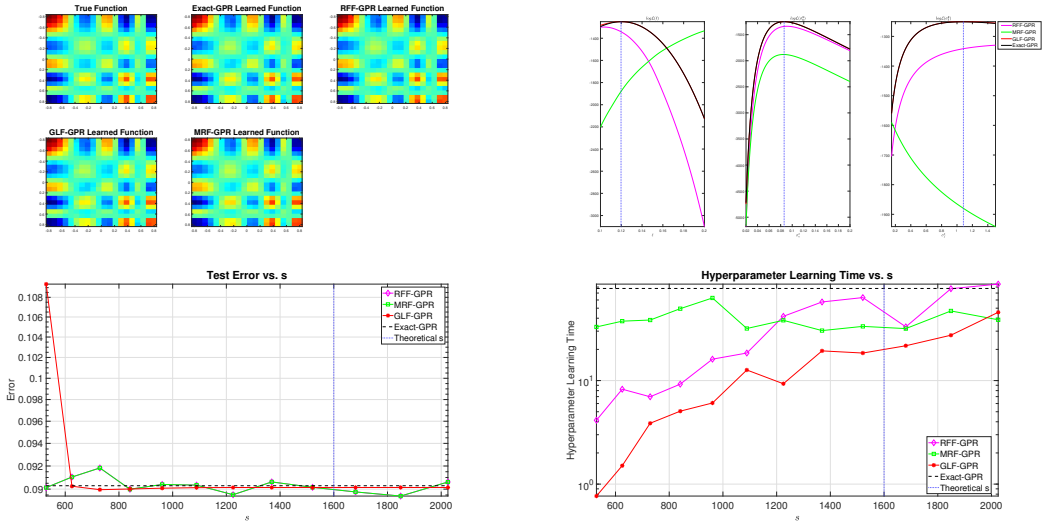

Figure 2: Results for data generated using the function $f_2^\star$. Top-left:true function, input data, and prediction. Top-right: log-likelihood for various sections of the hyperparameter range. Bottom-Left: Test error as a function of the number of features. Bottom-Right: hyperparameter learning time.

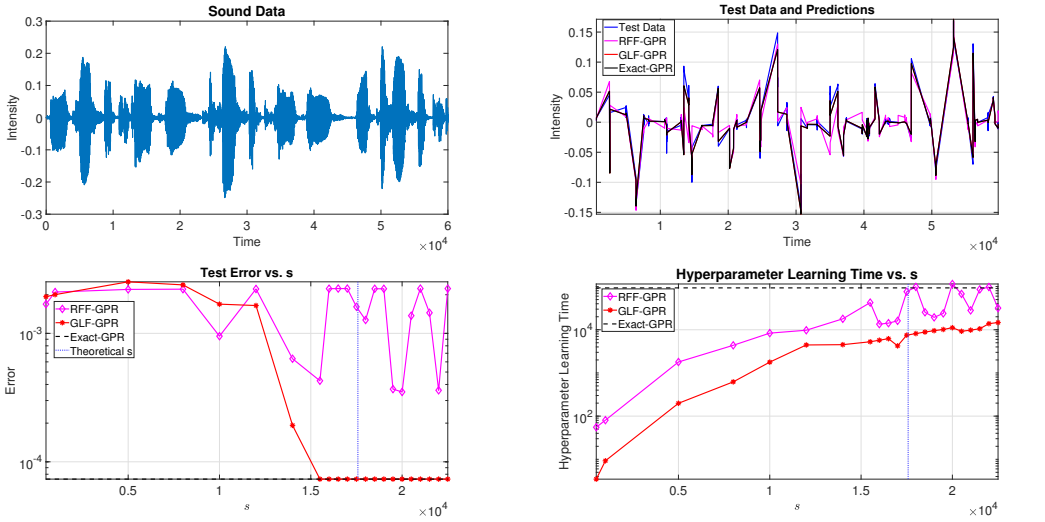

Figure 3: Results for the natural sound data. Top-left: full dataset (training and test). Top-right: test data and predictions, for the best smallest s we tested. Bottom-left: MSE of the test data for various sizes of s. The dashed vertical line is the value of the theoretical minimum s needed for spectral equivalence. Bottom-right: running time for various sizes of s.

7.3 Google Daily High Stock Price

We consider a time series data of the daily high stock price of Google spanning 3797 days from 19th August 2004 to 19th September 2019. We set the data as $x \in \{1, \ldots, 3797\}$ and $y = \log(Stock_{high})$.

The test is of size of 12% of the data, i.e., consists of 502 days. We use the Matèrn kernel with $\nu = 5/2$.

We note that the theoretical number of quadrature features s required for spectral equivalence is bigger than the number of training points. Possible reasons are: the hyperparameter σ_n^2 in these dataset is very small and that increases our bound in Eq. (21). This increases U which increases s. In addition, the weight function of the Matèrn kernel has a singularity point which leads to the ellipse parameter being pretty small. However, in practice, we see that the approximation convergences around the value $s = 1550$.

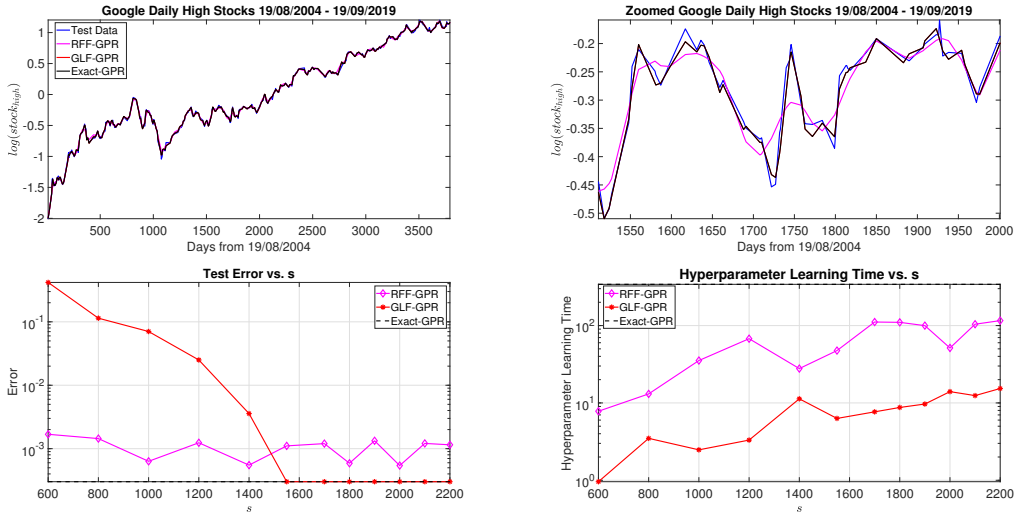

Figure 4: Results for the google stocks data. Top-left: test data and predictions. Top-right: zoomed test data and predictions for the best smallest s we tested. Bottom-left: MSE of the test data for various sizes of s. Bottom-right: running time for various sizes of s.

Results are reported in Figure 4. From the top-right and top-left plots, we see that RFF-GPR is producing a poor approximation while GLF-GPR approximation merges with Exact-GPR approximation. Also, from the bottom-left plot, we see that initially, GLF-GPR produces poor errors, but when s is large enough, the error stabilizes on the Exact-GPR error (see also the top-right plot).

7.4 Spatial Temperature Anomaly for East Africa in 2016

Similar to (Ton et al., 2018), we consider MOD11A2 Land Surface Temperature (LST) 8-day composite 2D data of synoptic yearly mean for 2016 in the East Africa region. For the training set, we randomly sample 77404 LST locations and set $\mathbf{x} \in \{(Longitude, Latitude)\}$ and $y = \{temperature\}$.We examine the MSE errors on the remaining 6005 locations but use all 83409 data points to draw maps. We also use the anisotropic Matèrn kernel with $\nu = 1$. Again, the theoretical number of quadrature features for spectral equivalence is bigger than the number of training points. However again in practice, we see that the approximation convergences with fewer features. Due to memory and time constraints, we were unable to use Exact-GPR, and RFF-GPR results are presented up to the computer's memory capacity.

Results are reported in Figure 5. In the top plot, we see that GLF-GPR approximates the true function well, unlike RFF-GPR. Also, from the bottom-left plot we see that around $s = 21025$, the GLF-GPR error stabilizes while RFF-GPR error is still suboptimal.

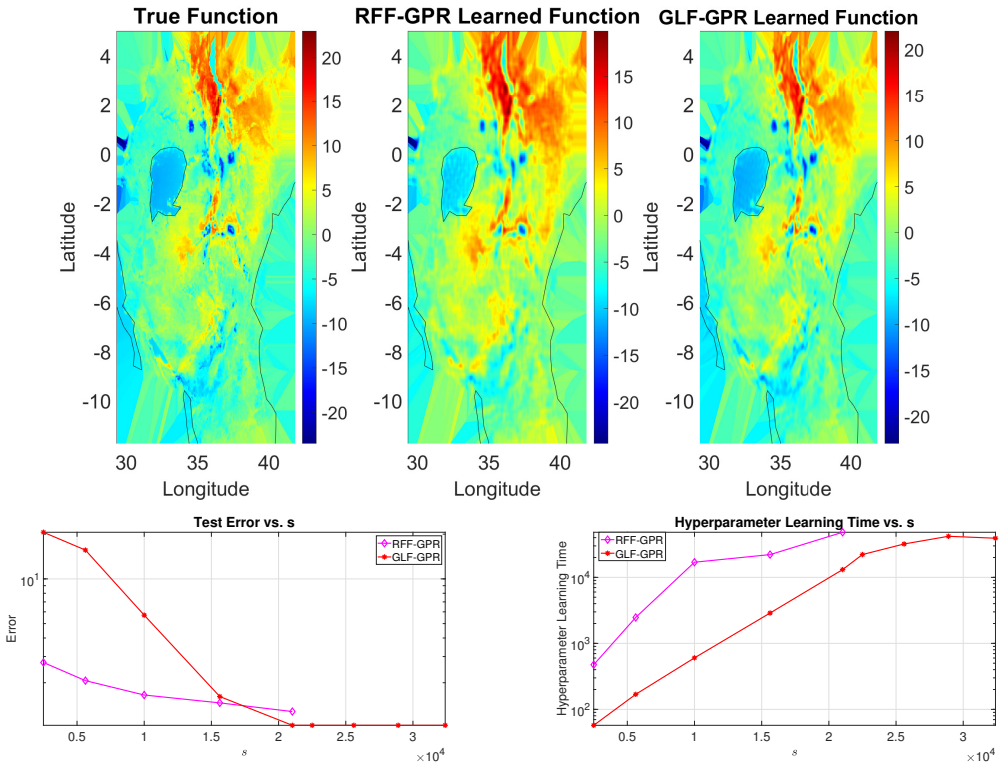

Figure 5: Results for the east Africa data. Top: all true data, and predictions. Bottom-left: MSE of the test data for various sizes of s. Bottom-right: running time for various sizes of s.

8. Conclusions and Future Work

In this paper, we proposed the use of Gauss-Legendre feature for large-scale Gaussian process regression. Our method is very much inspired by Random Fourier Features (Rahimi and Recht, 2008). However, our method replaces Monte-Carlo integration in RFF with a Gauss-Legendre quadrature of a truncated integral representation of the kernel function. With Gauss-Legendre quadrature our method is able to build spectrally equivalent kernel approximation with an number of features that is asymptotically poly-logarithmic in the training size. In contrast, with RFF the number of features for spectral equivalence must be at least linear. Sublinear number of features can also be obtained using a Gaussian quadrature (suggested by Dao et al. (Dao et al., 2017) in the context of kernel learning). However, this is problematic in the context of hyperparameter learning (see Section 4.3). RFF has a similar issue. In contrast, the use of Gauss-Legendre quadrature allows our method to keep the quadrature nodes and weights fixed, leading to simplified structural dependence of the kernel matrix on the hyperparameters which is more amenable to hyperparameter learning. Finally, we demonstrate the utility of our method on several real-world low-dimensional datasets.

We mention a few possible directions for future research:

- Asymptotically, our method requires a number of features that is poly-logarithmic in the training size. Yet, for some moderately sized datasets, our theoretical results required a number of features larger than the number of training points. However, in practice, the number of

features required for high quality results was much smaller than the bound. Closing this gap is an open problem.

- Our method is able to handle a rich family of kernels (see Eq. (8)) which includes stationary kernels and some non-stationary kernels (see Section 6.3). Extending our method to arbitrary non stationary kernels is an open problem.
- The number of features needed by our method is exponential in the dimension, i.e. we have not escaped from the curse of dimensionality. Future research directions are: replacing the tensorized multivariate quadrature with sparse grids, and using convolutional kernels. In doing so, we strive to avoid an exponential dependence on d.

Acknowledgements

This research was supported by BSF grant 2017698.

Appendix A. Further Details on Hyperparameters Learning

A.1 Derivation of Eq. (14)

Recall that,

$$\mathbf{F}(\boldsymbol{\theta}) = \sigma_f^2(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1})^{-1}\mathbf{Z}^*\mathbf{Z}\,.$$

First, from the Woodbury formula we obtain that

$$\begin{aligned}
\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\right) &= \mathbf{Tr}\left(\left(\sigma_f^2\mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^* + \sigma_n^2\mathbf{I}_n\right)^{-1}\right) \\
&= \sigma_n^{-2}\mathbf{Tr}\left(\mathbf{I}_n - \sigma_f^2\mathbf{Z}\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1}\right)^{-1}\mathbf{Z}^*\right) \\
&= \sigma_n^{-2}n - \sigma_n^{-2}\mathbf{Tr}\left(\sigma_f^2\mathbf{Z}\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1}\right)^{-1}\mathbf{Z}^*\right) \\
&= \sigma_n^{-2}n - \sigma_n^{-2}\mathbf{Tr}\left(\sigma_f^2\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1}\right)^{-1}\mathbf{Z}^*\mathbf{Z}\right) \\
&= \sigma_n^{-2}\left(n - \mathbf{Tr}\left(\mathbf{F}(\boldsymbol{\theta})\right)\right)\,.
\end{aligned}$$

Now, for the first term in Eq. (14) we have

$$\begin{aligned}
\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_f^2}\right) &= \mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^*\right) \\
&= \mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\sigma_f^{-2}\left(\sigma_f^2\mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^* + \sigma_n^2\mathbf{I}_n - \sigma_n^2\mathbf{I}_n\right)\right) \\
&= \sigma_f^{-2}\mathbf{Tr}\left(\mathbf{I}_n - \sigma_n^2\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\right) \\
&= \sigma_f^{-2}n - \sigma_f^{-2}\sigma_n^2\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\right) \\
&= \sigma_f^{-2}n - \sigma_f^{-2}\sigma_n^2\left(\sigma_n^{-2}n - \sigma_n^{-2}\mathbf{Tr}\left(\mathbf{F}(\boldsymbol{\theta})\right)\right) \\
&= \sigma_f^{-2}\mathbf{Tr}\left(\mathbf{F}(\boldsymbol{\theta})\right)\,.
\end{aligned}$$

For the next term in Eq. (14) we have

$$\begin{aligned}
\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_n^2}\right) &= \mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\right) \\
&= \sigma_n^{-2}(n - \mathbf{Tr}\left(\mathbf{F}(\boldsymbol{\theta})\right))
\end{aligned}$$

Finally, for the third term in Eq. (14) we have

$$\begin{aligned}
\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\theta_i}\right) &= \mathbf{Tr}\left(\left(\sigma_f^2\mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^*+\sigma_n^2\mathbf{I}_n\right)^{-1}\sigma_f^2\mathbf{Z}\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\right) \qquad (25)\\
&= \sigma_n^{-2}\mathbf{Tr}\left(\sigma_f^2\mathbf{Z}\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*-\sigma_f^4\mathbf{Z}\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z}+\sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1}\right)^{-1}\mathbf{Z}^*\mathbf{Z}\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\right)\\
&= \sigma_n^{-2}\sigma_f^2\mathbf{Tr}\left(\mathbf{Z}\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*-\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\right)\\
&= \sigma_n^{-2}\sigma_f^2\mathbf{Tr}\left(\mathbf{Z}\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\right)-\sigma_n^{-2}\sigma_f^2\mathbf{Tr}\left(\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\right)\\
&= \sigma_n^{-2}\sigma_f^2\mathbf{Tr}\left(\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\mathbf{Z}\right)-\sigma_n^{-2}\sigma_f^2\mathbf{Tr}\left(\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})\right)
\end{aligned}$$

where in the second equality we use the Woodbury formula.

A.2 Efficient Gaussian Process Regression using QR Decomposition

Here we present alternative formulas to the ones presented in Section 4.2 and based on QR decomposition instead of the normal equations. Such formulas are likely to be more numerically robust.

Let

$$\mathbf{Z}=\mathbf{Q}_\mathbf{Z}\mathbf{R}_\mathbf{Z}$$

be a thin QR decomposition of $\mathbf{Z}$, i.e. $\mathbf{Q}_\mathbf{Z}\in\mathbb{C}^{n\times s}$ is such that $\mathbf{Q}_\mathbf{Z}^*\mathbf{Q}_\mathbf{Z}=\mathbf{I}_n$ and $\mathbf{R}_\mathbf{Z}\in\mathbb{C}^{s\times s}$ is an upper triangular matrix. We suggest to compute the QR decomposition of $\mathbf{Z}$ in lieu of computing $\mathbf{Z}^*\mathbf{Z}$, and keeping only $\mathbf{R}_\mathbf{Z}$ and $\mathbf{Q}_\mathbf{Z}^*\mathbf{y}$ so still only $O(s^2)$ is needed. There is no asymptotic penalty in terms of arithmetic operation count since the decomposition can be computed in $O(ns^2)$ operations. However, $\mathbf{Z}^*\mathbf{Z}$ tends to be ill-conditioned due to the squaring of the condition number of $\mathbf{Z}$, so it is best to avoid computing it. Note that the QR decomposition is computed only once, and not per iteration.

Let us consider a specific iteration, and for conciseness, we omit $\boldsymbol{\theta}$ for the following formulas. Let

$$\mathbf{A}\coloneqq\begin{bmatrix}\mathbf{R}_\mathbf{Z}\\ \frac{\sigma_n}{\sigma_f}\mathbf{W}^{-1/2}\end{bmatrix}\in\mathbb{C}^{2s\times s}$$

We compute a thin QR decomposition $\mathbf{A}=\mathbf{Q}_\mathbf{A}\mathbf{R}_\mathbf{A}$ of $\mathbf{A}$ ($O(s^3)$ operations) and write

$$\mathbf{Q}_\mathbf{A}=\begin{bmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\end{bmatrix}$$

where $\mathbf{Q}_1,\mathbf{Q}_2\in\mathbb{C}^{s\times s}$, i.e.,

$$\mathbf{A}=\begin{bmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\end{bmatrix}\mathbf{R}_\mathbf{A}\,.$$

Hence,

$$\mathbf{R}_\mathbf{Z}=\mathbf{Q}_1\mathbf{R}_\mathbf{A}$$

which implies that

$$\mathbf{Z}=\mathbf{Q}_\mathbf{Z}\mathbf{Q}_1\mathbf{R}_\mathbf{A}\,.$$

Therefore,

$$\begin{bmatrix}\mathbf{Z}\\ \frac{\sigma_n}{\sigma_f}\mathbf{W}^{-1/2}\end{bmatrix}=\begin{bmatrix}\mathbf{Q}_\mathbf{Z}&\mathbf{0}\\ \mathbf{0}&\mathbf{I}_s\end{bmatrix}\mathbf{A}=\begin{bmatrix}\mathbf{Q}_\mathbf{Z}&\mathbf{0}\\ \mathbf{0}&\mathbf{I}_s\end{bmatrix}\begin{bmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\end{bmatrix}\mathbf{R}_\mathbf{A}=\begin{bmatrix}\mathbf{Q}_\mathbf{Z}\mathbf{Q}_1\\ \mathbf{Q}_2\end{bmatrix}\mathbf{R}_\mathbf{A}$$

which is a QR decomposition of

$$\mathbf{B} \coloneqq \begin{bmatrix} \mathbf{Z} \\ \frac{\sigma_n}{\sigma_f}\mathbf{W}^{-1/2} \end{bmatrix}$$

The crux is that given the QR decomposition of $\mathbf{Z}$, we can compute the QR decomposition of $\mathbf{B}$ in $O(s^3)$ arithmetic operations instead of $O(ns^2)$.

We now compute

$$\begin{aligned} \mathbf{w} &= \mathbf{W}^{-1}\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}^{-1}\right)^{-1}\mathbf{Z}^*\mathbf{y} \\ &= \sigma_f^{-2}\mathbf{W}^{-1}\begin{bmatrix} \mathbf{Z} \\ \frac{\sigma_n}{\sigma_f}\mathbf{W}^{-1/2} \end{bmatrix}^+ \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_{s\times 1} \end{bmatrix} \\ &= \sigma_f^{-2}\mathbf{W}^{-1}\mathbf{R}_{\mathbf{A}}^{-1}\left[\mathbf{Q}_1^*\mathbf{Q}_{\mathbf{Z}}^* \;\; \mathbf{Q}_2^*\right]\begin{bmatrix} \mathbf{y} \\ \mathbf{0}_{s\times 1} \end{bmatrix} \\ &= \sigma_f^{-2}\mathbf{W}^{-1}\mathbf{R}_{\mathbf{A}}^{-1}\mathbf{Q}_1^*\mathbf{Q}_{\mathbf{Z}}^*\mathbf{y} \end{aligned} \tag{26}$$

which implies that $\mathbf{w}$ can be computed in $O(ns)$ operations (since $\mathbf{W}$ is diagonal).

Similarly, we also have

$$\begin{aligned} \mathbf{F}(\boldsymbol{\theta}) &= \sigma_f^2\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta})^{-1}\right)^{-1}\mathbf{Z}^*\mathbf{Z} \\ &= \sigma_f^2\sigma_f^{-2}\mathbf{R}_{\mathbf{A}}^{-1}\mathbf{Q}_1^*\mathbf{Q}_{\mathbf{Z}}^*\mathbf{Z} \\ &= \mathbf{R}_{\mathbf{A}}^{-1}\mathbf{Q}_1^*\mathbf{Q}_{\mathbf{Z}}^*\mathbf{Q}_{\mathbf{Z}}\mathbf{R}_{\mathbf{Z}} \\ &= \mathbf{R}_{\mathbf{A}}^{-1}\mathbf{Q}_1^*\mathbf{R}_{\mathbf{Z}} \\ &= \mathbf{R}_{\mathbf{A}}^{-1}\mathbf{Q}_1^*\mathbf{Q}_1\mathbf{R}_{\mathbf{A}} \end{aligned}$$

i.e., $\mathbf{F}(\boldsymbol{\theta})$ can be computed in $O(s^3)$ operations. This allows us to compute the first two formulas in Eq. (14) in $O(s)$ time. As for the third formula, we have (25). Since $\partial\mathbf{W}(\boldsymbol{\theta})/\partial\theta_i$ is diagonal, we need to compute only the diagonal of $\mathbf{Z}^*\mathbf{Z}$ and $\mathbf{Z}^*\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})$. The diagonal of $\mathbf{Z}^*\mathbf{Z}$ is just the square norms of the columns of $\mathbf{Z}$, and can be precomputed in $O(ns)$. Furthermore, in some cases we know analytically the values of this norm. For example, for shift-invariant kernels we use $\varphi(\mathbf{x}, \boldsymbol{\eta}) = e^{-i\mathbf{x}^{\mathrm{T}}\boldsymbol{\eta}}$ so the squared norms of the columns of $\mathbf{Z}$ is equal to n. As for $\mathbf{Z}^*\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})$, we have:

$$\mathbf{Z}^*\mathbf{Z}\mathbf{F}(\boldsymbol{\theta}) = \mathbf{R}_{\mathbf{A}}^*(\mathbf{Q}_1^*\mathbf{Q}_1)^2\mathbf{R}_{\mathbf{A}} = \mathbf{R}_{\mathbf{Z}}^*\mathbf{Q}_1\mathbf{Q}_1^*\mathbf{R}_{\mathbf{Z}}\,.$$

Note that $\mathbf{Q}_1^*\mathbf{R}_{\mathbf{Z}}$ has already been computed for $\mathbf{F}(\boldsymbol{\theta})$. Since we only need the diagonal of $\mathbf{Z}^*\mathbf{Z}\mathbf{F}(\boldsymbol{\theta})$, and this is an Hermitian matrix, the diagonal is just the squared norms of the columns of $\mathbf{Q}_1^*\mathbf{R}_{\mathbf{Z}}$. Thus, after $O(s^3)$ preprocessing, for every θ_i the first term in Eq. (13) can be computed in $O(s)$ operations .

As for the first identity in Eq. (13), we still have

$$\boldsymbol{\alpha} = \sigma_n^{-2}\left(\mathbf{y} - \sigma_f^2\mathbf{Z}\mathbf{W}\mathbf{w}\right)$$

and $\mathbf{Z}^*\boldsymbol{\alpha} = \mathbf{w}$. So, the second identity can be computed as previously described using $\mathbf{w}$, which is computed according to Eq. (26).

Appendix B. Analysis of Multivariate Chebyshev Approximation in High Dimensions

The following theorems are generalizations of similar one-dimensional theorems. All the proofs rely on ideas similar to the ones presented in (Mason, 1980, 1982; Mason and Handscomb, 2002)

and (Trefethen, 2013, Theorem 3.1, Theorem 8.1). Note that a similar generalization can found in (Trefethen, 2017; Wang and Zhang, 2020). For completeness, we present our own proof, which is based on a different technique.

For convenience, denote:

$$E_{\boldsymbol{\rho}} \coloneqq \left\{ \mathbf{z} \in \mathbb{C}^d : \left| z_k + \sqrt{z_k^2 - 1} \right| < \rho_k \qquad \forall k = 1, \ldots, d \right\}$$

as the polyellipse, where each E_{ρ_k} is Bernstein ellipse with foci at ± 1 and the sum of of major and minor semiaxis lengths of the ellipse is ρ_k . Also denote

$$\mathcal{C}_{\mathbf{r}} \coloneqq \left\{ \mathbf{z} \in \mathbb{C}^d : |z_k| = r_k \qquad \forall k = 1, \ldots, d \right\}$$

as the polycircle centered at the origin, and simply denote $\mathcal{C}_1$ in the case where $r_1 = \ldots = r_d = 1$. Finally, denote

$$\mathcal{A}_{\mathbf{r},\mathbf{R}} \coloneqq \left\{ \mathbf{z} \in \mathbb{C}^d : 0 < r_k < |z_k| < R_k \, \forall k = 1, \ldots, d \right\}$$

as the polyannulus centered at the origin. The following proposition appears in (Scheidemann, 2005) as Theorem 1.5.26. See also (Bochner and Martin, 1948, Pages 32, 90-91) for further details.

Proposition 14 *Let $\mathcal{A}_{\mathbf{r},\mathbf{R}}$ be the polyannulus centered at the origin, and let f be an analytic complex function on $\mathcal{A}_{\mathbf{r},\mathbf{R}}$. Also, let $r_k < s_k < R_k$, $k = 1, \ldots, d$. Then f has a multivariate Laurent expansion*

$$f(\mathbf{z}) = \sum_{j_1, \ldots, j_d = -\infty}^{\infty} b_{j_1 \ldots j_d} z_1^{j_1} \cdots z_d^{j_d}$$

converging uniformly on $\mathcal{A}_{\mathbf{r},\mathbf{R}}$. The coefficients $b_{j_1 \ldots j_d}$ are given by

$$b_{j_1 \ldots j_d} = \frac{1}{(2i\pi)^d} \oint_{\mathcal{C}_{\mathbf{s}}} \frac{f(\mathbf{z})}{\mathbf{z}^{\alpha+1}} d\mathbf{z}$$

Note that the one dimensional Chebyshev polynomials in the complex plane are defined by

$$T_j(z) = \frac{w^j + w^{-j}}{2}$$

where

$$z = \frac{w + w^{-1}}{2} \, .$$

In addition, for a function f that is analytic in the interior and on the boundary of E_ρ in the complex plane, the complex Chebyshev series of f is $\sum_{j=0}^{\infty} a_j T_j(z)$ where

$$\forall j \neq 0 : \, a_j = \frac{2}{\pi \left(\rho^{2j} + \rho^{-2j} \right)} \oint_{E_\rho} f(z) \overline{T_j(z)} \left| \frac{dz}{\sqrt{1 - z^2}} \right| , \quad a_0 = \frac{1}{2\pi} \oint_{E_\rho} f(z) \overline{T_j(z)} \left| \frac{dz}{\sqrt{1 - z^2}} \right| .$$

The last definition was introduced in (Mason and Handscomb, 2002), and we generalize it to multivariate functions. The multivariate complex tensorized Chebyshev series of a multivariate complex function f that is analytic in the polyellipse $E_{\boldsymbol{\rho}}$ can be defined by

$$\sum_{j_1, \ldots, j_d = 0}^{\infty} a_{j_1 \ldots j_d} T_{j_1}(z_1) \cdots T_{j_d}(z_d) \, .$$

Proposition 15 *Let f be a multivariate complex function that is analytic in the polyellipse $E_{\boldsymbol{\rho}}$, where $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_d)$, $\rho_1, \ldots, \rho_d > 0$. Then, the coefficients of its multivariate complex tensorized Chebyshev series*

$$\sum_{j_1,\ldots,j_d=0}^{\infty} a_{j_1\ldots j_d} T_{j_1}(z_1)\cdots T_{j_d}(z_d) \tag{27}$$

are given by

$$a_{j_1\ldots j_d} = \frac{2^{d-m}}{\pi^d \left(\rho_1^{2j_1} + \rho_1^{-2j_1}\right)\cdots\left(\rho_d^{2j_d} + \rho_d^{-2j_d}\right)} \oint_{E_{\boldsymbol{\rho}}} f(z_1,\ldots,z_d)\,\overline{T_{j_1}(z_1)\cdots T_{j_d}(z_d)} \left| \frac{dz_1\cdots dz_d}{\sqrt{1-z_1^2}\cdots\sqrt{1-z_d^2}} \right|$$

where $m := \#\{j_k : j_k = 0\}$.

We remark that this Chebyshev series converges to f uniformly, as (Mason, 1982, Theorem 9.1) claims.

Proof We begin by mapping $f(\mathbf{z})$ on the contour of $E_{\boldsymbol{\rho}}$ into $g(\mathbf{w})$ on $\mathcal{C}_{\boldsymbol{\rho}}$. For $k = 1, \ldots, d$, define

$$z_k = \frac{w_k + w_k^{-1}}{2}$$

such that $g(w_1,\ldots,w_d) = f(z_1,\ldots,z_d) = f\left(\frac{w_1+w_1^{-1}}{2}, \ldots, \frac{w_d+w_d^{-1}}{2}\right)$. It follows that

$$g(w_1,\ldots,w_d) = g(w_1^{\alpha_1},\ldots,w_d^{\alpha_d}), \quad \alpha_1,\ldots,\alpha_d \in \{-1,1\}\,. \tag{28}$$

The equation for each w_k has two solutions

$$w_k = z_k \pm \sqrt{z_k^2 - 1}\,.$$

We choose the solutions $w_k = z_k + \sqrt{z_k^2-1}$, so $|w_k| = \rho_k > 1$ and thus the second solution for each $k = 1,\ldots,d$ is essentially w_k^{-1}. These relations imply that g is analytic in the polyannulus between $\mathcal{C}_{\boldsymbol{\rho}}$ and $\mathcal{C}_{\boldsymbol{\rho}^{-1}}$. We also have for each $k = 1,\ldots,d$

$$T_{j_k}(z_k) = \frac{w_k^{j_k} + w_k^{-j_k}}{2}\,.$$

Therefore, and since f is analytic in $E_{\boldsymbol{\rho}}$, we have

$$\begin{aligned} g(\mathbf{w}) &= f(\mathbf{z}) \\ &= \sum_{j_1,\ldots,j_d=0}^{\infty} a_{j_1\ldots j_d} T_{j_1}(z_1)\cdots T_{j_d}(z_d) \\ &= \sum_{j_1,\ldots,j_d=0}^{\infty} \frac{a_{j_1\ldots j_d}}{2^d}\left(w_k^{j_1} + w_k^{-j_1}\right)\cdots\left(w_d^{j_d} + w_d^{-j_d}\right). \end{aligned}$$

That is, the series given in Eq. (27) can be written as the Laurent series of g. Thus, by Proposition 14, the coefficients are given by

$$\frac{a_{j_1\ldots j_d}}{2^{d-m}} = \frac{1}{(2i\pi)^d} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-1-j_1}\cdots w_d^{-1-j_d} g(w_1,\ldots,w_d)\,dw_1\cdots dw_d\,.$$

which implies

$$a_{j_1\ldots j_d} = \frac{1}{2^m (i\pi)^d} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-1-j_1}\cdots w_d^{-1-j_d} g(w_1,\ldots,w_d)\,dw_1\cdots dw_d\,. \tag{29}$$

The last integral can be written also as

$$\begin{aligned}
&\oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-1-j_1}\cdots w_d^{-1-j_d} g\left(w_1,\ldots,w_d\right) dw_1\cdots dw_d = \\
&\oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-1-j_1}\cdots w_d^{-1-j_d} g\left(w_1^{\alpha_1},\ldots,w_d^{\alpha_d}\right) dw_1\cdots dw_d = \\
\oint_{|\widetilde{w}_1|=\rho_1^{\alpha_1}}\cdots\oint_{|\widetilde{w}_d|=\rho_d^{\alpha_d}} &\widetilde{w}_1^{-1+\alpha_1 j_1}\cdots\widetilde{w}_d^{-1+\alpha_d j_d} g\left(\widetilde{w}_1,\ldots,\widetilde{w}_d\right) d\widetilde{w}_1\cdots d\widetilde{w}_d = \\
&\oint_{\mathcal{C}_{\boldsymbol{\rho}}} \widetilde{w}_1^{-1+\alpha_1 j_1}\cdots\widetilde{w}_d^{-1+\alpha_d j_d} g\left(\widetilde{w}_1,\ldots,\widetilde{w}_d\right) d\widetilde{w}_1\cdots d\widetilde{w}_d
\end{aligned} \tag{30}$$

where the first equality follows from Eq. (28), the second equality is changing of variables from w_k to $\widetilde{w}_k = w_k^{\alpha_k}$, $\alpha_k \in \{-1,1\}$, and the last equality is due to Definition 14 which means the integral also can be considered on $\mathcal{C}_{\boldsymbol{\rho}^{-1}}$. Now we show that

$$\begin{aligned}
&\oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-1-j_1}\cdots w_d^{-1-j_d} g\left(w_1,\ldots,w_d\right) dw_1\cdots dw_d = \\
&\prod_{k=1}^{d}\frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}}\prod_{k=1}^{d}\left(\rho_k^{2j_k}w_k^{-j_k}+\rho_k^{-2j_k}w_k^{j_k}\right) g\left(w_1,\ldots,w_d\right)\frac{dw_1\cdots dw_d}{w_1\cdots w_d}
\end{aligned} \tag{31}$$

by induction on the number of changes of variables.

The base case: apply the change of variables only for one of the variables. Without loss of generality, we show it for w_1:

$$\begin{aligned}
&\oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-1-j_1}\cdots w_d^{-1-j_d} g\left(w_1,\ldots,w_d\right) dw_1\cdots dw_d = \\
&\frac{\rho_1^{2j_1}+\rho_1^{-2j_1}}{\rho_1^{2j_1}+\rho_1^{-2j_1}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-j_1}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right)\frac{dw_1\cdots dw_d}{w_1\cdots w_d} = \\
&\frac{1}{\rho_1^{2j_1}+\rho_1^{-2j_1}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}} \rho_1^{2j_1} w_1^{-j_1}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right)\frac{dw_1\cdots dw_d}{w_1\cdots w_d} + \\
&+\frac{1}{\rho_1^{2j_1}+\rho_1^{-2j_1}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}} \rho_1^{-2j_1} w_1^{-j_1}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right)\frac{dw_1\cdots dw_d}{w_1\cdots w_d} = \\
&\frac{1}{\rho_1^{2j_1}+\rho_1^{-2j_1}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}} \rho_1^{2j_1} w_1^{-j_1}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right)\frac{dw_1\cdots dw_d}{w_1\cdots w_d} + \\
&+\frac{1}{\rho_1^{2j_1}+\rho_1^{-2j_1}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}} \rho_1^{-2j_1} w_1^{j_1} w_2^{-j_2}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right)\frac{dw_1\cdots dw_d}{w_1\cdots w_d} = \\
&\frac{1}{\rho_1^{2j_1}+\rho_1^{-2j_1}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}} \left(\rho_1^{2j_1}w_1^{-j_1}+\rho_1^{-2j_1}w_1^{j_1}\right) w_2^{-j_2}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right)\frac{dw_1\cdots dw_d}{w_1\cdots w_d}
\end{aligned}$$

where in the third equality we use Eq. (30) with $\alpha_1 = 1$.

The inductive step: suppose that for $1 < n-1 < d$ changes of variables, the following holds:

$$\begin{aligned}
&\oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-1-j_1}\cdots w_d^{-1-j_d} g\left(w_1,\ldots,w_d\right) dw_1\cdots dw_d = \\
&\prod_{k=1}^{n-1}\frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}}\prod_{k=1}^{n-1}\left(\rho_k^{2j_k}w_k^{-j_k}+\rho_k^{-2j_k}w_k^{j_k}\right) w_n^{-j_n}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right)\frac{dw_1\cdots dw_d}{w_1\cdots w_d}
\end{aligned} \tag{32}$$

Then, we show that this is also true for n changes of variables:

$$\oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-1-j_1}\cdots w_d^{-1-j_d} g\left(w_1,\ldots,w_d\right) dw_1\cdots dw_d =$$

$$\frac{\rho_n^{2j_n}+\rho_n^{-2j_n}}{\rho_n^{2j_n}+\rho_n^{-2j_n}} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-j_1}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right) \frac{dw_1\cdots dw_d}{w_1\cdots w_d} =$$

$$\prod_{k=1}^{n} \frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}} \left(\rho_n^{2j_n}+\rho_n^{-2j_n}\right) \oint_{\mathcal{C}_{\boldsymbol{\rho}}} \left(\prod_{k=1}^{n-1}\left(\rho_k^{2j_k} w_k^{-j_k}+\rho_k^{-2j_k} w_k^{j_k}\right)\right) w_n^{-j_n}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right) \frac{dw_1\cdots dw_d}{w_1\cdots w_d} =$$

$$\prod_{k=1}^{n} \frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} \left(\prod_{k=1}^{n-1}\left(\rho_k^{2j_k} w_k^{-j_k}+\rho_k^{-2j_k} w_k^{j_k}\right)\right) \rho_n^{2j_n} w_n^{-j_n}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right) \frac{dw_1\cdots dw_d}{w_1\cdots w_d} +$$

$$+\prod_{k=1}^{n} \frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} \left(\prod_{k=1}^{n-1}\left(\rho_k^{2j_k} w_k^{-j_k}+\rho_k^{-2j_k} w_k^{j_k}\right)\right) \rho_n^{-2j_n} w_n^{-j_n}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right) \frac{dw_1\cdots dw_d}{w_1\cdots w_d} =$$

$$\prod_{k=1}^{n} \frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} \left(\prod_{k=1}^{n-1}\left(\rho_k^{2j_k} w_k^{-j_k}+\rho_k^{-2j_k} w_k^{j_k}\right)\right) \rho_n^{2j_n} w_n^{-j_n}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right) \frac{dw_1\cdots dw_d}{w_1\cdots w_d} +$$

$$+\prod_{k=1}^{n} \frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} \left(\prod_{k=1}^{n-1}\left(\rho_k^{2j_k} w_k^{-j_k}+\rho_k^{-2j_k} w_k^{j_k}\right)\right) \rho_n^{-2j_n} w_n^{j_n}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right) \frac{dw_1\cdots dw_d}{w_1\cdots w_d} =$$

$$\prod_{k=1}^{n} \frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} \left(\prod_{k=1}^{n}\left(\rho_k^{2j_k} w_k^{-j_k}+\rho_k^{-2j_k} w_k^{j_k}\right)\right) w_{n+1}^{-j_{n+1}}\cdots w_d^{-j_d} g\left(w_1,\ldots,w_d\right) \frac{dw_1\cdots dw_d}{w_1\cdots w_d}$$

where in the second equality we use Eq. (32), and in the fourth equality we use Eq. (30) with $\alpha_n = 1$. Therefore, by induction, for $n = d$ we obtain Eq. (31), and by Eq. (29):

$$a_{j_1\ldots j_d} = \frac{1}{2^m\left(i\pi\right)^d} \prod_{k=1}^{d} \frac{1}{\rho_k^{2j_k}+\rho_k^{-2j_k}} \oint_{\mathcal{C}_{\boldsymbol{\rho}}} \prod_{k=1}^{d}\left(\rho_k^{2j_k} w_k^{-j_k}+\rho_k^{-2j_k} w_k^{j_k}\right) g\left(w_1,\ldots,w_d\right) \frac{dw_1\cdots dw_d}{w_1\cdots w_d}.$$

Now, for each $k = 1,\ldots,d$ we have

$$\left|\frac{dz_k}{\sqrt{1-z_k^2}}\right| = \frac{dw_k}{iw_k} \tag{33}$$

and

$$\overline{T_{j_k}(z_k)} = \frac{\overline{w_k}^{j_k}+\overline{w_k}^{-j_k}}{2} = \frac{\rho_k^{2j_k} w_k^{-j_k}+\rho_k^{-2j_k} w_k^{j_k}}{2} \tag{34}$$

where $w_k = \rho_k e^{i\theta_k}$. Therefore, replacing $g\left(w_1,\ldots,w_d\right)$ by $f\left(z_1,\ldots,z_d\right)$ and recall that each w_k on $\mathcal{C}_{\rho_k}$ maps z_k on E_{ρ_k}, we obtain

$$a_{j_1\ldots j_d} = \frac{2^{d-m}}{\pi^d\left(\rho_1^{2j_1}+\rho_1^{-2j_1}\right)\cdots\left(\rho_d^{2j_d}+\rho_d^{-2j_d}\right)} \oint_{E_{\boldsymbol{\rho}}} f\left(z_1,\ldots,z_d\right)\overline{T_{j_1}(z_1)\cdots T_{j_d}(z_d)}\left|\frac{dz_1\cdots dz_d}{\sqrt{1-z_1^2}\cdots\sqrt{1-z_d^2}}\right|.$$

We proceed to the main theorem: ■

Theorem 16 *Let $f(x_1,\ldots,x_d)$ be an analytic function in $[-1,1]^d$ and analytically continuable to the polyellipse $E_{\boldsymbol{\rho}}$ where it satisfies $|f(x_1,\ldots,x_d)| \leq M$ for some $M > 0$. Let $T_j(x) := \cos(j\cos^{-1}(x))$ be the j degree one dimensional Chebyshev polynomial, and $E_{\rho_1},\ldots,E_{\rho_d}$ are the open Bernstein ellipses with major and minor semiaxis lengths correspondingly summing to $\rho_1,\ldots,\rho_d > 1$. Then:*

1. *The multivariate (real) Chebyshev coefficients of f are given by*

$$a_{j_1\ldots j_d} := \frac{2^{d-m}}{\pi^d}\int_{[-1,1]^d}\frac{f(x_1,\ldots,x_d)T_{j_1}(x_1)\cdots T_{j_d}(x_d)}{\sqrt{1-x_1^2}\cdots\sqrt{1-x_d^2}}dx_1\cdots dx_d$$

   where $m := \#\{j_k : j_k = 0\}$.

2. *The coefficients satisfy*

$$|a_{j_1\ldots j_d}| \le \frac{2^{d-m}M}{\rho_1^{j_1}\cdots\rho_d^{j_d}}\,.$$

Some versions of this theorem appear in (Bochner and Martin, 1948, Pages 32, 94-95) and (Trefethen, 2017), however without an explicit bound.

Proof As in the proof of Proposition 15, consider the analytic continuation $f(\mathbf{z})$ on the contour of $E_{\boldsymbol{\rho}}$ which we map into $g(\mathbf{w})$ on $\mathcal{C}_{\boldsymbol{\rho}}$, by defining for $k = 1,\ldots,d$

$$z_k = \frac{w_k + w_k^{-1}}{2}\,.$$

Then, we saw that

$$a_{j_1\ldots j_d} = \frac{2^{d-m}}{\pi^d\left(\rho_1^{2j_1}+\rho_1^{-2j_1}\right)\cdots\left(\rho_d^{2j_d}+\rho_d^{-2j_d}\right)}\oint_{E_{\boldsymbol{\rho}}} f\left(z_1,\ldots,z_d\right)\overline{T_{j_1}(z_1)\cdots T_{j_d}(z_d)}\left|\frac{dz_1\cdots dz_d}{\sqrt{1-z_1^2}\cdots\sqrt{1-z_d^2}}\right|$$

In particular, since $f(z_1,\ldots,z_d)$ is a continuation of $f(x_1,\ldots,x_d)$ to the complex plane, replacing each z_k with $x_k = Re(z_k)$ for $k = 1,\ldots d$ gives

$$a_{j_1\ldots j_d} = \frac{2^{d-m}}{\pi^d}\int_{[-1,1]^d}\frac{f\left(x_1,\ldots,x_d\right)T_{j_1}(x_1)\cdots T_{j_d}(x_d)}{\sqrt{1-x_1^2}\cdots\sqrt{1-x_d^2}}dx_1\cdots dx_d\,.$$

This completes the first part of the proof. For the second part of the proof, we use the bound on f representation in Eq. (29) for the coefficients to obtain:

$$\begin{aligned}
|a_{j_1\ldots j_d}| &= \frac{2^{-m}}{\pi^d}\left|\oint_{\mathcal{C}_{\boldsymbol{\rho}}} w_1^{-j_1-1}\cdots w_d^{-j_d-1}g\left(w_1,\ldots,w_d\right)dw_1\cdots dw_d\right| \\
&\le \frac{2^{-m}M}{\pi^d}\oint_{\mathcal{C}_{\boldsymbol{\rho}}}|w_1|^{-j_1-1}\cdots|w_d|^{-j_d-1}\,dw_1\cdots dw_d \\
&= \frac{2^{-m}M}{\pi^d\rho_1^{j_1+1}\cdots\rho_d^{j_d+1}}\oint_{\mathcal{C}_{\boldsymbol{\rho}}} dw_1\cdots dw_d \\
&= \frac{2^{-m}M}{\pi^d\rho_1^{j_1+1}\cdots\rho_d^{j_d+1}}\cdot 2^d\pi^d\rho_1\cdots\rho_d \\
&= \frac{2^{d-m}M}{\rho_1^{j_1}\cdots\rho_d^{j_d}}\,.
\end{aligned}$$

■

References

M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1972.

H. Avron and V. Sindhwani. High-performance kernel machines with implicit distributed optimization and randomization. *Technometrics*, 58(3):341–349, 2016. doi: 10.1080/00401706.2015.1111261. URL `https://doi.org/10.1080/00401706.2015.1111261`.

H. Avron, V. Sindhwani, J. Yang, and M. W. Mahoney. Quasi-Monte Carlo feature maps for shift-invariant kernels. *Journal of Machine Learning Research*, 17(1):4096–4133, 2016.

H. Avron, M. Kapralov, C. Musco, C. Musco, A. Velingker, and A. Zandieh. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 253–262. JMLR. org, 2017.

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(1):714–751, 2017.

S. Banerjee, A. E. Gelfand, A. O. Finley, and H. Sang. Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70 (4):825–848, 2008. doi: 10.1111/j.1467-9868.2008.00663.x. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2008.00663.x`.

C. Berg, Christensen, and P. Jens Peter Reus amd Ressel. *Harmonic analysis on semigroups: theory of positive definite and related functions*, volume 100. Springer, 1984.

L. Blumenson. A derivation of n-dimensional spherical coordinates. *The American Mathematical Monthly*, 67(1):63–66, 1960.

S. Bochner and W. Martin. *Several complex variables*. Princeton Univ Press, 1948.

K. Choromanski, M. Rowland, T. Sarlos, V. Sindhwani, R. Turner, and A. Weller. The geometry of random features. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84 of *Proceedings of Machine Learning Research*, pages 1–9. PMLR, 09–11 Apr 2018.

J. W. Craig. A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations. In *Proc. IEEE Milcom*, volume 91, pages 571–575, 1991.

N. Cressie and G. Johannesson. Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):209–226, 2008.

T. Dao, C. M. De Sa, and C. Ré. Gaussian quadrature for kernel features. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6107–6117, 2017.

K. Dong, D. Eriksson, H. Nickisch, D. Bindel, and A. G. Wilson. Scalable log determinants for Gaussian process kernel learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6327–6337, 2017.

J. Eidsvik, A. O. Finley, S. Banerjee, and H. Rue. Approximate Bayesian inference for large spatial datasets using predictive process models. *Computational Statistics & Data Analysis*, 56(6):1362–1380, 2012.

A. O. Finley, H. Sang, S. Banerjee, and A. E. Gelfand. Improving the performance of predictive process modeling for large datasets. *Computational statistics & data analysis*, 53(8):2873–2884, 2009.

N. Hale and A. Townsend. Fast and accurate computation of gauss–legendre and gauss–jacobi quadrature nodes and weights. *SIAM Journal on Scientific Computing*, 35(2):A652–A674, 2013.

P. Huang, H. Avron, T. N. Sainath, V. Sindhwani, and B. Ramabhadran. Kernel methods match deep neural networks on TIMIT. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 205–209, 2014. doi: 10.1109/ICASSP.2014.6853587.

M. Katzfuss and N. Cressie. Bayesian hierarchical spatio-temporal smoothing for very large datasets. *Environmetrics*, 23(1):94–107, 2012.

Z. Li, J.-F. Ton, D. Oglic, and D. Sejdinovic. Towards a unified analysis of random fourier features. In *International Conference on Machine Learning*, pages 3905–3914. PMLR, 2019.

J. Mason. Near-best multivariate approximation by Fourier series, Chebyshev series and Chebyshev interpolation. *Journal of Approximation Theory*, 28(4):349–358, 1980.

J. C. Mason. Minimal projections and near-best approximations by multivariate polynomial expansion and interpolation. In *Multivariate Approximation Theory II*, pages 241–254. Springer, 1982.

J. C. Mason and D. C. Handscomb. *Chebyshev polynomials*. Chapman and Hall/CRC, 2002.

M. Munkhoeva, Y. Kapushev, E. Burnaev, and I. Oseledets. Quadrature-based features for kernel approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9147–9156, 2018.

F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark. *NIST handbook of mathematical functions hardback and CD-ROM*. Cambridge university press, 2010.

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2008.

H. Sang and J. Z. Huang. A full scale approximation of covariance functions for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):111–132, 2012. doi: 10.1111/j.1467-9868.2011.01007.x. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2011.01007.x`.

H. Sang, M. Jun, and J. Z. Huang. Covariance approximation for large multivariate spatial data sets with an application to multiple climate model errors. *The Annals of Applied Statistics*, pages 2519–2548, 2011.

V. Scheidemann. *Introduction to complex analysis in several variables*. Springer, 2005.

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2951–2959, 2012.

B. Sriperumbudur and Z. Szabó. Optimal rates for random Fourier features. *Advances in Neural Information Processing Systems (NIPS)*, 28:1144–1152, 2015.

M. L. Stein. *Interpolation of spatial data: Some Theory for Kriging*. Springer Series in Statistics. Springer-Verlag, New York, 1999. ISBN 0-387-98629-4. doi: 10.1007/978-1-4612-1494-6. URL `http://dx.doi.org/10.1007/978-1-4612-1494-6`. Some theory for Kriging.

M. L. Stein. Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics*, 8:1–19, 2014.

M. L. Stein et al. Spatial variation of total column ozone on a global scale. *The Annals of Applied Statistics*, 1(1):191–210, 2007.

D. J. Sutherland and J. Schneider. On the error of random Fourier features. pages 862–871, 2015.

J.-F. Ton, S. Flaxman, D. Sejdinovic, and S. Bhatt. Spatial mapping with Gaussian processes and nonstationary Fourier features. *Spatial Statistics*, 28:59–78, 2018.

A. Townsend and L. N. Trefethen. An extension of chebfun to two dimensions. *SIAM Journal on Scientific Computing*, 35(6):C495–C518, 2013. doi: 10.1137/130908002. URL `https://doi.org/10.1137/130908002`.

L. Trefethen. Multivariate polynomial approximation in the hypercube. *Proceedings of the American Mathematical Society*, 145(11):4837–4844, 2017.

L. N. Trefethen. Is gauss quadrature better than clenshaw–curtis? *SIAM review*, 50(1):67–87, 2008.

L. N. Trefethen. *Approximation theory and approximation practice*, volume 128. Siam, 2013.

S. Ubaru, J. Chen, and Y. Saad. Fast estimation of $tr(f(a))$ via stochastic Lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017.

H. Wang and L. Zhang. Analysis of multivariate Gegenbauer approximation in the hypercube. *Adv Comput Math*, 46:53, 2020.

C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 682–688. MIT Press, 2001. URL `https://proceedings.neurips.cc/paper/2000/file/19de10adbaa1b2ee13f77f679fa1483a-Paper.pdf`.

C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA, 2006.

A. Wilson and H. Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning (ICML)*, pages 1775–1784, 2015.

J. Yang, V. Sindhwani, Q. Fan, H. Avron, and M. Mahoney. Random Laplace feature maps for semigroup kernels on histograms. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 971–978, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.129. URL `https://doi.org/10.1109/CVPR.2014.129`.

T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou. Nyström method vs random Fourier features: A theoretical and empirical comparison. In *Advances in Neural Information Processing Systems (NIPS)*, pages 476–484, 2012.