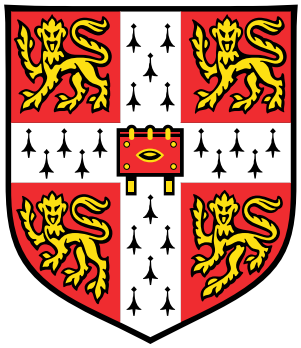
The Generalised Gaussian Process Convolution Model



Wessel Pieter Bruinsma

Department of Engineering

University of Cambridge

This dissertation is submitted for the degree of

Master of Philosophy

Sidney Sussex College

12 August 2016

Declaration

I, Wessel Pieter Bruinsma of Sidney Sussex College, being a candidate for the M.Phil in Machine Learning, Speech and Language Technology, hereby declare that this thesis and the work described in it are my own work, unaided except as may be specified below, and that the thesis does not contain material that has already been used to any substantial extent for a comparable purpose. This thesis, excluding appendices, contains approximately 7300 words.

Wessel Pieter Bruinsma
12 August 2016

Acknowledgements

First and foremost, I thank my supervisor Dr. Richard Turner. His guidance and support throughout the project have been indispensable.

I also thank Dr. Felipe Tobar, whose thoughts have been essential in the development of the project.

Abstract

This thesis formulates the Generalised Gaussian Process Convolution Model (GGPCM), which is a generalisation of the Gaussian Process Convolution Model presented by Tobar et al. [2015b]. The GGPCM provides a theoretical framework for nonparametric kernel models of multidimensional signals defined on multidimensional input spaces. We show that the GGPCM generalises and connects existing work; most notably, we derive a dual formulation of the cross-spectral mixture kernel presented by Ulrich et al. [2015]. Finally, we use the GGPCM to develop the Deep Kernel Model, which presents a new network structure for unsupervised learning.

Contents

List of Figures

List of Tables

List of Models

Notation

General

We use Lagrange's notation for differentation.

$(1,1),\dots,(N,M)$	Shorthand for $$(1,1),\dots,(N,1),(1,2),\dots,(N,2),\dots,(1,M),\dots,(N,M)$$
i	Imaginary unit or, depending on the context, index
$\cdot^*$	Complex conjugate
$\lvert\cdot\rvert$	Modulus or, depending on the context, determinant
$\mathbb{1}_X$	Indicator function
$\mathbb{1}$	Shorthand for $\mathbb{1}_{\{0\}}$
δ	Dirac delta function
R	Reversal function; for any function $f_1 : X \to Y$ the reversal function $R : Y^X \to Y^X$ yields the function $f_2 = R(f_1)$ such that $f_2(x) = f_1(-x)$ for all $x \in X$
$\mathcal{O}$	Bachmann-Landau notation for asymptotic behaviour

Probability Theory

p	Probability measure
$\overset{d}{=}$	Equality in distribution
$\mathbb{E}$	Expected value
$\mathcal{N}(\cdot\,;\boldsymbol{\mu},\boldsymbol{\Sigma})$	Multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$D_{KL}(\cdot\,\|\,\cdot)$	Kullback-Leibler divergence

Linear Algebra

Unbolded symbols x denote scalars, bolded lower-case symbols $\boldsymbol{x}$ vectors and bolded upper-case symbols $\boldsymbol{X}$ matrices or arrays—by array we mean the multidimensional generalisation of the two-dimensional matrix. Vectors are column vectors unless specified otherwise.

$\boldsymbol{I}$	Identity matrix
$\boldsymbol{I}_N$	$N \times N$ identity matrix
$.^T$	Transpose
$.^H$	Conjugate transpose
$\lvert \cdot \rvert$	Determinant or, depending on the context, modulus
$\lVert \cdot \rVert_p$	p-norm
$\lVert \cdot \rVert_F$	Frobenius norm
$\otimes$	Kronecker product
$\operatorname{diag}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$	Block diagonal matrix; yields

$$\begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_2 & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{X}_{N-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{X}_N \end{bmatrix}$$

$\operatorname{circ}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$	Block circulant matrix; yields

$$\begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_N & \cdots & \boldsymbol{X}_3 & \boldsymbol{X}_2 \\ \boldsymbol{X}_2 & \boldsymbol{X}_1 & \cdots & \boldsymbol{X}_4 & \boldsymbol{X}_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{X}_{N-1} & \boldsymbol{X}_{N-2} & \cdots & \boldsymbol{X}_1 & \boldsymbol{X}_N \\ \boldsymbol{X}_N & \boldsymbol{X}_{N-1} & \cdots & \boldsymbol{X}_2 & \boldsymbol{X}_1 \end{bmatrix}$$

$\text{stoep}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$ Symmetric block Toeplitz matrix; yields

$$\begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 & \cdots & \boldsymbol{X}_{N-1} & \boldsymbol{X}_N \\ \boldsymbol{X}_2 & \boldsymbol{X}_1 & \cdots & \boldsymbol{X}_{N-2} & \boldsymbol{X}_{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{X}_{N-1} & \boldsymbol{X}_{N-2} & \cdots & \boldsymbol{X}_1 & \boldsymbol{X}_2 \\ \boldsymbol{X}_N & \boldsymbol{X}_{N-1} & \cdots & \boldsymbol{X}_2 & \boldsymbol{X}_1 \end{bmatrix}$$

vec Matrix vectorisation function; if $\boldsymbol{X} \in \mathbb{R}^{N \times M}$, then

$$\text{vec}\, \boldsymbol{X} = \begin{bmatrix} X_{1,1} & \cdots & X_{N,M} \end{bmatrix}^T$$

$X_{i_1,\ldots,i_N}$ Vector, matrix and array indexing; yields element $(i_1, \ldots, i_N)$ of the array $\boldsymbol{X}$

$\boldsymbol{X}_{i,:}$ Fixation of first dimension of a matrix and ranging over the second; yields vector containing the i'th row of $\boldsymbol{X}$

$\boldsymbol{X}_{:,i}$ Fixation of second dimension of a matrix and ranging over the first; yields vector containing the i'th column of $\boldsymbol{X}$

$\boldsymbol{X}_{\underbrace{i_1,\ldots,i_n,:,\ldots,:}_{N \text{ indices}}}$ Fixation of an arbitrary subset of dimensions and ranging over the remaining; yields the array whose $(i_{n+1}, \ldots, i_N)$'th element is given by $X_{i_1,\ldots,i_N}$

$f(\boldsymbol{X})$ Element-wise function application; if $\boldsymbol{X}$ is M-dimensional—$\boldsymbol{X} \in \mathbb{R}^{N_1 \times \cdots \times N_M}$—but f takes a $(M-m)$-dimensional object—$f : \mathbb{R}^{N_{m+1} \times \cdots \times N_M} \to C$—then $f(\boldsymbol{X})$ is the m-dimensional array whose $(i_1, \ldots, i_m)$'th element is given by $f(\boldsymbol{X}_{i_1,\ldots,i_m,:,\ldots,:})$

$f(\boldsymbol{X}, \boldsymbol{Y})$ Element-wise function application; if $\boldsymbol{X} \in \mathbb{R}^{N_1 \times \cdots \times N_M}$ and $\boldsymbol{X} \in \mathbb{R}^{N_1 \times \cdots \times N_K}$, but $f : \mathbb{R}^{N_{m+1} \times \cdots \times N_M} \times \mathbb{R}^{N_{k+1} \times \cdots \times N_K} \to C$, then $f(\boldsymbol{X}, \boldsymbol{Y})$ is the array whose $(i_1, \ldots, i_m, j_1, \ldots, j_k)$'th element is given by $f(\boldsymbol{X}_{i_1,\ldots,i_m,:,\ldots,:}, \boldsymbol{Y}_{i_1,\ldots,i_k,:,\ldots,:})$

Miscellaneous

$*$	Convolution; for matrix-valued functions $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ defined as $$(\boldsymbol{F}_1 * \boldsymbol{F}_2)(\boldsymbol{x}) = \int \boldsymbol{F}_1(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{F}_2(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}$$
$\boldsymbol{\mathcal{F}}_N$	$N \times N$ unitary discrete Fourier transform matrix
DFT	Discrete Fourier transform
$\mathcal{F}_{\boldsymbol{y}}\{f\}(\boldsymbol{x})$	Continuous Fourier transform of f; defined as $$\mathcal{F}_{\boldsymbol{y}}\{f\}(\boldsymbol{x}) = \int f(\boldsymbol{y}) \exp(-2\pi i \boldsymbol{x}^T\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}$$
$\mathcal{F}_f(\boldsymbol{x})$	Shorthand for $\mathcal{F}_{\boldsymbol{y}}\{f\}(\boldsymbol{x})$
$\mathcal{S}_{\boldsymbol{y}}\{f,g\}(\boldsymbol{x})$	Cross-spectral density between two wide-sense stationary processes f and g; equal to $$\mathcal{S}_{\boldsymbol{y}}\{f,g\}(\boldsymbol{x}) = \mathcal{F}_{\boldsymbol{z}}\{\mathbb{E}[f(\boldsymbol{x}+\boldsymbol{z})g^*(\boldsymbol{x})]\}(\boldsymbol{y})$$
$\mathcal{S}_{f,g}(\boldsymbol{y})$	Shorthand for $\mathcal{S}_{\boldsymbol{y}}\{f,g\}(\boldsymbol{x})$
$(C, \boldsymbol{A}, \boldsymbol{b}, c)$	See Appendix F.2; shorthand for $$C\exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{b} + c\right)$$
$(C, \boldsymbol{A})$	See Appendix F.3; shorthand for $(C, \boldsymbol{A}\otimes\boldsymbol{I}, \boldsymbol{0}, 0)$

Acronyms

AKM	Approximate Kernel Model (Model 6)
CGPM	Collaborative Gaussian processes model [Nguyen and Bonilla, 2014]
CMOGPM	Convolved multi-output Gaussian process model [Álvarez and Lawrence, 2011]
CSMK	Cross-spectral mixture kernel [Ulrich et al., 2015]
GGPCM	Generalised Gaussian Process Convolution Model (Model 4)
GPCM	Gaussian Process Convolution Model [Tobar et al., 2015b]
GPRN	Gaussian process regression network [Wilson et al., 2012]
ICM	Intrinsic coregionalisation model [Goovaerts, 1997]

LCM	Linear coregionalisation model [Goovaerts, 1997]
LFM	Latent force model [Álvarez et al., 2009]
MTGPM	Multi-task Gaussian process [Bonilla et al., 2008]
NKM	Nonparametric Kernel Model (Model 5)
PSD	Power spectral density
SLFM	Semiparametric latent factor model [Teh and Seeger, 2005]
SMK	Spectral mixture kernel [Wilson and Adams, 2013]

1 Introduction

1.1 Motivation

Gaussian processes elegantly provide means to model an unknown function. They give rise to Bayesian regression models in which one maintains a posterior distribution over the unknown function as evidence is accumulated; Figure 1.1 illustrates this process. Gaussian processes have been succesfully applied in a wide variety of contexts. Rasmussen and Williams [2006] provide an excellent overview.

Gaussian processes are *nonparametric* models. As opposed to *parametric* models, there is no finite number of parameters that parametrises a Gaussian processes. Instead, the number of parameters grows with the amount of evidence that is accumulated. This property allows Gaussian processes to learn complex functions if plenty of evidence is available. Conversely, this property makes them robust against overfitting if only little evidence is available.

Their expressiveness and robustness however, come at a cost: Gaussian process models are often computationally expensive and one is forced to choose a *kernel*. The kernel of a Gaussian process reflects one's assumption on how the unknown function autocovaries.[1] The choice of the kernel is crucial; Figure 1.2 illustrates that the posterior distribution can wildly vary for different kernels. Unfortunately, determining which kernel to use is hard; the kernel thus poses a difficult design problem. We specifically refer to this problem as the *kernel design problem*.

A number of recent works address the kernel design problem: Duvenaud [2014] presents a way to search over a space of kernels through composition of existing kernels, and Wilson and Adams [2013] present a flexible kernel by modellings its power spectral

[1] The autocovariance of an unknown function f specifies the covariance between any two function values $f(t_1)$ and $f(t_2)$.

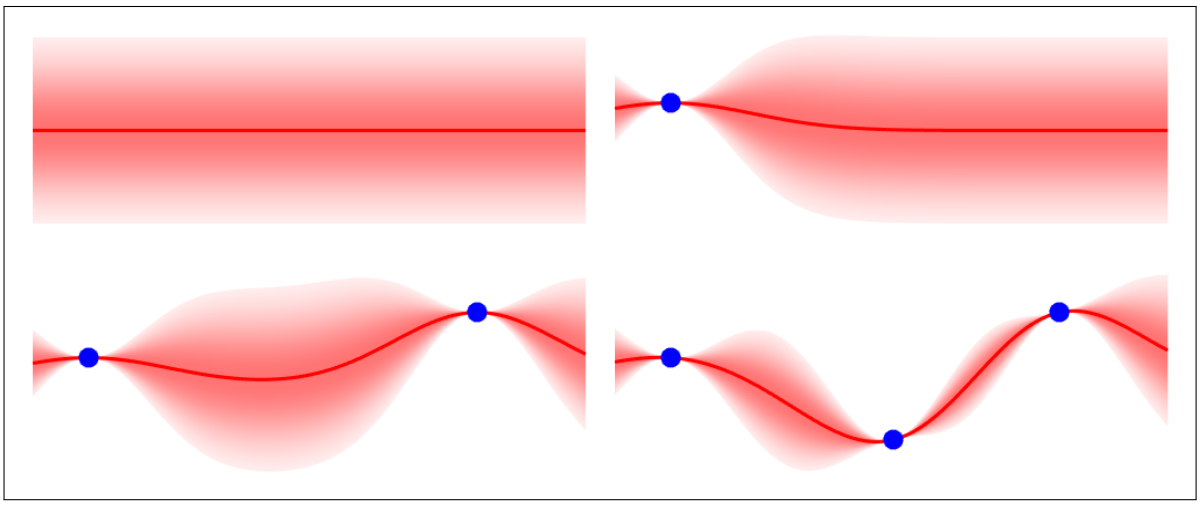


Figure 1.1: Posterior distribution over an unknown function as evidence is accumulated. The evidence is represented by blue dots. The red line represents the mean of the posterior distribution and the gradient indicates the marginal variance up to two standard deviations.

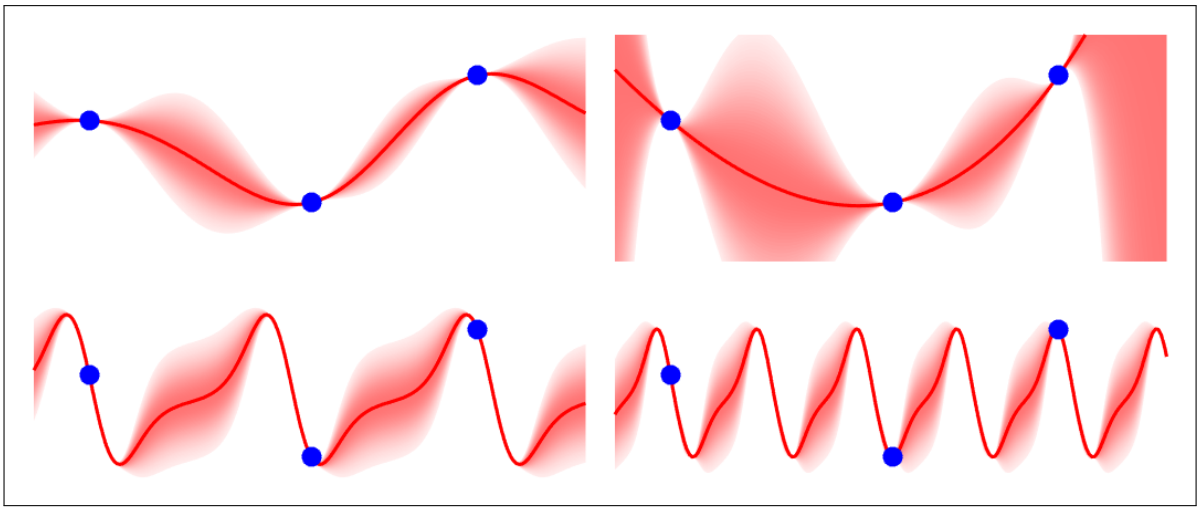


Figure 1.2: Posterior distribution over an unknown function for different kernels. The evidence is represented by the blue dots and is the same as in Figure 1.1. The red line represents the mean of the posterior distribution and the gradient indicates the marginal variance up to two standard deviations.

density with a sum of Gaussians. As noted by Tobar et al. [2015b], these approaches do not completely resolve the kernel design problem: computational expenses restrict the space of kernels that can be searched, and the large number of parameters introduced by the sum of Gaussians reintroduces the problem of overfitting. Instead, Tobar et al. [2015b] note that the kernel constitutes just another unknown function and thus propose to model it by another Gaussian; the results are encouraging. They call their model the Gaussian Process Convolution Model (GPCM).

The GPCM has only been formulated for one-dimensional signals defined on one-dimensional input spaces. Extending this model to multidimensional signals defined on multidimensional input spaces would enable numerous applications in econometrics, geostatistics and signal processing, among others.

1.2 Contribution of Thesis and Outline

The main contribution of this thesis is the formulation of the Generalised Gaussian Process Convolution Model (GGPCM). The GGPCM provides a theoretical framework for nonparametric kernel models of multidimensional signals defined on multidimensional input spaces.

Chapter 2 develops a general-purpose model of dynamical signals. This model will suggest a generalisation of the GPCM: the GGPCM. Thus, in hindsight, Chapter 2 motivates why this particular generalisation is appropriate.

Chapter 3 develops the GGPCM. We use the GGPCM to address the kernel design problem in multi-output Gaussian processes on multidimensional input spaces. Furthermore, we develop a dual formulation of the cross-spectral mixture kernel [Ulrich et al., 2015] and we show how the GGPCM connects to existing work.

Chapter 4 examines the GGPCM's connection to existing multi-output Gaussian process models. We provide an overview of the current literature.

Chapter 5 develops the Deep Kernel Model, which is a new network structure for unsupervised learning. We sample from the Deep Kernel Model and investigate its properties.

2 | Modelling Dynamical Signals

2.1 Introduction

For many years scientists have used the concept of a *system* to study dynamic processes of diverse nature. A system is a mathematical abstraction that is used to describe properties of the process we intend to study. *Dynamical systems* describe processes that evolve in time; the system then usually describes a transformation $T : (\mathbb{R}^M)^{\mathbb{R}} \to (\mathbb{R}^N)^{\mathbb{R}}$ from an input signal $\boldsymbol{x}$ to an output signal $\boldsymbol{f} = T(\boldsymbol{x})$—the latter is also called the *system response*.

This chapter develops a general-purpose model of dynamical signals. In doing so we take a systems-modelling perspective by postulating that every signal is the response of a particular system. Thus, equivalently, we develop a model of system responses. We therefore utilise a general description of dynamical *systems* to construct the desired model of dynamical *signals*.

2.2 The Linear State-Space Model

A widely used dynamical system model is one that describes mechanical, electrical, hydraulic, and thermal systems, among others. These systems are all compositions of resistive, inductive, capactive and memristive elements that are connected through lossless transfer of energy; hence, they allow for a unified description. This description embodies the concept of *system state*, whose manifestation is a vector $\boldsymbol{s}$ that at any time determines future output given future input, thus rendering past input irrelevant. More precisely, the description is a so-called *state-space model* whose general *linear* form is given by

$$\boldsymbol{s}'(t) = \boldsymbol{A}(t)\boldsymbol{s}(t) + \boldsymbol{B}(t)\boldsymbol{x}(t), \tag{2.1}$$

$$\boldsymbol{f}(t) = \boldsymbol{C}(t)\boldsymbol{s}(t) + \boldsymbol{D}(t)\boldsymbol{x}(t). \tag{2.2}$$

Examples of systems that are succesfully described by linear state-space models are ubiquitous: linear state-space models describe electronics that are essential in everyday life; systems that control your car, airplanes and even rockets; and many phenomena in nature.

2.3 A General-Purpose Model of Dynamical Signals

Suppose that $\boldsymbol{f}$ is a dynamical signal. By the postulate that every signal is the response of some system, we can assume that some system generated $\boldsymbol{f}$. Now, Section 2.2 suggests that we can safely assume this system to be of the form of Equations (2.1) and (2.2). In that case Appendix A.1 shows that $\boldsymbol{f}$ admits the following parametrisation:

Model 1 (Time-Variant Model)**.**

$$\boldsymbol{f}(t) = \int_{\mathbb{R}} \boldsymbol{H}(t,\tau)\boldsymbol{x}(\tau)\,\mathrm{d}\boldsymbol{\tau}.$$

Observe that $\boldsymbol{f}$ is parametrised in terms of some matrix-valued function $\boldsymbol{H}$—the *impulse response*[1]—and the postulated input $\boldsymbol{x}$.

In many applications of state-space models $\boldsymbol{A}(t)$, $\boldsymbol{B}(t)$, $\boldsymbol{C}(t)$ and $\boldsymbol{D}(t)$ are approximately constant.[2] In that case Appendix A.2 shows that $\boldsymbol{H}(t,\tau) = \boldsymbol{H}(t-\tau)$. Model 1 then reduces to the following model:

Model 2 (Time-Invariant Model)**.**

$$\boldsymbol{f}(t) = \int_{\mathbb{R}} \boldsymbol{H}(t-\tau)\boldsymbol{x}(\tau)\,\mathrm{d}\tau = (\boldsymbol{H} * \boldsymbol{x})(t).$$

Observe that Model 2 attains the form of a convolution.

[1] Letting $\boldsymbol{x}(t) = \delta(t-t_0)\boldsymbol{x}_0$, $t \in \mathbb{R}$ yields the response $\boldsymbol{f}(t) = \boldsymbol{H}(t,t_0)$, $t \in \mathbb{R}$. That is, $\boldsymbol{H}$ is the system response of an *impulse excitation*; in this sense $\boldsymbol{H}$ is called the *impulse response*.

[2] For example, consider an electric circuit. In that case, approximating $\boldsymbol{A}(t)$, $\boldsymbol{B}(t)$, $\boldsymbol{C}(t)$ and $\boldsymbol{D}(t)$ as time invariant roughly corresponds to approximating resistances, capacitances and inductances as time invariant. This approximation is often reasonable.

2.4 Conclusion

We have developed a general-purpose model of dynamical signals: Model 2. Unfortunately, Model 2 leaves the impulse reponse $\boldsymbol{H}$ and postulated input $\boldsymbol{x}$ unspecified. This issue will be addressed in Chapter 3.

3 | The Generalised Gaussian Process Convolution Model

3.1 Introduction

Chapter 2 developed a general-purpose model of dynamical signals: the Time-Invariant Model (Model 2). By extending Model 2's input space to $\mathbb{R}^K$ we derive the Multidimensional Time-Invariant Model:

Model 3 (Multidimensional Time-Invariant Model)**.**

$$\boldsymbol{f}(\boldsymbol{t}) = \int_{\mathbb{R}^K} \boldsymbol{H}(\boldsymbol{t} - \boldsymbol{\tau})\boldsymbol{x}(\boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau} = (\boldsymbol{H} * \boldsymbol{x})(\boldsymbol{t})$$

Model 3 is a deterministic model of multidimensional signals defined on multidimensional input spaces. This chapter shows that a stochastic version of Model 3 can be used to address the kernel design problem in multi-output Gaussian processes on multidimensional input spaces.

Recall that $\boldsymbol{f} : \mathbb{R}^K \to \mathbb{R}^N$, $\boldsymbol{H} : \mathbb{R}^K \to \mathbb{R}^{M\times N}$ and $\boldsymbol{x} : \mathbb{R}^K \to \mathbb{R}^M$.

3.2 Gaussian Process Regression

A Gaussian process defines a distribution over functions. More precisely, a stochastic process $f(t)$, $t \in \mathbb{R}$ is Gaussian if and only if for every $\boldsymbol{t} = \begin{bmatrix} t_1 & \cdots & t_T \end{bmatrix}^T$ it holds that $f(\boldsymbol{t}) = \begin{bmatrix} f(t_1) & \cdots & f(t_T) \end{bmatrix}^T$ is multivariate Gaussian distributed. We denote $f \sim \mathcal{GP}(m_f, \mathcal{K}_f)$ where m_f and $\mathcal{K}_f$ denote respectively the mean function and kernel of f. This implies that $f(\boldsymbol{t}) \sim \mathcal{N}[m_f(\boldsymbol{t}), \mathcal{K}_f(\boldsymbol{t}, \boldsymbol{t})]$. Furthermore, we usually let $m_f = 0$ without loss of generality [Rasmussen and Williams, 2006] and denote $\boldsymbol{K}_{\boldsymbol{f}_1,\boldsymbol{f}_2} = \mathcal{K}_f(\boldsymbol{t}_1, \boldsymbol{t}_2)$ where $\boldsymbol{f}_1 = f(\boldsymbol{t}_1)$ and $\boldsymbol{f}_2 = f(\boldsymbol{t}_2)$.

Suppose that we observe data $\boldsymbol{y}$ at $\boldsymbol{t}_y$ generated by the process $y(t) = f(t) + \varepsilon(t)$, $t \in \mathbb{R}$ where $\varepsilon(t) \sim \mathcal{N}(0, \sigma^2)$, $t \in \mathbb{R}$. For unobserved function values $\boldsymbol{f}^*$ at $\boldsymbol{t}_{f^*}$ it then holds that

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}^* \end{bmatrix} \Bigg| \begin{bmatrix} \boldsymbol{t}_y \\ \boldsymbol{t}_{f^*} \end{bmatrix}, \boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{y},\boldsymbol{y}} + \sigma^2 \boldsymbol{I} & \boldsymbol{K}_{\boldsymbol{y},\boldsymbol{f}^*} \\ \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{y}} & \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} \end{bmatrix}\right)$$

where $\boldsymbol{\theta}$ denotes σ and the parameters of $\mathcal{K}_f$ and m_f—$\boldsymbol{\theta}$ are also called the *hyperparameters*. We usually omit explicit conditioning $\boldsymbol{t}_y$ and $\boldsymbol{t}_{f^*}$. According to Appendix B.1 we can perform prediction via

$$\boldsymbol{f}^* \,|\, \boldsymbol{y}, \boldsymbol{\theta} \sim \mathcal{N}[\boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{y}}(\boldsymbol{K}_{\boldsymbol{y},\boldsymbol{y}} + \sigma^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} - \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{y}}(\boldsymbol{K}_{\boldsymbol{y},\boldsymbol{y}} + \sigma^2 \boldsymbol{I})^{-1}\boldsymbol{K}_{\boldsymbol{y},\boldsymbol{f}^*}].$$

This is the Gaussian process regression framework.

The values of the hyperparameters should be chosen such that the model best fits the true data-generating process. In this sense, as argued by Rasmussen and Williams [2006], their values can be determined by maximising the *marginal likelihood*—also called the *evidence*—given by

$$p(\boldsymbol{y} \,|\, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \underbrace{|\boldsymbol{\Sigma}|^{-1/2}}_{\text{complexity penalty}} \underbrace{\exp\left[-\frac{1}{2}\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{y} - \boldsymbol{\mu})\|_2^2\right]}_{\text{compatability with data}}$$

for some $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Observe that the marginal likelihood is a trade-off between the model's compatibility with the data and $|\boldsymbol{\Sigma}|^{-1/2}$ being small; that is, the evidence is large if the model is compatible with the data, but only if the model is not too *complex* in the sense that it is compatible with *any* data—if $\boldsymbol{\Sigma}^{-1/2}$ is such that $\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{y} - \boldsymbol{\mu})\|_2^2$ is small for any $\boldsymbol{y}$, then $|\boldsymbol{\Sigma}^{-1/2}| = |\boldsymbol{\Sigma}|^{-1/2}$ is small and thereby the evidence must be small. Thus maximising the evidence tends to explains the data in a *simple* way. This tendency is commonly recognised as a manifestation of *Occam's razor* [MacKay, 2002].

Finally, Appendix C shows how Gaussian processes can be used to define distributions over multivariate matrix-valued functions.

3.3 The Generalised Gaussian Process Convolution Model

Model 3 leaves the input signal $\boldsymbol{x}$ and impulse response $\boldsymbol{H}$ unspecified. Since they constitute unknown functions, it is sensible to model them by two Gaussian processes; we then obtain the Generalised Gaussian Process Convolution Model (GGPCM):

Model 4 (Generalised Gaussian Process Convolution Model (GGPCM))**.** *Draw*

$$\begin{aligned}\boldsymbol{H} &\sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\mathcal{K}}_{\boldsymbol{H}}),\\ \boldsymbol{x} &\sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\mathcal{K}}_{\boldsymbol{x}}),\\ \boldsymbol{\varepsilon} &\sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\Lambda}^2)\end{aligned}$$

independently for some kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{x}}$, some kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}$, and some diagonal matrix $\boldsymbol{\Lambda}$. Then observations are generated by $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\varepsilon} = \boldsymbol{H} * \boldsymbol{x} + \boldsymbol{\varepsilon}$.

In the remainder of this chapter we study the case that $\boldsymbol{\mathcal{K}}_{\boldsymbol{x}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \delta(\boldsymbol{t}_1 - \boldsymbol{t}_2)\boldsymbol{I}$—we let $\boldsymbol{x}$ be white noise. The general case will be studied in Chapter 4.

Observe that $\boldsymbol{f} \mid \boldsymbol{H}$ is a linear combination of Gaussian processes. Hence $\boldsymbol{f} \mid \boldsymbol{H}$ is another Gaussian process, which thus can be identified by its mean function and kernel:

$$\begin{aligned}\mathbb{E}[\boldsymbol{f}(\boldsymbol{t}) \mid \boldsymbol{H}] &= \int_{\mathbb{R}^K} \boldsymbol{H}(\boldsymbol{t} - \boldsymbol{\tau})\mathbb{E}[\boldsymbol{x}(\boldsymbol{\tau})] \,\mathrm{d}\boldsymbol{\tau} = \boldsymbol{0},\\ \boldsymbol{\mathcal{K}}_{\boldsymbol{f} \mid \boldsymbol{H}}(\boldsymbol{t}_1, \boldsymbol{t}_2) &= \mathbb{E}[\boldsymbol{f}(\boldsymbol{t}_1)\boldsymbol{f}^T(\boldsymbol{t}_2) \mid \boldsymbol{H}]\\ &= \int_{\mathbb{R}^K}\int_{\mathbb{R}^K} \boldsymbol{H}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1) \underbrace{\mathbb{E}[\boldsymbol{x}(\boldsymbol{\tau}_1)\boldsymbol{x}^T(\boldsymbol{\tau}_2)]}_{\delta(\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2)\boldsymbol{I}} \boldsymbol{H}^T(\boldsymbol{t}_2 - \boldsymbol{\tau}_2) \,\mathrm{d}\boldsymbol{\tau}_1 \,\mathrm{d}\boldsymbol{\tau}_2\\ &= \int_{\mathbb{R}^K} \boldsymbol{H}[\boldsymbol{\tau} - (\boldsymbol{t}_2 - \boldsymbol{t}_1)]\boldsymbol{H}^T(\boldsymbol{\tau}) \,\mathrm{d}\boldsymbol{\tau}\\ &= [R(\boldsymbol{H}) * \boldsymbol{H}^T](\boldsymbol{t}_2 - \boldsymbol{t}_1)\\ &= \boldsymbol{\mathcal{K}}_{\boldsymbol{f} \mid \boldsymbol{H}}(\boldsymbol{t}_1 - \boldsymbol{t}_2)\end{aligned} \tag{3.1}$$

where R is the reversal function. We have established the following equivalent model:

Model 5 (Nonparametric Kernel Model (NKM))**.** *Draw*

$$\begin{aligned}\boldsymbol{H} &\sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\mathcal{K}}_{\boldsymbol{H}}),\\ \boldsymbol{\varepsilon} &\sim \mathcal{GP}[\boldsymbol{0}, \delta(\boldsymbol{t}_1 - \boldsymbol{t}_2)\boldsymbol{\Lambda}^2]\end{aligned}$$

independently for some kernel $\mathcal{K}_{\boldsymbol{H}}$ *and some diagonal matrix* $\boldsymbol{\Lambda}$*. Afterwards draw*

$$\boldsymbol{f} \,|\, \boldsymbol{H} \sim \mathcal{GP}\{\boldsymbol{0}, [R(\boldsymbol{H}) * \boldsymbol{H}^T](\boldsymbol{t}_2 - \boldsymbol{t}_1)\}.$$

Then observations are generated by $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\varepsilon}$*.*

This equivalent formulation reveals that white noise excitation in Model 4 yields ordinary Gaussian process regression in which the kernel is modelled nonparametrically. We cannot model the kernel of $\boldsymbol{f}$ simply with a sample from a Gaussian process—a kernel has to be positive semidefinite, which in general a sample from a Gaussian process is not. Thus Model 5 shows a way to nonparametrically model a positive-semidefinite matrix-valued function. We verifiy that $\boldsymbol{\mathcal{K}}_{\boldsymbol{f}\,|\,\boldsymbol{H}} = R(\boldsymbol{H}) * \boldsymbol{H}^T$ is indeed positive semidefinite: Let $\boldsymbol{g} : \mathbb{R}^K \to \mathbb{R}^N$ be square integrable. Then

$$\begin{aligned}\int_{\mathbb{R}^{2K}} \boldsymbol{g}^T(\boldsymbol{t}_1)\boldsymbol{\mathcal{K}}(\boldsymbol{t}_1, \boldsymbol{t}_2)\boldsymbol{g}(\boldsymbol{t}_2)\,\mathrm{d}\boldsymbol{t}_1\,\mathrm{d}\boldsymbol{t}_2 &= \int_{\mathbb{R}^{3K}} \boldsymbol{g}^T(\boldsymbol{t}_1)\boldsymbol{H}[\boldsymbol{\tau} - (\boldsymbol{t}_2 - \boldsymbol{t}_1)]\boldsymbol{H}^T(\boldsymbol{\tau})\boldsymbol{g}(\boldsymbol{t}_2)\,\mathrm{d}\boldsymbol{t}_1\,\mathrm{d}\boldsymbol{t}_2\,\mathrm{d}\boldsymbol{\tau}\\ &= \int_{\mathbb{R}^{3K}} \boldsymbol{g}^T(\boldsymbol{t}_1)\boldsymbol{H}(\boldsymbol{t}_1 - \boldsymbol{\tau})\,\mathrm{d}\boldsymbol{t}_1\boldsymbol{H}^T(\boldsymbol{t}_2 - \boldsymbol{\tau})\boldsymbol{g}(\boldsymbol{t}_2)\,\mathrm{d}\boldsymbol{t}_2\,\mathrm{d}\boldsymbol{\tau}\\ &= \int_{\mathbb{R}^K} \left\| \int_{\mathbb{R}^K} \boldsymbol{H}^T(\boldsymbol{t} - \boldsymbol{\tau})\boldsymbol{g}(\boldsymbol{t})\,\mathrm{d}\boldsymbol{t} \right\|^2 \mathrm{d}\boldsymbol{\tau}\\ &\geq 0.\end{aligned}$$

Alternatively, Model 5 can be interpreted in the frequency domain. It is clear that $\boldsymbol{f} \,|\, \boldsymbol{H}$

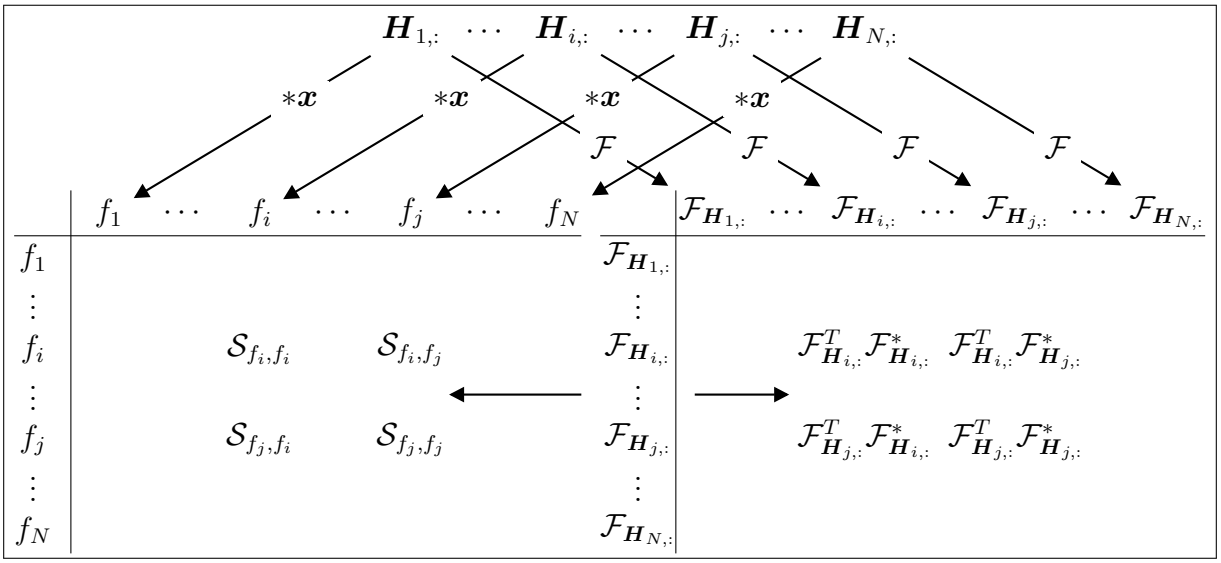


Figure 3.1: Generation of the outputs of in Model 5 and relationships between their cross-spectral densities

is wide-sense stationary; hence,

$$\begin{aligned}
\mathcal{S}_{f_i,f_j\,|\,\boldsymbol{H}}(\boldsymbol{\xi}) &= \mathcal{F}_{\boldsymbol{\tau}}\{\mathcal{K}_{f_i,f_j\,|\,\boldsymbol{H}}(\boldsymbol{t}+\boldsymbol{\tau},\boldsymbol{t})\}(\boldsymbol{\xi}) \\
&= \sum_{m=1}^{M}\int_{\mathbb{R}^{2K}} H_{i,m}(\boldsymbol{\tau}'+\boldsymbol{\tau})H_{j,m}(\boldsymbol{\tau}')\exp(-2\pi i\boldsymbol{\xi}^T\boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau}'\,\mathrm{d}\boldsymbol{\tau} \\
&= \sum_{m=1}^{M}\int_{\mathbb{R}^{2K}} H_{i,m}(\boldsymbol{\tau})\exp(-2\pi i\boldsymbol{\xi}^T\boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau}[H_{j,m}(\boldsymbol{\tau}')\exp(-2\pi i\boldsymbol{\xi}^T\boldsymbol{\tau}')]^*\,\mathrm{d}\boldsymbol{\tau}' \\
&= \sum_{m=1}^{M}\mathcal{F}_{H_{i,m}}(\boldsymbol{\xi})\mathcal{F}^*_{H_{j,m}}(\boldsymbol{\xi}) \qquad (3.2)\\
&= \mathcal{F}^T_{\boldsymbol{H}_{i,:}}(\boldsymbol{\xi})\mathcal{F}^*_{\boldsymbol{H}_{j,:}}(\boldsymbol{\xi}). \qquad (3.3)
\end{aligned}$$

Since each $H_{i,j}$ is modelled nonparametrically, each $\mathcal{F}_{H_{i,j}}$ is modelled nonparametrically. Model 5 therefore models the cross-spectral densities $\mathcal{S}_{f_i,f_j}$ nonparametrically by complex inner products between the stacked spectra $\mathcal{F}_{\boldsymbol{H}_{i,:}}$ and $\mathcal{F}_{\boldsymbol{H}_{j,:}}$ associated to f_i and f_j. This provides insight in how correlations between the outputs are induced; Figure 3.1 depicts in more detail how Model 5 generates outputs and how their cross-spectral densities relate.

Finally, Model 5 can be interpreted as a continuous-time moving average model. Equivalently, consider white noise excitation in Model 4. Then discretising the input space

results in a model of the form

$$\boldsymbol{f_t} = \sum_{\boldsymbol{\tau} \in \mathbb{R}^K} \boldsymbol{H_{t-\tau} x_\tau}$$

where $\boldsymbol{x}$ is white noise. This is exactly a multi-output moving average model on a multidimensional input space. Hence Model 5 is a continuous-time Bayesian multi-output moving average model on a multidimensional input space.

3.3.1 Interpretation and Choice of the Kernel in the Nonparametric Kernel Model

In this section we resolve an apparent modelling issue concerning Model 5: it is not clear how to interpret and choose $\boldsymbol{\mathcal{K}_H}$ and thereby M—recall that $\boldsymbol{\mathcal{K}_H} : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}^{NM \times NM}$.

First, we examine M. We already determined that $\boldsymbol{f \mid H}$ is a Gaussian process with zero mean function. Thus Model 5 attains its full expressive power if it can generate any $\boldsymbol{\mathcal{K}_{f \mid H}}$, or equivalenty any $\mathcal{F}\{\boldsymbol{\mathcal{K}_{f \mid H}}\} = \boldsymbol{S_{f \mid H}}$. To begin with,

$$\boldsymbol{S^H_{f \mid H}} = \mathcal{F}^*_{\boldsymbol{\tau}}\{\mathbb{E}^T[\boldsymbol{f(t+\tau)f}^T\boldsymbol{(t)}]\} = \mathcal{F}_{\boldsymbol{\tau}}\{\mathbb{E}[\boldsymbol{f(t)f}^T\boldsymbol{(t-\tau)}]\} = \boldsymbol{S_{f \mid H}}$$

shows that $\boldsymbol{S_{f \mid H}}$ is Hermitian. Therefore, by the Complex Spectral Theorem, generating any $\boldsymbol{S_{f \mid H}}$ is equivalent to generating any spectral decomposition. Now, utilising Equation (3.2) to express

$$\boldsymbol{S_{f \mid H}(\xi)} = \sum_{m=1}^{M} \mathcal{F}_{\boldsymbol{H}_{:,m}}(\boldsymbol{\xi}) \mathcal{F}^H_{\boldsymbol{H}_{:,m}}(\boldsymbol{\xi})$$

shows that any spectral decomposition can be generated only if $M \geq N$. Thus $M = N$ is the minimal M for which Model 5 attains its full expressive power. Conversely, Model 5 has limited expressive power if $M < N$. More precisely, Equation (3.3) then shows that at each frequency $\boldsymbol{\xi}$ at most M outputs can be independent; f_i and f_j are independent if and only if $\mathcal{S}_{f_i, f_j \mid \boldsymbol{H}}(\boldsymbol{\xi}) = \mathcal{F}^H_{\boldsymbol{H}_{j,:}} \mathcal{F}_{\boldsymbol{H}_{i,:}} = 0$, and at most M vectors $\mathcal{F}_{\boldsymbol{H}_{i,:}}$ can simultaneously be orthogonal as they are elements of $\mathbb{C}^M$. We let $M = N$ in further development of Model 5.

Second, we examine $\boldsymbol{\mathcal{K}_H}$. It turns out that we must be careful in choosing $\boldsymbol{\mathcal{K}_H}$. Suppose the simplest case; that is, let the components of $\boldsymbol{H}$ be independent and share an

exponentiated-quadratic kernel:

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \sigma_h^2 \exp(-\gamma \|\boldsymbol{t}_1 - \boldsymbol{t}_2\|^2) \boldsymbol{I}.$$

Having already observed that Model 5 induces correlations between its outputs justifies modelling the components of $\boldsymbol{H}$ independently. We call $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}$ the diagonal multi-output exponentiated-quadratic kernel. Now, by equivalently considering white noise excitation in Model 4,

$$\begin{aligned}
\mathbb{E}[f_i^2(\boldsymbol{t})] &= \int_{\mathbb{R}^{2K}} \mathbb{E}[\boldsymbol{H}_{i,:}^T(\boldsymbol{t} - \boldsymbol{\tau}_1)\boldsymbol{x}(\boldsymbol{\tau}_1)\boldsymbol{x}(\boldsymbol{\tau}_2)^T \boldsymbol{H}_{i,:}(\boldsymbol{t} - \boldsymbol{\tau}_2)] \, \mathrm{d}\boldsymbol{\tau}_1 \, \mathrm{d}\boldsymbol{\tau}_2 \\
&= \int_{\mathbb{R}^{2K}} \operatorname{tr} \mathbb{E}[\underbrace{\boldsymbol{H}_{i,:}(\boldsymbol{t} - \boldsymbol{\tau}_1)\boldsymbol{H}_{i,:}^T(\boldsymbol{t} - \boldsymbol{\tau}_2)}_{\boldsymbol{\mathcal{K}}_{\boldsymbol{H}_{i,:}}[(\boldsymbol{t}-\boldsymbol{\tau}_1)-(\boldsymbol{t}-\boldsymbol{\tau}_2)]} \underbrace{\boldsymbol{x}(\boldsymbol{\tau}_1)\boldsymbol{x}^T(\boldsymbol{\tau}_2)}_{\delta(\boldsymbol{\tau}_1-\boldsymbol{\tau}_2)\boldsymbol{I}}] \, \mathrm{d}\boldsymbol{\tau}_1 \, \mathrm{d}\boldsymbol{\tau}_2 \\
&= \int_{\mathbb{R}^K} \operatorname{tr}(\sigma_h^2 \boldsymbol{I}) \, \mathrm{d}\boldsymbol{\tau} \\
&= \infty.
\end{aligned}$$

Intuitively, this means that the model prior assigns each f_i infinitely large error bars, or equivalently assigns each f_i infinite signal power. This is unlike real-world signals, which have finite power. We therefore let the components of $\boldsymbol{H}$ have a decaying exponentiated-quadratic kernel [Tobar et al., 2015b] instead:

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \sigma_h^2 \exp(-\alpha \|\boldsymbol{t}_1\|^2 - \alpha \|\boldsymbol{t}_2\|^2 - \gamma \|\boldsymbol{t}_1 - \boldsymbol{t}_2\|^2) \boldsymbol{I}.$$

We call $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}$ the diagonal multi-output decaying exponentiated-quadratic kernel. Now, using the notation and identities from Appendix F with composite vector $\begin{bmatrix} \boldsymbol{t}^T & \boldsymbol{\tau}^T \end{bmatrix}^T$,

$$\begin{aligned}
\mathbb{E}[f_i^2(\boldsymbol{t})] &= \int_{\mathbb{R}^K} \operatorname{tr}[\sigma_h^2 \exp(-2\alpha \|\boldsymbol{t} - \boldsymbol{\tau}\|^2) \boldsymbol{I}] \, \mathrm{d}\boldsymbol{\tau} \\
&= I_{\boldsymbol{\tau}}[(K\sigma_h^2, 4 \begin{bmatrix} \alpha & -\alpha \\ -\alpha & \alpha \end{bmatrix})] \\
&= K\sigma_h^2 \frac{\pi^{K/2}}{(2\alpha)^{K/2}}
\end{aligned}$$

and so our modelling issue is resolved. We decide to use the diagonal multi-output decaying exponentiated-quadratic kernel in further development of Model 5.

Note that the components of $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}(\boldsymbol{t}_1, \boldsymbol{t}_2)$ are small for $\|\boldsymbol{t}_1\|$ or $\|\boldsymbol{t}_2\|$ much larger than

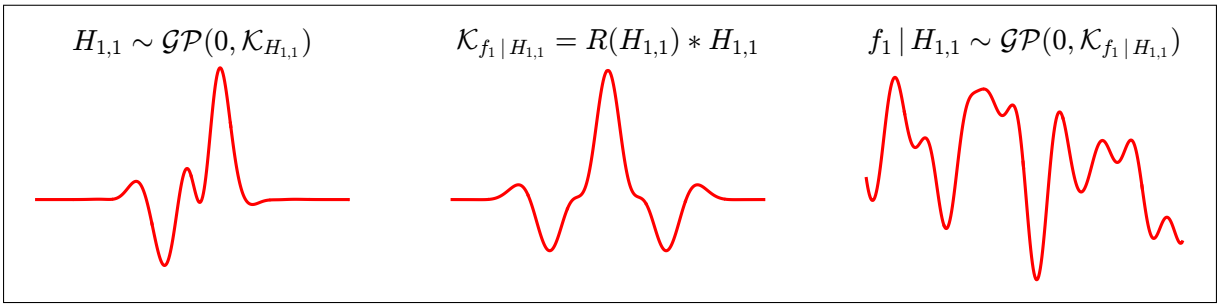


Figure 3.2: Generative process of Model 5 in the case that $N = M = K = 1$. Illustrates the interpretation that Model 5 corresponds ordinary Gaussian process regression in which the kernel is modelled nonparametrically.

$1/\sqrt{\alpha}$. Therefore for such $\boldsymbol{t}_1$ or $\boldsymbol{t}_2$ the posteriors of the components of $\boldsymbol{H}$ are likely to be near zero; that is, $1/\sqrt{\alpha}$ is the effective extent of the components of $\boldsymbol{H}$. Similarly, the components of $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}(\boldsymbol{t}_1, \boldsymbol{t}_2)$ are small for $\|\boldsymbol{t}_1 - \boldsymbol{t}_2\|$ much larger than $1/\sqrt{\gamma}$; thus $1/\sqrt{\gamma}$ determines the effective length scale on which the components of $\boldsymbol{H}$ vary.

We can let the effective extent and effective length scale be different for different dimensions of the input space. We then obtain

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \sigma_h^2 \exp[-\boldsymbol{t}_1^T \boldsymbol{A} \boldsymbol{t}_1 - \boldsymbol{t}_2^T \boldsymbol{A} \boldsymbol{t}_2 - (\boldsymbol{t}_1 - \boldsymbol{t}_2)^T \boldsymbol{\Gamma} (\boldsymbol{t}_1 - \boldsymbol{t}_2)] \boldsymbol{I}$$

for positive-semidefinite matrices $\boldsymbol{A}$ and $\boldsymbol{\Gamma}$. Since $\boldsymbol{A}$ and $\boldsymbol{\Gamma}$ reflect the relevance of each input variable, we call $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}$ the diagonal multi-output decaying automatic relevance determination kernel [MacKay, 2002].

3.3.2 Illustrative Samples of the Nonparametric Kernel Model

Figure 3.2 shows the generative process of Model 5 in the simplest case that $M = N = 1$ and $K = 1$; that is, a sample is defined on a one-dimensional space—for example, time—and constitutes a single output. This figure illustrates the interpretation that Model 5 corresponds ordinary Gaussian process regression in which the kernel is modelled nonparametrically: First, $H_{1,1}$ is generated. Then, a kernel for $f_1 \mid H_{1,1}$ is constructed by computing $R(H_{1,1}) * H_{1,1}$. Finally, $f_1 \mid H_{1,1}$ is generated by drawing from a Gaussian process with $f_1 \mid H_{1,1}$'s generated kernel.

Furthermore, Figure 3.3 shows samples from Model 5 in the case that $M = N = 2$ and $K = 1$—a sample now constitutes two outputs. First, observe that the sampled kernel

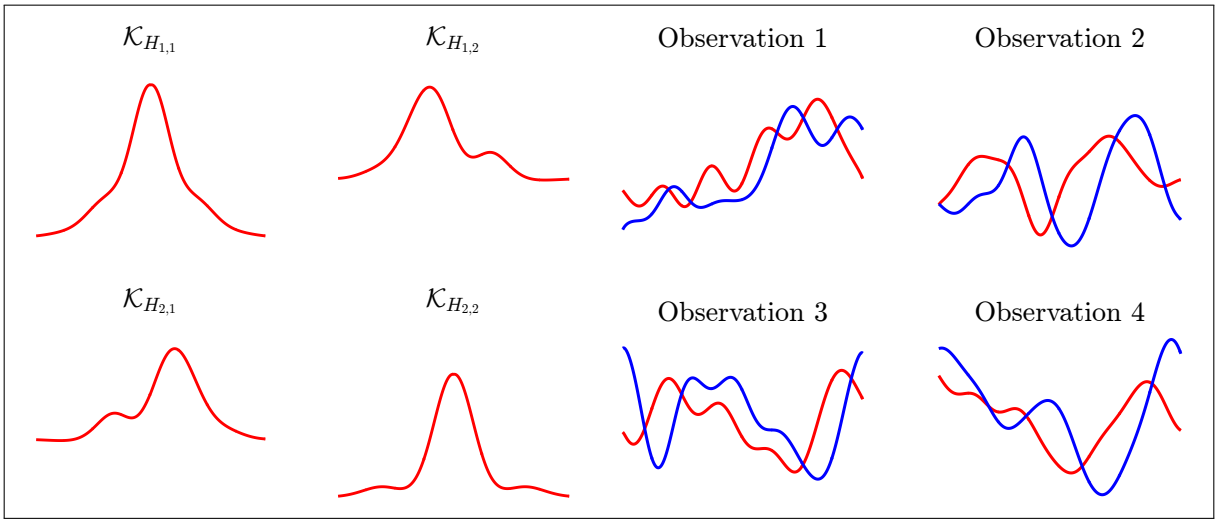


Figure 3.3: Samples from Model 5 in the case that $N = M = 2$ and $K = 1$. Shows the sampled kernel and four observations.

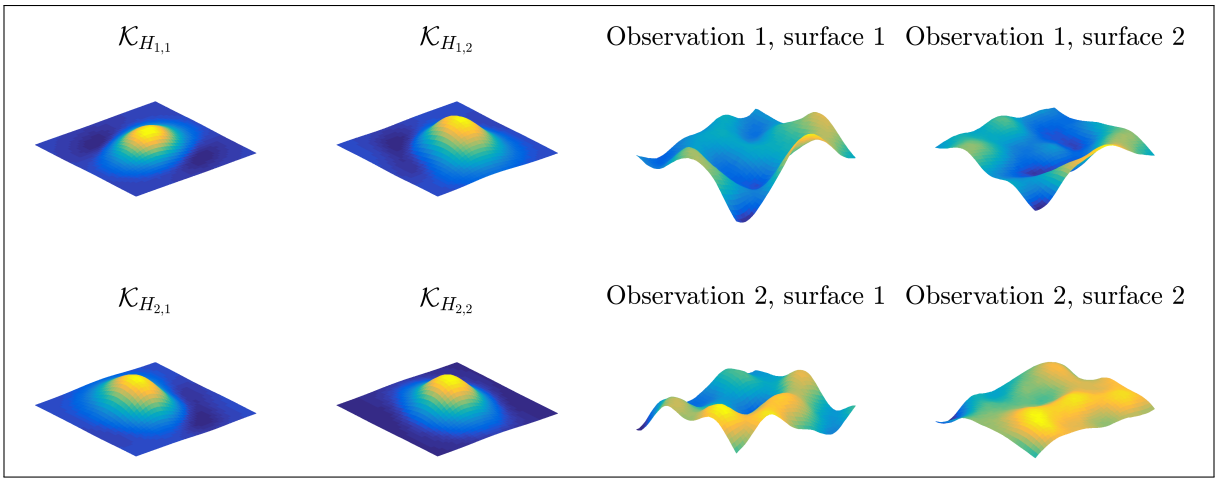


Figure 3.4: Samples from Model 5 in the case that $N = M = 2$ and $K = 2$. Shows the sampled kernel and two observations.

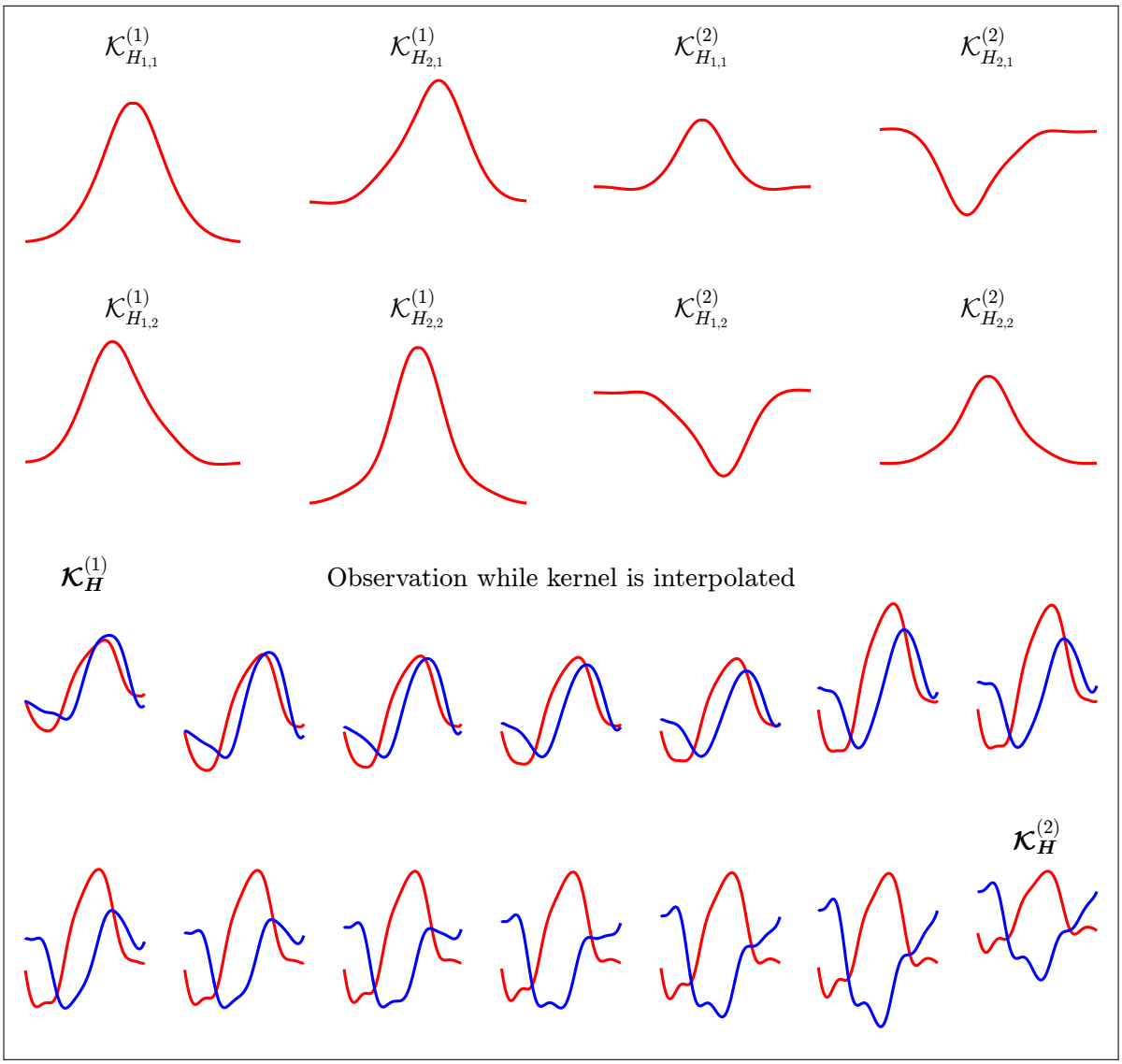


Figure 3.5: Interpolation between two kernels sampled from Model 5 in the case that $N = M = 2$ and $K = 1$. Shows the way an observation changes as its kernel is interpolated between the two kernels.

is indeed symmetric: $\mathcal{K}_{H_{1,1}}$ and $\mathcal{K}_{H_{2,2}}$ are symmetric and $\mathcal{K}_{H_{1,2}} = R(\mathcal{K}_{H_{2,1}})$. Second, observe that $\mathcal{K}_{H_{1,2}}$ and $\mathcal{K}_{H_{1,2}}$ attain reasonably peaked maxima at a nonzero lag. Hence we expect the outputs of a sample to be positively correlated at a nonzero lag; indeed, Figure 3.3 shows that the red lines essentially lead the blue lines. Thus Model 5 shows an certain ability to model multi-output time series. In the same way Model 5 is also able to model multi-output signals defined on multidimensional spaces; Figure 3.4 illustrates that the case $K = 2$ generates correlated surfaces.

Finally, Figure 3.5 illustrates that different samples from Model 5 yield different correlation structures between the outputs; in the case that $M = N = 2$ and $K = 1$, Figure 3.5 shows the way an observation changes as its kernel is interpolated between two kernels sampled from Model 5. Observe that kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}^{(1)}$ corresponds to positively correlated outputs, whereas kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}^{(2)}$ corresponds to negatively corresponds outputs; indeed, as the kernel of the observation is interpolated, the outputs change from being positively correlated to being negatively correlated.

Appendix D was utilised in generating Figures 3.2 to 3.5.

3.3.3 Expressivity of White Noise Excitation

Thus far is not clear how Model 5's expressivity compares to that of Model 4. In this section we shed some light on this matter.

Model 4 shows that $\boldsymbol{f}$ is generated by filtering $\boldsymbol{x}$ with $\boldsymbol{H}$. This means that frequencies in $\boldsymbol{x}$ are attenuated according to the spectrum of $\boldsymbol{H}$. In other words, $\boldsymbol{f}$ consists only of frequencies that are also present in $\boldsymbol{x}$. Thus, if $\boldsymbol{x}$'s spectrum has effectively limited support, then $\boldsymbol{f}$ is limited to only those frequencies. This makes choosing a white noise kernel for $\boldsymbol{x}$ a great choice; a white noise process has a constant power spectrum and therefore contains all frequencies, meaning that $\boldsymbol{f}$ can also contain all frequencies.

More concretely, we show that an instance $(\boldsymbol{\mathcal{K}}_{\boldsymbol{x}^{(1)}}^{(1)}, \boldsymbol{\mathcal{K}}_{\boldsymbol{H}^{(1)}}^{(1)}, \boldsymbol{\Lambda}^{(1)})$ of Model 4 can be expressed as an instance $(\boldsymbol{\mathcal{K}}_{\boldsymbol{H}^{(2)}}^{(2)}, \boldsymbol{\Lambda}^{(2)})$ of Model 5 if $\boldsymbol{\mathcal{K}}_{\boldsymbol{x}^{(1)}}^{(1)}(\boldsymbol{t}_1, \boldsymbol{t}_2) = [R(\boldsymbol{R}) * \boldsymbol{R}^T](\boldsymbol{t}_2 - \boldsymbol{t}_1)$ for some $\boldsymbol{R}$—we call $\boldsymbol{R}$ the *root* of $\boldsymbol{\mathcal{K}}_{\boldsymbol{x}^{(1)}}^{(1)}$. This is equivalent to showing that $(\boldsymbol{\mathcal{K}}_{\boldsymbol{x}^{(1)}}^{(1)}, \boldsymbol{\mathcal{K}}_{\boldsymbol{H}^{(1)}}^{(1)}, \boldsymbol{\Lambda}^{(1)})$ can be expressed as another instance $(\boldsymbol{\mathcal{K}}_{\boldsymbol{x}^{(2)}}^{(2)}, \boldsymbol{\mathcal{K}}_{\boldsymbol{H}^{(2)}}^{(2)}, \boldsymbol{\Lambda}^{(2)})$ of Model 4 where $\boldsymbol{x}^{(2)}$ is white noise.

Observe that $\boldsymbol{R} * \boldsymbol{x}^{(2)}$ and $\boldsymbol{H}^{(1)} * \boldsymbol{R}$ are linear combinations of Gaussian processes. Hence $\boldsymbol{R} * \boldsymbol{x}^{(2)}$ and $\boldsymbol{H}^{(1)} * \boldsymbol{R}$ are also Gaussian processes, which thus can be identified by their mean functions and kernels. It is readily verified that $\boldsymbol{R} * \boldsymbol{x}^{(2)}$ and $\boldsymbol{H}^{(1)} * \boldsymbol{R}$

have zero mean function. The kernel of $\boldsymbol{R} * \boldsymbol{x}^{(2)}$ is derived in a similar fashion as Equation (3.1), yielding $\boldsymbol{\mathcal{K}}_{\boldsymbol{R}*\boldsymbol{x}^{(2)}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = [R(\boldsymbol{R}) * \boldsymbol{R}^T](\boldsymbol{t}_2 - \boldsymbol{t}_1)$. Thus $\boldsymbol{x}^{(1)} \overset{d}{=} \boldsymbol{R} * \boldsymbol{x}^{(2)}$. Now let $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}^{(2)}} = \boldsymbol{\mathcal{K}}_{\boldsymbol{H}^{(1)}*\boldsymbol{R}}$ and $\boldsymbol{\Lambda}^{(2)} = \boldsymbol{\Lambda}^{(1)}$. Then also $\boldsymbol{H}^{(1)} * \boldsymbol{R} \overset{d}{=} \boldsymbol{H}^{(2)}$ and $\boldsymbol{\varepsilon}^{(1)} \overset{d}{=} \boldsymbol{\varepsilon}^{(2)}$. Consequently, by associativity of the convolution operator,

$$\begin{aligned}
\boldsymbol{f}^{(1)} &= \boldsymbol{H}^{(1)} * \boldsymbol{x}^{(1)} + \boldsymbol{\varepsilon}^{(1)} \\
&\overset{d}{=} \boldsymbol{H}^{(1)} * (\boldsymbol{R} * \boldsymbol{x}^{(2)}) + \boldsymbol{\varepsilon}^{(2)} \\
&= (\boldsymbol{H}^{(1)} * \boldsymbol{R}) * \boldsymbol{x}^{(2)} + \boldsymbol{\varepsilon}^{(2)} \\
&\overset{d}{=} \boldsymbol{H}^{(2)} * \boldsymbol{x}^{(2)} + \boldsymbol{\varepsilon}^{(2)} \\
&= \boldsymbol{f}^{(2)}
\end{aligned}$$

so that $\boldsymbol{f}^{(1)} \overset{d}{=} \boldsymbol{f}^{(2)}$.

We have established that an instance of Model 4 can be expressed of an instance of Model 5 if $\boldsymbol{x}$'s kernel has a root and is thereby stationary. Appendix G shows that any kernel of exponentiated quadratic form has a root, that the diagonal multi-output exponentiated-quadratic kernel has a root and that the white noise kernel is its own root. This sheds some light on the class of kernels of $\boldsymbol{x}$ for which Model 4 and Model 5 have equal expressivity.

We can more generally identify which kernels have roots in the case that $M = N = 1$. Suppose that $\mathcal{K}_x$ is stationary and Schwartz—that is, rapidly decreasing. Then, by Bochner's Theorem, its Fourier transform $\mathcal{F}_{\mathcal{K}_x}$ exists and is real. Therefore $\mathcal{F}_{\mathcal{K}_x}^{1/2}$ is well defined. As $\mathcal{K}_x$ is Schwartz, $\mathcal{F}_{\mathcal{K}_x}$ is also Schwartz and thereby absolutely integrable. Thus, by

$$\int_{\mathbb{R}^K} |\mathcal{F}_{\mathcal{K}_x}(\boldsymbol{\xi})| \, \mathrm{d}\boldsymbol{\xi} = \int_{\mathbb{R}^K} |\mathcal{F}_{\mathcal{K}_x}^{1/2}(\boldsymbol{\xi})|^2 \, \mathrm{d}\boldsymbol{\xi} < \infty,$$

$\mathcal{F}_{\mathcal{K}_x}^{1/2}$ is square integrable and so $\mathcal{F}^{-1}\{\mathcal{F}_{\mathcal{K}_{\boldsymbol{x}}}^{1/2}\}$ exists. Let $r = \mathcal{F}^{-1}\{\mathcal{F}_{\mathcal{K}_{\boldsymbol{x}}}^{1/2}\}$. Then

$$\begin{aligned}
[R(r) * r](\boldsymbol{\tau}) &= \int_{\mathbb{R}^{3K}} \mathcal{F}_{\mathcal{K}_x}^{1/2}(\boldsymbol{\xi}_1) \exp[2\pi i \boldsymbol{\xi}_1^T(\boldsymbol{\tau}' - \boldsymbol{\tau})] \mathcal{F}_{\mathcal{K}_x}^{1/2}(\boldsymbol{\xi}_2) \exp(2\pi i \boldsymbol{\xi}_2^T \boldsymbol{\tau}') \,\mathrm{d}\boldsymbol{\xi}_1 \,\mathrm{d}\boldsymbol{\xi}_2 \,\mathrm{d}\boldsymbol{\tau}' \\
&= \int_{\mathbb{R}^{3K}} \mathcal{F}_{\mathcal{K}_x}^{1/2}(\boldsymbol{\xi}_1) \mathcal{F}_{\mathcal{K}_x}^{1/2}(\boldsymbol{\xi}_2) \exp(2\pi i \boldsymbol{\xi}_2^T \boldsymbol{\tau}) \underbrace{\exp[2\pi i (\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2)^T \boldsymbol{\tau}']}_{\delta(-\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2)} \,\mathrm{d}\boldsymbol{\xi}_1 \,\mathrm{d}\boldsymbol{\xi}_2 \,\mathrm{d}\boldsymbol{\tau}' \\
&= \int_{\mathbb{R}^K} \mathcal{F}_{\mathcal{K}_x}(\boldsymbol{\xi}) \exp(2\pi i \boldsymbol{\xi}^T \boldsymbol{\tau}) \,\mathrm{d}\boldsymbol{\xi} \\
&= \mathcal{K}_x(\boldsymbol{\tau})
\end{aligned}$$

and so $\mathcal{K}_x$ has a root.

In summary, white noise excitation in Model 4 yields great expressivity because the spectrum of white noise has infinite support. Concretely, an instance of Model 4 can be expressed as an instance of Model 5 if $\boldsymbol{x}$'s kernel has a root. In the case that $M = N = 1$, a kernel has a root if it is stationary and rapidly decreasing.

3.4 The Approximate Kernel Model

Model 5 is troublesome from a computational perspective. Namely, $\boldsymbol{\mathcal{K}}_{\boldsymbol{f}\,|\,\boldsymbol{H}} = R(\boldsymbol{H}) * \boldsymbol{H}^T$ is a convolution between two matrix-valued stochastic processes and it is not clear how such a quantity can be computed. We therefore develop a model that is approximate to Model 5, but whose computation is tractable.

Recall that $\boldsymbol{H}$ is smooth and that its support is effectively limited due to use of the decaying exponentiated-quadratic kernel. In that case, as demonstrated by Minka [2000], we can numerically approximate the integral $R(\boldsymbol{H}) * \boldsymbol{H}^T$ by its expectation conditioned on some finite number of observations $\boldsymbol{H}(\boldsymbol{T})$ where $\boldsymbol{T} \in \mathbb{R}^{T \times K}$—we call these observations *inducing points*. Symbolically,

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{f}\,|\,\boldsymbol{H}} = R(\boldsymbol{H}) * \boldsymbol{H}^T \approx \mathbb{E}[R(\boldsymbol{H}) * \boldsymbol{H}^T \,|\, \boldsymbol{H}(\boldsymbol{T})] = \tilde{\boldsymbol{\mathcal{K}}}_{\boldsymbol{f}\,|\,\boldsymbol{H}(\boldsymbol{T})}.$$

Note that $\tilde{\boldsymbol{\mathcal{K}}}_{\boldsymbol{f}\,|\,\boldsymbol{H}(\boldsymbol{T})}$ numerically approximates $\boldsymbol{\mathcal{K}}_{\boldsymbol{f}\,|\,\boldsymbol{H}}$ and at the same time maintains knowledge of its own uncertainty; in this sense $\tilde{\boldsymbol{\mathcal{K}}}_{\boldsymbol{f}\,|\,\boldsymbol{H}(\boldsymbol{T})}$ is called a *Bayesian numerical approximation* of $\boldsymbol{\mathcal{K}}_{\boldsymbol{f}\,|\,\boldsymbol{H}}$ [Tobar et al., 2015b]. Appendix H shows that the analytical

form of $\tilde{\boldsymbol{\mathcal{K}}}_{\boldsymbol{f}\,|\,\boldsymbol{H}(\boldsymbol{T})}$ is given by

$$\tilde{\mathcal{K}}_{f_i,f_j\,|\,\boldsymbol{H}(\boldsymbol{T})} = \sum_{k=1}^{M} \Big\{ \underbrace{\mathbb{1}(i-j) I^{(1,H_{i,k})}}_{\text{prior}} - \underbrace{\mathbb{1}(i-j)\,\mathrm{tr}(\boldsymbol{K}^{-1}_{H_{i,k}(\boldsymbol{T})} \boldsymbol{I}^{(2,H_{i,k},H_{i,k})})}_{\text{conditioning on inducing points}} + \underbrace{H^T_{i,k}(\boldsymbol{T}) \boldsymbol{K}^{-1}_{H_{i,k}(\boldsymbol{T})} \boldsymbol{I}^{(2,H_{i,k},H_{j,k})} \boldsymbol{K}^{-1}_{H_{j,k}(\boldsymbol{T})} H_{j,k}(\boldsymbol{T})}_{\text{learned through inducing points}} \Big\}.$$

We clearly distinguish the prior covariance structure, the covariance structure due to conditioning on the inducing points and the covariance structure learned through the inducing points.

We have derived the following approximation of Model 5:

Model 6 (Approximate Kernel Model (AKM))**.** *Draw*

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{T}) &\sim \mathcal{N}[\boldsymbol{0}, \boldsymbol{\mathcal{K}}_{\boldsymbol{H}}(\boldsymbol{T},\boldsymbol{T})], \\ \boldsymbol{\varepsilon} &\sim \mathcal{GP}[\boldsymbol{0}, \boldsymbol{\Lambda}^2 \delta(\boldsymbol{t}_1 - \boldsymbol{t}_2)] \end{aligned}$$

independently for some diagonal kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}$ *and some diagonal matrix* $\boldsymbol{\Lambda}$*. Afterwards let* $\tilde{\boldsymbol{\mathcal{K}}}_{\boldsymbol{f}\,|\,\boldsymbol{H}(\boldsymbol{T})}$ *be such that*

$$\mathcal{K}_{f_i,f_j\,|\,\boldsymbol{H}(\boldsymbol{T})} = \sum_{k=1}^{M} \Big\{ \mathbb{1}(i-j)[I^{(1,H_{i,k})} - \mathrm{tr}(\boldsymbol{K}^{-1}_{H_{i,k}(\boldsymbol{T})} \boldsymbol{I}^{(2,H_{i,k},H_{i,k})})] + H^T_{i,k}(\boldsymbol{T}) \boldsymbol{K}^{-1}_{H_{i,k}(\boldsymbol{T})} \boldsymbol{I}^{(2,H_{i,k},H_{j,k})} \boldsymbol{K}^{-1}_{H_{j,k}(\boldsymbol{T})} H_{j,k}(\boldsymbol{T}) \Big\}.$$

Finally draw

$$\boldsymbol{f}\,|\,\boldsymbol{H}(\boldsymbol{T}) \sim \mathcal{GP}(\boldsymbol{0}, \tilde{\boldsymbol{\mathcal{K}}}_{\boldsymbol{f}\,|\,\boldsymbol{H}(\boldsymbol{T})}).$$

Then observations are generated by $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\varepsilon}$*.*

Observe that Model 6 is ordinary Gaussian process regression with a specific kernel and a prior over the hyperparameters. Thus Model 6 can take advantage of existing literature on ordinary Gaussian process regression.

Furthermore, the fact that Model 6 approximates Model 5 implies that the prior over the

hyperparameters specified by Model 6 is one that is sensible to use. This is an advantage of Model 6 over most other multi-output kernels; priors over hyperparameters are usually chosen without principled motivation.

3.4.1 Interpretation of the Kernel Approximation

We investigate the kernel approximation in the case that $M = N = 1$. Denote $f = f_1$, $h = H_{1,1}$ and $h(\boldsymbol{T}) = \boldsymbol{h}$. Let $\mathcal{K}_{\tilde{h} \mid \boldsymbol{A}}$ be the posterior kernel of the sparse approximation of h according to [Titsias, 2009] where $\boldsymbol{h}$ are the approximation's inducing points and $\boldsymbol{A}$ is their posterior covariance matrix. We compare $\tilde{\mathcal{K}}_{f \mid \boldsymbol{h}}$ to $\mathcal{K}_{\tilde{h} \mid \boldsymbol{A}}$:

$$\begin{aligned}
\tilde{\mathcal{K}}_{f \mid \boldsymbol{h}}(\boldsymbol{t}_1, \boldsymbol{t}_2) &= I^{(1,h)}(\boldsymbol{t}_1, \boldsymbol{t}_2) - \operatorname{tr}[\boldsymbol{K}_{\boldsymbol{h}}^{-1} \boldsymbol{I}^{(2,h,h)}(\boldsymbol{t}_1, \boldsymbol{t}_2)] \\
&\quad + \operatorname{tr}[\boldsymbol{K}_{\boldsymbol{h}}^{-1} \boldsymbol{I}^{(2,h,h)}(\boldsymbol{t}_1, \boldsymbol{t}_2) \boldsymbol{K}_{\boldsymbol{h}}^{-1} \boldsymbol{h}\boldsymbol{h}^T], \\
\mathcal{K}_{\tilde{h} \mid \boldsymbol{A}}(\boldsymbol{t}_1, \boldsymbol{t}_2) &= \mathcal{K}_h(\boldsymbol{t}_1, \boldsymbol{t}_2) - \operatorname{tr}[\boldsymbol{K}_{\boldsymbol{h}}^{-1} \mathcal{K}_h(\boldsymbol{T}, \boldsymbol{t}_2) \mathcal{K}_h(\boldsymbol{t}_1, \boldsymbol{T})] \\
&\quad + \operatorname{tr}[\boldsymbol{K}_{\boldsymbol{h}}^{-1} \mathcal{K}_h(\boldsymbol{T}, \boldsymbol{t}_2) \mathcal{K}_h(\boldsymbol{t}_1, \boldsymbol{T}) \boldsymbol{K}_{\boldsymbol{h}}^{-1} \boldsymbol{A}].
\end{aligned}$$

Thus, by Equations (H.1) and (H.2), we establish that

$$\tilde{\mathcal{K}}_{f \mid \boldsymbol{h}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \int_{\mathbb{R}^K} \mathcal{K}_{\tilde{h} \mid \boldsymbol{A} = \boldsymbol{h}\boldsymbol{h}^T}(\boldsymbol{t}_1 - \boldsymbol{\tau}, \boldsymbol{t}_2 - \boldsymbol{\tau}) \, \mathrm{d}\boldsymbol{\tau}.$$

This means that the approximate integration performed by $\tilde{\mathcal{K}}_{f \mid \boldsymbol{h}}$ corresponds to first sparsely approximating h and then exactly integrating its posterior kernel.

3.4.2 The Case of the Diagonal Multi-Output Decaying Exponentiated-Quadratic Kernel

Let $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}$ be a diagonal multi-output decaying exponentiated-quadratic kernel. It then turns out that $I^{(1,H_{i,k})}$, $\boldsymbol{I}^{(2,H_{i,k},H_{j,k})}$ and their Fourier transforms admit simple forms. We now compute these simple forms, which in Section 3.5 will reveal Model 6's connection

to existing work. To begin with, Appendices I.4.3 and I.4.5 show that

$$\begin{aligned}
I^{(1,H_{i,k})}(\boldsymbol{t}_1,\boldsymbol{t}_2) &= \underbrace{\sigma_h^2\frac{\pi^{K/2}}{(2\alpha)^{K/2}}}_{C^{(1)}}\exp\Big[-\frac{1}{2}\underbrace{(\alpha+2\gamma)}_{l^{-(1)}}\|\boldsymbol{t}_1-\boldsymbol{t}_2\|^2\Big],\\
&= C^{(1)}\exp\Big(-\frac{1}{2}l^{-(1)}\|\boldsymbol{t}_1-\boldsymbol{t}_2\|^2\Big)\\
&= I^{(1)}(\boldsymbol{t}_1-\boldsymbol{t}_2),\\
I^{(2,H_{i,k},H_{j,k})}_{m,n}(\boldsymbol{t}_1,\boldsymbol{t}_2) &= \underbrace{\sigma_h^4\frac{\pi^{K/2}}{(2\alpha+2\gamma)^{K/2}}\exp\left[-\frac{1}{2}\frac{(\alpha+\gamma)^2-\gamma^2}{\alpha+\gamma}(\|\boldsymbol{T}_{m,:}-\boldsymbol{T}_{n,:}\|^2\right.}_{C^{(2)}_{m,n}}\\
&\qquad\qquad\left.-\|\boldsymbol{T}_{m,:}+\boldsymbol{T}_{n,:}\|^2)\right]\\
&\qquad \exp\left[-\frac{1}{2}\underbrace{(\alpha+\gamma)}_{l^{-(2)}}\Big\|(\boldsymbol{t}_1-\boldsymbol{t}_2)-\underbrace{\frac{\gamma}{\alpha+\gamma}(\boldsymbol{T}_{m,:}-\boldsymbol{T}_{n,:})}_{\boldsymbol{\mu}^{(2)}_{m,n}}\Big\|^2\right]\\
&= C^{(2)}_{m,n}\exp\Big[-\frac{1}{2}l^{-(2)}\|(\boldsymbol{t}_1-\boldsymbol{t}_2)-\boldsymbol{\mu}^{(2)}_{m,n}\|^2\Big]\\
&= I^{(2)}_{m,n}(\boldsymbol{t}_1-\boldsymbol{t}_2).
\end{aligned}$$

Therefore

$$\mathcal{F}_{\boldsymbol{t}_1-\boldsymbol{t}_2}\{I^{(1)}\}(\boldsymbol{f}) = C^{(1)}\sqrt{2\pi l^{(1)}}\exp(-2\pi^2 l^{(1)}\|\boldsymbol{f}\|^2), \tag{3.4}$$

$$\mathcal{F}_{\boldsymbol{t}_1-\boldsymbol{t}_2}\{I^{(2)}_{m,n}\}(\boldsymbol{f}) = C^{(2)}_{m,n}\sqrt{2\pi l^{(2)}}\exp(-2\pi^2 l^{(2)}\|\boldsymbol{f}\|^2+2\pi i\boldsymbol{\mu}^T_{m,n}\boldsymbol{f}). \tag{3.5}$$

Now, consider the case that $i=j$. First, all $\mathcal{K}_{H_{i,j}}$ are equal and so all $\boldsymbol{K}_{H_{i,j}(\boldsymbol{T})}=\boldsymbol{K}$; thus the kernel approximation simplifies to

$$\mathcal{K}_{f_i,f_i\,|\,\boldsymbol{H}(\boldsymbol{T})} = M[I^{(1)}-\operatorname{tr}(\boldsymbol{K}^{-1}\boldsymbol{I}^{(2)})]+\sum_{k=1}^{M}H^T_{i,k}(\boldsymbol{T})\boldsymbol{K}^{-1}\boldsymbol{I}^{(2)}\boldsymbol{K}^{-1}H_{i,k}(\boldsymbol{T}), \tag{3.6}$$

which shows that we can equivalently let $\boldsymbol{I}^{(2)}\leftarrow(\boldsymbol{I}^{(2)}+\boldsymbol{I}^{(2)T})/2$. Second, note that $C^{(2)}_{m,n}=C^{(2)}_{n,m}$ and $-\boldsymbol{\mu}^{(2)}_{n,m}=\boldsymbol{\mu}^{(2)}_{m,n}$; hence $\mathcal{F}\{I^{(2)}_{m,n}\}=\mathcal{F}^*\{I^{(2)}_{n,m}\}$. Therefore, if $i=j$, then

equivalently

$$\begin{aligned}\mathcal{F}_{\boldsymbol{t}_1-\boldsymbol{t}_2}\{I^{(2)}_{m,n}\}(\boldsymbol{f}) &= \frac{1}{2}[\mathcal{F}_{\boldsymbol{t}_1-\boldsymbol{t}_2}\{I^{(2)}_{m,n}\}(\boldsymbol{f}) + \mathcal{F}_{\boldsymbol{t}_1-\boldsymbol{t}_2}\{I^{(2)}_{n,m}\}(\boldsymbol{f})] \\ &= \frac{1}{2}[\mathcal{F}_{\boldsymbol{t}_1-\boldsymbol{t}_2}\{I^{(2)}_{m,n}\}(\boldsymbol{f}) + \mathcal{F}^{*}_{\boldsymbol{t}_1-\boldsymbol{t}_2}\{I^{(2)}_{m,n}\}(\boldsymbol{f})] \\ &= C^{(2)}_{m,n}\sqrt{2\pi l^{(2)}}\exp(-2\pi^2 l^{(2)}\|\boldsymbol{f}\|^2)\cos(2\pi i\boldsymbol{\mu}^T_{m,n}\boldsymbol{f}). \qquad (3.7)\end{aligned}$$

3.5 Related Work

Model 5 and its approximation Model 6 are closely related to recent work.

First, Model 5 is a generalisation of the Gaussian Process Convolution Model (GPCM) [Tobar et al., 2015b]; the case $N = M = K = 1$ recovers their model exactly.

Second, Equations (3.4), (3.6) and (3.7) show that the power spectral densities of the kernel in Model 6 take the form of spectral mixture kernels (SMKs) [Wilson and Adams, 2013]. In other words, the diagonal entries of the kernel in Model 6 form *Fourier pairs* with spectral mixture kernels. Therefore the kernel in Model 6 is a *dual* multi-output generalisation of the spectral mixture kernel. Furthermore, recall that Model 6 approximates Model 5 more accurately as the number of inducing points increases. This reveals the kernel in Model 5 as a multi-output generalisation of the spectral mixture kernel with an infinite number of components.

Third, Equations (3.4) to (3.6) show that the kernel in Model 6 is a dual formulation of the cross-spectral mixture kernel (CSMK) [Ulrich et al., 2015]. Similar to the spectral mixture kernel, this reveals the kernel in Model 5 as a cross-spectral mixture kernel with an infinite number of components. Furthermore, the connection between Model 6 and the cross-spectral mixture kernel suggests that results in [Ulrich et al., 2015] might carry over to Model 6; indeed, inspired by [Ulrich et al., 2015], Appendix E presents an approximation of stationary multi-output kernel matrices that can be used to efficiently perform inference in Model 6 if $\boldsymbol{H}(\boldsymbol{T})$ are instead treated as hyperparameters.

Figure 3.6 summarises how Model 5 and Model 6 relate to current literature on flexible kernel models.

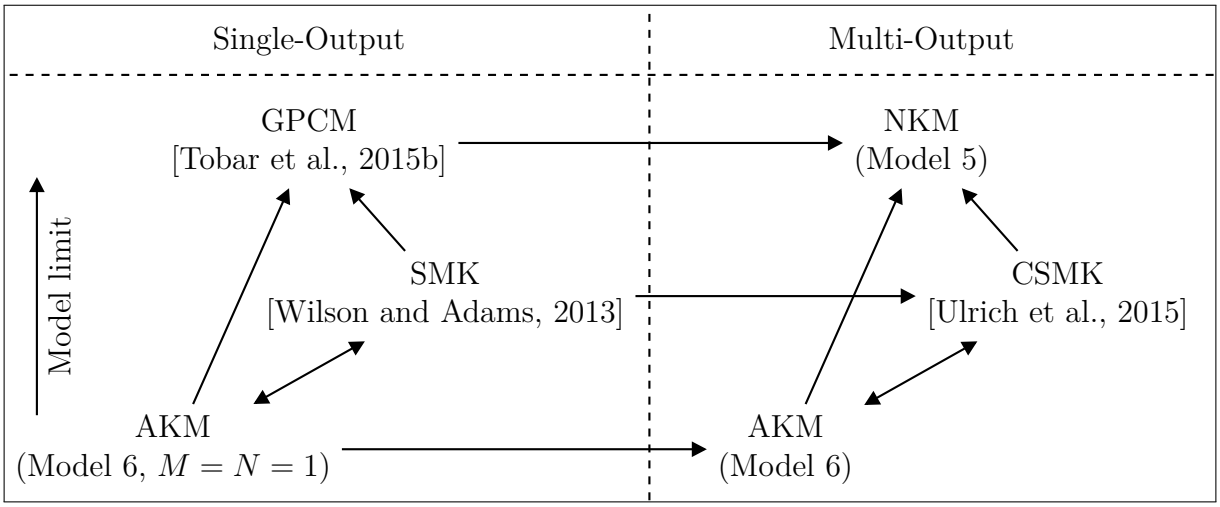


Figure 3.6: Relationship of Model 5 and Model 6 to current literature on flexible kernel models. Single-headed arrows indicate generalisation and double-headed arrows indicate duality. "Model limit" refers to taking the model's number of components to infinity.

3.6 Inference in the Nonparametric Kernel Model

Performing inference in Model 5 directly is troublesome, because $\boldsymbol{H}$ parametrises the kernel of Model 5's likelihood $p(\boldsymbol{y} \mid \boldsymbol{H})$ which thus is intractable. Instead, equivalently consider white noise excitation in Model 4. Model 4's likelihood $p(\boldsymbol{y} \mid \boldsymbol{H}, \boldsymbol{x})$ is Gaussian with mean function $\boldsymbol{f} = \boldsymbol{H} * \boldsymbol{x}$. As we show shortly, this likelihood is manageable.

We perform inference in Model 5 by inferring $\boldsymbol{H}$ and $\boldsymbol{x}$ in Model 4 where $\boldsymbol{x}$ is white noise. This immediately raises a concern: $\boldsymbol{f} = \boldsymbol{H} * \boldsymbol{x}$ requires knowledge of the whole processes $\boldsymbol{H}$ and $\boldsymbol{x}$, but $\boldsymbol{x}$ cannot wholly be learned because $\boldsymbol{x}(\boldsymbol{t}_1)$ and $\boldsymbol{x}(\boldsymbol{t}_2)$ are independent for $\boldsymbol{t}_1 \neq \boldsymbol{t}_2$; that is, we would have to learn $\boldsymbol{x}(\boldsymbol{t})$ for every $\boldsymbol{t}$ separately, but a computer can only store $\boldsymbol{x}(\boldsymbol{t})$ for finitely many $\boldsymbol{t}$.

To resolve this issue, consider another instance of Model 4 whose excitation is denoted by $\tilde{\boldsymbol{x}}$. Let $\tilde{\boldsymbol{x}}$ have a diagonal multi-output exponentiated-quadratic kernel. In that case $\tilde{\boldsymbol{x}}$ is smooth and can therefore be learned by conditioning on $\tilde{\boldsymbol{x}}(\boldsymbol{t})$ for finitely many $\boldsymbol{t}$ [Titsias, 2009]. Thus we should be able to perform inference in this new instance. Now, Appendix G shows that $\tilde{\boldsymbol{x}}$'s kernel has a root $\boldsymbol{R}$. Therefore, by Section 3.3.3, Model 5 can be expressed in terms of this new instance where $\tilde{\boldsymbol{x}} = \boldsymbol{R} * \boldsymbol{x}$. Crucially, the fact that we can learn this new instance then implies that we can learn Model 5 by learning

$\boldsymbol{R} * \boldsymbol{x}$ instead of $\boldsymbol{x}$.

Learning $\boldsymbol{R} * \boldsymbol{x}$ instead of $\boldsymbol{x}$ has a number of equivalent interpretations. First, $\boldsymbol{R} * \boldsymbol{x}$ represents a smoothed version of $\boldsymbol{x}$ where each point is a linear combination of the points of $\boldsymbol{x}$. Thus learning one point on $\boldsymbol{R} * \boldsymbol{x}$ induces knowledge about the whole process $\boldsymbol{x}$. Second, $\boldsymbol{R}$ acts as a filter, which means that $\boldsymbol{R} * \boldsymbol{x}$ represents a band-limited version of $\boldsymbol{x}$. Hence we restrict ourselves to only learning frequencies in the band of $\boldsymbol{R}$; this is an easier problem than learning all frequencies.

The latter interpretation raises another concern; if $\boldsymbol{H}$ admits frequencies not in the band of $\boldsymbol{R}$, then those frequencies cannot be learned. Thus $\boldsymbol{R}$ should be chosen carefully; in the case that $M = N = K = 1$, Tobar et al. [2015a] consider how $\boldsymbol{R}$ can be designed to promote training performance. We decide to simply use

$$\boldsymbol{R}(\boldsymbol{t}) = \exp(-\omega \|\boldsymbol{t}\|^2)\boldsymbol{I}.$$

This choice should enable us to learn $\boldsymbol{x}$; namely, Appendix G shows that this $\boldsymbol{R}$ is a root of a diagonal multi-output exponentiated-quadratic kernel, and we previously determined that a process with a diagonal multi-output exponentiated-quadratic kernel can be learned.

Concretely, we perform approximate inference in Model 5 via a structured mean field approximation in which we sparsely approximate $\boldsymbol{H}$ and $\boldsymbol{R} * \boldsymbol{x}$ through inducing points [Titsias, 2009]. Let $\boldsymbol{Y}$ denote Y observations. Consider the family $\mathcal{Q}$ of distributions of the form

$$q(\boldsymbol{H}, \boldsymbol{x}, \underbrace{\boldsymbol{u}_{H_{1,1}}, \ldots, \boldsymbol{u}_{H_{N,M}}}_{\boldsymbol{U}_{\boldsymbol{H}}}, \underbrace{\boldsymbol{u}_{\tilde{x}_1}, \ldots, \boldsymbol{u}_{\tilde{x}_M}}_{\boldsymbol{U}_{\tilde{\boldsymbol{x}}}}) = p(\boldsymbol{H}, \boldsymbol{x} \,|\, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}) \prod_{i=1,j=1}^{N,M} q(\boldsymbol{u}_{H_{i,j}}) \prod_{j=1}^{M} q(\boldsymbol{u}_{\tilde{x}_j})$$

where $\boldsymbol{u}_{H_{i,j}} = H_{i,j}(\boldsymbol{T}_{H_{i,j}})$ and $\boldsymbol{u}_{\tilde{x}_j} = \tilde{x}_j(\boldsymbol{T}_{\tilde{x}_j})$ are inducing points for respectively the processes $H_{i,j}$ and $\tilde{x}_j = \exp(-\omega\|\boldsymbol{t}\|^2) * x_j$. Let

$$q(\boldsymbol{u}_{H_{i,j}}) = \mathcal{N}(\boldsymbol{u}_{H_{i,j}}; \boldsymbol{\mu}_{H_{i,j}}, \boldsymbol{\Sigma}_{H_{i,j}}),$$
$$q(\boldsymbol{u}_{\tilde{x}_j}) = \mathcal{N}(\boldsymbol{u}_{\tilde{x}_j}; \boldsymbol{\mu}_{\tilde{x}_j}, \boldsymbol{\Sigma}_{\tilde{x}_j}).$$

We then perform approximate inference via

$$
\begin{aligned}
&p(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}} \,|\, \boldsymbol{Y}) \\
&\quad = \operatorname*{argmin}_{q} D_{KL}[q(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}) \,\|\, p(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}} \,|\, \boldsymbol{Y})] \\
&\quad \approx \operatorname*{argmin}_{q \in \mathcal{Q}} D_{KL}[q(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}) \,\|\, p(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}} \,|\, \boldsymbol{Y})] \\
&\quad = \operatorname*{argmax}_{q \in \mathcal{Q}} \{\log p(\boldsymbol{Y}) - D_{KL}[q(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}) \,\|\, p(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}} \,|\, \boldsymbol{Y})]\} \\
&\quad = \operatorname*{argmax}_{q \in \mathcal{Q}} \mathcal{F}(q).
\end{aligned}
$$

Since $D_{KL}(\cdot \,\|\, \cdot) \geq 0$, $\mathcal{F}(q)$—also called the *variational free energy*, or *free energy* in short—is a lower bound on the marginal likelihood; hence performing inference is equivalent to maximising this lower bound. Now, recall that $\boldsymbol{H}$'s diagonal multi-output decaying exponentiated-quadratic kernel implies independence between different $H_{i,j}$'s; as a result,

$$
p(\boldsymbol{H}, \boldsymbol{x} \,|\, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}) = p(\boldsymbol{H} \,|\, \boldsymbol{U}_{\boldsymbol{H}}) p(\boldsymbol{x} \,|\, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}) = \prod_{i=1,j=1}^{N,M} p(H_{i,j} \,|\, \boldsymbol{u}_{H_{i,j}}) \prod_{j=1}^{M} p(x_j \,|\, \boldsymbol{u}_{\tilde{x}_j}).
$$

Then

$$
\begin{aligned}
\mathcal{F}(q) &= \mathbb{E}_q[\log p(\boldsymbol{Y})] + \mathbb{E}_q\left[\log \frac{p(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}} \,|\, \boldsymbol{Y})}{q(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}})}\right] \\
&= \mathbb{E}_q\left[\log \frac{p(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}, \boldsymbol{Y})}{q(\boldsymbol{H}, \boldsymbol{x}, \boldsymbol{U}_{\boldsymbol{H}}, \boldsymbol{U}_{\tilde{\boldsymbol{x}}})}\right] \\
&= \mathbb{E}_q\left[\log \frac{p(\boldsymbol{Y} \,|\, \boldsymbol{H}, \boldsymbol{x}) p(\boldsymbol{H} \,|\, \boldsymbol{U}_{\boldsymbol{H}}) p(\boldsymbol{x} \,|\, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}) \prod_{i=1,j=1}^{N,M} p(\boldsymbol{u}_{H_{i,j}}) \prod_{j=1}^{M} p(\boldsymbol{u}_{\tilde{x}_j})}{p(\boldsymbol{H} \,|\, \boldsymbol{U}_{\boldsymbol{H}}) p(\boldsymbol{x} \,|\, \boldsymbol{U}_{\tilde{\boldsymbol{x}}}) \prod_{i=1,j=1}^{N,M} q(\boldsymbol{u}_{H_{i,j}}) \prod_{j=1}^{M} q(\boldsymbol{u}_{\tilde{x}_j})}\right] \\
&= \mathbb{E}_q[\log p(\boldsymbol{Y} \,|\, \boldsymbol{H}, \boldsymbol{x})] - \sum_{i=1,j=1}^{N,M} D_{KL}[q(\boldsymbol{u}_{H_{i,j}}) \,\|\, p(\boldsymbol{u}_{H_{i,j}})] - \sum_{j=1}^{M} D_{KL}[q(\boldsymbol{u}_{\tilde{x}_j}) \,\|\, p(\boldsymbol{u}_{\tilde{x}_j})] \\
&= \underbrace{-\frac{1}{2}\log[(2\pi)^N |\boldsymbol{\Lambda}|^2] - \frac{1}{2}\sum_{i=1}^{Y} \mathbb{E}_q\{\|\boldsymbol{\Lambda}^{-1}[\boldsymbol{Y}_{i,:} - \boldsymbol{y}(\boldsymbol{T}_{i,:})]\|^2\}}_{\text{data reconstruction cost}} \\
&\quad \underbrace{- \sum_{i=1,j=1}^{N,M} D_{KL}[q(\boldsymbol{u}_{H_{i,j}}) \,\|\, p(\boldsymbol{u}_{H_{i,j}})] - \sum_{j=1}^{M} D_{KL}[q(\boldsymbol{u}_{\tilde{x}_j}) \,\|\, p(\boldsymbol{u}_{\tilde{x}_j})]}_{\text{divergence from model}}.
\end{aligned}
$$

We observe that the variational free energy can interpreted as negative a cost of reconstructing the data regularised by a divergence from the model prior; thus, maximisation of the free energy seeks an explanation of the data that is compatible with the model prior.

Appendix I derives an analytical expression for the variational free energy. Appendix D discusses some useful techniques concerning implementation of the free energy.

Furthermore, we consider the asymptotic time complexity of computing the free energy. Denote the number of inducing points for the processes $H_{i,j}$ and $\tilde{x}_j$ by $T_{H_{i,j}}$ and $T_{\tilde{x}_j}$ respectively. Let $T_H = \max_{i,j} T_{H_{i,j}}$ and $T_{\tilde{x}} = \max_j T_{\tilde{x}_j}$. Then Appendix I.5 shows that the time complexity of computing the free energy is given by

$$\mathcal{O}[NMT_H^3 + MT_{\tilde{x}}^3 + YNM(T_H^2 T_{\tilde{x}} + KT_H^2 + KT_H T_{\tilde{x}} + KT_{\tilde{x}}^2 + T_H T_{\tilde{x}}^2) + YNM^2].$$

Recall that $\boldsymbol{H}$ has effectively limited support due to use of the decaying exponentiated-quadratic kernel; thus likely $T_{\tilde{x}} \gg T_H$. Also, likely $T_{\tilde{x}} \gg M$. In that case the asymptotic time complexity of computing the free energy simplies to $\mathcal{O}[MT_{\tilde{x}}^3 + YNM(K + T_H)T_{\tilde{x}}^2]$.

Finally, we discuss initialisation of the parameters of the variational free energy—that is, initialisation of $\boldsymbol{\mu}_{H_{i,j}}$, $\boldsymbol{\Sigma}_{H_{i,j}}$, $\boldsymbol{\mu}_{\tilde{x}_j}$ and $\boldsymbol{\Sigma}_{\tilde{x}_j}$. Optimisation of the free energy is usually performed via gradient-based methods. This means that the optimiser is likely to become trapped in a local minimum if the parameters of the free energy are initialised far from their optimal values. Hence proper initialisation of the parameters is of crucial importance.

We propose the following initialisation: To begin with, fit the predictive mean to the data using weighted least squares [Tobar and Turner, 2016]. Appendix I.2 shows that this is equivalent to performing inference in the following model:

Model 7 (Basis Function Model)**.** *Let* $\boldsymbol{f}$ *be such that*

$$f_i(\boldsymbol{t}) = \sum_{j=1}^{M} \boldsymbol{\mu}_{H_{i,j}}^T \boldsymbol{K}_{\boldsymbol{u}_{H_{i,j}}}^{-1} \boldsymbol{I}^{(L,H_{i,j},x_j)}(\boldsymbol{t}) \boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_j}}^{-1} \boldsymbol{\mu}_{\tilde{x}_j}$$

for some kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{x}}$*, some kernel* $\boldsymbol{\mathcal{K}}_{\boldsymbol{H}}$ *and some vectors* $\boldsymbol{\mu}_{H_{i,j}}$ *and* $\boldsymbol{\mu}_{\tilde{x}_j}$*. Afterwards draw* $\boldsymbol{\varepsilon} \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\Lambda}^2)$ *for some diagonal matrix* $\boldsymbol{\Lambda}$*. Then observations are generated by*

$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\varepsilon}$.

Learning $\boldsymbol{\mu}_{H_{i,j}}$ and $\boldsymbol{\mu}_{\tilde{x}_j}$ through Model 7 yields arguably sensible initial values. Finally, we initialise $\boldsymbol{\Sigma}_{H_{i,j}}$ and $\boldsymbol{\Sigma}_{\tilde{x}_j}$ to fractions of their priors $\boldsymbol{K}_{\boldsymbol{u}_{H_{i,j}}}$ and $\boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_j}}$.

Observe that Model 7 is a multi-output basis function model that is structured in its parametrisation. Model 7 is a generalisation of the basis function model presented by Tobar and Turner [2016]; the case $N = M = K = 1$ recovers their model exactly.

3.7 Conclusion

The Generalised Gaussian Process Convolution Model enabled us to formulate Model 5. Model 5 addresses the kernel design problem in multi-output Gaussian processes on multidimensional input spaces by parameterising the kernel with another Gaussian process. By modelling the kernel nonparametrically we avoid choosing a kernel of parametric form; instead, we infer its form from the data.

Finally, Model 5 is intimately connected to other models (Section 3.5). Most notably, Model 5 enabled us to formulate Model 6, which forms a dual formulation of the cross-spectral mixture kernel and reveals Model 5 as a cross-spectral mixture kernel with an infinite number of components.

3.8 Discussion

If $T_{\tilde{x}} \gg T_H$ and $T_{\tilde{x}} \gg M$, then Section 3.6 shows that the variational free energy of Model 5 can be computed in $\mathcal{O}[MT_{\tilde{x}}^3 + YNM(K + T_H)T_{\tilde{x}}^2]$ time. This complexity scales linearly in Y. However, in many settings—for example, time series—the size of the data is proportional to the size of the space the data occupies. Now, the inducing points for $\boldsymbol{x}$ must cover the space the data occupies. Therefore, in such settings, $T_{\tilde{x}}$ effectively scales with Y, which means that cost of computing the free energy effectively scales with Y^3.

4 | Multi-Task Learning

4.1 Introduction

Many inference problems involve dealing with multiple signals. These problems are best solved not only by learning signals individually, but also by simultaneously exploiting dependencies between signals. The signals can be interpreted as the output of multiple *tasks*; a problem involving multiple signals is therefore commonly referred to as a *multi-task* problem. We will use the terms "multi-output" and "multi-task" interchangeably.

A prominent example of a multi-task problem is the prediction of concentration levels of pollutants in geostatistics [Álvarez and Lawrence, 2011; Álvarez et al., 2009; Wilson et al., 2012]. Pollutants are often expensive to sample, which hinders accurately predicting their concentration levels. Fortunately, pollutants often strongly correlate with other substances, which can be cheap to sample. The multi-task predictor aims to improve predicting concentration levels of scarcely sampled pollutants by exploiting their correlation with densely sampled cheap substances.

Numerous multi-task models have been developed, many of which are Gaussian process models. Now, the Generalised Gaussian Process Convolution Model (Model 4) is also a multi-output model. This chapter provides an overview of multi-output models from the geostatistics and machine learning literature and shows how Model 4 fits in.

4.2 Mixing Models

A model for a multi-output signal $\boldsymbol{f}$ can be constructed by assuming that at any $\boldsymbol{t}$ the output $\boldsymbol{f}(\boldsymbol{t})$ is explained by the value $\boldsymbol{x}(\boldsymbol{t})$ of some latent process $\boldsymbol{x}$ with independent components. If the relationship between $\boldsymbol{f}(\boldsymbol{t})$ and $\boldsymbol{x}(\boldsymbol{t})$ is linear, then we arrive at the Instantaneous Mixing Model (IMM):

Model 8 (Instantaneous Mixing Model (IMM))**.** *Let* $\boldsymbol{H}$ *be a matrix. Draw*

$$\begin{aligned}\boldsymbol{x} &\sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\mathcal{K}}_{\boldsymbol{x}}),\\ \boldsymbol{\varepsilon} &\sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\Lambda}^2)\end{aligned}$$

independently for some diagonal kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{x}}$ *and some diagonal matrix* $\boldsymbol{\Lambda}$*. Then observations are generated by* $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\varepsilon} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\varepsilon}$*.*

Observe that Model 8 corresponds to ordinary Gaussian process regression with a kernel of the form $\boldsymbol{\mathcal{K}}_{\boldsymbol{f}} = \boldsymbol{H}\boldsymbol{\mathcal{K}}_{\boldsymbol{x}}\boldsymbol{H}^T$ for some matrix $\boldsymbol{H}$ and some diagonal kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{x}}$. Alternatively, Model 8 can be interpreted as a factor analysis model at all $\boldsymbol{t}$.

Model 8 assumes that $\boldsymbol{f}(\boldsymbol{t})$ depends only on $\boldsymbol{x}(\boldsymbol{t})$, meaning that $\boldsymbol{f}(\boldsymbol{t})$ is independent of $\boldsymbol{x}(\boldsymbol{t}')$ for $\boldsymbol{t}' \neq \boldsymbol{t}$. This assumption limits the expressivity of $\boldsymbol{f}$ significantly; for example, $\boldsymbol{f}$ cannot be a smoothed version of $\boldsymbol{x}$. We therefore relax Model 8 by instead letting $\boldsymbol{f}(\boldsymbol{t})$ depend on $\boldsymbol{x}(\boldsymbol{t})$ for all $\boldsymbol{t}$. If their relationship is again linear, then we arrive at the Convolutional Mixing Model (CMM):

Model 9 (Convolutional Mixing Model (CMM))**.** *Let* $\boldsymbol{H}$ *be a matrix-valued function. Draw*

$$\begin{aligned}\boldsymbol{x} &\sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\mathcal{K}}_{\boldsymbol{x}}),\\ \boldsymbol{\varepsilon} &\sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\Lambda}^2)\end{aligned}$$

independently for some diagonal kernel $\boldsymbol{\mathcal{K}}_{\boldsymbol{x}}$ *and some diagonal matrix* $\boldsymbol{\Lambda}$*. Then observations are generated by* $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\varepsilon} = \boldsymbol{H} * \boldsymbol{x} + \boldsymbol{\varepsilon}$*.*

Observe that Model 9 is a generalisation of Model 8, since letting $\boldsymbol{H} = \delta\boldsymbol{H}'$ for some constant matrix $\boldsymbol{H}'$ yields that

$$\boldsymbol{f}(\boldsymbol{t}) = (\delta\boldsymbol{H}' * \boldsymbol{x})(\boldsymbol{t}) = \int_{\mathbb{R}^K} \delta(\boldsymbol{t} - \boldsymbol{\tau})\boldsymbol{H}'\boldsymbol{x}(\boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau} = \boldsymbol{H}'\boldsymbol{x}(\boldsymbol{t}).$$

4.3 The Mixing Model Hierarchy

Many multi-output Gaussian process models from the geostatistics and machine learning literature can be identified as specialisations of Models 8 and 9; Table 4.1 shows the identification of the intrinsic coregionalisation model (ICM) [Goovaerts, 1997], the lin-

Model	Form of $\boldsymbol{H}$	Form of $\boldsymbol{\mathcal{K}_x}$	Mixing type
ICM [Goovaerts, 1997]	$\boldsymbol{H}$	$(\sum_{q=1}^{Q} k_x^{(q)})\boldsymbol{I}$	Instantaneous
LCM [Goovaerts, 1997]	$\begin{bmatrix}\boldsymbol{H}_1 & \cdots & \boldsymbol{H}_Q\end{bmatrix}$	$\operatorname{diag}(k_x^{(1)}\boldsymbol{I}, \ldots, k_x^{(Q)}\boldsymbol{I})$	Instantaneous
SLFM [Teh and Seeger, 2005]	$\boldsymbol{H}$	$\boldsymbol{\mathcal{K}_x}$	Instantaneous
MTGPM [Bonilla et al., 2008]	$\boldsymbol{H}$	$k_x\boldsymbol{I}$	Instantaneous
LFM [Álvarez et al., 2009]	Green's function	$\boldsymbol{\mathcal{K}_x}$	Convolutional
CMOGPM [Álvarez and Lawrence, 2011]	$\begin{bmatrix}\boldsymbol{H}_1 & \cdots & \boldsymbol{H}_Q\end{bmatrix}$	$\operatorname{diag}(k_x^{(1)}\boldsymbol{I}, \ldots, k_x^{(Q)}\boldsymbol{I})$	Convolutional
CGPM [Nguyen and Bonilla, 2014]	$\begin{bmatrix}\boldsymbol{H} & \boldsymbol{I}\end{bmatrix}$	$\boldsymbol{\mathcal{K}_x}$	Instantaneous

Table 4.1: Identification of multi-output models from the geostatistics and machine learning literature as specialisations of Models 8 and 9

ear coregionalisation model (LCM) [Goovaerts, 1997], the semiparametric latent factor model (SLFM) [Teh and Seeger, 2005], the multi-task Gaussian process model (MTGPM) [Bonilla et al., 2008], the latent force model (LFM) [Álvarez et al., 2009], the convolved multi-output Gaussian process model (CMOGPM) [Álvarez and Lawrence, 2011] and the collaborative Gaussian processes model (CGPM) [Nguyen and Bonilla, 2014].

Model 9 is also closely related to Model 4. Specifically, Model 4 is a generalisation of Model 9 where $\boldsymbol{H}$ is modelled stochastically. This shows that Models 4, 8 and 9 form a hierarchy—the *mixing model hierarchy*—in which Model 4 generalises Model 9 and Model 9 generalises Model 8.

Figure 4.1 organises the main models presented in this thesis, the cross-spectral mixture kernel (CSMK) [Ulrich et al., 2015], and the models in Table 4.1 according to the mixing model hierarchy. This yields an overview of many multi-output Gaussian process models from the geostatistics and machine learning literature that emphasises their distinctive modelling assumptions. An immediate result is that the cross-spectral mixture kernel can now be connected to other multi-output models.

An important model that does not fit in the mixing model hierarchy is the Gaussian process regression network (GPRN) [Wilson et al., 2012].

4.4 Conclusion

We have presented the mixing model model hierarchy, which organises many multi-output Gaussian process from the geostatistics and machine learning literature according

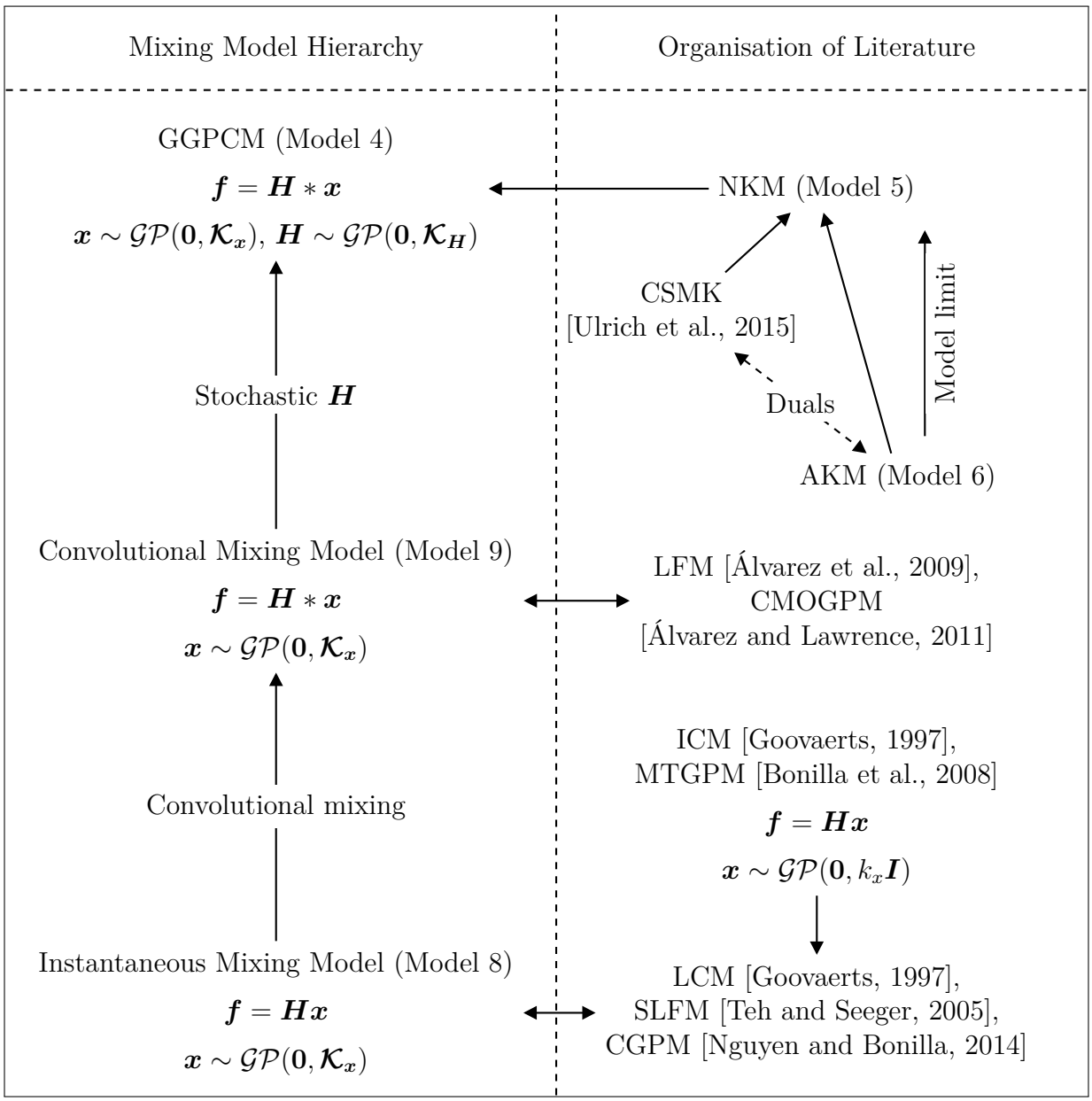


Figure 4.1: Organisation of multi-output models from the geostatistics and machine learning literature according to the mixing model hierarchy. Single-header arrows indicate generalisation and double-headed arrows indicate equivalence. "Model limit" refers to taking the model's number of components to infinity.

to their distinctive modelling assumptions.

Model 4 fits in as a generalisation of Model 9 and Model 8. This shows that many multi-output Gaussian process models from the current literature can be derived from Model 4, with the exception being the Gaussian process regression network.

5 | The Deep Kernel Model

5.1 Introduction

Chapter 3 developed the idea that the kernel design problem can be addressed by modelling the kernel nonparametrically. In retrospect, this approach actually replaces the kernel design problem with another one: we now have to design the *kernel of the kernel*. Fortunately, we know already how to solve this kernel design problem: we can parametrise the *kernel of the kernel* with another Gaussian process. But then we have to design the *kernel of the kernel of the kernel*...

This chapter investigates a model that not only models the kernel nonparametrically, but also the *kernel of the kernel*, the *kernel of the kernel of the kernel*, and further *"deeper" kernels*. We confine the presentation to one-dimensional signals f on one-dimensional input spaces.

5.2 The Deep Kernel Model

Consider the Nonparametric Kernel Model (Model 5). In the case of one-dimensional signals on one-dimensional input spaces Model 5 recovers the Gaussian Process Convolution Model (GPCM) [Tobar et al., 2015b] (Section 3.5):

Model 10 (Gaussian Process Convolution Model [Tobar et al., 2015b])**.** *Draw*

$$\begin{aligned} h &\sim \mathcal{GP}(0, \mathcal{K}_h), \\ x &\sim \mathcal{GP}[0, \delta(t_1 - t_2)], \\ \varepsilon &\sim \mathcal{GP}[0, \sigma^2 \delta(t_1 - t_2)] \end{aligned}$$

independently for some kernel $\mathcal{K}_h$ and some constant σ. Then observations are generated by $y = f + \varepsilon = h * x + \varepsilon$.

Recall that Model 10 corresponds to ordinary Gaussian process regression in which the kernel is modelled nonparametrically (Section 3.3). We immediately recognise that the kernel design problem is replaced with another one: we now have to choose $\mathcal{K}_h$—the *kernel of the kernel*. We investigate whether $\mathcal{K}_h$ can be modelled nonparametrically.

Modelling $\mathcal{K}_h$ nonparametrically immediately presents a problem. In Section 3.3.1 we ensured that f has finite variance by letting $\mathcal{K}_h$ be a *decaying* exponentiated-quadratic kernel. However, if we model $\mathcal{K}_h$ nonparametrically, then it is not clear how we must restrict the prior on $\mathcal{K}_h$ to ensure that f has finite variance. We investigate by examining the case that $\mathcal{K}_h$ is a *non-decaying* exponentiated-quadratic kernel. In that case f has infinite variance (Section 3.3.1). Now, let $w(t) = \exp(-\alpha t^2)$, $t \in \mathbb{R}$ and consider the process wh. It holds that

$$\begin{aligned}
\mathcal{K}_{wh}(t_1, t_2) &= \mathbb{E}[w(t_1)h(t_1)w(t_2)h(t_2)] \\
&= w(t_1)w(t_2)\mathbb{E}[h(t_1)h(t_2)] \\
&= \exp(-\alpha t_1^2)\exp(-\alpha t_2^2)\sigma_h^2 \exp[-\gamma(t_1 - t_2)^2] \\
&= \sigma_h^2 \exp[-\alpha t_1^2 - \alpha t_2^2 - \gamma(t_1 - t_2)^2],
\end{aligned}$$

which is a *decaying* exponentiated-quadratic kernel. In other words, despite h having a kernel for which f has infinite variance, multiplication by w yields a kernel for which f has finite variance. This suggests the following, more robust formulation of the GPCM:

Model 11 (Gaussian Process Convolution Model (Explicit Decay))**.** *Draw*

$$\begin{aligned}
h &\sim \mathcal{GP}(0, \mathcal{K}_h), \\
x &\sim \mathcal{GP}[0, \delta(t_1 - t_2)], \\
\varepsilon &\sim \mathcal{GP}[0, \sigma^2\delta(t_1 - t_2)]
\end{aligned}$$

independently for some kernel $\mathcal{K}_h$ and some constant σ. Let $w(t) = \exp(-\alpha t^2)$, $w \in \mathbb{R}$. Then observations are generated by $y = f + \varepsilon = wh * x + \varepsilon$.

It now holds that

$$\begin{aligned}\mathbb{E}[f^2(t)] &= \int_{\mathbb{R}^2} w(t-\tau_1)w(t-\tau_2)\mathbb{E}[h(t-\tau_1)h(t-\tau_2)]\underbrace{\mathbb{E}[x(\tau_1)x(\tau_2)]}_{\delta(\tau_1-\tau_2)}\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \\ &= \int_{\mathbb{R}} w^2(\tau)\mathcal{K}_h(\tau,\tau)\,\mathrm{d}\tau,\end{aligned}$$

which clearly is finite if $\mathcal{K}_h$ is bounded. In other words, if h has finite variance, then f has finite variance. Therefore, we can safely model $\mathcal{K}_h$ nonparametrically if we ensure that h has finite variance.

To model $\mathcal{K}_h$ nonparametrically, consider the following extension of Model 11:

Model 12 (Deep Gaussian Process Convolution Model)**.** *Let N be the model's order. Draw*

$$\begin{aligned}h_1 &\sim \mathcal{GP}(0,\mathcal{K}_h),\\ x_1 &\sim \mathcal{GP}[0,\delta(t_1-t_2)],\\ &\vdots\\ x_N &\sim \mathcal{GP}[0,\delta(t_1-t_2)],\\ \varepsilon &\sim \mathcal{GP}[0,\sigma^2\delta(t_1-t_2)]\end{aligned}$$

independently for some kernel $\mathcal{K}_h$ and some constant σ. Let $w(t) = \exp(-\alpha t^2)$, $t \in \mathbb{R}$. Then observations are generated by y where

$$\begin{aligned}h_1 &= h,\\ h_2 &= wh_1 * x_1,\\ &\vdots\\ h_{N+1} &= wh_N * x_N,\\ f &= h_{N+1},\\ y &= f + \varepsilon.\end{aligned}$$

Observe that each $h_{i+1} \,|\, h_i$ is a linear combination of Gaussian processes. Hence $h_{i+1} \,|\, h_i$ is another Gaussian process, which thus can be identified by its mean function and

kernel:

$$\begin{aligned}
\mathbb{E}[h_{i+1}(t) \,|\, h_i] &= \int_{\mathbb{R}} w(t-\tau) h_i(t-\tau) \mathbb{E}[x_i(\tau)] \,\mathrm{d}\tau = 0, \\
\mathcal{K}_{h_{i+1} \,|\, h_i}(t_1, t_2) &= \mathbb{E}[h_{i+1}(t_1) h_{i+1}(t_2) \,|\, h_i] \\
&= \int_{\mathbb{R}^2} w(t_1-\tau_1) h_i(t_1-\tau_1) w(t_2-\tau_2) h_i(t_2-\tau_2) \underbrace{\mathbb{E}[x_i(\tau_1) x_i(\tau_2)]}_{\delta(\tau_1-\tau_2)} \,\mathrm{d}\tau_1 \,\mathrm{d}\tau_2 \\
&= \int_{\mathbb{R}} w[\tau - (t_2-t_1)] h_i[\tau - (t_2-t_1)] w(\tau) h_i(\tau) \,\mathrm{d}\tau \\
&= [R(wh_i) * wh_i](t_2 - t_1) \\
&= \mathcal{K}_{h_{i+1}}(t_1 - t_2).
\end{aligned}$$

We have established the following equivalent model:

Model 13 (Deep Kernel Model)**.** *Let N be the model's order. Draw*

$$\begin{aligned}
h &\sim \mathcal{GP}(0, \mathcal{K}_h), \\
\varepsilon &\sim \mathcal{GP}[0, \sigma^2 \delta(t_1 - t_2)]
\end{aligned}$$

independently for some kernel $\mathcal{K}_h$ and some constant σ. Let $h = h_1$ and $w(t) = \exp(-\alpha t^2)$, $t \in \mathbb{R}$. Afterwards draw

$$h_{i+1} \,|\, h_i \sim \mathcal{GP}\{0, [R(wh_i) * wh_i](t_2 - t_1)\}$$

in the order $i = 1, \ldots, N$. Then observations are generated by $y = f + \varepsilon = h_{N+1} + \varepsilon$.

Model 13 shows that the kernel of $f = h_{N+1}$ is now parametrised by h_N, whose kernel is parametrised by h_{N-1}, whose kernel is parametrised by h_{N-2}, et cetera. That is, we have successfully formulated a nonparametric model of the kernel, the *kernel of the kernel*, the *kernel of the kernel of the kernel*, and further *"deeper" kernels*. Furthermore, it holds

that

$$\begin{aligned}
\mathbb{E}[f^2(t)] &= \mathbb{E}[(w\{\cdots[w(wh_1 * x_1) * x_2]\cdots\} * x_N)^2(t)] \\
&= \int \Big[\prod_{i=1}^{N} w\Big(t - \sum_{j=i}^{N} \tau_j^{(1)}\Big) w\Big(t - \sum_{j=i}^{N} \tau_j^{(2)}\Big)\Big] \\
&\qquad \mathbb{E}\Big[h_1\Big(t - \sum_{i=1}^{N} \tau_i^{(1)}\Big) h_1\Big(t - \sum_{i=1}^{N} \tau_i^{(2)}\Big)\Big] \prod_{i=1}^{N} \underbrace{\mathbb{E}[x_i(\tau_i^{(1)}) x_i(\tau_i^{(2)})]}_{\delta(\tau_i^{(1)} - \tau_i^{(2)})} \,\mathrm{d}\tau_i^{(1)}\,\mathrm{d}\tau_i^{(2)} \\
&= \int \Big[\prod_{i=1}^{N} w^2\Big(t - \sum_{j=i}^{N} \tau_j\Big)\Big] \mathcal{K}_{h_1}\Big(t - \sum_{i=1}^{N} \tau_i, t - \sum_{i=1}^{N} \tau_i\Big) \prod_{i=1}^{N} \mathrm{d}\tau_i \\
&= \int_{\mathbb{R}} w^2(\tau_1) \int_{\mathbb{R}} w^2(\tau_1 + \tau_2) \cdots \int_{\mathbb{R}} w^2\Big(\sum_{i=1}^{N} \tau_i\Big) \mathcal{K}_{h_1}\Big(\sum_{i=1}^{N} \tau_i, \sum_{i=1}^{N} \tau_i\Big)\,\mathrm{d}\tau_N \cdots \mathrm{d}\tau_2\,\mathrm{d}\tau_1.
\end{aligned}$$

Observe that the most inner integral is finite if $\mathcal{K}_{h_1}$ is bounded, in which case all integrals are finite. In other words, if h_1 has finite variance, then f has finite variance. Therefore Models 12 and 13 are well defined.

5.2.1 Network Interpretation

The left side of Figure 5.1 depicts the graphical model of Model 13. We recognise the "deep" kernel structure by the chain of latent h_i's.

The right side of Figure 5.1 depicts the graphical model of Model 13 if the latent processes h_i are expanded into their function values; that is, any h_i is instead represented by $h_i(t)$ for all t—uncountably many t. Now, since the kernel of h_{i+1} is a convolution of h_i, it holds that every $h_{i+1}(t)$ depends on $h_i(t')$ for all t'. Hence the resulting graphical model is a fully connected network with layers of infinite size. This is very reminiscent of neural networks. The key difference between a neural network and Model 13 is that in a neural network layer i forms the input of layer $i+1$, whereas in Model 13 layer i *parametrises* layer $i+1$.

We can truncate these layers of infinite size to finite size by approximating the kernels in Model 13 by their expectations conditioned on some finite number of observations. We then obtain the Approximate Deep Kernel Model:

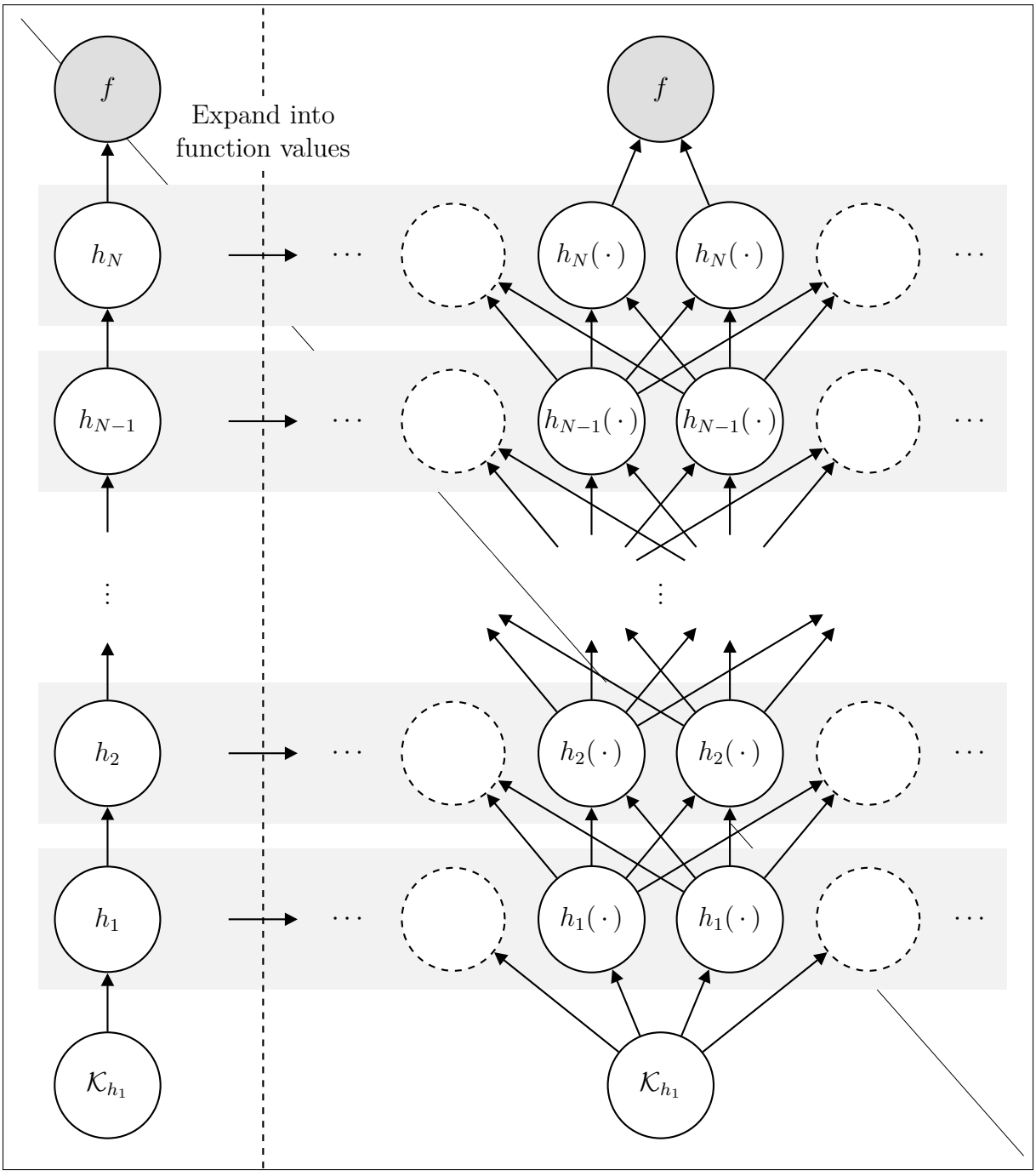


Figure 5.1: Graphical model of Model 13. Shows the resulting network structure if the latent h_i are expanded into their function values.

Model 14 (Approximate Deep Kernel Model)**.** *Let N be the model's order. Draw*

$$\begin{aligned} h &\sim \mathcal{GP}(0, \mathcal{K}_h), \\ \varepsilon &\sim \mathcal{GP}[0, \sigma^2 \delta(t_1 - t_2)] \end{aligned}$$

independently for some kernel $\mathcal{K}_h$ and some constant σ. Let $h = h_1$ and $w(t) = \exp(-\alpha t^2)$, $w \in \mathbb{R}$. Afterwards draw

$$\begin{aligned} \mathcal{K}_{h_{i+1} \,|\, h_i}(t_1, t_2) &= \mathbb{E}\{[R(wh_i) * wh_i](t_2 - t_1) \,|\, h_i(\boldsymbol{t})\}, \\ h_{i+1} \,|\, h_i &\sim \mathcal{GP}(0, \mathcal{K}_{h_{i+1} \,|\, h_i}) \end{aligned}$$

in the order $i = 1, \ldots, N$. Then observations are generated by $y = f + \varepsilon = h_{N+1} + \varepsilon$.

Note that Model 14 approximates Model 13 in same way that Model 6 approximates Model 5.

5.3 Illustrative Samples

In the following we let h_1 have an exponentiated-quadratic kernel.

Figure 5.2 illustrates the generative process of Model 13 in the case that $N = 10$. Observe that the produced $h_i \,|\, h_{i-1}$'s and $\mathcal{K}_{h_i \,|\, h_{i-1}}$'s change in complexity as the generation progresses; that is, if by chance $h_i \,|\, h_{i-1}$ is slightly more complicated, then $\mathcal{K}_{h_{i+1} \,|\, h_i}$ and thereby $h_{i+1} \,|\, h_i$ tend to be more complicated as well. Thus, in each generation step Model 13 can either simplify or complicate the kernel, which means that $\mathcal{K}_{f \,|\, h_{10}}$ can be of a vastly different form than the kernel produced by h_1; Model 13 therefore seems to be less restricted by h_1's kernel than Model 10 is by h's kernel.

Figure 5.3 shows observations from Model 13 in the cases that $N = 1$ and $N = 20$—a *shallow* and *deep* model respectively. Consider the shallow model. Observe that its observations exhibit barely any variation in smoothness and degree of periodicity; that is, their kernels are all of a similar form and their power spectral densities all show a similar low-pass structure. This is due h_1's exponentiated-quadratic kernel, which imposes smoothness on h_1 and thereby directly on the observation's kernel. On the other hand, consider the deep model. Observe that its observations are much richer: they can be periodic, aperiodic, smoothly varying, or anywhere inbetween. This is reflected in the generated kernels, which now show great diversity. Furthermore, instead of the low-pass structure observed for the shallow model, the power spectral densities

now show low-pass structures, high-pass structures, band-pass structures, and structures inbetween.

5.4 Conclusion

We have presented Model 13, which not only models the kernel nonparametrically, but also the *kernel of the kernel*, the *kernel of the kernel of the kernel*, and further *"deeper" kernels*. Experiments showed that Model 13 exhibits greatly increased expressivity compared to its shallow counterpart Model 10, or equivalently Model 5 where $M = N = K = 1$.

5.5 Discussion

We can perform approximate inference in Model 13 along the lines of Section 3.6. Although the free energy allows to be solved for analytically, its resulting complexity is exponential in the number of layers. Therefore, to perform inference in Model 13, additional approximations are probably necessary.

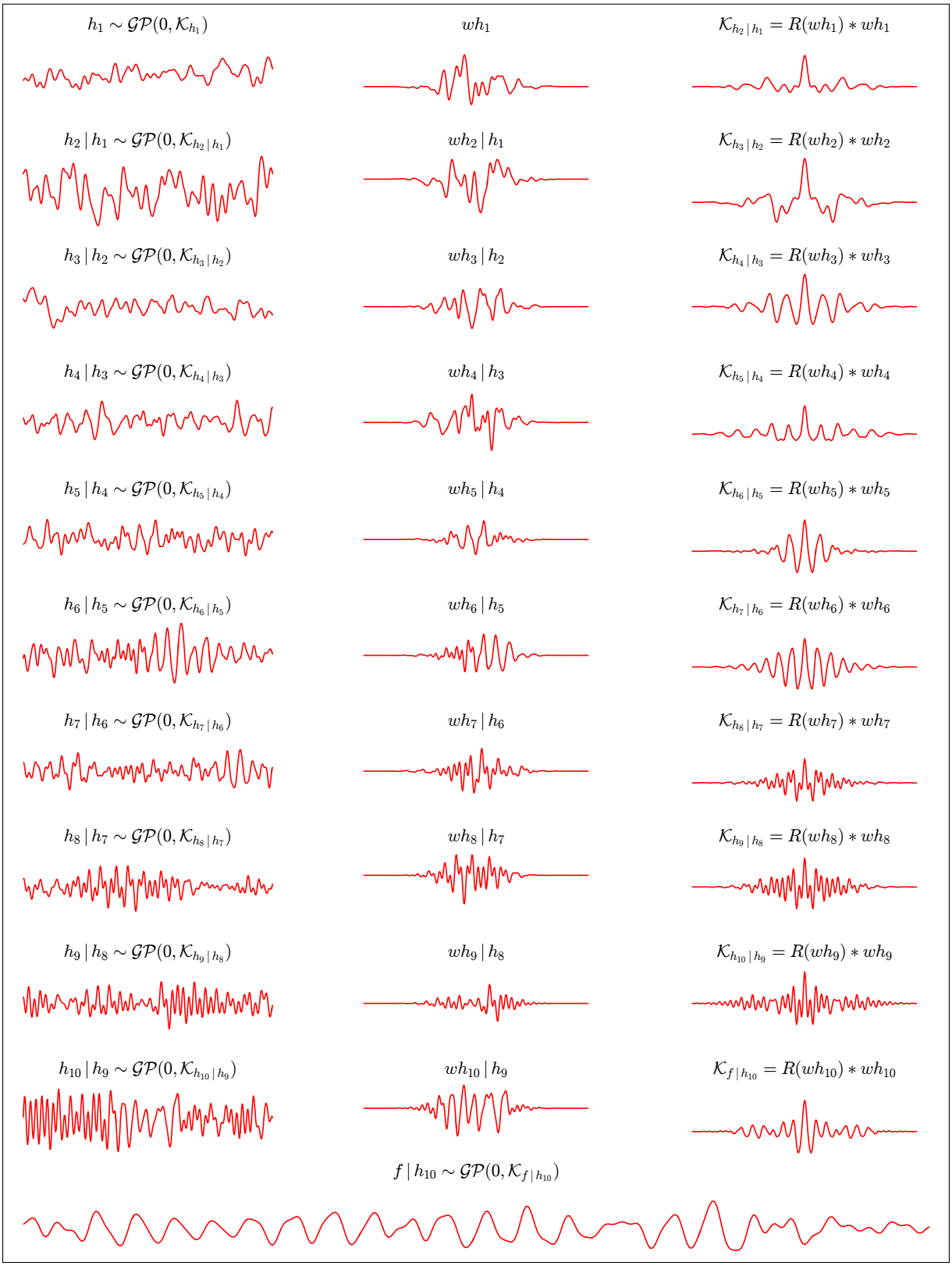


Figure 5.2: Generative process of Model 13 in the case that $N = 10$

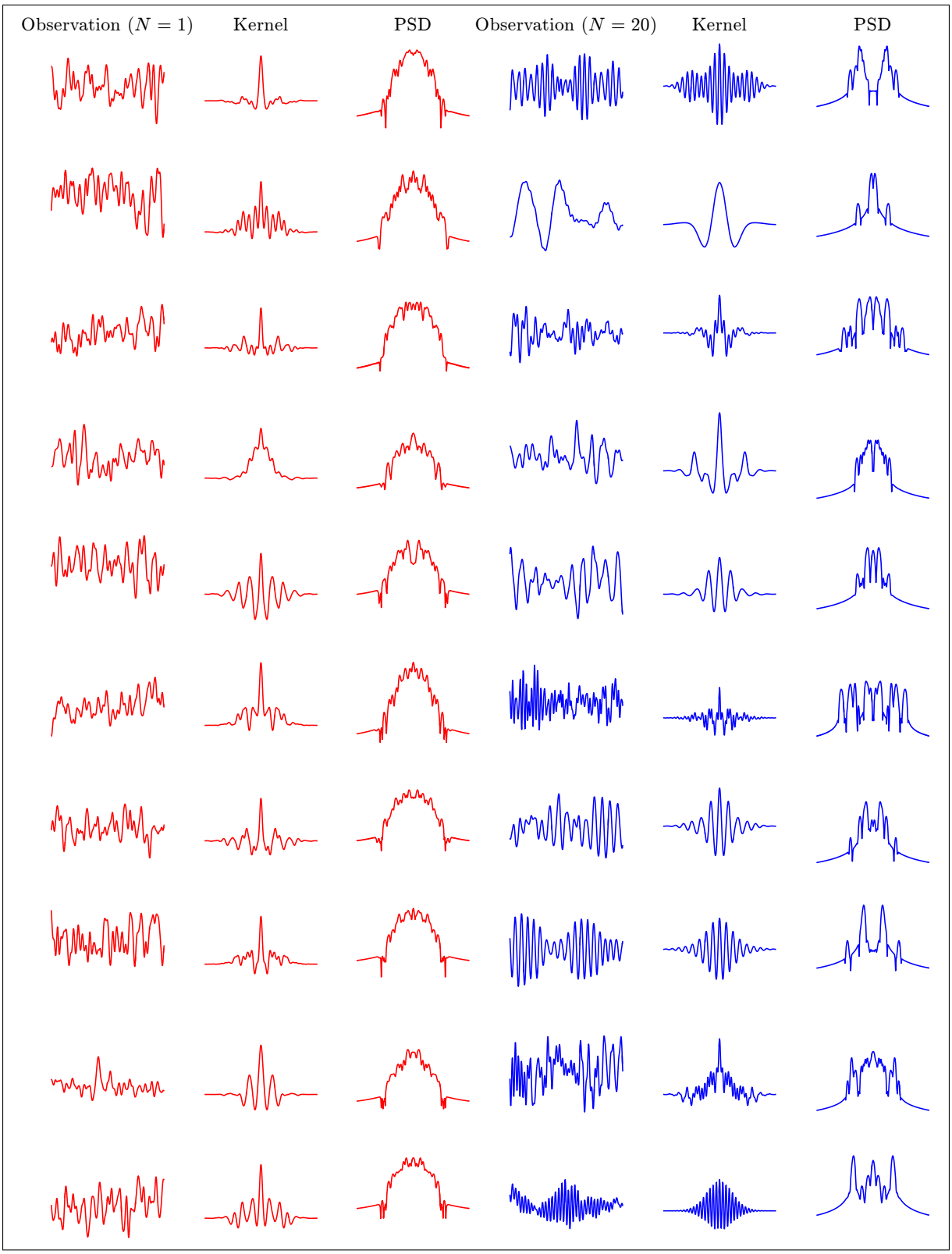


Figure 5.3: Observations from Model 13 in the cases that $N = 1$ and $N = 20$. Also shows the kernels and power spectral densities (PSDs) of the observations.

A | Solution of the Linear State-Space Model

A.1 Time-Variant Solution

Consider

$$\begin{aligned} \boldsymbol{s}'(t) &= \boldsymbol{A}(t)\boldsymbol{s}(t) + \boldsymbol{B}(t)\boldsymbol{x}(t), \\ \boldsymbol{f}(t) &= \boldsymbol{C}(t)\boldsymbol{s}(t) + \boldsymbol{D}(t)\boldsymbol{x}(t). \end{aligned} \tag{A.1}$$

Let $\boldsymbol{\Psi}(t)$ be a fundamental matrix of the homogeneous system—that is, $\boldsymbol{\Psi}'(t) = \boldsymbol{A}(t)\boldsymbol{\Psi}(t)$ where $\boldsymbol{\Psi}(t)$'s columns are independent. We proceed to find the general solution via variation of parameters. Assume a solution of the form $\boldsymbol{s}(t) = \boldsymbol{\Psi}(t)\boldsymbol{c}(t)$. Then application of the product rule yields that

$$\boldsymbol{s}'(t) = \boldsymbol{\Psi}'(t)\boldsymbol{c}(t) + \boldsymbol{\Psi}(t)\boldsymbol{c}'(t) = \boldsymbol{A}(t)\boldsymbol{\Psi}(t)\boldsymbol{c}(t) + \boldsymbol{\Psi}(t)\boldsymbol{c}'(t)$$

while substitution into Equation (A.1) yields that

$$\boldsymbol{s}'(t) = \boldsymbol{A}(t)\boldsymbol{\Psi}(t)\boldsymbol{c}(t) + \boldsymbol{B}(t)\boldsymbol{x}(t).$$

Hence $\boldsymbol{\Psi}(t)\boldsymbol{c}'(t) = \boldsymbol{B}(t)\boldsymbol{x}(t)$. Since $\boldsymbol{\Psi}(t)$ is invertible we have that $\boldsymbol{c}'(t) = \boldsymbol{\Psi}^{-1}(t)\boldsymbol{B}(t)\boldsymbol{x}(t)$. Therefore

$$\begin{aligned}
\boldsymbol{f}(t) &= \boldsymbol{C}(t)\boldsymbol{\Psi}(t)\boldsymbol{c}(t) + \boldsymbol{D}(t)\boldsymbol{x}(t) \\
&= \boldsymbol{C}(t)\boldsymbol{\Psi}(t)\int_{-\infty}^{t} \boldsymbol{\Psi}^{-1}(\tau)\boldsymbol{x}(\tau)\boldsymbol{B}(t)\,\mathrm{d}\tau + \boldsymbol{D}(t)\boldsymbol{x}(t) \\
&= \int_{-\infty}^{t} \boldsymbol{C}(t)\boldsymbol{\Psi}(t)\boldsymbol{\Psi}^{-1}(\tau)\boldsymbol{B}(t)\boldsymbol{x}(\tau)\,\mathrm{d}\tau + \boldsymbol{D}(t)\boldsymbol{x}(t) \\
&= \int_{\mathbb{R}} \underbrace{[\mathbb{1}(\tau \leq t)\boldsymbol{C}(t)\boldsymbol{\Psi}(t)\boldsymbol{\Psi}^{-1}(\tau)\boldsymbol{B}(t) + \delta(t-\tau)\boldsymbol{D}(t)]}_{\boldsymbol{H}(t,\tau)}\boldsymbol{x}(\tau)\,\mathrm{d}\tau \\
&= \int_{\mathbb{R}} \boldsymbol{H}(t,\tau)\boldsymbol{x}(\tau)\,\mathrm{d}\tau.
\end{aligned}$$

A.2 Time-Invariant Solution

Let $\boldsymbol{A}(t)$, $\boldsymbol{B}(t)$, $\boldsymbol{C}(t)$ and $\boldsymbol{D}(t)$ be independent of t; that is, let $\boldsymbol{A}(t) = \boldsymbol{A}$, $\boldsymbol{B}(t) = \boldsymbol{B}$, $\boldsymbol{C}(t) = \boldsymbol{C}$ and $\boldsymbol{D}(t) = \boldsymbol{D}$. The fundamental matrix is then given by the matrix exponential

$$\boldsymbol{\Psi}(t) = \sum_{i=0}^{\infty} \frac{t^i \boldsymbol{A}^i}{i!},$$

denoted as $\boldsymbol{\Psi}(t) = \exp t\boldsymbol{A}$. To show this, note that by the binomial theorem and a change of variables

$$\begin{aligned}
\exp[(s+t)\boldsymbol{A}] &= \sum_{i=0}^{\infty} \frac{(s+t)^i \boldsymbol{A}^i}{i!} \\
&= \sum_{i=0,j=0}^{\infty,\infty} \frac{s^{i-j}t^j \boldsymbol{A}^i}{(i-j)!j!} \\
&= \sum_{i=0,j=0}^{\infty,\infty} \frac{s^i t^j \boldsymbol{A}^{i+j}}{i!j!} \\
&= (\exp s\boldsymbol{A})(\exp t\boldsymbol{A})
\end{aligned}$$

so that

$$\begin{aligned}\boldsymbol{\Psi}'(t) &= \lim_{h\to 0} \frac{\exp[(s+h)\boldsymbol{A}] - \exp s\boldsymbol{A}}{h} \\ &= \left(\lim_{h\to 0} \frac{\exp h\boldsymbol{A} - \boldsymbol{I}}{h}\right) \exp s\boldsymbol{A} \\ &= \boldsymbol{A}\boldsymbol{\Psi}(t).\end{aligned}$$

Now

$$\boldsymbol{I} = \exp \boldsymbol{0} = \exp(t\boldsymbol{A} - t\boldsymbol{A}) = (\exp t\boldsymbol{A})[\exp(-t\boldsymbol{A})]$$

implies that $(\exp t\boldsymbol{A})^{-1} = \exp(-t\boldsymbol{A})$. Consequently

$$\begin{aligned}\boldsymbol{H}(t,\tau) &= \mathbb{1}(t-\tau \geq 0)\boldsymbol{C}\boldsymbol{\Psi}(t)\boldsymbol{\Psi}^{-1}(\tau)\boldsymbol{B} + \delta(t-\tau)\boldsymbol{D} \\ &= \mathbb{1}(t-\tau \geq 0)\boldsymbol{C}(\exp t\boldsymbol{A})(\exp \tau\boldsymbol{A})^{-1}\boldsymbol{B} + \delta(t-\tau)\boldsymbol{D} \\ &= \mathbb{1}(t-\tau \geq 0)\boldsymbol{C}(\exp t\boldsymbol{A})[\exp(-\tau\boldsymbol{A})]\boldsymbol{B} + \delta(t-\tau)\boldsymbol{D} \\ &= \mathbb{1}(t-\tau \geq 0)\boldsymbol{C}\exp[(t-\tau)\boldsymbol{A}]\boldsymbol{B} + \delta(t-\tau)\boldsymbol{D} \\ &= \boldsymbol{H}(t-\tau).\end{aligned}$$

B | Properties of the Multivariate Gaussian Distribution

B.1 Marginal and Conditional Distribution

Let

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2} \end{bmatrix} \right).$$

Then Murphy [2012] shows that $\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{1,1})$ and

$$\boldsymbol{x}_1 \,|\, \boldsymbol{x}_2 \sim \mathcal{N}[\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1}].$$

B.2 Kullback-Leibler Divergence

Let $\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ attain values in $\mathbb{R}^N$. Then Murphy [2012] shows that

$$D_{KL}[p(\boldsymbol{x}_1) \,\|\, p(\boldsymbol{x}_2)] = \frac{1}{2}\left[\operatorname{tr}(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} - N\right].$$

C | Multivariate Matrix-Valued Gaussian Processes

A stochastic process $\boldsymbol{F}(\boldsymbol{t}) \in \mathbb{R}^{N\times M}$, $\boldsymbol{t} \in \mathbb{R}^K$ is Gaussian if and only if for every $\boldsymbol{T} \in \mathbb{R}^{T\times K}$ it holds that $(\operatorname{vec}\boldsymbol{F})(\boldsymbol{T})$ is multivariate Gaussian distributed where

$$\begin{aligned}(\operatorname{vec}\boldsymbol{F})(\boldsymbol{T}) &= \begin{bmatrix}\operatorname{vec}\boldsymbol{F}(\boldsymbol{T}_{1,:}) \\ \vdots \\ \operatorname{vec}\boldsymbol{F}(\boldsymbol{T}_{T,:})\end{bmatrix} \\ &= \begin{bmatrix}F_{1,1}(\boldsymbol{T}_{1,:}) & \cdots & F_{N,M}(\boldsymbol{T}_{1,:}) & \cdots & F_{1,1}(\boldsymbol{T}_{T,:}) & \cdots & F_{N,M}(\boldsymbol{T}_{T,:})\end{bmatrix}^T.\end{aligned}$$

The mean function is then defined by

$$\begin{aligned}\boldsymbol{m}_{\boldsymbol{F}}(\boldsymbol{T}) &= \begin{bmatrix}\boldsymbol{m}_{\boldsymbol{F}}^T(\boldsymbol{T}_{1,:}) & \cdots & \boldsymbol{m}_{\boldsymbol{F}}^T(\boldsymbol{T}_{T,:})\end{bmatrix}^T \\ &= \begin{bmatrix}m_{F_{1,1}}(\boldsymbol{T}_{1,:}) & \cdots & m_{F_{N,M}}(\boldsymbol{T}_{1,:}) & \cdots & m_{F_{1,1}}(\boldsymbol{T}_{T,:}) & \cdots & m_{F_{N,M}}(\boldsymbol{T}_{T,:})\end{bmatrix}^T\end{aligned}$$

and the kernel by

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{F}}(\boldsymbol{T}^{(1)}, \boldsymbol{T}^{(2)}) = \begin{bmatrix}\boldsymbol{\mathcal{K}}_{\boldsymbol{F}}(\boldsymbol{T}^{(1)}_{1,:}, \boldsymbol{T}^{(2)}_{1,:}) & \cdots & \boldsymbol{\mathcal{K}}_{\boldsymbol{F}}(\boldsymbol{T}^{(1)}_{1,:}, \boldsymbol{T}^{(2)}_{T,:}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{K}}_{\boldsymbol{F}}(\boldsymbol{T}^{(1)}_{T,:}, \boldsymbol{T}^{(2)}_{1,:}) & \cdots & \boldsymbol{\mathcal{K}}_{\boldsymbol{F}}(\boldsymbol{T}^{(1)}_{T,:}, \boldsymbol{T}^{(2)}_{T,:})\end{bmatrix}$$

where

$$\boldsymbol{\mathcal{K}}_{\boldsymbol{F}}(\boldsymbol{T}^{(1)}_{i,:}, \boldsymbol{T}^{(2)}_{j,:}) = \begin{bmatrix}\mathcal{K}_{F_{1,1},F_{1,1}}(\boldsymbol{T}^{(1)}_{i,:}, \boldsymbol{T}^{(2)}_{j,:}) & \cdots & \mathcal{K}_{F_{1,1},F_{N,M}}(\boldsymbol{T}^{(1)}_{i,:}, \boldsymbol{T}^{(2)}_{j,:}) \\ \vdots & \ddots & \vdots \\ \mathcal{K}_{F_{N,M},F_{1,1}}(\boldsymbol{T}^{(1)}_{i,:}, \boldsymbol{T}^{(2)}_{j,:}) & \cdots & \mathcal{K}_{F_{N,M},F_{N,M}}(\boldsymbol{T}^{(1)}_{i,:}, \boldsymbol{T}^{(2)}_{j,:})\end{bmatrix}.$$

D | Gaussian Processes in Practice

D.1 Implementation of Gaussian Process Models

We briefly discuss some useful techniques concerning the implementation of Gaussian process models. Assume that all matrices have shape $N \times N$.

First, carefully considering the memory layout of large matrices can be beneficial. Namely, contiguous memory is accessed more cheaply than non-contiguous memory. This means that the cost of operations on large matrices greatly depends on the memory layout of the matrices.

Second, a covariance matrix $\boldsymbol{\Sigma}$ can be made positive semidefinite by adding a small diagonal. In theory, $\boldsymbol{\Sigma}$ is always positive semidefinite. However, when it is evaluated numerically, it can be indefinite or negative definite due round-off errors introduced by the floating-point representation of numbers. In that case $\boldsymbol{\Sigma}$'s positive semidefiniteness can be ensured by adding a diagonal comparable to $\varepsilon\|\boldsymbol{\Sigma}\|_\infty$ where ε is the machine epsilon. Usually the noise introduced by the diagonal is negligible.

In the case that $\boldsymbol{\Sigma}$ is indefinite or negative definite due to noise other than round-off errors, the diagonal required to make $\boldsymbol{\Sigma}$ positive semidefinite might be so large that the introduced noise becomes significant. In that case Theorem 1 can be used to compute the symmetric positive semidefinite matrix nearest to $\boldsymbol{\Sigma}$ in Frobenius norm.

Third, quantities of the form $\boldsymbol{A}\boldsymbol{\Sigma}^{-1}\boldsymbol{B}$ where $\boldsymbol{\Sigma}$ is a covariance matrix should not be computed by first computing $\boldsymbol{\Sigma}^{-1} = \boldsymbol{X}$ explicity. Namely, the error in $\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}$ due to round-off errors can be large, especially when $\boldsymbol{\Sigma}$ is ill-conditioned [Trefethen and Bau, 1997]. A numerically more stable approach is to first compute the Cholesky decomposi-

tion $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$ where $\boldsymbol{L}$ is lower triangular. Then

$$\boldsymbol{A}\boldsymbol{\Sigma}^{-1}\boldsymbol{B} = \boldsymbol{A}(\boldsymbol{L}\boldsymbol{L}^T)^{-1}\boldsymbol{B} = (\boldsymbol{A}\boldsymbol{L}^{-T})(\boldsymbol{L}^{-1}\boldsymbol{B})$$

where the systems $\boldsymbol{A}\boldsymbol{L}^{-T}$ and $\boldsymbol{L}^{-1}\boldsymbol{B}$ can be solved using respectively back substitution and forward substitution. The Cholesky decomposition has time complexity $\mathcal{O}(N^3)$ and back substitution and forward substitution have time complexity $\mathcal{O}(N^2)$ [Trefethen and Bau, 1997]. Thus $\boldsymbol{A}\boldsymbol{\Sigma}^{-1}\boldsymbol{B}$ can be computed in $\mathcal{O}(N^3)$ time.

Fourth, the determinant $\boldsymbol{\Sigma}$ of a covariance matrix can efficiently be computed via its Cholesky decomposition $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$. Specifically, as $\boldsymbol{L}$ is lower triangular,

$$|\boldsymbol{\Sigma}| = |\boldsymbol{L}|^2 = \prod_{i=1}^{N} L_{i,i}^2.$$

Thus the determinant can be computed in $\mathcal{O}(N^3)$ time.

D.2 Nearest Symmetric Positive-Semidefinite Matrix

Lemma 1 (Polar Decomposition of Symmetric Matrix)**.** *Let $\boldsymbol{A}$ be symmetrix and let $\boldsymbol{B} = \boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^T$ be its spectral decomposition. Then there exists an orthogonal matrix $\boldsymbol{U}$ such that*

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{Z}\boldsymbol{S}\boldsymbol{Z}^T$$

where $\boldsymbol{S} = \operatorname{diag}(|\Lambda_{1,1}|, \ldots, |\Lambda_{N,N}|)$.

Proof. Let $\boldsymbol{L}$ be diagonal such that $L_{i,i} = \operatorname{sign}\Lambda_{i,i}$. Then $\boldsymbol{L}\boldsymbol{L} = \boldsymbol{I}$. Therefore

$$\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^T = \boldsymbol{Z}\boldsymbol{L}\boldsymbol{L}\boldsymbol{\Lambda}\boldsymbol{Z}^T = \underbrace{\boldsymbol{Z}\boldsymbol{L}\boldsymbol{Z}^T}_{\boldsymbol{U}}\boldsymbol{Z}\underbrace{\boldsymbol{L}\boldsymbol{\Lambda}}_{\boldsymbol{S}}\boldsymbol{Z}^T$$

where $\boldsymbol{L}\boldsymbol{\Lambda} = \operatorname{diag}(|\Lambda_{1,1}|, \ldots, |\Lambda_{N,N}|)$ and

$$\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{Z}\boldsymbol{L}\boldsymbol{Z}^T\boldsymbol{Z}\boldsymbol{L}\boldsymbol{Z}^T = \boldsymbol{Z}\boldsymbol{L}\boldsymbol{L}\boldsymbol{Z}^T = \boldsymbol{Z}\boldsymbol{Z}^T = \boldsymbol{I}.$$

□

Lemma 2. *Let $\boldsymbol{A}$ be symmetric and let $\boldsymbol{B}$ be antisymmetric. Then $\|\boldsymbol{A}+\boldsymbol{B}\|_F^2 = \|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}\|_F^2$.*

Proof. To begin with, $\boldsymbol{A}^T\boldsymbol{B} = -\boldsymbol{A}\boldsymbol{B}^T$; thus $\operatorname{tr}(\boldsymbol{A}^T\boldsymbol{B}) + \operatorname{tr}(\boldsymbol{A}\boldsymbol{B}^T) = 0$. Therefore

$$\begin{aligned}\|\boldsymbol{A}+\boldsymbol{B}\|_F^2 &= \operatorname{tr}[(\boldsymbol{A}+\boldsymbol{B})^T(\boldsymbol{A}+\boldsymbol{B})] \\ &= \operatorname{tr}(\boldsymbol{A}^T\boldsymbol{A}) + \operatorname{tr}(\boldsymbol{A}^T\boldsymbol{B}) + \operatorname{tr}(\boldsymbol{B}^T\boldsymbol{A}) + \operatorname{tr}(\boldsymbol{B}^T\boldsymbol{B}) \\ &= \operatorname{tr}(\boldsymbol{A}^T\boldsymbol{A}) + \operatorname{tr}(\boldsymbol{B}^T\boldsymbol{B}) \\ &= \|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}\|_F^2.\end{aligned}$$

□

Lemma 3. *Let $\boldsymbol{A}$ be a matrix and let $\boldsymbol{U}$ be orthogonal. Then $\|\boldsymbol{U}\boldsymbol{A}\boldsymbol{U}^T\|_F^2 = \|\boldsymbol{A}\|_F^2$.*

Proof. Direct computation yields that

$$\begin{aligned}\|\boldsymbol{U}\boldsymbol{A}\boldsymbol{U}^T\|_F^2 &= \operatorname{tr}(\boldsymbol{U}\boldsymbol{A}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{A}^T\boldsymbol{U}^T) \\ &= \operatorname{tr}(\boldsymbol{U}\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{U}^T) \\ &= \operatorname{tr}(\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{A}\boldsymbol{A}^T) \\ &= \operatorname{tr}(\boldsymbol{A}\boldsymbol{A}^T) \\ &= \|\boldsymbol{A}\|_F^2.\end{aligned}$$

□

The following proof is based on the proof by Higham [1988].

Theorem 1 (Nearest Symmetric Positive-Semidefinite Matrix)**.** *Let $\boldsymbol{A}$ be a matrix and $\boldsymbol{B} = (\boldsymbol{A}+\boldsymbol{A}^T)/2$. Furthermore, let $\boldsymbol{B} = \boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^T$ be $\boldsymbol{B}$'s spectral decomposition and let $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{Z}\boldsymbol{S}\boldsymbol{Z}^T$ be $\boldsymbol{B}$'s polar decomposition (Lemma 1). Finally, let $\boldsymbol{A}_F = (\boldsymbol{B} + \boldsymbol{Z}\boldsymbol{S}\boldsymbol{Z}^T)/2$. Then*

1. *$\boldsymbol{A}_F$ is symmetric positive semidefinite and*
2. *for any other symmetric positive semidefinite matrix $\boldsymbol{X}$ it holds that $\|\boldsymbol{A}-\boldsymbol{X}\|_F^2 \geq \|\boldsymbol{A}-\boldsymbol{A}_F\|_F^2$.*

Proof. First, we verify that $\boldsymbol{A}_F$ is symmetric positive semidefinite. Trivially, $\boldsymbol{A}_F$ is symmetric. To see that $\boldsymbol{A}_F$ is positive semidefinite, consider

$$\boldsymbol{A}_F = \frac{1}{2}(\boldsymbol{B} + \boldsymbol{Z}\boldsymbol{S}\boldsymbol{Z}^T) = \frac{1}{2}(\boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^T + \boldsymbol{Z}\boldsymbol{S}\boldsymbol{Z}^T) = \boldsymbol{Z}\frac{\boldsymbol{\Lambda}+\boldsymbol{S}}{2}\boldsymbol{Z}^T. \tag{D.1}$$

By Lemma 1,

$$\frac{\boldsymbol{\Lambda}+\boldsymbol{S}}{2} = \operatorname{diag}\left(\frac{\Lambda_{1,1}+|\Lambda_{1,1}|}{2}, \ldots, \frac{\Lambda_{N,N}+|\Lambda_{N,N}|}{2}\right), \tag{D.2}$$

whose diagonal elements are all nonnegative. Therefore, for any $\boldsymbol{x}$, it holds that

$$\boldsymbol{x}^T\boldsymbol{A}_F\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{Z}\left(\frac{\boldsymbol{\Lambda}+\boldsymbol{S}}{2}\right)^{1/2}\left(\frac{\boldsymbol{\Lambda}+\boldsymbol{S}}{2}\right)^{1/2}\boldsymbol{Z}\boldsymbol{x}^T = \left\|\left(\frac{\boldsymbol{\Lambda}+\boldsymbol{S}}{2}\right)^{1/2}\boldsymbol{Z}^T\boldsymbol{x}\right\|_2^2 \geq 0.$$

Second, let $\boldsymbol{X}$ be symmetric positive semidefinite. We then show that $\|\boldsymbol{A}-\boldsymbol{X}\|_F^2 \geq \|\boldsymbol{A}-\boldsymbol{A}_F\|_F^2$. To begin with, let $\boldsymbol{C} = (\boldsymbol{A}-\boldsymbol{A}^T)/2$. Then $\boldsymbol{A} = \boldsymbol{B}+\boldsymbol{C}$ where $\boldsymbol{B}$ is symmetric and $\boldsymbol{C}$ is antisymmetric. Thus, by Lemma 2,

$$\|\boldsymbol{A}-\boldsymbol{X}\|_F^2 = \|(\boldsymbol{B}-\boldsymbol{X})+\boldsymbol{C}\|_F^2 = \|\boldsymbol{B}-\boldsymbol{X}\|_F^2 + \|\boldsymbol{C}\|_F^2 \geq \|\boldsymbol{B}-\boldsymbol{X}\|_F^2.$$

Let $\boldsymbol{Y} = \boldsymbol{Z}^T\boldsymbol{X}\boldsymbol{Z}$. As $\boldsymbol{X}$ is positive semidefinite, $\boldsymbol{Y}$ is also positive semidefinite and so $Y_{i,i} \geq 0$. Therefore, by Lemma 3,

$$\begin{aligned}
\|\boldsymbol{A}-\boldsymbol{X}\|_F^2 &\geq \|\boldsymbol{B}-\boldsymbol{X}\|_F^2 \\
&= \|\boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^T - \boldsymbol{X}\|_F^2 \\
&= \|\boldsymbol{Z}(\boldsymbol{\Lambda} - \boldsymbol{Z}^T\boldsymbol{X}\boldsymbol{Z})\boldsymbol{Z}^T\|_F^2 \\
&= \|\boldsymbol{\Lambda}-\boldsymbol{Y}\|_F^2 \\
&\geq \sum_{i:\Lambda_{i,i}<0} (\Lambda_{i,i}-Y_{i,i})^2 \\
&\geq \sum_{i:\Lambda_{i,i}<0} \Lambda_{i,i}^2.
\end{aligned}$$

This lower bound holds for any positive semidefinite $\boldsymbol{X}$. Therefore, if $\boldsymbol{A}_F$ achieves this lower bound, then the result follows.

To this end, let $\boldsymbol{X} = \boldsymbol{A}_F$. As $\boldsymbol{A}_F$ is symmetric, $\boldsymbol{C} = \boldsymbol{0}$ and so

$$\|\boldsymbol{A} - \boldsymbol{A}_F\|_F^2 = \|\boldsymbol{\Lambda} - \boldsymbol{Y}\|_F^2$$

where $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{A}_F\boldsymbol{Z}^T$. Now, Equation (D.1) and Equation (D.2) show that

$$\boldsymbol{Y} = \operatorname{diag}\left(\frac{\Lambda_{1,1} + |\Lambda_{1,1}|}{2}, \ldots, \frac{\Lambda_{N,N} + |\Lambda_{N,N}|}{2}\right).$$

Hence

$$\begin{aligned}
\|\boldsymbol{A} - \boldsymbol{A}_F\|_F^2 &= \|\boldsymbol{\Lambda} - \boldsymbol{Y}\|_F^2 \\
&= \sum_{i=1}^{N} \left(\Lambda_{i,i} - \frac{\Lambda_{i,i} + |\Lambda_{i,i}|}{2}\right)^2 \\
&= \sum_{i:\Lambda_{i,i}<0} \left(\Lambda_{i,i} - \frac{\Lambda_{i,i} + |\Lambda_{i,i}|}{2}\right)^2 \\
&= \sum_{i:\Lambda_{i,i}<0} \Lambda_{i,i}^2.
\end{aligned}$$

□

E | Circulant Approximation of Stationary Multi-Output Kernel Matrices

E.1 Introduction

Let $\boldsymbol{K} : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}^{N \times N}$ be a stationary multi-output kernel and let $\boldsymbol{K_F}$ be $\boldsymbol{K}$ evaluated for points that lie on an evenly-spaced grid $\{t_1^{(1)}, \ldots, t_{Y_1}^{(1)}\} \times \cdots \times \{t_1^{(K)}, \ldots, t_{Y_K}^{(K)}\}$. The total number of points is then given by $Y_1 \cdots Y_K = Y$. Assume that all Y_i are even.

Consider computation of $|\boldsymbol{K_F}|$ and a product of the form $\boldsymbol{P} = \boldsymbol{U}^T \boldsymbol{K_F}^{-1} \boldsymbol{V}$ for some $NY \times P$ matrix $\boldsymbol{U}$ and $NY \times Q$ matrix $\boldsymbol{V}$. Computing $|\boldsymbol{K_F}|$ directly costs $\mathcal{O}(N^3 Y^3)$ time (Appendix D). This does not scale for large Y. Furthermore, inverting $\boldsymbol{K_F}$ costs $\mathcal{O}(N^3 Y^3)$ time and then computing $\boldsymbol{U}^T \boldsymbol{K_F}^{-1} \boldsymbol{V}$ costs $\mathcal{O}(PQN^2 Y^2)$ time, resulting in time complexity $\mathcal{O}(PQN^2 Y^2 + N^3 Y^3)$. This also does not scale for large Y.

This chapter develops an approximation of $\boldsymbol{K_F}$ based on the facts that $\boldsymbol{K_F}$ is stationary and is evaluated for points on a grid [Ulrich et al., 2015]. We show that this approximation can be leveraged to compute $|\boldsymbol{K_F}|$ and $\boldsymbol{U}^T \boldsymbol{K_F}^{-1} \boldsymbol{V}$ with complexities that scale favourably in Y.

E.2 Circulant Approximation of Toeplitz Matrices

Let $k : \mathbb{R} \to \mathbb{R}$ be a stationary kernel and let $\boldsymbol{K_f}$ be k evaluated for points that lie on an evenly-spaced grid $\{t_1, \ldots, t_Y\}$. Then, by k's stationary and the fact that the points

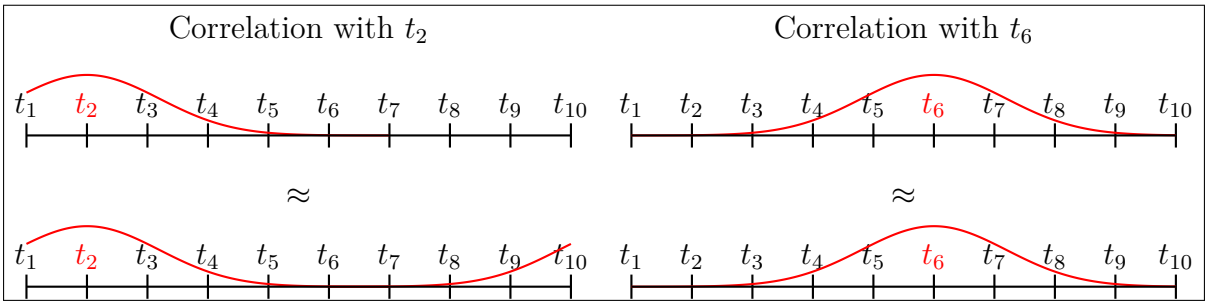


Figure E.1: Circulant approximation of a stationary kernel. Shows how the space "wraps around" at the edge of the grid. Also shows that the approximation is excellent for points not in the vicinity of the edge.

lie on a grid,

$$\boldsymbol{K}_{\boldsymbol{f}} = \text{stoep}[k(t_1 - t_1), \ldots, k(t_1 - t_Y)].$$

A banded Toeplitz matrix can be approximated by a particular circulant matrix where the approximation becomes more accurate as the matrix becomes large [Gray, 2006]. More specifically, if k has effectively limited support, then we can approximate

$$\begin{aligned}\boldsymbol{K}_{\boldsymbol{f}} &= \text{stoep}[k(t_1 - t_1), \ldots, k(t_1 - t_{Y/2}), k(t_1 - t_{Y/2+1}), k(t_1 - t_{Y/2+2}), \ldots, k(t_1 - t_Y)] \\ &\approx \text{stoep}[k(t_1 - t_1), \ldots, k(t_1 - t_{Y/2}), k(t_1 - t_{Y/2+1}), k(t_1 - t_{Y/2}), \ldots, k(t_1 - t_2)] \\ &= \text{circ}[k(t_1 - t_1), \ldots, k(t_1 - t_{Y/2}), k(t_1 - t_{Y/2+1}), k(t_1 - t_{Y/2}), \ldots, k(t_1 - t_2)]. \quad \text{(E.1)}\end{aligned}$$

We call this approximation the *circulant approximation* of $\boldsymbol{K}_{\boldsymbol{f}}$. Essentially, the approximation lets the space in which distance is measured now "wrap around" at the edge of the grid. As a consequence, points near the edge of the grid correlate. However, if the kernel is local, then correlations between points not in the vicinity of the edge are preserved; in that case $\boldsymbol{K}_{\boldsymbol{f}}$'s circulant approximation is an excellent approximation for most points. Figure E.1 illustrates the circulant approximation of a stationary kernel.

Finally, the $Y \times Y$ unitary discrete Fourier transform matrix $\boldsymbol{\mathcal{F}}_Y$ diagonalises any $Y \times Y$ circulant matrix [Gray, 2006]. Thus, by $\boldsymbol{K}_{\boldsymbol{f}} = \boldsymbol{K}_{\boldsymbol{f}}^T$, it holds that $\boldsymbol{K}_{\boldsymbol{f}} \approx \boldsymbol{\mathcal{F}}_Y \boldsymbol{\Lambda} \boldsymbol{\mathcal{F}}_Y^H = \boldsymbol{\mathcal{F}}_Y^H \boldsymbol{\Lambda} \boldsymbol{\mathcal{F}}_Y$ where $\boldsymbol{\Lambda}$ is diagonal.

E.3 Circulant Approximation of Stationary Multi-Output Kernel Matrices

In the same way that a Toeplitz matrix has a circulant approximation, a block Toeplitz matrix has a block circulant approximation. We use this observation to approximate $\boldsymbol{K_F}$.

We construct $\boldsymbol{K_F}$ in a particular way. Specifically, let $\boldsymbol{K}^{(0)}(\boldsymbol{t}, \boldsymbol{t}') = \boldsymbol{K}(\boldsymbol{t}, \boldsymbol{t}')$ and let

$$\begin{aligned}
\boldsymbol{t}^{(>i)} &= \begin{bmatrix} t_{i+1} & \cdots & t_K \end{bmatrix}^T, \\
\boldsymbol{t}^{(>i)'} &= \begin{bmatrix} t'_{i+1} & \cdots & t'_K \end{bmatrix}^T, \\
\boldsymbol{K}^{(i)}(\boldsymbol{t}^{(>i)}, \boldsymbol{t}^{(>i)'}) &= \begin{bmatrix} \boldsymbol{K}^{(i-1)}(t_1^{(i)}, \boldsymbol{t}^{(>i)}, t_1^{(i)}, \boldsymbol{t}^{(>i)'}) & \cdots & \boldsymbol{K}^{(i-1)}(t_1^{(i)}, \boldsymbol{t}^{(>i)}, t_{Y_i}^{(i)}, \boldsymbol{t}^{(>i)'}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{K}^{(i-1)}(t_{Y_i}^{(i)}, \boldsymbol{t}^{(>i)}, t_1^{(i)}, \boldsymbol{t}^{(>i)'}) & \cdots & \boldsymbol{K}^{(i-1)}(t_{Y_i}^{(i)}, \boldsymbol{t}^{(>i)}, t_{Y_i}^{(i)}, \boldsymbol{t}^{(>i)'}) \end{bmatrix}
\end{aligned}$$

for $i = 1, \ldots, K$. Then $\boldsymbol{K_F} = \boldsymbol{K}^{(K)}$. Now, $\boldsymbol{K}$ is stationary and the points for which we evaluate $\boldsymbol{K}$ lie on a grid; thus we can use Equation (E.1) to approximate

$$\begin{aligned}
&\boldsymbol{K}^{(i)}(\boldsymbol{t}^{(>i)}, \boldsymbol{t}^{(>i)'}) \\
&\quad = \text{stoep}[\boldsymbol{K}^{(i-1)}(t_1^{(i)}, \boldsymbol{t}^{(>i)}, t_1^{(i)}, \boldsymbol{t}^{(>i)'}), \ldots, \boldsymbol{K}^{(i-1)}(t_1^{(i)}, \boldsymbol{t}^{(>i)}, t_{Y_i}^{(i)}, \boldsymbol{t}^{(>i)'})] \\
&\quad \approx \text{circ}[\boldsymbol{K}^{(i-1)}(t_1^{(i)}, \boldsymbol{t}^{(>i)}, t_{w^{(i)}(1)}^{(i)}, \boldsymbol{t}^{(>i)'}), \ldots, \boldsymbol{K}^{(i-1)}(t_1^{(i)}, \boldsymbol{t}^{(>i)}, t_{w^{(i)}(Y_i)}^{(i)}, \boldsymbol{t}^{(>i)'})]
\end{aligned}$$

where $w^{(i)}(j_i) = \min\{j, Y_i - j_i + 2\}$. Then, by decomposing each $\boldsymbol{K}^{(i)}(\boldsymbol{t}^{(>i)}, \boldsymbol{t}^{(>i)'})$ as

$$\boldsymbol{K}^{(i)}(\boldsymbol{t}^{(>i)}, \boldsymbol{t}^{(>i)'}) \approx \sum_{j_i=1}^{Y_i} \overbrace{\text{circ}(\underbrace{0, \ldots, 0}_{j_i - 1 \text{ times}}, 1, \underbrace{0, \ldots, 0}_{Y_i - j_i \text{ times}})}^{\boldsymbol{C}^{(i,j_i)}} \otimes \boldsymbol{K}^{(i-1)}(t_1^{(i)}, \boldsymbol{t}^{(>i)}, t_{w^{(i)}(j_i)}^{(i)}, \boldsymbol{t}^{(>i)'}),$$

we obtain that

$$\begin{aligned}
\boldsymbol{K}^{(K)} &\approx \sum_{j_K=1}^{Y_K} \boldsymbol{C}^{(K,j_K)} \otimes \boldsymbol{K}^{(K-1)}(t_1^{(K)}, t_{w^{(K)}(j_K)}^{(K)}) \\
&\approx \sum_{j_K=1, j_{K-1}=1}^{Y_K, Y_{K-1}} \boldsymbol{C}^{(K,j_K)} \otimes \boldsymbol{C}^{(K-1,j_{K-1})} \otimes \boldsymbol{K}^{(K-1)}(t_1^{(K-1)}, t_1^{(K)}, t_{w^{(K-1)}(j_{K-1})}^{(K-1)}, t_{w^{(K)}(j_K)}^{(K)}) \\
&\vdots \\
&\approx \sum_{j_K=1,\ldots,j_1=1}^{Y_K,\ldots,Y_1} \boldsymbol{C}^{(K,j_K)} \otimes \cdots \otimes \boldsymbol{C}^{(1,j_1)} \otimes \boldsymbol{K}^{(0)}(\underbrace{t_1^{(1)}, \ldots, t_1^{(K)}}_{\boldsymbol{t}_1}, \underbrace{t_{w^{(1)}(j_1)}^{(1)}, \ldots, t_{w^{(K)}(j_K)}^{(K)}}_{\boldsymbol{t}_{w(\boldsymbol{j})}}) \\
&= \sum_{j_K=1,\ldots,j_1=1}^{Y_K,\ldots,Y_1} \boldsymbol{C}^{(K,j_K)} \otimes \cdots \otimes \boldsymbol{C}^{(1,j_1)} \otimes \boldsymbol{K}(\boldsymbol{t}_1, \boldsymbol{t}_{w(\boldsymbol{j})}).
\end{aligned}$$

Let $\boldsymbol{\mathcal{F}}^{(i)} = \boldsymbol{\mathcal{F}}_{Y_i} \otimes \cdots \otimes \boldsymbol{\mathcal{F}}_{Y_1} \otimes \boldsymbol{I}$. Then

$$\begin{aligned}
&\boldsymbol{\mathcal{F}}^{(K)} \boldsymbol{K}^{(K)} \boldsymbol{\mathcal{F}}^{(K)H} \\
&\quad \approx \sum_{j_K=1,\ldots,j_1=1}^{Y_K,\ldots,Y_1} (\underbrace{\boldsymbol{\mathcal{F}}_{Y_K} \boldsymbol{C}^{(K,j_K)} \boldsymbol{\mathcal{F}}_{Y_K}^H}_{\boldsymbol{\Lambda}^{(K,j_k)}}) \otimes \cdots \otimes (\underbrace{\boldsymbol{\mathcal{F}}_{Y_1} \boldsymbol{C}^{(1,j_1)} \boldsymbol{\mathcal{F}}_{Y_1}^H}_{\boldsymbol{\Lambda}^{(1,j_k)}}) \otimes \boldsymbol{K}(\boldsymbol{t}_1, \boldsymbol{t}_{w(\boldsymbol{j})}) \qquad \text{(E.2)}
\end{aligned}$$

where all $\boldsymbol{\Lambda}^{(i,j)}$ are diagonal because all $\boldsymbol{C}^{(i,j)}$ are circulant by definition. Thus $\boldsymbol{\mathcal{F}}^{(K)} \boldsymbol{K}^{(K)} \boldsymbol{\mathcal{F}}^{(K)H}$ is approximately block diagonal. Let

$$\boldsymbol{B}_{m_1,\ldots,m_K} = \sum_{j_K=1,\ldots,j_1=1}^{Y_K,\ldots,Y_1} \Lambda_{m_K,m_K}^{(K,j_k)} \cdots \Lambda_{m_1,m_1}^{(1,j_1)} \boldsymbol{K}(\boldsymbol{t}_1, \boldsymbol{t}_{w(\boldsymbol{j})}). \qquad \text{(E.3)}$$

Then Equation (E.2) tells us that $\boldsymbol{B}_{m_1,\ldots,m_K}$ is the $d(m_1,\ldots,m_K)$'th block on the diagonal of $\boldsymbol{\mathcal{F}}^{(K)} \boldsymbol{K}^{(K)} \boldsymbol{\mathcal{F}}^{(K)H}$ where

$$d(m_1,\ldots,m_K) = (m_K - 1)(Y_{K-1} \cdots Y_1) + \ldots + (m_2 - 1)Y_1 + (m_1 - 1) + 1.$$

Finally, let $m^{(i)}(d)$ be such that $m^{(i)}[d(m_1, \ldots, m_K)] = m_i$.[1] Then $\boldsymbol{K_F}$ is approximated by

$$\boldsymbol{K_F} \approx \boldsymbol{\mathcal{F}}^{(K)H} \operatorname{diag}(\boldsymbol{B}_{m^{(1)}(1),\ldots,m^{(K)}(1)}, \ldots, \boldsymbol{B}_{m^{(1)}(Y),\ldots,m^{(K)}(Y)}) \boldsymbol{\mathcal{F}}^{(K)}. \tag{E.4}$$

E.4 Approximating Determinants

We can use Equation (E.4) to efficiently approximate $|\boldsymbol{K_F}|$. We first derive an approximation of $|\boldsymbol{K_F}|$ and then show that this approximation can be computed efficiently.

Note that $\boldsymbol{\mathcal{F}}^{(K)H}\boldsymbol{\mathcal{F}}^{(K)} = \boldsymbol{I}$. Thus, by Equation (E.4),

$$\begin{aligned} |\boldsymbol{K_F}| &\approx |\boldsymbol{\mathcal{F}}^{(K)H}|\,|\operatorname{diag}(\boldsymbol{B}_{m^{(1)}(1),\ldots,m^{(K)}(1)}, \ldots, \boldsymbol{B}_{m^{(1)}(Y),\ldots,m^{(K)}(Y)})|\,|\boldsymbol{\mathcal{F}}^{(K)}| \\ &= \prod_{d=1}^{Y} |\boldsymbol{B}_{m^{(1)}(d),\ldots,m^{(K)}(d)}|. \end{aligned} \tag{E.5}$$

We now show that Equation (E.5) can be computed efficiently. To begin with, we show that all $\boldsymbol{B}_{m_1,\ldots,m_K}$ can be computed efficiently. Let $\boldsymbol{S}^{(0)}(\boldsymbol{t}) = \boldsymbol{K}(\boldsymbol{t}_1, \boldsymbol{t})$ and let

$$\begin{aligned} \boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})} &= \begin{bmatrix} t^{(i+1)}_{w^{(i+1)}(j_{i+1})} & \cdots & t^{(K)}_{w^{(K)}(j_K)} \end{bmatrix}^T, \\ \boldsymbol{S}^{(i)}_{m_1,\ldots,m_i}(\boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})}) &= \sum_{j_i=1}^{Y_i} \Lambda^{(i,j_i)}_{m_i,m_i} \boldsymbol{S}^{(i-1)}_{m_1,\ldots,m_{i-1}}(t^{(i)}_{w^{(i)}(j_i)}, \boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})}) \end{aligned} \tag{E.6}$$

for $i = 1, \ldots, K$. Then one verifies that $\boldsymbol{S}^{(K)}_{m_1,\ldots,m_K} = \boldsymbol{B}_{m_1,\ldots,m_K}$. Now, fix i and fix

[1] That is, let $m^{(i)}(d) = 1 + [\lfloor (d-1)/(Y_{i-1} \cdots Y_i) \rfloor \bmod Y_i]$. Then

$$\begin{aligned} m^{(i)}(d)\Big|_{d=d(m_1,\ldots,m_K)} &= 1 + \left(\left\lfloor \frac{d-1}{Y_{i-1}\cdots Y_1} \right\rfloor \bmod Y_i \right)\Bigg|_{d=d(m_1,\ldots,m_K)} \\ &= 1 + \Bigg(\Bigg\lfloor (m_K - 1)(Y_K \cdots Y_i) + \ldots + (m_{i+1} - 1)Y_i \\ &\qquad + (m_i - 1) + \cancel{\frac{m_{i-1} - 1}{Y_{i-1}}} + \ldots + \cancel{\frac{m_1 - 1}{Y_{i-1}\cdots Y_1}} \Bigg\rfloor \bmod Y_i \Bigg) \\ &= 1 + \Big[(m_K - 1)(Y_K \cdots Y_i) + \ldots + (m_{i+1} - 1)Y_i + m_i - 1 \Big] \bmod Y_i \\ &= m_i. \end{aligned}$$

$m_1, \ldots, m_{i-1}$ and $j_{i+1}, \ldots, j_K$. It holds that [Gray, 2006]

$$\begin{aligned}\Lambda^{(i,j_i)}_{m_i,m_i} &= (\boldsymbol{\mathcal{F}}_{Y_i} \boldsymbol{C}^{(i,j_i)} \boldsymbol{\mathcal{F}}^H_{Y_i})_{m_i,m_i} \\ &= \sum_{n=1}^{Y_i} \mathbb{1}(j_i - n) \exp\left[-2\pi\sqrt{-1}\frac{(m_i-1)(n-1)}{Y_i}\right] \\ &= \exp\left[-2\pi\sqrt{-1}\frac{(m_i-1)(j_i-1)}{Y_i}\right].\end{aligned}$$

Hence, by Equation (E.6),

$$\underbrace{\boldsymbol{S}^{(i)}_{m_1,\ldots,m_i}(\boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})})}_{\boldsymbol{A}_{m_i}} = \sum_{j_i=1}^{Y_i} \underbrace{\boldsymbol{S}^{(i-1)}_{m_1,\ldots,m_{i-1}}(t^{(i)}_{w^{(i)}(j_i)}, \boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})})}_{\boldsymbol{B}_{j_i}} \exp\left[-2\pi\sqrt{-1}\frac{(m_i-1)(j_i-1)}{Y_i}\right].$$

Note that all $\boldsymbol{A}_{m_i}$ and $\boldsymbol{B}_{j_i}$ are $N \times N$ matrices. Importantly, observe that $\boldsymbol{A}_{:,n,n'} = \mathrm{DFT}\, \boldsymbol{B}_{:,n,n'}$. Thus, the fast Fourier transform algorithm can be used to compute a single $\boldsymbol{A}_{:,n,n'}$ in $\mathcal{O}(Y_i \log Y_i)$ time. By doing this for every (n, n') we can compute $\boldsymbol{S}^{(i)}_{m_1,\ldots,m_i}(\boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})})$ for all m_i in $\mathcal{O}(N^2 Y_i \log Y_i)$ time. Hence, we can compute $\boldsymbol{S}^{(i)}_{m_1,\ldots,m_i}(\boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})})$ for all $m_1, \ldots, m_i$ and $j_{i+1}, \ldots, j_K$ in

$$\mathcal{O}(Y_1 \cdots Y_{i-1} Y_{i+1} \cdots Y_K N^2 Y_i \log Y_i) = \mathcal{O}(N^2 Y \log Y_i)$$

time, which shows that recursively applying Equation (E.6) to compute $\boldsymbol{S}^{(K)}_{m_1,\ldots,m_K} = \boldsymbol{B}_{m_1,\ldots,m_K}$ for all $m_1, \ldots, m_K$ costs

$$\mathcal{O}[N^2 Y(\log Y_1 + \ldots + \log Y_K)]$$

time. This time complexity is bounded by $\mathcal{O}(N^2 Y \log Y)$.

Finally, Equation (E.5) shows that after all $\boldsymbol{B}_{m_1,\ldots,m_K}$ have been computed, $|\boldsymbol{K}_{\boldsymbol{F}}|$ can be computed in $\mathcal{O}(N^3 Y)$ time (Appendix D). Therefore, the resulting complexity of computing $|\boldsymbol{K}_{\boldsymbol{F}}|$ is $\mathcal{O}(N^2 Y \log Y + N^3 Y)$. The complete procedure to compute $|\boldsymbol{K}_{\boldsymbol{F}}|$ is outlined in Algorithm 1.

Algorithm 1 Efficient approximation of the determinant of a multi-output stationary kernel matrix. Runs in $\mathcal{O}(N^2 Y \log Y + N^3 Y)$ time.

function DETERMINANT($\boldsymbol{K}$, $\{t_1^{(1)}, \ldots, t_{Y_1}^{(1)}\} \times \cdots \times \{t_1^{(K)}, \ldots, t_{Y_K}^{(K)}\}$)
  for $i = 1, \ldots, K$ **do**
    for $m_1 = 1, \ldots, Y_1$ to $m_{i-1} = 1, \ldots, Y_{i-1}$ and $j_{i+1} = 1, \ldots, Y_{i+1}$ to $j_K = 1, \ldots, Y_K$ **do**
      for $n = 1, \ldots, N$ and $n' = 1, \ldots, N$ **do**
        $\boldsymbol{A}_{:,n,n'} \leftarrow \mathrm{DFT}\, \boldsymbol{B}_{:,n,n'}$
          ▷ Via fast Fourier transform algorithm
          ▷ $\boldsymbol{S}^{(0)}(\boldsymbol{t}) = \boldsymbol{K}(\boldsymbol{t}_1, \boldsymbol{t})$, $\boldsymbol{A}_{m_i} = \boldsymbol{S}^{(i)}_{m_1,\ldots,m_i}(\boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})})$ and $\boldsymbol{B}_{j_i} = \boldsymbol{S}^{(i-1)}_{m_1,\ldots,m_{i-1}}(t^{(i)}_{w^{(i)}(j_i)}, \boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})})$ where $w^{(i)}(j_i) = \min\{j, Y_i - j_i + 2\}$
  for $m_1 = 1, \ldots, Y_1$ to $m_K = 1, \ldots, Y_K$ **do**
    Compute $|\boldsymbol{S}^{(K)}_{m_1,\ldots,m_K}|$
  return $\prod_{d=1}^{Y} |\boldsymbol{S}^{(K)}_{m^{(1)}(d),\ldots,m^{(K)}(d)}|$ ▷ $m^{(i)}(d) = 1 + [\lfloor (d-1)/(Y_{i-1} \cdots Y_i) \rfloor \bmod Y_i]$

E.5 Approximating Products Involving an Inverse

We can also use Equation (E.4) to efficiently approximate a product of the form $\boldsymbol{P} = \boldsymbol{U}^T \boldsymbol{K}_{\boldsymbol{F}}^{-1} \boldsymbol{V}$ for some $NY \times P$ matrix $\boldsymbol{U}$ and $NY \times Q$ matrix $\boldsymbol{V}$. We first derive an approximation of $\boldsymbol{P}$ and then show that this approximation can be computed efficiently.

Consider $P_{p,q}$. Recall that by Equation (E.4), $\boldsymbol{\mathcal{F}}^{(K)} \boldsymbol{K}^{(K)} \boldsymbol{\mathcal{F}}^{(K)H}$ is approximately block diagonal with blocks $\boldsymbol{B}_{m_1,\ldots,m_K}$; hence $\boldsymbol{\mathcal{F}}^{(K)} \boldsymbol{K}^{-(K)} \boldsymbol{\mathcal{F}}^{(K)H}$ is approximately block diagonal with blocks $\boldsymbol{B}^{-1}_{m_1,\ldots,m_K}$. Thus

$$
\begin{aligned}
P_{p,q} &= \underbrace{(\boldsymbol{U}_{:,p})^T}_{\boldsymbol{u}^T} \boldsymbol{K}^{-(K)} \underbrace{\boldsymbol{V}_{:,q}}_{\boldsymbol{v}} \\
&= (\boldsymbol{\mathcal{F}}^{(K)} \boldsymbol{u})^H (\boldsymbol{\mathcal{F}}^{(K)} \boldsymbol{K}^{-(K)} \boldsymbol{\mathcal{F}}^{(K)H}) (\boldsymbol{\mathcal{F}}^{(K)} \boldsymbol{v}) \\
&\approx \sum_{d=1}^{Y} \begin{bmatrix} u_{(d-1)N+1} & \cdots & u_{dN} \end{bmatrix} \boldsymbol{B}^{-1}_{m^{(1)}(d),\ldots,m^{(K)}(d)} \begin{bmatrix} v_{(d-1)N+1} & \cdots & v_{dN} \end{bmatrix}^T . \qquad \text{(E.7)}
\end{aligned}
$$

We now show that Equation (E.7) can be computed efficiently. First, we show that all $\boldsymbol{B}^{-1}_{m_1,\ldots,m_K}$ can be computed efficiently. Appendix E.4 showed that all $\boldsymbol{B}_{m_1,\ldots,m_K}$ can be computed in $\mathcal{O}(N^2 Y \log Y)$ time. Therefore all $\boldsymbol{B}^{-1}_{m_1,\ldots,m_K}$ can be computed in $\mathcal{O}(N^2 Y \log Y + N^3 Y)$ time.

Second, we show that $\boldsymbol{\mathcal{F}}^{(K)}\boldsymbol{u}$ and thereby $\boldsymbol{\mathcal{F}}^{(K)}\boldsymbol{v}$ from Equation (E.7) can be computed efficiently. It holds that

$$\begin{aligned}\boldsymbol{\mathcal{F}}^{(K)}\boldsymbol{u} &= (\boldsymbol{\mathcal{F}}_{Y_K} \otimes \cdots \otimes \boldsymbol{\mathcal{F}}_{Y_1} \otimes \boldsymbol{I})\boldsymbol{u} \\ &= \underbrace{(\boldsymbol{\mathcal{F}}_{Y_K} \otimes \boldsymbol{I}_{Y_{K-1}} \otimes \cdots \otimes \boldsymbol{I}_{Y_1} \otimes \boldsymbol{I}_N)}_{\boldsymbol{T}_{Y_K}} \cdots \underbrace{(\boldsymbol{I}_{Y_K} \otimes \cdots \otimes \boldsymbol{I}_{Y_2} \otimes \boldsymbol{\mathcal{F}}_{Y_1} \otimes \boldsymbol{I}_N)}_{\boldsymbol{T}_{Y_1}} \boldsymbol{u}.\end{aligned}$$

Observe that each $\boldsymbol{T}_{Y_i}$ is of the form $\boldsymbol{I}_{Z_i^{(1)}} \otimes \boldsymbol{\mathcal{F}}_{Y_i} \otimes \boldsymbol{I}_{Z_i^{(2)}}$ where $Z_i^{(1)} = Y_K \cdots Y_{i+1}$ and $Z_i^{(2)} = Y_{i-1} \cdots Y_1 N$. Inspection of

$$(\boldsymbol{\mathcal{F}}_{Y_i} \otimes \boldsymbol{I}_{Z_i^{(2)}})\boldsymbol{x} = \begin{bmatrix} \mathcal{F}_{Y_i,1,1}\boldsymbol{I}_{Z_i^{(2)}} & \cdots & \mathcal{F}_{Y_i,1,Y_i}\boldsymbol{I}_{Z_i^{(2)}} \\ \vdots & \ddots & \vdots \\ \mathcal{F}_{Y_i,Y_i,1}\boldsymbol{I}_{Z_i^{(2)}} & \cdots & \mathcal{F}_{Y_i,Y_i,Y_i}\boldsymbol{I}_{Z_i^{(2)}} \end{bmatrix} \boldsymbol{x} = \boldsymbol{y}$$

shows that

$$\boldsymbol{\mathcal{F}}_{Y_i} \underbrace{\begin{bmatrix} x_z & x_{z+Z_i^{(2)}} & \cdots & x_{z+(Y_i-1)Z_i^{(2)}} \end{bmatrix}^T}_{\text{every } Z_i^{(2)}\text{'th element of } \boldsymbol{x}\text{, starting at } z} = \underbrace{\begin{bmatrix} y_z & y_{z+Z_i^{(2)}} & \cdots & y_{z+(Y_i-1)Z_i^{(2)}} \end{bmatrix}^T}_{\text{every } Z_i^{(2)}\text{'th element of } \boldsymbol{y}\text{, starting at } z} \tag{E.8}$$

for $z = 1, \ldots, Z_i^{(2)}$, which completely specifies $\boldsymbol{y}$. The fast Fourier transform algorithm can be used to compute Equation (E.8) in $\mathcal{O}(Y_i \log Y_i)$ time, which means that $\boldsymbol{y}$ can be computed in $\mathcal{O}(Z_i^{(2)} Y_i \log Y_i)$ time. Similarly, by directly applying the definition of the Kronecker product we see that $(\boldsymbol{I}_{Z_i^{(1)}} \otimes \boldsymbol{\mathcal{F}}_{Y_i} \otimes \boldsymbol{I}_{Z_i^{(2)}})\boldsymbol{x} = \boldsymbol{y}$ yields the $\boldsymbol{y}$ such that

$$\begin{aligned}(\boldsymbol{\mathcal{F}}_{Y_i} \otimes \boldsymbol{I}_{Z_i^{(2)}}) &\underbrace{\begin{bmatrix} x_{(z-1)Z_i^{(2)}Y_i+1} & x_{(z-1)Z_i^{(2)}Y_i+2} & \cdots & x_{zZ_i^{(2)}Y_i} \end{bmatrix}^T}_{z\text{'th consecutive group of } Z_i^{(2)}Y_i \text{ consecutive elements of } x} \\ &= \underbrace{\begin{bmatrix} y_{(z-1)Z_i^{(2)}Y_i+1} & y_{(z-1)Z_i^{(2)}Y_i+2} & \cdots & y_{zZ_i^{(2)}Y_i} \end{bmatrix}^T}_{z\text{'th consecutive group of } Z_i^{(2)}Y_i \text{ consecutive elements of } y}\end{aligned}$$

for $z = 1, \ldots, Z_{Y_i}^{(1)}$, which again completely specifies $\boldsymbol{y}$. Hence multiplication by any $\boldsymbol{T}_{Y_i}$ can be done in $\mathcal{O}(Z_{Y_i}^{(1)} Z_{Y_i}^{(2)} Y_i \log Y_i) = \mathcal{O}(NY \log Y_i)$ time. Thus $\boldsymbol{\mathcal{F}}^{(K)}\boldsymbol{u}$ and $\boldsymbol{\mathcal{F}}^{(K)}\boldsymbol{v}$ can be computed in

$$\mathcal{O}[NY(\log Y_1 + \ldots + \log Y_K)].$$

time. This complexity is bounded by $\mathcal{O}(NY \log Y)$.

Finally, Equation (E.7) shows that after all $\boldsymbol{B}^{-1}_{m_1,\dots,m_K}$, $\boldsymbol{\mathcal{F}}^{(K)}\boldsymbol{u}$ and $\boldsymbol{\mathcal{F}}^{(K)}\boldsymbol{v}$ have been computed, $P_{p,q}$ can be computed in $\mathcal{O}(N^2 Y)$ time. Therefore, since all $\boldsymbol{B}^{-1}_{m_1,\dots,m_K}$ have to be computed only once, the resulting complexity of computing $\boldsymbol{P}$ is $\mathcal{O}[N^2 Y \log Y + N^3 Y + PQ(NY \log Y + N^2 Y)]$. The complete procedure to compute $\boldsymbol{P}$ is outlined in Algorithm 2.

Algorithm 2 Efficient approximation of a product involving an inverse stationary multi-output kernel matrix. Runs in $\mathcal{O}[N^2 Y \log Y + N^3 Y + PQ(NY \log Y + N^2 Y)]$ time.

function PRODUCT($\boldsymbol{K}$, $\{t_1^{(1)}, \dots, t_{Y_1}^{(1)}\} \times \cdots \times \{t_1^{(K)}, \dots, t_{Y_K}^{(K)}\}$, $\boldsymbol{U}$, $\boldsymbol{V}$)
  for $i = 1, \dots, K$ **do**
    for $m_1 = 1, \dots, Y_1$ to $m_{i-1} = 1, \dots, Y_{i-1}$ and $j_{i+1} = 1, \dots, Y_{i+1}$ to $j_K = 1, \dots, Y_K$ **do**
      for $n = 1, \dots, N$ and $n' = 1, \dots, N$ **do**
        $\boldsymbol{A}_{:,n,n'} \leftarrow \mathrm{DFT}\, \boldsymbol{B}_{:,n,n'}$
          ▷ Via fast Fourier transform algorithm
          ▷ $\boldsymbol{S}^{(0)}(\boldsymbol{t}) = \boldsymbol{K}(\boldsymbol{t}_1, \boldsymbol{t})$, $\boldsymbol{A}_{m_i} = \boldsymbol{S}^{(i)}_{m_1,\dots,m_i}(\boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})})$ and $\boldsymbol{B}_{j_i} = \boldsymbol{S}^{(i-1)}_{m_1,\dots,m_{i-1}}(t^{(i)}_{w^{(i)}(j_i)}, \boldsymbol{t}^{(>i)}_{w(\boldsymbol{j})})$ where $w^{(i)}(j_i) = \min\{j, Y_i - j_i + 2\}$
  for $m_1 = 1, \dots, Y_1$ to $m_K = 1, \dots, Y_K$ **do**
    Compute $\boldsymbol{S}^{-(K)}_{m_1,\dots,m_K}$
  for $p = 1, \dots, P$ and $q = 1, \dots, Q$ **do**
    $\boldsymbol{U}_{:,p} \leftarrow$ TRANSFORM $\boldsymbol{U}_{:,p}$
    $\boldsymbol{U}_{:,q} \leftarrow$ TRANSFORM $\boldsymbol{U}_{:,q}$
    $P_{p,q} \leftarrow \sum_{d=1}^{Y} \begin{bmatrix} U_{(d-1)N+1,p} & \cdots & U_{dN,p} \end{bmatrix} \boldsymbol{S}^{-(K)}_{m^{(1)}(d),\dots,m^{(K)}(d)} \begin{bmatrix} V_{(d-1)N+1,q} & \cdots & V_{dN,q} \end{bmatrix}^T$
      ▷ $m^{(i)}(d) = 1 + [\lfloor (d-1)/(Y_{i-1} \cdots Y_i) \rfloor \bmod Y_i]$
  return $\boldsymbol{P}$
function TRANSFORM($\boldsymbol{x}$)
  for $i = 1, \dots, K$ **do**
    for $z_1 = 1, \dots, Z_1$ **do** ▷ $Z_1 = Y_K \cdots Y_{i+1}$
      for $z_2 = 1, \dots, Z_2$ **do** ▷ $Z_2 = Y_{i-1} \cdots Y_1 N$
        $\begin{bmatrix} y_{z_2} & y_{z_2+Z_2} & \cdots & y_{z_2+(Y_i-1)Z_2} \end{bmatrix}^T$
          $\leftarrow Y_i^{-1/2}\, \mathrm{DFT} \begin{bmatrix} y_{z_2} & y_{z_2+Z_2} & \cdots & y_{z_2+(Y_i-1)Z_2} \end{bmatrix}^T$
          ▷ Via fast Fourier transform algorithm
          ▷ $\boldsymbol{y} = \begin{bmatrix} x_{(z_1-1)Z_2Y_i+1} & x_{(z_1-1)Z_2Y_i+2} & \cdots & x_{z_1 Z_2 Y_i} \end{bmatrix}^T$
  return $\boldsymbol{x}$

E.6 Conclusion

We have derived procedures to efficiently approximate the determinant of a stationary multi-output kernel matrix and a product involving an inverse stationary multi-output kernel matrix. These procedures have respectively time complexities $\mathcal{O}(N^2Y \log Y + N^3Y)$ and $\mathcal{O}[N^2Y \log Y + N^3Y + PQ(NY \log Y + N^2Y)]$. Importantly, these complexities scale favourably in Y.

F | Exponentiated Quadratic Forms

F.1 Introduction

This chapter develops notation that makes working with exponentiated quadratic forms more convenient.

F.2 General Form

Notation 1 (Exponentiated Quadratic Form)**.** *Let $\boldsymbol{x} \in \mathbb{R}^N$. Then denote an exponentiated quadratic form*

$$p(\boldsymbol{x}) = C \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{b} + c\right)$$

by $(C, \boldsymbol{A}, \boldsymbol{b}, c)$.

We call $\boldsymbol{x}$ the *composite vector of variables*, or *composite vector* in short.

Lemma 4 (Product Identity (General Form))**.** *Let p_1 and p_2 be two exponentiated quadratic forms. Then*

$$p_1 p_2 = (C_1, \boldsymbol{A}_1, \boldsymbol{b}_1, c_1)(C_2, \boldsymbol{A}_2, \boldsymbol{b}_2, c_2) = (C_1 C_2, \boldsymbol{A}_1 + \boldsymbol{A}_2, \boldsymbol{b}_1 + \boldsymbol{b}_2, c_1 + c_2).$$

Proof. Direct computation yields that

$$\begin{aligned} p_1 p_2 &= C_1 C_2 \exp\Big(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{b}_1 - \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}_2\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{b}_2 + c\Big) \\ &= C_1 C_2 \exp\Big[-\frac{1}{2}\boldsymbol{x}^T(\boldsymbol{A}_1 + \boldsymbol{A}_2)\boldsymbol{x} + \boldsymbol{x}^T(\boldsymbol{b}_1 + \boldsymbol{b}_2) + (c_1 + c_2)\Big] \\ &= (C_1 C_2, \boldsymbol{A}_1 + \boldsymbol{A}_2, \boldsymbol{b}_1 + \boldsymbol{b}_2, c_1 + c_2). \end{aligned}$$

□

Lemma 5 (Integration Identity (General Form))**.** *Let the composite vector be* $\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T$ *and consider*

$$p = (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2\end{bmatrix}, c).$$

Denote integration over the subset of variables $\boldsymbol{x}_1$ *by* $I_{\boldsymbol{x}_1}(p)$*:*

$$I_{\boldsymbol{x}_1}(p)(\boldsymbol{x}_2) = \int (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2\end{bmatrix}, c)\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T\Big)\,\mathrm{d}\boldsymbol{x}_1.$$

Then

$$I_{\boldsymbol{x}_1}(p) = \left(C\frac{(2\pi)^{N/2}}{|\boldsymbol{A}_{1,1}|^{1/2}}, \boldsymbol{A}_{2,2} - \boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,2}, \boldsymbol{b}_2 - \boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{b}_1, c + \frac{1}{2}\boldsymbol{b}_1^T\boldsymbol{A}_{1,1}^{-1}\boldsymbol{b}_1\right).$$

Proof. An exponentiated quadratic form can be integrated as follows:

$$\begin{aligned} &\int_{\mathbb{R}^N} \exp\Big(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c\Big)\,\mathrm{d}\boldsymbol{x} \\ &= \exp\Big(\frac{1}{2}\boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b} + c\Big)\int_{\mathbb{R}^N} \exp\Big(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} - \frac{1}{2}\boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b}\Big)\,\mathrm{d}\boldsymbol{x} \\ &= \frac{(2\pi)^{N/2}}{|\boldsymbol{A}|^{1/2}}\exp\Big(\frac{1}{2}\boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b} + c\Big)\int_{\mathbb{R}^N} \frac{(2\pi)^{-N/2}}{|\boldsymbol{A}|^{-1/2}} \exp\Big(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} - \frac{1}{2}\boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b}\Big)\,\mathrm{d}\boldsymbol{x} \\ &= \frac{(2\pi)^{N/2}}{|\boldsymbol{A}|^{1/2}}\exp\Big(\frac{1}{2}\boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b} + c\Big)\int_{\mathbb{R}^N} \mathcal{N}(\boldsymbol{x}; \boldsymbol{A}^{-1}\boldsymbol{b}, \boldsymbol{A})\,\mathrm{d}\boldsymbol{x} \\ &= \frac{(2\pi)^{N/2}}{|\boldsymbol{A}|^{1/2}}\exp\Big(\frac{1}{2}\boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b} + c\Big). \end{aligned}$$

Now,

$$
\begin{aligned}
&-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}+\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}+c\\
&\quad=-\frac{1}{2}\boldsymbol{x}_1^T\boldsymbol{A}_{1,1}\boldsymbol{x}_1+\boldsymbol{x}_1^T\Big[\boldsymbol{b}_1-\underbrace{\frac{1}{2}(\boldsymbol{A}_{1,2}+\boldsymbol{A}_{2,1}^T)}_{\boldsymbol{A}_{1,2}}\boldsymbol{x}_2\Big]+\Big(\boldsymbol{x}_2^T\boldsymbol{b}_2-\frac{1}{2}\boldsymbol{x}_2^T\boldsymbol{A}_{2,2}\boldsymbol{x}_2+c\Big).
\end{aligned}
$$

Thus

$$
\begin{aligned}
I_{\boldsymbol{x}_1}(p)(\boldsymbol{x}_2) &= C\frac{(2\pi)^{N/2}}{|\boldsymbol{A}_{1,1}|^{1/2}}\exp\Big[\frac{1}{2}(\boldsymbol{b}_1-\boldsymbol{A}_{1,2}\boldsymbol{x}_2)^T\boldsymbol{A}_{1,1}^{-1}(\boldsymbol{b}_1-\boldsymbol{A}_{1,2}\boldsymbol{x}_2)\\
&\qquad\qquad+(\boldsymbol{x}_2^T\boldsymbol{b}_2-\frac{1}{2}\boldsymbol{x}_2^T\boldsymbol{A}_{2,2}\boldsymbol{x}_2+c)\Big]\\
&= C\frac{(2\pi)^{N/2}}{|\boldsymbol{A}_{1,1}|^{1/2}}\exp\Big[-\frac{1}{2}\boldsymbol{x}_2^T(\boldsymbol{A}_{2,2}-\boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,2})\boldsymbol{x}_2\\
&\qquad\qquad+\boldsymbol{x}_2^T(\boldsymbol{b}_2-\boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{b}_1)+(c+\frac{1}{2}\boldsymbol{b}_1^T\boldsymbol{A}_{1,1}^{-1}\boldsymbol{b}_1)\Big]
\end{aligned}
$$

and so the result follows. □

Lemma 6 (Linear Expansion Property (General Form))**.** *Let the composite vector be* $\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T$ *and consider*

$$
p_0 = (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}, c).
$$

Then the exponentiated quadratic forms

$$
\begin{aligned}
p_1\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \boldsymbol{x}_3^T\end{bmatrix}^T\Big) &= p_0\Big(\begin{bmatrix}(\boldsymbol{x}_1+\boldsymbol{B}\boldsymbol{x}_3)^T & \boldsymbol{x}_2^T\end{bmatrix}^T\Big),\\
p_2\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \boldsymbol{x}_3^T\end{bmatrix}^T\Big) &= p_0\Big(\begin{bmatrix}\boldsymbol{x}_1^T & (\boldsymbol{x}_2+\boldsymbol{B}\boldsymbol{x}_3)^T\end{bmatrix}^T\Big)
\end{aligned}
$$

are given by

$$
p_i = (C, \begin{bmatrix}\boldsymbol{A} & & \boldsymbol{A}_{1,i}\boldsymbol{B}\\ & & \boldsymbol{A}_{2,i}\boldsymbol{B}\\ \boldsymbol{B}^T\boldsymbol{A}_{i,1} & \boldsymbol{B}^T\boldsymbol{A}_{i,2} & \boldsymbol{B}^T\boldsymbol{A}_{i,i}\boldsymbol{B}\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\\ \boldsymbol{B}^T\boldsymbol{b}_i\end{bmatrix}, c)
$$

for respectively $i = 1, 2$.

Proof. The exponent of p_1 can be simplified as follows:

$$
\begin{aligned}
&-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_1+\boldsymbol{B}\boldsymbol{x}_3\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1+\boldsymbol{B}\boldsymbol{x}_3\\ \boldsymbol{x}_2\end{bmatrix}+\begin{bmatrix}\boldsymbol{x}_1+\boldsymbol{B}\boldsymbol{x}_3\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}+c\\
&\quad=-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}-\frac{1}{2}\begin{bmatrix}\boldsymbol{B}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}\\
&\qquad-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{B}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}-\frac{1}{2}\begin{bmatrix}\boldsymbol{B}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{B}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}\\
&\qquad+\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}+\begin{bmatrix}\boldsymbol{B}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}+c\\
&\quad=-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{0} & \boldsymbol{B}^T\boldsymbol{A}_{1,2}\\ \boldsymbol{0} & \boldsymbol{B}^T\boldsymbol{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}\\
&\qquad-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1}\boldsymbol{B} & \boldsymbol{0}\\ \boldsymbol{A}_{2,1}\boldsymbol{B} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{B}^T\boldsymbol{A}_{1,1}\boldsymbol{B} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}\\
&\qquad+\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix}+\begin{bmatrix}\boldsymbol{x}_3\\ \boldsymbol{0}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{B}^T\boldsymbol{b}_1\\ \boldsymbol{0}\end{bmatrix}+c\\
&\quad=-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \boldsymbol{x}_3\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{1,1}\boldsymbol{B}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{2,1}\boldsymbol{B}\\ \boldsymbol{B}^T\boldsymbol{A}_{1,1} & \boldsymbol{B}^T\boldsymbol{A}_{1,2} & \boldsymbol{B}^T\boldsymbol{A}_{1,1}\boldsymbol{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \boldsymbol{x}_3\end{bmatrix}+\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \boldsymbol{x}_3\end{bmatrix}^T\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\\ \boldsymbol{B}^T\boldsymbol{b}_1\end{bmatrix}+c.
\end{aligned}
$$

Thus

$$
p_1=(C,\begin{bmatrix}\boldsymbol{A} & \boldsymbol{A}_{1,1}\boldsymbol{B}\\ & \boldsymbol{A}_{2,1}\boldsymbol{B}\\ \boldsymbol{B}^T\boldsymbol{A}_{1,1} & \boldsymbol{B}^T\boldsymbol{A}_{1,2} \quad \boldsymbol{B}^T\boldsymbol{A}_{1,1}\boldsymbol{B}\end{bmatrix},\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\\ \boldsymbol{B}^T\boldsymbol{b}_1\end{bmatrix},c).
$$

The case p_2 follows similarly. □

Lemma 7 (Compression Property (General Form))**.** *Let the composite vector be* $\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \boldsymbol{x}_2^T\end{bmatrix}^T$ *and consider*

$$
p_1=(C,\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{1,3}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{2,3}\\ \boldsymbol{A}_{3,1} & \boldsymbol{A}_{3,2} & \boldsymbol{A}_{3,3}\end{bmatrix},\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\\ \boldsymbol{b}_3\end{bmatrix},c).
$$

Then the exponentiated quadratic form

$$p_2\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T\Big) = p_1\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \boldsymbol{x}_2^T\end{bmatrix}^T\Big)$$

is given by

$$p_2 = (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}+\boldsymbol{A}_{1,3} \\ \boldsymbol{A}_{2,1}+\boldsymbol{A}_{3,1} & \boldsymbol{A}_{2,2}+\boldsymbol{A}_{2,3}+\boldsymbol{A}_{3,2}+\boldsymbol{A}_{3,3}\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2+\boldsymbol{b}_3\end{bmatrix}, c).$$

Proof. The exponent of p_1 can be simplified as follows:

$$\begin{aligned}
&\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_2\end{bmatrix}^T \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{1,3} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{2,3} \\ \boldsymbol{A}_{3,1} & \boldsymbol{A}_{3,2} & \boldsymbol{A}_{3,3}\end{bmatrix} \begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_2\end{bmatrix} + \begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_2\end{bmatrix}^T \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3\end{bmatrix} + c \\
&\quad = \boldsymbol{x}_1^T\boldsymbol{A}_{1,1}\boldsymbol{x} + \boldsymbol{x}_1^T(\boldsymbol{A}_{1,2}+\boldsymbol{A}_{1,3})\boldsymbol{x}_2 + \boldsymbol{x}_2^T(\boldsymbol{A}_{2,1}+\boldsymbol{A}_{2,3})\boldsymbol{x}_1 \\
&\qquad + \boldsymbol{x}_2^T(\boldsymbol{A}_{2,2}+\boldsymbol{A}_{2,3}+\boldsymbol{A}_{3,2}+\boldsymbol{A}_{3,3})\boldsymbol{x}_2 + \boldsymbol{x}_1^T\boldsymbol{b}_1 + \boldsymbol{x}_2^T(\boldsymbol{b}_2+\boldsymbol{b}_3) + c \\
&\quad = \begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix}^T \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}+\boldsymbol{A}_{1,3} \\ \boldsymbol{A}_{2,1}+\boldsymbol{A}_{3,1} & \boldsymbol{A}_{2,2}+\boldsymbol{A}_{2,3}+\boldsymbol{A}_{3,2}+\boldsymbol{A}_{3,3}\end{bmatrix} \begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} + \begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix}^T \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2+\boldsymbol{b}_3\end{bmatrix} + c.
\end{aligned}$$

Hence the result follows. □

F.3 Kronecker-Structured Form

Notation 2 (Kronecker-Structured Exponentiated Quadratic Form)**.** *Let the composite vector be* $\begin{bmatrix}\boldsymbol{x}_1^T & \cdots & \boldsymbol{x}_M^T\end{bmatrix}^T \in \mathbb{R}^{MN}$*. Then denote a Kronecker-structured exponentiated quadratic form*

$$(C, \boldsymbol{A} \otimes \boldsymbol{I}_N, \boldsymbol{0}, 0)$$

by $(C, \boldsymbol{A})$*.*

Lemma 8 (Product Identity (Kronecker-Structured Form))**.** *Let* p_1 *and* p_2 *be two Kronecker-structured exponentiated quadratic forms. Then*

$$p_1 p_2 = (C_1, \boldsymbol{A}_1)(C_2, \boldsymbol{A}_2) = (C_1C_2, \boldsymbol{A}_1 + \boldsymbol{A}_2).$$

Proof. Follows from Lemma 4 and bilinearity of the Kronecker product. □

Lemma 9 (Integration Identity (Kronecker-Structured Form))**.** *Let the composite vector be* $\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T$ *where* $\boldsymbol{x}_1 \in \mathbb{R}^{KN}$ *and* $\boldsymbol{x}_2 \in \mathbb{R}^{MN}$*. Consider*

$$p = (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}).$$

Denote integration over the subset of variables $\boldsymbol{x}_1$ *by* $I_{\boldsymbol{x}_1}(p)$*:*

$$I_{\boldsymbol{x}_1}(p)(\boldsymbol{x}_2) = \int (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2\end{bmatrix}, c)\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T\Big)\,\mathrm{d}\boldsymbol{x}_1.$$

Then

$$I_{\boldsymbol{x}_1}(p) = \left(C\frac{(2\pi)^{KN/2}}{|\boldsymbol{A}_{1,1}|^{N/2}}, \boldsymbol{A}_{2,2} - \boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,2}\right).$$

Proof. Direct computation yields that

$$\begin{aligned}
I_{\boldsymbol{x}_1}[(C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix})] &= I_{\boldsymbol{x}_1}[(C, \begin{bmatrix}\boldsymbol{A}_{1,1}\otimes\boldsymbol{I} & \boldsymbol{A}_{1,2}\otimes\boldsymbol{I} \\ \boldsymbol{A}_{2,1}\otimes\boldsymbol{I} & \boldsymbol{A}_{2,2}\otimes\boldsymbol{I}\end{bmatrix}, \boldsymbol{0}, 0)] \\
&= \Bigg(C\frac{(2\pi)^{KN/2}}{|\boldsymbol{A}_{1,1}\otimes\boldsymbol{I}|^{1/2}}, \\
&\qquad \boldsymbol{A}_{2,2}\otimes\boldsymbol{I} - (\boldsymbol{A}_{2,1}\otimes\boldsymbol{I})(\boldsymbol{A}_{1,1}\otimes\boldsymbol{I})^{-1}(\boldsymbol{A}_{1,2}\otimes\boldsymbol{I}), \boldsymbol{0}, 0\Bigg) \\
&= \left(C\frac{(2\pi)^{KN/2}}{|\boldsymbol{A}_{1,1}|^{N/2}}, (\boldsymbol{A}_{2,2} - \boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,2})\otimes\boldsymbol{I}, \boldsymbol{0}, 0\right) \\
&= \left(C\frac{(2\pi)^{KN/2}}{|\boldsymbol{A}_{1,1}|^{N/2}}, \boldsymbol{A}_{2,2} - \boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,2}\right).
\end{aligned}$$

□

Lemma 10 (Linear Expansion Property (Kronecker-Structured Form))**.** *Let the composite vector be* $\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T$ *and consider*

$$p_0 = (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix}).$$

Then the exponentiated quadratic forms

$$p_1\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \boldsymbol{x}_3^T\end{bmatrix}^T\Big) = p_0\Big(\begin{bmatrix}(\boldsymbol{x}_1 + \boldsymbol{B}\boldsymbol{x}_3)^T & \boldsymbol{x}_2^T\end{bmatrix}^T\Big),$$
$$p_2\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \boldsymbol{x}_3^T\end{bmatrix}^T\Big) = p_0\Big(\begin{bmatrix}\boldsymbol{x}_1^T & (\boldsymbol{x}_2 + \boldsymbol{B}\boldsymbol{x}_3)^T\end{bmatrix}^T\Big)$$

are given by

$$p_i = (C, \begin{bmatrix}\boldsymbol{A} & \boldsymbol{A}_{1,i}\boldsymbol{B} \\ & \boldsymbol{A}_{2,i}\boldsymbol{B} \\ \boldsymbol{B}^T\boldsymbol{A}_{i,1} & \boldsymbol{B}^T\boldsymbol{A}_{i,2} & \boldsymbol{B}^T\boldsymbol{A}_{i,i}\boldsymbol{B}\end{bmatrix})$$

for respectively $i = 1, 2$.

Proof. Follows from Lemma 6 and bilinearity of the Kronecker product. □

Lemma 11 (Compression Property (Kronecker-Structured Form))**.** *Let the composite vector be* $\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \boldsymbol{x}_2^T\end{bmatrix}^T$ *and consider*

$$p_1 = (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{1,3} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{2,3} \\ \boldsymbol{A}_{3,1} & \boldsymbol{A}_{3,2} & \boldsymbol{A}_{3,3}\end{bmatrix}).$$

Then the exponentiated quadratic form

$$p_2\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T\Big) = p_1\Big(\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \boldsymbol{x}_2^T\end{bmatrix}^T\Big)$$

is given by

$$p_2 = (C, \begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} + \boldsymbol{A}_{1,3} \\ \boldsymbol{A}_{2,1} + \boldsymbol{A}_{3,1} & \boldsymbol{A}_{2,2} + \boldsymbol{A}_{2,3} + \boldsymbol{A}_{3,2} + \boldsymbol{A}_{3,3}\end{bmatrix}).$$

Proof. Follows from Lemma 7 and bilinearity of the Kronecker product. □

Finally, we prove one additional identity.

Lemma 12 (Direct Integration Identity (Kronecker-Structured Form))**.** *Let the com-*

posite vector be $\begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T\end{bmatrix}^T$ *where* $\boldsymbol{x}_1 \in \mathbb{R}^{KN}$ *and* $\boldsymbol{x}_2 = \begin{bmatrix}\boldsymbol{y}_1^T & \cdots \boldsymbol{y}_M^T\end{bmatrix}^T \in \mathbb{R}^{MN}$. *Then*

$$I_{\boldsymbol{x}_1}[(C, 2\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix})](\boldsymbol{x}_2) = C\frac{\pi^{KN/2}}{|\boldsymbol{A}_{1,1}|^{N/2}}\exp\left\{-\sum_{m_1=1,m_2=1}^{M,M} A_{2,2,m_1,m_2}\boldsymbol{y}_{m_1}^T\boldsymbol{y}_{m_2} + \sum_{k_1=1,k_2=1}^{K,K} A_{1,1,k_1,k_2}^{-1}\left(\sum_{m=1}^{M} A_{2,1,k_1,m}\boldsymbol{y}_m\right)^T \left(\sum_{m=1}^{M} A_{2,1,k_2,m}\boldsymbol{y}_m\right)\right\}.$$

Proof. Direct computation yields that

$$\begin{aligned} I_{\boldsymbol{x}_1}[(C, 2\begin{bmatrix}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2}\\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2}\end{bmatrix})] &= \left(C\frac{\pi^{KN/2}}{|\boldsymbol{A}_{1,1}|^{N/2}}, 2(\boldsymbol{A}_{2,2} - \boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,2})\right) \\ &= C\frac{\pi^{KN/2}}{|\boldsymbol{A}_{1,1}|^{N/2}}\exp\{-{\boldsymbol{x}_2}^T[(\boldsymbol{A}_{2,2} - \boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,2}) \otimes \boldsymbol{I}_N]\boldsymbol{x}_2\} \\ &= C\frac{\pi^{KN/2}}{|\boldsymbol{A}_{1,1}|^{N/2}}\exp\{-{\boldsymbol{x}_2}^T(\boldsymbol{A}_{2,2} \otimes \boldsymbol{I}_N)\boldsymbol{x}_2 \\ &\qquad + \boldsymbol{x}_2^T(\boldsymbol{A}_{2,1} \otimes \boldsymbol{I}_N)(\boldsymbol{A}_{1,1}^{-1} \otimes \boldsymbol{I}_N) \\ &\qquad [(\boldsymbol{A}_{1,2} \otimes \boldsymbol{I}_N)\boldsymbol{x}_2]\}. \end{aligned}$$

Hence the result follows. □

F.4 Conclusion

We have developed notation that makes working with exponentiated quadratic forms more convenient. Importantly, the notation reduces manipulation of exponentiated quadratic forms to relatively simple matrix operations.

G | Roots of Kernels

First, using the notation and identities from Appendix F with composite vector $\begin{bmatrix}\boldsymbol{t}^T & \boldsymbol{\tau}^T\end{bmatrix}^T$, any kernel of exponentiated quadratic form has a root:

$$
\begin{aligned}
&R\left[\left(\sigma\frac{|2\boldsymbol{A}|^{1/4}}{\pi^{K/4}}, 2\boldsymbol{A}, \mathbf{0}, 0\right)\right] * \left(\sigma\frac{|2\boldsymbol{A}|^{1/4}}{\pi^{K/4}}, 2\boldsymbol{A}, \mathbf{0}, 0\right) \\
&\quad = I_{\boldsymbol{\tau}}\left[\left(\sigma\frac{|2\boldsymbol{A}|^{1/4}}{\pi^{K/4}}, 2\begin{bmatrix}\boldsymbol{A} & -\boldsymbol{A}\\ -\boldsymbol{A} & \boldsymbol{A}\end{bmatrix}, \mathbf{0}, 0\right)\left(\sigma\frac{|2\boldsymbol{A}|^{1/4}}{\pi^{K/4}}, 2\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{A}\end{bmatrix}, \mathbf{0}, 0\right)\right] \\
&\quad = I_{\boldsymbol{\tau}}\left[\left(\sigma^2\frac{|2\boldsymbol{A}|^{1/2}}{\pi^{K/2}}, 2\begin{bmatrix}\boldsymbol{A} & -\boldsymbol{A}\\ -\boldsymbol{A} & 2\boldsymbol{A}\end{bmatrix}, \mathbf{0}, 0\right)\right] \\
&\quad = \left(\sigma^2\frac{|2\boldsymbol{A}|^{1/2}}{\pi^{K/2}}\frac{\pi^{K/2}}{|2\boldsymbol{A}|^{1/2}}, 2[\boldsymbol{A} - \boldsymbol{A}(2\boldsymbol{A})^{-1}\boldsymbol{A}], \mathbf{0}, 0\right) \\
&\quad = (\sigma^2, \boldsymbol{A}, \mathbf{0}, 0).
\end{aligned}
$$

Second, consider the diagonal multi-output exponentiated-quadratic kernel. Its diagonal entries are of exponentiated quadratic form and thus have roots. Now, for any diagonal $\boldsymbol{H}$ it holds that

$$R(\boldsymbol{H}) * \boldsymbol{H}^T = \mathrm{diag}[R(H_{1,1}) * H_{1,1}, \ldots, R(H_{N,N})].$$

Thus the diagonal entries having roots implies that the diagonal multi-output exponentiated-quadratic kernel has a root. Specifically, if

$$\boldsymbol{\mathcal{K}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \sigma^2 \exp(-\gamma\|\boldsymbol{t}_1 - \boldsymbol{t}_2\|^2)\boldsymbol{I},$$

then $\boldsymbol{\mathcal{K}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = [R(\boldsymbol{R}) * \boldsymbol{R}](\boldsymbol{t}_2 - \boldsymbol{t}_1)$ where

$$\boldsymbol{R}(\boldsymbol{t}) = \sigma \frac{(4\gamma)^{K/4}}{\pi^{K/4}} \exp(-2\gamma \|\boldsymbol{t}\|^2) \boldsymbol{I}.$$

Finally, the white noise kernel is its own root:

$$[R(\delta \boldsymbol{I}) * \delta \boldsymbol{I}](\boldsymbol{t}) = \int_{\mathbb{R}^K} \delta(\boldsymbol{t}' - \boldsymbol{t}) \boldsymbol{I} \delta(\boldsymbol{t}') \boldsymbol{I} \, \mathrm{d}\boldsymbol{t}' = \delta(\boldsymbol{t}) \boldsymbol{I}.$$

H | Approximate Kernel Model

We derive an analytical expression for $\tilde{\mathcal{K}}_{\boldsymbol{f}\,|\,\boldsymbol{H}(\boldsymbol{T})}$. Recall that $\boldsymbol{H}$'s components are modelled independently. Then

$$\tilde{\mathcal{K}}_{f_i,f_j\,|\,\boldsymbol{H}(\boldsymbol{T})} = \sum_{k=1}^{M} \int_{\mathbb{R}^K} \mathbb{E}[H_{i,k}(\boldsymbol{t}_1 - \boldsymbol{\tau}) H_{j,k}(\boldsymbol{t}_2 - \boldsymbol{\tau}) \,|\, \boldsymbol{H}(\boldsymbol{T})]\,\mathrm{d}\boldsymbol{\tau}$$

where, denoting $h_i = H_{i,k}$, $h_j = H_{j,k}$, $\boldsymbol{u}_{h_i} = h_i(\boldsymbol{T})$ and $\boldsymbol{u}_{h_j} = h_j(\boldsymbol{T})$,

$$\begin{aligned}
&\int_{\mathbb{R}^K} \mathbb{E}[H_{i,k}(\boldsymbol{t}_1 - \boldsymbol{\tau}) H_{j,k}(\boldsymbol{t}_2 - \boldsymbol{\tau}) \,|\, \boldsymbol{H}(\boldsymbol{T})]\,\mathrm{d}\boldsymbol{\tau} \\
&\quad = \int_{\mathbb{R}^K} \{\mathcal{K}_{h_i,h_j\,|\,\boldsymbol{u}_{h_i},\boldsymbol{u}_{h_j}}(\boldsymbol{t}_1 - \boldsymbol{\tau}, \boldsymbol{t}_2 - \boldsymbol{\tau}) \\
&\qquad\qquad - \mathbb{E}[h_i(\boldsymbol{t}_1 - \boldsymbol{\tau}) \,|\, \boldsymbol{u}_{h_i}] \mathbb{E}[h_j(\boldsymbol{t}_1 - \boldsymbol{\tau}) \,|\, \boldsymbol{u}_{h_j}]\}\,\mathrm{d}\boldsymbol{\tau} \\
&\quad = \int_{\mathbb{R}^{2K}} [\mathbb{1}(i-j)\mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}, \boldsymbol{t}_2 - \boldsymbol{\tau}) \\
&\qquad\qquad + \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}, \boldsymbol{T}) \boldsymbol{M}^{(h_i,h_j)} \mathcal{K}_{h_j}(\boldsymbol{T}, \boldsymbol{t}_2 - \boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau} \\
&\quad = \mathbb{1}(i-j) \underbrace{\int_{\mathbb{R}^K} \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}, \boldsymbol{t}_2 - \boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau}}_{I^{(1,h_i)}(\boldsymbol{t}_1,\boldsymbol{t}_2)} \qquad \text{(H.1)} \\
&\qquad + \sum_{m=1,n=1}^{T,T} M_{m,n}^{(h_i,h_j)} \underbrace{\int_{\mathbb{R}^K} \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}, \boldsymbol{T}_{m,:}) \mathcal{K}_{h_j}(\boldsymbol{T}_{n,:}, \boldsymbol{t}_2 - \boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau}}_{I_{m,n}^{(2,h_i,h_j)}(\boldsymbol{t}_1,\boldsymbol{t}_2)} \qquad \text{(H.2)}
\end{aligned}$$

and

$$\boldsymbol{M}^{(h_i,h_j)} = \boldsymbol{K}_{\boldsymbol{u}_{h_i}}^{-1} \boldsymbol{u}_{h_i} \boldsymbol{u}_{h_j}^T \boldsymbol{K}_{\boldsymbol{u}_{h_j}}^{-1} - \mathbb{1}(i-j) \boldsymbol{K}_{\boldsymbol{u}_{h_i}}^{-1}.$$

Hence,

$$\begin{aligned}\tilde{\mathcal{K}}_{f_i,f_j\,|\,\boldsymbol{H}(\boldsymbol{T})} &= \sum_{k=1}^{M}\Big[\mathbb{1}(i-j)I^{(1,H_{i,k})} + \mathrm{tr}(\boldsymbol{M}^{(H_{i,k},H_{j,k})^T}\boldsymbol{I}^{(2,H_{i,k},H_{j,k})})\Big] \\ &= \sum_{k=1}^{M}\Big\{\mathbb{1}(i-j)[I^{(1,H_{i,k})} - \mathrm{tr}(\boldsymbol{K}^{-1}_{H_{i,k}(\boldsymbol{T})}\boldsymbol{I}^{(2,H_{i,k},H_{i,k})})] \\ &\qquad + H^T_{i,k}(\boldsymbol{T})\boldsymbol{K}^{-1}_{H_{i,k}(\boldsymbol{T})}\boldsymbol{I}^{(2,H_{i,k},H_{j,k})}\boldsymbol{K}^{-1}_{H_{j,k}(\boldsymbol{T})}H_{j,k}(\boldsymbol{T})\Big\}.\end{aligned}$$

I | Variational Free Energy of the Nonparametric Kernel Model

I.1 Introduction

This chapter derives an analytical expression for the variational free energy of Model 5. We also determine the asymptotic time complexity of computing the free energy.

Recall that the free energy is given by

$$\mathcal{F}(q) = \underbrace{-\frac{1}{2}\log[(2\pi)^N|\boldsymbol{\Lambda}|^2] - \frac{1}{2}\sum_{i=1}^{Y}\mathbb{E}_q\{\|\boldsymbol{\Lambda}^{-1}[\boldsymbol{Y}_{i,:} - \boldsymbol{y}(\boldsymbol{T}_{i,:})]\|^2\}}_{\mathbb{E}_q[\log p(\boldsymbol{Y}\,|\,\boldsymbol{H},\boldsymbol{x})]} - \sum_{i=1,j=1}^{N,M} D_{KL}[q(\boldsymbol{u}_{H_{i,j}}) \,\|\, p(\boldsymbol{u}_{H_{i,j}})] - \sum_{j=1}^{M} D_{KL}[q(\boldsymbol{u}_{\tilde{x}_j}) \,\|\, p(\boldsymbol{u}_{\tilde{x}_j})].$$

In more detail, the Kullback-Leibler divergences are given by (Appendix B.2)

$$D_{KL}[q(\boldsymbol{u}_{H_{i,j}}) \,\|\, p(\boldsymbol{u}_{H_{i,j}})] = \frac{1}{2}\left[\mathrm{tr}(\boldsymbol{K}_{\boldsymbol{u}_{H_{i,j}}}^{-1}\boldsymbol{\Sigma}_{H_{i,j}}) + \boldsymbol{\mu}_{H_{i,j}}^T\boldsymbol{K}_{\boldsymbol{u}_{H_{i,j}}}^{-1}\boldsymbol{\mu}_{H_{i,j}} + \log\frac{|\boldsymbol{K}_{\boldsymbol{u}_{H_{i,j}}}|}{|\boldsymbol{\Sigma}_{H_{i,j}}|} - N\right], \tag{I.1}$$

$$D_{KL}[q(\boldsymbol{u}_{\tilde{x}_j}) \,\|\, p(\boldsymbol{u}_{\tilde{x}_j})] = \frac{1}{2}\left[\mathrm{tr}(\boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_j}}^{-1}\boldsymbol{\Sigma}_{\tilde{x}_j}) + \boldsymbol{\mu}_{\tilde{x}_j}^T\boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_j}}^{-1}\boldsymbol{\mu}_{\tilde{x}_j} + \log\frac{|\boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_j}}|}{|\boldsymbol{\Sigma}_{\tilde{x}_j}|} - N\right] \tag{I.2}$$

and the likelihood is given by

$$\mathbb{E}_q[\log p(\boldsymbol{Y}\,|\,\boldsymbol{H},\boldsymbol{x})] = -\frac{1}{2}\log[(2\pi)^N|\boldsymbol{\Lambda}|^2] - \frac{1}{2}\sum_{i=1}^{Y}\Big\{\boldsymbol{Y}_{i,:}^T\boldsymbol{\Lambda}^{-2}\boldsymbol{Y}_{i,:} - 2\boldsymbol{Y}_{i,:}^T\boldsymbol{\Lambda}^{-2}\mathbb{E}_q[\boldsymbol{y}(\boldsymbol{T}_{i,:})] + \mathrm{tr}\{\boldsymbol{\Lambda}^{-2}\mathbb{E}_q[\boldsymbol{y}(\boldsymbol{T}_{i,:})\boldsymbol{y}^T(\boldsymbol{T}_{i,:})]\}\Big\}.$$

It thus remains to determine the predictive mean $\mathbb{E}_q[\boldsymbol{y}(\boldsymbol{t})]$ and predictive autocovariance $\mathbb{E}_q[\boldsymbol{y}(\boldsymbol{t}_1)\boldsymbol{y}^T(\boldsymbol{t}_2)]$. Note that the latter is only evaluated on its diagonal.

Recall that $H_{i,j}$ and $\tilde{x}_j$ denote the number of inducing points for the processes $T_{H_{i,j}}$ and $T_{\tilde{x}_j}$ respectively. Also recall that $T_H = \max_{i,j} T_{H_{i,j}}$ and $T_{\tilde{x}} = \max_j T_{\tilde{x}_j}$. Finally, recall that Y denotes the number of observed data points. Then Equations (I.1) and (I.2) have respectively time complexities $\mathcal{O}(T_H^3)$ and $\mathcal{O}(T_{\tilde{x}}^3)$ (Appendix D). Thus $\mathcal{F}(q)$ has time complexity $\mathcal{O}[NMT_H^3 + MT_{\tilde{x}}^3 + YN(C_\mu + C_\Sigma)]$ where C_μ and C_Σ represent the number of operations required to compute respectively $\mathbb{E}_q[y_i(\boldsymbol{t})]$ and $\mathbb{E}_q[y_i(\boldsymbol{t})y_i(\boldsymbol{t})]$.

In the remainder of this chapter we utilise the notation and identities from Appendix F without further notion.

I.2 Predictive Mean

We have that

$$\mathbb{E}_q[y_i(\boldsymbol{t})] = \mathbb{E}_q\{[(\boldsymbol{H} * \boldsymbol{x})(\boldsymbol{t})]_i\} = \sum_{j=1}^{M} \underbrace{\int_{\mathbb{R}^K} \mathbb{E}_q[H_{i,j}(\boldsymbol{t}-\boldsymbol{\tau})x_j(\boldsymbol{\tau})]\,\mathrm{d}\boldsymbol{\tau}}_{T_{i,j}^{(1)}}$$

where, denoting $h = H_{i,j}$ and $x = x_j$,

$$\begin{aligned} T_{i,j}^{(1)} &= \int h(\boldsymbol{t}-\boldsymbol{\tau})p(h \mid \boldsymbol{u}_h)\,\mathrm{d}h q(\boldsymbol{u}_h)\,\mathrm{d}\boldsymbol{u}_h \\ &\qquad x(\boldsymbol{\tau})p(x \mid \boldsymbol{u}_{\tilde{x}})\,\mathrm{d}x q(\boldsymbol{u}_{\tilde{x}})\,\mathrm{d}\boldsymbol{u}_{\tilde{x}}\,\mathrm{d}\boldsymbol{\tau} \\ &= \int_{\mathbb{R}^2} \mathcal{K}_h(\boldsymbol{t}-\boldsymbol{\tau}, \boldsymbol{T}_h)\boldsymbol{K}_{\boldsymbol{u}_h}^{-1}\boldsymbol{\mu}_h \mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau}, \boldsymbol{T}_{\tilde{x}})\boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}}}^{-1}\boldsymbol{\mu}_{\tilde{x}}\,\mathrm{d}\boldsymbol{\tau} \\ &= \boldsymbol{\mu}_h^T\boldsymbol{K}_{\boldsymbol{u}_h}^{-1} \underbrace{\int_{\mathbb{R}^K} \mathcal{K}_h(\boldsymbol{T}_h, \boldsymbol{t}-\boldsymbol{\tau})\mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau}, \boldsymbol{T}_{\tilde{x}})\,\mathrm{d}\boldsymbol{\tau}}_{\boldsymbol{I}^{(L,h,x)}(\boldsymbol{t})} \boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}}}^{-1}\boldsymbol{\mu}_{\tilde{x}}. \end{aligned}$$

In summary,

$$\mathbb{E}_q[y_i(\boldsymbol{t})] = \sum_{j=1}^{M} \boldsymbol{\mu}_{H_{i,j}}^T \boldsymbol{K}_{\boldsymbol{u}_{H_{i,j}}}^{-1} \boldsymbol{I}^{(L,H_{i,j},x_j)}(\boldsymbol{t}) \boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_j}}^{-1} \boldsymbol{\mu}_{\tilde{x}_j}.$$

The integral $\boldsymbol{I}^{(L,\cdot,\cdot)}$ is computed in Appendix I.4.

To begin with, there is a one-time cost of constructing and inverting all $\boldsymbol{K}_{\boldsymbol{u}_{H_{i,j}}}$ and $\boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_j}}$, which has time complexity $\mathcal{O}[M(T_H^3 + KT_H^2 + KT_{\tilde{x}}^2 + T_{\tilde{x}}^3)]$. Afterwards, all $\boldsymbol{I}^{(L,H_{i,j},\tilde{x}_j)}$ are

constructed in $\mathcal{O}(MKT_HT_{\tilde{x}})$ time and the predictive mean is computed in $\mathcal{O}[M(T_H^2 + T_HT_{\tilde{x}} + T_{\tilde{x}}^2)]$. Hence $\mathcal{O}(C_\mu) = \mathcal{O}[M(T_H^2 + KT_HT_{\tilde{x}} + T_{\tilde{x}}^2)]$.

I.3 Predictive Autocovariance

We have that

$$\begin{aligned}
\mathbb{E}_q[y_i(\boldsymbol{t}_1)y_j(\boldsymbol{t}_2)] &= \sum_{k=1,l=1}^{M,M} \int_{\mathbb{R}^{2K}} \mathbb{E}_q[H_{i,k}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1)x_k(\boldsymbol{\tau}_1)H_{j,l}(\boldsymbol{t}_2 - \boldsymbol{\tau}_2)x_l(\boldsymbol{\tau}_2)]\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2 \\
&= \sum_{k=1}^{M} \int_{\mathbb{R}^K} \mathbb{E}_q[H_{i,k}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1)x_k(\boldsymbol{\tau})]\,\mathrm{d}\boldsymbol{\tau} \sum_{l=1,l\neq k}^{M} \int_{\mathbb{R}^K} \mathbb{E}_q[H_{j,l}(\boldsymbol{t}_2 - \boldsymbol{\tau})x_l(\boldsymbol{\tau})]\,\mathrm{d}\boldsymbol{\tau} \\
&\quad + \sum_{k=1}^{M} \underbrace{\int_{\mathbb{R}^{2K}} \mathbb{E}_q[H_{i,k}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1)x_k(\boldsymbol{\tau}_1)H_{j,k}(\boldsymbol{t}_2 - \boldsymbol{\tau}_2)x_k(\boldsymbol{\tau}_2)]\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2}_{T_{i,j,k}^{(2)}} \\
&= \sum_{k=1}^{M} T_{i,k}^{(1)} \sum_{l=1:l\neq k}^{M} T_{j,l}^{(1)} + \sum_{k=1}^{M} T_{i,j,k}^{(2)}.
\end{aligned}$$

Then, denoting $h_i = H_{i,k}$, $h_j = H_{j,k}$ and $x = x_k$,

$$\begin{aligned}
T_{i,j,k}^{(2)} &= \int h_i(\boldsymbol{t}_1 - \boldsymbol{\tau}_1)h_j(\boldsymbol{t}_2 - \boldsymbol{\tau}_2)p(h_i, h_j \,|\, \boldsymbol{u}_{h_i}, \boldsymbol{u}_{h_j})\,\mathrm{d}(h_i, h_j)q(\boldsymbol{u}_{h_i}, \boldsymbol{u}_{h_j})\,\mathrm{d}(\boldsymbol{u}_{h_i}, \boldsymbol{u}_{h_j}) \\
&\qquad x(\boldsymbol{\tau}_1)x(\boldsymbol{\tau}_2)p(x \,|\, \boldsymbol{u}_{\tilde{x}})\,\mathrm{d}x q(\boldsymbol{u}_{\tilde{x}})\,\mathrm{d}\boldsymbol{u}_{\tilde{x}}\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2 \\
&= \int_{\mathbb{R}^{2K}} [\mathbb{1}(i-j)\mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1, \boldsymbol{t}_2 - \boldsymbol{\tau}_2) + \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1, \boldsymbol{T}_{h_i})\boldsymbol{M}^{(h_i,h_j)}\mathcal{K}_{h_j}(\boldsymbol{T}_{h_j}, \boldsymbol{t}_2 - \boldsymbol{\tau}_2)] \\
&\qquad [\mathcal{K}_x(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) + \mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau}_1, \boldsymbol{T}_{\tilde{x}})\boldsymbol{M}^{(\tilde{x})}\mathcal{K}_{\tilde{x},x}(\boldsymbol{T}_{\tilde{x}}, \boldsymbol{\tau}_2)]\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{M}^{(h_i,h_j)} &= \boldsymbol{K}_{\boldsymbol{u}_{h_i}}^{-1}[\mathbb{1}(i-j)\boldsymbol{\Sigma}_{h_i} + \boldsymbol{\mu}_{h_i}\boldsymbol{\mu}_{h_j}^T]\boldsymbol{K}_{\boldsymbol{u}_{h_j}}^{-1} - \mathbb{1}(i-j)\boldsymbol{K}_{\boldsymbol{u}_{h_i}}^{-1}, \\
\boldsymbol{M}^{(\tilde{x})} &= \boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}}}^{-1}[\boldsymbol{\Sigma}_{\tilde{x}} + \boldsymbol{\mu}_{\tilde{x}}\boldsymbol{\mu}_{\tilde{x}}^T]\boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}}}^{-1} - \boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}}}^{-1}.
\end{aligned}$$

Thus $T^{(2)}_{i,j,k} = Q_1 + Q_2 + Q_3 + Q_4$ where

$$Q_1 = \mathbb{1}(i-j) \underbrace{\int_{\mathbb{R}^{2K}} \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1, \boldsymbol{t}_2 - \boldsymbol{\tau}_2)\mathcal{K}_x(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2)\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2}_{I^{(Q_1,h_i,x)}(\boldsymbol{t}_1,\boldsymbol{t}_2)},$$

$$Q_2 = \mathbb{1}(i-j) \sum_{m,n} M^{(\tilde{x})}_{m,n} \underbrace{\int_{\mathbb{R}^{2K}} \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1, \boldsymbol{t}_2 - \boldsymbol{\tau}_2)\mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau}_1, \boldsymbol{T}_{\tilde{x},m,:})\mathcal{K}_{\tilde{x},x}(\boldsymbol{T}_{\tilde{x},n,:}, \boldsymbol{\tau}_2)\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2}_{I^{(Q_2,h_i,x)}_{m,n}(\boldsymbol{t}_1,\boldsymbol{t}_2)},$$

$$Q_3 = \sum_{m,n} M^{(h_i,h_j)}_{m,n} \underbrace{\int_{\mathbb{R}^{2K}} \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1, \boldsymbol{T}_{h_i,m,:})\mathcal{K}_{h_j}(\boldsymbol{T}_{h_j,n,:}, \boldsymbol{t}_2 - \boldsymbol{\tau}_2)\mathcal{K}_x(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2)\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2}_{I^{(Q_3,h_i,h_j,x)}_{m,n}(\boldsymbol{t}_1,\boldsymbol{t}_2)},$$

$$\begin{aligned} Q_4 &= \sum_{m,n,o,p} M^{(h_i,h_j)}_{m,n} M^{(\tilde{x})}_{o,p} \int_{\mathbb{R}^{2K}} \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}_1, \boldsymbol{T}_{h_i,m,:})\mathcal{K}_{h_j}(\boldsymbol{T}_{h_j,n,:}, \boldsymbol{t}_2 - \boldsymbol{\tau}_2) \\ &\qquad \mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau}_1, \boldsymbol{T}_{\tilde{x},o,:})\mathcal{K}_{\tilde{x},x}(\boldsymbol{T}_{\tilde{x},p,:}, \boldsymbol{\tau}_2)\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2 \\ &= \sum_{m,n,o,p} M^{(h_i,h_j)}_{m,n} M^{(\tilde{x})}_{o,p} \int_{\mathbb{R}^K} \mathcal{K}_{h_i}(\boldsymbol{t}_1 - \boldsymbol{\tau}, \boldsymbol{T}_{h_i,m,:})\mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau}, \boldsymbol{T}_{\tilde{x},o,:})\,\mathrm{d}\boldsymbol{\tau} \\ &\qquad \underbrace{\int_{\mathbb{R}^K} \mathcal{K}_{h_j}(\boldsymbol{t}_2 - \boldsymbol{\tau}, \boldsymbol{T}_{h_j,n,:})\mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau}, \boldsymbol{T}_{\tilde{x},p,:})\,\mathrm{d}\boldsymbol{\tau}}_{I^{(Q_4,h_j,x)}_{n,p}(\boldsymbol{t}_2)} \\ &= \sum_{m,n,o,p} M^{(h_i,h_j)}_{m,n} M^{(\tilde{x})}_{o,p} I^{(Q_4,h_i,x)}_{m,o}(\boldsymbol{t}_1) I^{(Q_4,h_j,x)}_{n,p}(\boldsymbol{t}_2). \end{aligned}$$

In summary,

$$\begin{aligned}\mathbb{E}_q[y_i(\boldsymbol{t}_1)y_j(\boldsymbol{t}_2)] = \sum_{k=1}^{M} \boldsymbol{\mu}_{H_{i,k}}^T \boldsymbol{K}_{\boldsymbol{u}_{H_{i,k}}}^{-1} \boldsymbol{I}^{(L,H_{i,k},x_k)}(\boldsymbol{t}_1) \boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_k}}^{-1} \boldsymbol{\mu}_{\tilde{x}_k} \\ \underbrace{\sum_{l=1,l\neq k}^{M} \boldsymbol{\mu}_{H_{j,l}}^T \boldsymbol{K}_{\boldsymbol{u}_{H_{j,l}}}^{-1} \boldsymbol{I}^{(L,H_{j,l},x_l)}(\boldsymbol{t}_2) \boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_l}}^{-1} \boldsymbol{\mu}_{\tilde{x}_l}}_{T^{(1)}} \\ + \mathbb{1}(i-j) \sum_{k=1}^{M} I^{(Q_1,H_{i,k},x_k)}(\boldsymbol{t}_1,\boldsymbol{t}_2) \\ \underbrace{\begin{aligned} &+ \mathbb{1}(i-j) \sum_{k=1}^{M} \operatorname{tr}[\boldsymbol{M}^{(\tilde{x}_k)T} \boldsymbol{I}^{(Q_2,H_{i,k},x_k)}(\boldsymbol{t}_1,\boldsymbol{t}_2)] \\ &+ \sum_{k=1}^{M} \operatorname{tr}[\boldsymbol{M}^{(H_{i,k})T} \boldsymbol{I}^{(Q_3,H_{i,k},H_{j,k},x_k)}(\boldsymbol{t}_1,\boldsymbol{t}_2)] \\ &+ \sum_{k=1}^{M} \operatorname{tr}[\boldsymbol{M}^{(H_{i,k},H_{j,k})T} \boldsymbol{I}^{(Q_4,H_{i,k},x_k)}(\boldsymbol{t}_1) \boldsymbol{M}^{(\tilde{x}_k)} \boldsymbol{I}^{(Q_4,H_{j,k},x_k)T}(\boldsymbol{t}_2)]. \end{aligned}}_{T^{(2)}}\end{aligned}$$

The integrals $I_{\cdot}^{(\cdot)}$ are computed in Appendix I.4.

To begin with, there is a one-time cost of constructing and inverting all $\boldsymbol{K}_{\boldsymbol{u}_{H_{i,j}}}$ and $\boldsymbol{K}_{\boldsymbol{u}_{\tilde{x}_j}}$, which has time complexity $\mathcal{O}[M(T_H^3 + KT_H^2 + KT_{\tilde{x}}^2 + T_{\tilde{x}}^3)]$, and a one-time cost of computing all $\boldsymbol{M}^{(H_{i,j},H_{i,j})}$ and $\boldsymbol{M}^{(\tilde{x}_j)}$, which has time complexity $\mathcal{O}[M(T_H^3 + T_{\tilde{x}}^3)]$. Thus, the resulting one-time cost has time complexity $\mathcal{O}[M(T_H^3 + KT_H^2 + KT_{\tilde{x}}^2 + T_{\tilde{x}}^3)]$. Afterwards, all $\boldsymbol{I}^{(Q_2,H_{i,j},x_j)}$, $\boldsymbol{I}^{(Q_3,H_{i,j},H_{i,j},x_j)}$ and $\boldsymbol{I}^{(Q_4,H_{i,j},x_j)}$ are constructed in $\mathcal{O}[M(KT_H^2 + KT_H T_{\tilde{x}} + KT_{\tilde{x}}^2)]$ time and $T^{(1)}$ and $T^{(2)}$ are computed in respectively $\mathcal{O}[M(T_H^2 + T_H T_{\tilde{x}} + T_{\tilde{x}}^2) + M^2]$ and $\mathcal{O}(M[T_H^2 T_{\tilde{x}} + T_H T_{\tilde{x}}^2])$ time. Hence

$$\mathcal{O}(C_\Sigma) = \mathcal{O}[M(T_H^2 T_{\tilde{x}} + KT_H^2 + KT_H T_{\tilde{x}} + KT_{\tilde{x}}^2 + T_H T_{\tilde{x}}^2) + M^2].$$

I.4 Integrals $I_{\cdot}^{(\cdot)}$

In this section we compute the integrals $I_{\cdot}^{(\cdot)}$ from Appendix I.3.

I.4.1 Kernels

First, let the composite vector be $\begin{bmatrix} \boldsymbol{t}_1^T & \boldsymbol{t}_2^T & \boldsymbol{\tau} \end{bmatrix}^T$. Then

$$\begin{aligned}
\mathcal{K}_{\tilde{x}_j}(\boldsymbol{t}_1, \boldsymbol{t}_2) &= \int_{\mathbb{R}^{2K}} \exp(-\omega\|\boldsymbol{t}_1 - \boldsymbol{\tau}_1\|^2) \underbrace{\mathbb{E}[x_j(\boldsymbol{\tau}_1)x_j(\boldsymbol{\tau}_2)]}_{\delta(\boldsymbol{\tau}_1-\boldsymbol{\tau}_2)} \exp(-\omega\|\boldsymbol{t}_2 - \boldsymbol{\tau}_2\|^2)\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2 \\
&= \int_{\mathbb{R}^K} \exp(-\omega\|\boldsymbol{t}_1 - \boldsymbol{\tau}\|^2 - \omega\|\boldsymbol{t}_2 - \boldsymbol{\tau}\|^2)\,\mathrm{d}\boldsymbol{\tau} \\
&= I_{\boldsymbol{\tau}}[(1, 2 \left[\begin{array}{cc|c} \omega & 0 & -\omega \\ 0 & \omega & -\omega \\ \hline -\omega & -\omega & 2\omega \end{array}\right])] \\
&= \left(\frac{\pi^{K/2}}{(2\omega)^{K/2}}, 2 \begin{bmatrix} \omega & 0 \\ 0 & \omega \end{bmatrix} - \frac{1}{\omega} \begin{bmatrix} \omega \\ \omega \end{bmatrix} \begin{bmatrix} \omega \\ \omega \end{bmatrix}^T \right) \\
&= \left(\frac{\pi^{K/2}}{(2\omega)^{K/2}}, \begin{bmatrix} \omega & -\omega \\ -\omega & \omega \end{bmatrix} \right) \\
&= \frac{\pi^{K/2}}{(2\omega)^{K/2}} \exp\left(-\frac{1}{2}\omega\|\boldsymbol{t}_1 - \boldsymbol{t}_2\|^2\right) \\
&= \mathcal{K}_{\tilde{x}_j}(\boldsymbol{t}_1 - \boldsymbol{t}_2).
\end{aligned}$$

Second, let the composite vector be $\begin{bmatrix} \boldsymbol{t}_1^T & \boldsymbol{t}_2^T \end{bmatrix}^T$. Then

$$\begin{aligned}
\mathcal{K}_{x_j,\tilde{x}_j}(\boldsymbol{t}_1, \boldsymbol{t}_2) &= \int_{\mathbb{R}^K} \underbrace{\mathbb{E}[x_j(\boldsymbol{t}_1)x_j(\boldsymbol{t}_2 - \boldsymbol{\tau})]}_{\delta(\boldsymbol{t}_1-\boldsymbol{t}_2+\boldsymbol{\tau})} \exp(-\omega\|\boldsymbol{\tau}\|^2)\,\mathrm{d}\boldsymbol{\tau} \\
&= \exp(-\omega\|\boldsymbol{t}_1 - \boldsymbol{t}_2\|^2) \\
&= (1, 2 \begin{bmatrix} \omega & -\omega \\ -\omega & \omega \end{bmatrix}) \\
&= \mathcal{K}_{x_j,\tilde{x}_j}(\boldsymbol{t}_1 - \boldsymbol{t}_2)
\end{aligned}$$

and $\mathcal{K}_{\tilde{x}_j,x_j}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \mathcal{K}_{x_j,\tilde{x}_j}(\boldsymbol{t}_1, \boldsymbol{t}_2)$.

Finally, let the composite vector be $\begin{bmatrix}\boldsymbol{t}_1^T & \boldsymbol{t}_2^T\end{bmatrix}^T$. Then

$$\begin{aligned}\mathcal{K}_{H_{i,j}}(\boldsymbol{t}_1,\boldsymbol{t}_2) &= \sigma_h^2\exp(-\alpha\|\boldsymbol{t}_1\|^2-\alpha\|\boldsymbol{t}_2\|^2-\gamma\|\boldsymbol{t}_2-\boldsymbol{t}_1\|^2)\\ &= (\sigma_h^2, 2\begin{bmatrix}\alpha+\gamma & -\gamma\\ -\gamma & \alpha+\gamma\end{bmatrix}).\end{aligned}$$

I.4.2 Integral $I_{\cdot,\cdot}^{(L,\cdot,\cdot)}$

Let the composite vector be $\begin{bmatrix}\boldsymbol{t}^T & \boldsymbol{T}_{h,i,:}^T & \boldsymbol{T}_{\tilde{x},j,:}^T & \boldsymbol{\tau}^T\end{bmatrix}^T$ and let $L=\alpha+\gamma+\omega$. Then

$$\begin{aligned}I_{i,j}^{(L,h,x)}(\boldsymbol{t}) &= \int_{\mathbb{R}^K}\mathcal{K}_h(\boldsymbol{T}_{h,i,:},\boldsymbol{t}-\boldsymbol{\tau})\mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau},\boldsymbol{T}_{\tilde{x},j,:})\,\mathrm{d}\boldsymbol{\tau}\\ &= I_{\boldsymbol{\tau}}[(\sigma_h^2, 2\begin{bmatrix}\alpha+\gamma & -\gamma & 0 & -\alpha-\gamma\\ -\gamma & \alpha+\gamma & 0 & \gamma\\ 0 & 0 & 0 & 0\\ -\alpha-\gamma & \gamma & 0 & \alpha+\gamma\end{bmatrix})(1, 2\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&\omega&-\omega\\0&0&-\omega&\omega\end{bmatrix})],\\ &= I_{\boldsymbol{\tau}}[(\sigma_h^2, 2\left[\begin{array}{ccc|c}\alpha+\gamma & -\gamma & 0 & -\alpha-\gamma\\ -\gamma & \alpha+\gamma & 0 & \gamma\\ 0 & 0 & \omega & -\omega\\ \hline -\alpha-\gamma & \gamma & -\omega & L\end{array}\right])]\\ &= \sigma_h^2\frac{\pi^{K/2}}{L^{K/2}}\exp(-\alpha\|\boldsymbol{t}\|^2-\alpha\|\boldsymbol{T}_{h,i,:}\|^2-\gamma\|\boldsymbol{t}-\boldsymbol{T}_{h,i,:}\|^2-\omega\|\boldsymbol{T}_{\tilde{\boldsymbol{x}},j,:}\|^2\\ &\qquad + L^{-1}\|(\alpha+\gamma)\boldsymbol{t}-\gamma\boldsymbol{T}_{h,i,:}+\omega\boldsymbol{T}_{\tilde{x},j,:}\|^2).\end{aligned}$$

I.4.3 Integral $I^{(Q_1,\cdot,\cdot)}$

Let the composite vector be $\begin{bmatrix}\boldsymbol{t}_1^T & \boldsymbol{t}_2^T & \boldsymbol{\tau}^T\end{bmatrix}^T$. Then

$$
\begin{aligned}
I^{(Q_1,\cdot,\cdot)}(\boldsymbol{t}_1,\boldsymbol{t}_2) &= \int_{\mathbb{R}^K} \mathcal{K}_h(\boldsymbol{t}_1-\boldsymbol{\tau},\boldsymbol{t}_2-\boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau} \\
&= (\sigma_h^2, 2\begin{bmatrix}\gamma & -\gamma\\ -\gamma & \gamma\end{bmatrix}) I_{\boldsymbol{\tau}}[(1, 2\left[\begin{array}{cc|c}\alpha & 0 & -\alpha\\ 0 & \alpha & -\alpha\\ \hline -\alpha & -\alpha & 2\alpha\end{array}\right])] \\
&= (\sigma_h^2, 2\begin{bmatrix}\gamma & -\gamma\\ -\gamma & \gamma\end{bmatrix})\left(\frac{\pi^{K/2}}{(2\alpha)^{K/2}}, 2\begin{bmatrix}\alpha & 0\\ 0 & \alpha\end{bmatrix} - \frac{1}{\alpha}\begin{bmatrix}-\alpha\\ -\alpha\end{bmatrix}\begin{bmatrix}-\alpha\\ -\alpha\end{bmatrix}^T\right) \\
&= (\sigma_h^2, 2\begin{bmatrix}\gamma & -\gamma\\ -\gamma & \gamma\end{bmatrix})\left(\frac{\pi^{K/2}}{(2\alpha)^{K/2}}, \begin{bmatrix}\alpha & -\alpha\\ -\alpha & \alpha\end{bmatrix}\right) \\
&= \left(\sigma_h^2\frac{\pi^{K/2}}{(2\alpha)^{K/2}}, \begin{bmatrix}\alpha+2\gamma & -\alpha-2\gamma\\ -\alpha-2\gamma & \alpha+2\gamma\end{bmatrix}\right) \\
&= \sigma_h^2\frac{\pi^{K/2}}{(2\alpha)^{K/2}}\exp\left[-\frac{1}{2}(\alpha+2\gamma)\|\boldsymbol{t}_1-\boldsymbol{t}_2\|^2\right].
\end{aligned}
$$

I.4.4 Integral $I_{\cdot,\cdot}^{(Q_2,\cdot,\cdot)}$

Let the composite vector be $= \begin{bmatrix} \boldsymbol{t}_1^T & \boldsymbol{t}_2^T & \boldsymbol{T}_{\tilde{x},i,:}^T & \boldsymbol{T}_{\tilde{x},j,:}^T & \boldsymbol{\tau}_1^T & \boldsymbol{\tau}_2^T \end{bmatrix}^T$. Then

$$
\begin{aligned}
&I_{i,j}^{(Q_2,h,x)}(\boldsymbol{t}_1,\boldsymbol{t}_2) \\
&\quad= \int_{\mathbb{R}^{2K}} \mathcal{K}_h(\boldsymbol{t}_1-\boldsymbol{\tau}_1,\boldsymbol{t}_2-\boldsymbol{\tau}_2)\mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau}_1,\boldsymbol{T}_{\tilde{x},i,:})\mathcal{K}_{\tilde{x},x}(\boldsymbol{T}_{\tilde{x},j,:},\boldsymbol{\tau}_2)\,\mathrm{d}\boldsymbol{\tau}_1\,\mathrm{d}\boldsymbol{\tau}_2 \\
&\quad= I_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2}[(\sigma_h^2, 2\begin{bmatrix} \alpha+\gamma & -\gamma & 0 & 0 & -\alpha-\gamma & \gamma \\ -\gamma & \alpha+\gamma & 0 & 0 & \gamma & -\alpha-\gamma \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\alpha-\gamma & \gamma & 0 & 0 & \alpha+\gamma & \gamma \\ \gamma & -\alpha-\gamma & 0 & 0 & \gamma & \alpha+\gamma \end{bmatrix}) \\
&\qquad\qquad (1, 2\begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&\omega&0&-\omega&0 \\ 0&0&0&0&0&0 \\ 0&0&-\omega&0&\omega&0 \\ 0&0&0&0&0&0 \end{bmatrix})(1, 2\begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&\omega&0&-\omega \\ 0&0&0&0&0&0 \\ 0&0&0&-\omega&0&\omega \end{bmatrix})], \\
&\quad= I_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2}[(\sigma_h^2, 2\left[\begin{array}{cccc|cc} \alpha+\gamma & -\gamma & 0 & 0 & -\alpha-\gamma & \gamma \\ -\gamma & \alpha+\gamma & 0 & 0 & \gamma & -\alpha-\gamma \\ 0 & 0 & \omega & 0 & -\omega & 0 \\ 0 & 0 & 0 & \omega & 0 & -\omega \\ \hline -\alpha-\gamma & \gamma & -\omega & 0 & L & \gamma \\ \gamma & -\alpha-\gamma & 0 & -\omega & \gamma & L \end{array}\right]) \\
&\quad= \sigma_h^2 \frac{\pi^K}{(L^2-\gamma^2)^{K/2}} \\
&\qquad \exp\Big[-\alpha\|\boldsymbol{t}_1\|^2 - \alpha\|\boldsymbol{t}_2\|^2 - \gamma\|\boldsymbol{t}_1-\boldsymbol{t}_2\|^2 - \omega\|\boldsymbol{T}_{\tilde{x},i,:}\|^2 - \omega\|\boldsymbol{T}_{\tilde{x},j,:}\|^2 \\
&\qquad\quad + (L^2-\gamma^2)^{-1}\Big\{L\|(\alpha+\gamma)\boldsymbol{t}_1 - \gamma\boldsymbol{t}_2 + \omega\boldsymbol{T}_{\tilde{x},i,:}\|^2 \\
&\qquad\qquad\quad + L\|-\gamma\boldsymbol{t}_1 + (\alpha+\gamma)\boldsymbol{t}_2 + \omega\boldsymbol{T}_{\tilde{x},j,:}\|^2 \\
&\qquad\qquad\quad + 2\gamma[(\alpha+\gamma)\boldsymbol{t}_1 - \gamma\boldsymbol{t}_2 + \omega\boldsymbol{T}_{\tilde{x},i,:})^T(-\gamma\boldsymbol{t}_1 + (\alpha+\gamma)\boldsymbol{t}_2 + \omega\boldsymbol{T}_{\tilde{x},j,:}]\Big\}\Big].
\end{aligned}
$$

I.4.5 Integral $I_{\cdot,\cdot}^{(Q_3,\cdot,\cdot,\cdot)}$

Let the composite vector be $= \begin{bmatrix} \boldsymbol{t}_1^T & \boldsymbol{t}_2^T & \boldsymbol{T}_{h_k,i,:}^T & \boldsymbol{T}_{h_l,j,:}^T & \boldsymbol{\tau}^T \end{bmatrix}^T$. Then

$$
\begin{aligned}
&I_{i,j}^{(Q_3,h_k,h_l,x)}(\boldsymbol{t}_1,\boldsymbol{t}_2) \\
&\quad= \int_{\mathbb{R}^K} \mathcal{K}_h(\boldsymbol{t}_1-\boldsymbol{\tau},\boldsymbol{T}_{h_k,i,:})\mathcal{K}_h(\boldsymbol{T}_{h_l,j,:},\boldsymbol{t}_2-\boldsymbol{\tau})\,\mathrm{d}\boldsymbol{\tau} \\
&\quad= I_{\boldsymbol{\tau}}[(\sigma_h^2, 2\begin{bmatrix} \alpha+\gamma & 0 & -\gamma & 0 & -\alpha-\gamma \\ 0 & 0 & 0 & 0 & 0 \\ -\gamma & 0 & \alpha+\gamma & 0 & \gamma \\ 0 & 0 & 0 & 0 & 0 \\ -\alpha-\gamma & 0 & \gamma & 0 & \alpha+\gamma \end{bmatrix})(\sigma_h^2, 2\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha+\gamma & 0 & -\gamma & -\alpha-\gamma \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\gamma & 0 & \alpha+\gamma & \gamma \\ 0 & -\alpha-\gamma & 0 & \gamma & \alpha+\gamma \end{bmatrix})] \\
&\quad= I_{\boldsymbol{\tau}}[(\sigma_h^4, 2\left[\begin{array}{cccc|c} \alpha+\gamma & 0 & -\gamma & 0 & -\alpha-\gamma \\ 0 & \alpha+\gamma & 0 & -\gamma & -\alpha-\gamma \\ -\gamma & 0 & \alpha+\gamma & 0 & \gamma \\ 0 & -\gamma & 0 & \alpha+\gamma & \gamma \\ \hline -\alpha-\gamma & -\alpha-\gamma & \gamma & \gamma & 2\alpha+2\gamma \end{array}\right])] \\
&\quad= \left(\sigma_h^4 \frac{\pi^{K/2}}{(2\alpha+2\gamma)^{K/2}}, 2\begin{bmatrix} \alpha+\gamma & 0 & -\gamma & 0 \\ 0 & \alpha+\gamma & 0 & -\gamma \\ -\gamma & 0 & \alpha+\gamma & 0 \\ 0 & -\gamma & 0 & \alpha+\gamma \end{bmatrix} - \frac{1}{\alpha+\gamma}\begin{bmatrix} \alpha+\gamma \\ \alpha+\gamma \\ -\gamma \\ -\gamma \end{bmatrix}\begin{bmatrix} \alpha+\gamma \\ \alpha+\gamma \\ -\gamma \\ -\gamma \end{bmatrix}^T\right).
\end{aligned}
$$

Now

$$
\begin{aligned}
&2\begin{bmatrix}\alpha+\gamma & 0 & -\gamma & 0\\ 0 & \alpha+\gamma & 0 & -\gamma\\ -\gamma & 0 & \alpha+\gamma & 0\\ 0 & -\gamma & 0 & \alpha+\gamma\end{bmatrix} - \frac{1}{\alpha+\gamma}\begin{bmatrix}\alpha+\gamma\\ \alpha+\gamma\\ -\gamma\\ -\gamma\end{bmatrix}\begin{bmatrix}\alpha+\gamma\\ \alpha+\gamma\\ -\gamma\\ -\gamma\end{bmatrix}^T\\
&\quad = 2\begin{bmatrix}\alpha+\gamma & 0 & -\gamma & 0\\ 0 & \alpha+\gamma & 0 & -\gamma\\ -\gamma & 0 & \alpha+\gamma & 0\\ 0 & -\gamma & 0 & \alpha+\gamma\end{bmatrix} - \begin{bmatrix}\alpha+\gamma & \alpha+\gamma & -\gamma & -\gamma\\ \alpha+\gamma & \alpha+\gamma & -\gamma & -\gamma\\ -\gamma & -\gamma & \alpha+\gamma & \alpha+\gamma\\ -\gamma & -\gamma & \alpha+\gamma & \alpha+\gamma\end{bmatrix}\\
&\qquad - \frac{\gamma^2-(\alpha+\gamma)^2}{\alpha+\gamma}\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & 1\end{bmatrix}\\
&\quad = \begin{bmatrix}\alpha+\gamma & -\alpha-\gamma & -\gamma & \gamma\\ -\alpha-\gamma & \alpha+\gamma & \gamma & -\gamma\\ -\gamma & \gamma & \alpha+\gamma & -\alpha-\gamma\\ \gamma & -\gamma & -\alpha-\gamma & \alpha+\gamma\end{bmatrix} - \frac{\gamma^2-(\alpha+\gamma)^2}{\alpha+\gamma}\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & 1\end{bmatrix}
\end{aligned}
$$

and so

$$
\begin{aligned}
I_{i,j}^{(Q_3,h_k,h_l,x)}(\boldsymbol{t}_1,\boldsymbol{t}_2) = \sigma_h^4\frac{\pi^{K/2}}{(2\alpha+2\gamma)^{K/2}}\exp\Bigg[&-\frac{1}{2}\alpha\|\boldsymbol{t}_1-\boldsymbol{t}_2\|^2 - \frac{1}{2}\alpha\|\boldsymbol{T}_{h_k,i,:}-\boldsymbol{T}_{h_l,j,:}\|^2\\
&-\frac{1}{2}\gamma\|(\boldsymbol{t}_1-\boldsymbol{t}_2)-(\boldsymbol{T}_{h_k,i,:}-\boldsymbol{T}_{h_l,j,:})\|^2\\
&+\frac{1}{2}\frac{\gamma^2-(\alpha+\gamma)^2}{\alpha+\gamma}\|\boldsymbol{T}_{h_k,i,:}+\boldsymbol{T}_{h_l,j,:}\|^2\Bigg].
\end{aligned}
$$

Furthermore, denote $\boldsymbol{t}_1-\boldsymbol{t}_2=\boldsymbol{t}$ and $\boldsymbol{T}_{h_k,i,;}-\boldsymbol{T}_{h_l,j,:}=\boldsymbol{T}$. Then

$$
\begin{aligned}
\alpha\|\boldsymbol{t}\|^2+\alpha\|\boldsymbol{T}\|^2+\gamma\|\boldsymbol{t}-\boldsymbol{T}\|^2 &= (\alpha+\gamma)\|\boldsymbol{t}\|^2 - 2\gamma\boldsymbol{t}^T\boldsymbol{T}+(\alpha+\gamma)\|\boldsymbol{T}\|^2\\
&= (\alpha+\gamma)\left\|\boldsymbol{t}-\frac{\gamma}{\alpha+\gamma}\boldsymbol{T}\right\|^2 + \frac{(\alpha+\gamma)^2-\gamma^2}{\alpha+\gamma}\|\boldsymbol{T}\|^2
\end{aligned}
$$

so that

$$I_{i,j}^{(Q_3,h_k,h_l,x)}(\boldsymbol{t}_1,\boldsymbol{t}_2) = \sigma_h^4 \frac{\pi^{K/2}}{(2\alpha+2\gamma)^{K/2}} \exp\left[-\frac{1}{2}(\alpha+\gamma)\left\|(\boldsymbol{t}_1-\boldsymbol{t}_2) - \frac{\gamma}{\alpha+\gamma}(\boldsymbol{T}_{h_k,i,:}-\boldsymbol{T}_{h_l,j,:})\right\|^2 + \frac{\gamma^2-(\alpha+\gamma)^2}{\alpha+\gamma}(\|\boldsymbol{T}_{h_k,i,:}\|^2+\|\boldsymbol{T}_{h_l,j,:}\|^2)\right].$$

I.4.6 Integral $I_{\cdot,\cdot}^{(Q_4,\cdot,\cdot)}$

Let the composite vector be $\begin{bmatrix}\boldsymbol{t}^T & \boldsymbol{T}_{h,i,:}^T & \boldsymbol{T}_{\tilde{x},j,:}^T & \boldsymbol{\tau}^T\end{bmatrix}^T$. Then

$$\begin{aligned}
I_{i,j}^{(Q_4,h,x)}(\boldsymbol{t}) &= \int_{\mathbb{R}^K} \mathcal{K}_h(\boldsymbol{t}-\boldsymbol{\tau},\boldsymbol{T}_{h,i,:})\mathcal{K}_{x,\tilde{x}}(\boldsymbol{\tau},\boldsymbol{T}_{\tilde{x},j,:})\,\mathrm{d}\boldsymbol{\tau} \\
&= I_{\boldsymbol{\tau}}[(\sigma_h^2, 2\begin{bmatrix}\alpha+\gamma & -\gamma & 0 & -\alpha-\gamma\\ -\gamma & \alpha+\gamma & 0 & \gamma\\ 0 & 0 & 0 & 0\\ -\alpha-\gamma & \gamma & 0 & \alpha+\gamma\end{bmatrix})(1, 2\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&\omega&-\omega\\0&0&-\omega&\omega\end{bmatrix})] \\
&= I_{\boldsymbol{\tau}}[(\sigma_h^2, 2\left[\begin{array}{ccc|c}\alpha+\gamma & -\gamma & 0 & -\alpha-\gamma\\ -\gamma & \alpha+\gamma & 0 & \gamma\\ 0 & 0 & \omega & -\omega\\ \hline -\alpha-\gamma & \gamma & -\omega & L\end{array}\right])] \\
&= \sigma_h^2 \frac{\pi^{K/2}}{L^{K/2}} \exp[-\alpha\|\boldsymbol{t}\|^2 - \alpha\|\boldsymbol{T}_{h,i,:}\|^2 - \gamma\|\boldsymbol{t}-\boldsymbol{T}_{h,i,:}\|^2 - \omega\|\boldsymbol{T}_{\tilde{x},j,:}\|^2 \\
&\qquad + L^{-1}\|(\alpha+\gamma)\boldsymbol{t} - \gamma\boldsymbol{T}_{h,i,:} + \omega\boldsymbol{T}_{\tilde{x},j,:}\|^2].
\end{aligned}$$

I.5 Conclusion

We have derived an analytical expression for the variational free energy of Model 5. The asymptotic time complexity of computing the free energy including one-time costs is given by

$$\begin{aligned}
&\mathcal{O}[NMT_H^3 + MT_{\tilde{x}}^3 + YN(C_\mu + C_\Sigma) + \text{one-time costs}] \\
&\quad = \mathcal{O}[NMT_H^3 + MT_{\tilde{x}}^3 + YNM(T_H^2T_{\tilde{x}} + KT_H^2 + KT_HT_{\tilde{x}} + KT_{\tilde{x}}^2 + T_HT_{\tilde{x}}^2) + YNM^2 \\
&\qquad + M(T_H^3 + KT_H^2 + KT_{\tilde{x}}^2 + T_{\tilde{x}}^3)] \\
&\quad = \mathcal{O}[NMT_H^3 + MT_{\tilde{x}}^3 + YNM(T_H^2T_{\tilde{x}} + KT_H^2 + KT_HT_{\tilde{x}} + KT_{\tilde{x}}^2 + T_HT_{\tilde{x}}^2) + YNM^2].
\end{aligned}$$

References

Álvarez, M. A. and Lawrence, N. D. (2011). Computationally Efficient Convolved Multiple Output Gaussian Processes. *Journal of Machine Learning Research*, (12):1459–1500. (Pages xxii, 31, 33, and 34)

Álvarez, M. A., Luengo, D., and Lawrence, N. D. (2009). Latent Force Models. *Artificial Intelligence and Statistics*, 5:9–16. (Pages xxiii, 31, 33, and 34)

Bonilla, E. V., Chai, K. M., and Williams, C. K. I. (2008). Multi-Task Gaussian Process Prediction. *Advances in Neural Information Processing Systems*, 20:153–160. (Pages xxiii, 33, and 34)

Duvenaud, D. (2014). *Automatic Model Construction with Gaussian Processes*. PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge. (Page 1)

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation*. Oxford University Press. (Pages xxii, xxiii, 32, 33, and 34)

Gray, R. M. (2006). Toeplitz and Circulant Matrices: A Review. *Foundations and Trends in Communications and Information Theory*, 2(3):155–239. (Pages 62 and 66)

Higham, N. J. (1988). Computing a Nearest Symmetric Positive Semidefinite Matrix. *Linear Algebra and its Applications*, 103:103–118. (Page 57)

MacKay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press. (Pages 10 and 16)

Minka, T. (2000). Deriving Quadrature Rules from Gaussian Processes. Technical report, Statistics Department, Carnegie Mellon University. (Page 21)

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press. (Page 51)

Nguyen, T. V. and Bonilla, E. V. (2014). Collaborative Multi-Output Gaussian Processes. *Conference on Uncertainty in Artificial Intelligence*, 30. (Pages xxii, 33, and 34)

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press. (Pages 1, 9, and 10)

Teh, Y. W. and Seeger, M. (2005). Semiparametric Latent Factor Models. *International Workshop on Artificial Intelligence and Statistics*, 10. (Pages xxiii, 33, and 34)

Titsias, M. K. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. *Artificial Intelligence and Statistics*, 12:567–574. (Pages 23, 26, and 27)

Tobar, F., Bui, T. D., and Turner, R. E. (2015a). Design of Covariance Functions Using Inter-Domain Inducing Variables. *Time Series Workshop on Advances in Neural Information Processing Systems*. (Page 27)

Tobar, F., Bui, T. D., and Turner, R. E. (2015b). Learning Stationary Time Series using Gaussian Processes with Nonparametric Kernels. *Advances in Neural Information Processing Systems*, 29:3501–3509. (Pages vii, xvii, xxii, 3, 15, 21, 25, 26, and 37)

Tobar, F. and Turner, R. E. (2016). Modelling Time Series via Automatic Learning of Basis Functions. *Ninth IEEE Sensor Array and Multichannel Signal Processing Workshop*. (Pages 29 and 30)

Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics. (Pages 55 and 56)

Ulrich, K., Carlson, D. E., Dzirasa, K., and Carin, L. (2015). GP Kernels for Cross-Spectrum Analysis. *Advances in Neural Information Processing Systems*, 28:1999–2007. (Pages vii, xxii, 3, 25, 26, 33, 34, and 61)

Wilson, A. G. and Adams, R. P. (2013). Gaussian Process Kernels for Pattern Discovery and Extrapolation. *International Conference on Machine Learning*, 3:1067–1075. (Pages xxiii, 1, 25, and 26)

Wilson, A. G., Knowles, D. A., and Ghahramani, Z. (2012). Gaussian process Regression Networks. *International Conference on Machine Learning*, 29.

(Pages xxii, 31, and 33)

Index